An addendum, titled *Addendum to Rev. 2 of the i.MX53 Applications Processor Reference Manual (Rev. 2.1)* has been added at the end of this document.

i.MX53 Multimedia Applications Processor Reference Manual

Document Number: iMX53RM
Rev. 2.1, 06/2012

freescale™

Contents

Section Number **Title** **Page**

Chapter 1
Introduction

Chapter 2
Memory Map

Chapter 3
Interrupts and SDMA Events

Chapter 4
External Signals and Pin Multiplexing

Chapter 5
External Memory

Chapter 6
System Debug

Chapter 7
System Boot

Chapter 8
Multimedia

Chapter 9
Power Management

Chapter 10
System Security

Chapter 11
ARM Cortex A8 Platform (ARM Platform)

Chapter 12
ARM Platform Debug

Chapter 13
Multi-Layer AHB Crossbar Switch (AHBMAX)

Chapter 14
AHB to IP Bridge (AIPSTZ)

Chapter 15
Asynchronous Sample Rate Converter (ASRC)

Chapter 16
Digital Audio Multiplexer (AUDMUX)

Chapter 17
Clock Amplifier (CAMP)

Chapter 18
Clock Control Module (CCM)

Chapter 19
Configurable Serial Peripheral Interface (CSPI)

Chapter 20
Central Security Unit (CSU)

Chapter 21
DPLL Controller (DPLLC)

Chapter 22
Dynamic Voltage and Frequency Scaling Core (DVFSC)

Chapter 23
Dynamic Voltage and Frequency Scaling for Peripherals (DVFSP)

Chapter 24
Enhanced Configurable SPI (ECSPI)

Chapter 25
External Interface Module (EIM)

Chapter 26
Enhanced Periodic Interrupt Timer (EPIT)

Chapter 27
Enhanced Serial Audio Interface (ESAI)

Chapter 28
Enhanced SDRAM Controller (ESDCTL)

Chapter 29
Enhanced Secured Digital Host Controller (eSDHCv2)

Chapter 30
Enhanced Secured Digital Host Controller (eSDHCv3)

Chapter 31
External Memory Controller (EXTMC)

Chapter 32
Fast Ethernet Controller (FEC)

Chapter 33
Fast Infrared Interface (FIRI)

Chapter 34
Flexible Controller Area Network (FLEXCAN)

Chapter 35
Electrical Fuse Array (FUSEBOX)

Chapter 36
General Power Controller (GPC)

Chapter 37
General Purpose Input/Output (GPIO)

Chapter 38
General Purpose Timer (GPT)

Chapter 39
2D Graphics Processing Unit (GPU2D)

Chapter 40
3D Graphics Processing Unit (GPU3D)

Chapter 41
I2C Controller (I2C)

Chapter 42
IC Identification Module (IIM)

Chapter 43
IOMUX Controller (IOMUXC)

Chapter 44
IEEE 1588 Precision Time Protocol Assist (IPTP)

Chapter 45
Image Processing Unit (IPU)

Chapter 46
Keypad Port (KPP)

Chapter 47
LVDS Display Bridge (LDB)

Chapter 48
Low-Dropout Regulator (LDO)

Chapter 49
Multi Master Multi Memory Interface (M4IF)

Chapter 50
Media Local Bus (MediaLB) Block (MLB)

Chapter 51
NAND Flash Controller (NFC)

Chapter 52
On-Chip RAM Memory Controller (OCRAM)

Chapter 53
1-Wire Block (OWIRE)

Chapter 54
Parallel Advanced Technology Attachment (PATA)

Chapter 55
Power Fail Detector (PFD)

Chapter 56
PL301 4x1 AXI Arbiter (PLARB1)

Chapter 57
PL301 2x2 Arbiter (PLARB2)

Chapter 58
Power On Reset (POR)

Chapter 59
Pulse Width Modulation (PWM)

Chapter 60
ROM Controller with Patch (ROMC)

Chapter 61
Run-Time Integrity Checker (RTIC)

Chapter 62
SAHARA Security Accelerator (SAHARA)

Chapter 63
Serial Advanced Technology Attachment Controller (SATA)

Chapter 64
Serial Advanced Technology Attachment PHY (SATA PHY)

Chapter 65
Security Controller (SCC)

Chapter 66
Smart Direct Memory Access Controller (SDMA)

Chapter 67
System JTAG Controller (SJC)

Chapter 68
Shared Peripheral Bus Arbiter (SPBA)

Chapter 69
Sony/Philips Digital Interface (SPDIF)

Chapter 70
System Reset Controller (SRC)

Chapter 71
State Retention Power Gating Controller (SRPGC)

Chapter 72
Secure Real Time Clock (SRTC)

Chapter 73
Synchronous Serial Interface (SSI)

Chapter 74
Television Encoder (TVEv2)

Chapter 75
TrustZone Aware Interrupt Controller (TZIC)

Chapter 76
Universal Asynchronous Receiver/Transmitter (UART)

Chapter 77
Universal Serial Bus Controller (USB)

Chapter 78
Video Processing Unit (VPU)

Chapter 79
Watchdog Timer (WDOG-1)

Chapter 80
Crystal Oscillator 24 MHz (XTALOSC)

Chapter 81
Crystal Oscillator 32K (XTALOSC32K)

Chapter 1
Introduction

1.1 About This Document

This reference manual describes the functionality of the i.MX53 multimedia applications processor. The i.MX53 is Freescale Semiconductor's latest addition to a growing family of multimedia-focused products offering high performance processing optimized for lowest power consumption. The i.MX53 processors feature Freescale's advanced implementation of the ARM®Cortex™-A8 core, which operates at speeds as high as 1.2 GHz and interfaces with DDR2-800, LVDDR2-800 and DDR3-800 DRAM memory devices. This product is suitable for applications such as:

- Automotive navigation and entertainment
- High-end Mobile Internet Devices and high-end PDAs
- Netbooks
- Nettops
- High-end portable media players with HD video capability
- Portable navigation devices
- Gaming Consoles

The i.MX53 is a system on a chip (SoC) which integrates on one integrated circuit a complete microcomputer platform based on the ARM Cortex-A8 including an integrated NEON coprocessor, a vector floating point unit, an Execution Trace Module (ETM), separate 32 Kbyte instruction and data L1 caches and a unified 256 Kbyte L2 cache. Supporting the ARM Platform are a large array of system-level components such as system RAM, processor code ROM, Smart DMA controller (SDMA) and system control devices such as a power-on reset controller, voltage regulators, power management, on-board fuse array (FUSEBOX) and clock generation circuitry. Coprocessors including two Graphics Processing Units (GPUs), an Image Processin Unit (IPU), Video Processing Unit (VPU) and Asynchronous Sample Rate Converter (ASRC) accelerate computationally intensive tasks offloading the ARM Platform.

Included in the SoC are I/O devices such as USB controllers, UARTs, Ethernet controllers, General Purpose I/O (GPIO), Multimedia processor, , general purpose and special purpose timers, and system security devices. An integrated memory controller supports the attachment of external DRAM. A Flash controller supports both NAND and NOR Flash devices. Integrated external storage controllers support devices such as SATA hard disks and SD/SDIO/MMC/eMMC memory cards.

1.1.1 Audience

This manual is intended to be used by board-level product designers and product software developers. This manual assumes that the reader has a background in computer engineering and/or software engineering and understands concepts of digital system design, microprocessor architecture, Input / Output (I/O) devices and industry standard communication and device interface protocols.

1.1.2 Organization

This document is organized in two main sections called Book I and Book II.

Book I covers the i.MX53 at a system level and provides an architectural overview. Also covered are system memory map, system-level interrupt events, external pins and pin multiplexing, external memory, system debug, system boot, multimedia subsystem, power management, and system security.

Book II describes the ARM Platform, ARM Platform debug, and an array of internal functional blocks.

1.1.3 Suggested Reading

This section lists additional reading that provides background for the information in this manual as well as general information about the architecture.

1.1.3.1 General Information

The following documentation provides useful background information about the ARM processor and computer architecture in general:

- For information about the ARM Cortex-A8 processor see http://www.arm.com/products/processors/cortex-a/cortex-a8.php

- Computer Architecture: A Quantitative Approach, Fourth Edition, by John L. Hennessy and David A. Patterson
- *Computer Organization and Design: The Hardware/Software Interface*, Second Edition, by David A. Patterson and John L. Hennessy

1.1.3.2 Related Documentation

Freescale documentation is available from the sources listed on the back cover of this manual; the document order numbers are included in parentheses for ease in ordering:

For a current list of documentation, refer to www.freescale.com.

1.1.4 Conventions

This document uses the following notational conventions:

cleared / set
When a bit takes the value zero, it is said to be cleared; when it takes a value of one, it is said to be set.

mnemonics
Instruction mnemonics are shown in lowercase bold

italics
Italics indicate variable command parameters, for example, **bcctr***x*

Book titles in text are set in italics

15
An integer in decimal

0x
Prefix to denote hexadecimal number

0b
Prefix to denote binary number. Binary 0 and 1 are written without the prefix.

n'H4000CA00
n-bit Hexadecimal number

BLK_REG_NAME
Register names are all uppercase. The block mnemonic is prepended with an underscore delimiter (_).

BLK_REG[FIELD]
Fields within registers appear in brackets. For example, ESR[RLS] refers to the Receive Last Slot field of the ESAI Status Register.

BLK_REG[*n*]

Bit number *n* within register BLK.REG. Bit numbering is little endian.

BLK_REG[*l:r*]

Register bit ranges. Ranges are indicated by the left-most bit number *l* and the right-most bit number *r* separated by a colon (:). For example, ESR[15:0] refers to the lower half word in the ESAI Status Register.

x, U

In some contexts, such as signal encodings, an unitalicized x indicates a don't care or uninitialized. The binary value could be 1 or 0.

x

An italicized *x* indicates an alphanumeric variable

n, m

Italicized *n or m* represent integer variables

!

Binary logic operator NOT

&&

Binary logic operator AND

||

Binary logic operator OR

^ or <O+>

Binary logic operator XOR

|

Bit-wise OR. For example, 0b0001 | 0b1000 yields the value 0b1001.

&

Bit-wise AND. For example, 0b0001 & 0b1000 yields the value 0b0000.

{A,B}

Concatenation, where the *n*-bit value A is prepended to the *m*-bit value B to form an (*n* +*m*)-bit value. For example, {0, REG*m* [14:0]} yeilds a 16-bit value with 0 in the most significant bit.

- or grey fill

Indicates a reserved bit field in an register. Although these bits can be written to as ones or zeros, they are always read as zeros.

>>

Shift right logical one position

<<

Shift left logical one posiiton

= <left arrow>

Assignment

==

Compare equal

!=

Compare not equal

>
Greater than
<
Less than

1.1.5 Register Diagram Field Access Type Legend

The following figure provides the interpretation of the notation used in register diagrams for a number of common field access types.

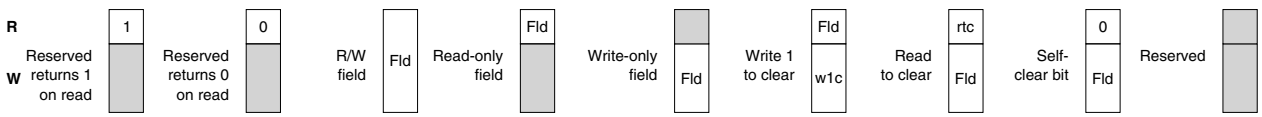


Figure 1-1. Register Field Conventions

NOTE

For reserved register fields, software should mask off the data in the field after read (software can not rely on the contents of data read from a reserved field) and always write all zeros.

1.1.6 Signal Conventions

_b, _B
When appended to a signal name, indicates that a signal is active-low

NEG_ACTIVE
Overbar also denotes a negative active signal

UPPERCASE
Package pin names, Block I/O signals

lowercase
Lowercase is used to indicate internal signals

1.1.7 Acronyms and Abbreviations

Table 1- contains acronyms and abbreviations used in this document.

Acronyms and Abbreviated Terms

Term	Meaning
BIST	Built-in self test
DDR	Double data rate (of Dynamic RAM)

Table continues on the next page...

Term	Meaning
FIFO	First In / First Out (of a queue)
DMA	Direct memory access
DPLL	Digital phase-locked loop
DRAM	Dynamic random access memory
EPROM	Erasable programmable read-only memory
GPIO	General-purpose I/O
GPR	General-purpose register
GPU	Graphicd Processing Unit
I2C or I2C	Inter-integrated circuit
IEEE	Institute of Electrical and Electronics Engineers
IrDA	Infrared Data Association
JTAG	Joint Test Action Group (a serial bus protocol usually used for test purposes)
LIFO	Last-in-first-out
LRU	Least recently used
LSB	Least-significant byte
lsb	Least-significant bit
MSB	Most-significant byte
msb	Most-significant bit
PCI	Peripheral Component Interconnect
PCI-X	PCI extended
PCIe	PCI enhanced
PCMCIA	Personal Computer Memory Card International Association
PIC	Programmable interrupt controller
POR	Power-on reset
RISC	Reduced instruction set computing
RTOS	Real-time operating system
Rx	Receive
SDLC	Synchronous data link control
SDMA	Serial DMA
SPDIF	Sony Phillips Digital Interface
SPI	Serial peripheral interface
SRAM	Static random access memory
Tx	Transmit
UART	Universal asynchronous receiver/transmitter
USB	Universal serial bus

1.2 Overview

This chapter introduces the architecture of the i.MX53 Multimedia Applications Processor. The i.MX53 processor represents Freescale Semiconductor's latest achievement in multimedia integrated applications processors that are part of a growing family of multimedia-focused products offering high performance processing optimized for lowest power consumption.

1.3 Target Applications

The primary market focus of the i.MX53 applications processor is high-end Mobile Internet Devices (MIDs) and Automotive Infotainment. A secondary focus is High End Portable Multimedia Players (PMPs) with HD video capability, as well as markets with similar requirements in terms of package type, thermal limits and I/O. The flexibility of the i.MX53 architecture permits in a wide variety of other applications. The i.MX53 processor provides all of the interfaces for connecting peripherals such as WLAN, Bluetooth, GPS, camera sensors, and dual displays.

The i.MX53 application processor is a follow-on to the i.MX51 with improved performance, power efficiency and multimedia capabilities.

1.4 Features

The i.MX53 ARM Platform (AP) is based on the ARM Cortex A8™ Architecture, which has the following features:

- ARM Cortex A8™ Processor (with TrustZone)
- 32 Kbyte L1 instruction cache
- 32 Kbyte L1 data cache
- 256 Kbyte unified instruction and data L2 cache
- A range of core processor clock speeds (up to 1.2 GHz) based on part number.
- Neon coprocessor
  - SIMD Media Processing Architecture
  - NEON register file with 32x64-bit general-purpose registers
  - NEON Integer execute pipeline (ALU, Shift, MAC)
  - NEON dual, single-precision floating point execute pipeline (FADD, FMUL)
  - NEON load/store and permute pipeline
  - Non-pipelined Vector Floating Point (VFP) coprocessor (VFPv3)

The i.MX53 makes use of dedicated hardware (H/W) accelerators in order to meet specific targeted multimedia performance requirements. The use of H/W accelerators is key factor in obtaining high performance at low power consumption while leaving the ARM platform core relatively free for performing other tasks.

The i.MX53 incorporates the following hardware accelerators:

- VPU-Video Processing Unit
- IPU-Image Processing Unit, version 3M
- GPU3D-3D Graphics Processing Unit (OpenGL ES 2.0), version 3
- GPU2D-2D Graphics accelerator, (OpenVG 1.1), version 1
- ASRC-Asynchronous Sample Rate Converter

Security features are enabled and accelerated by the following hardware:

- ARM TrustZone including the TZ architecture (separation of interrupts, memory mapping, etc.)
- SJC-System JTAG Controller. Protects JTAG from debug port attacks by regulating or blocking access to the system debug features
- SRTC-Secure Real-Time Clock (RTC). Tamper resisted RTC with it's own power domain and mechanism to detect voltage and clock glitches
- RTIC-Real-Time Integrity Checker, version 3, RTIC type 1, enhanced with SHA-256 engine
- SAHARA-Cryptographic accelerator that includes true random number generator (TRNG)
- SCC-Security Controller, type 2. Improved SCC v1 with AES engine, Secure/Non-Secure RAM and support for multiple keys as well as TZ/non-TZ separation
- CSU-Central Security Unit. Enhancement for the IC Identification Module (IIM) configured during boot using e-fuses. Determines the security level operation mode as well as the TZ policy
- AHAB-Advanced High Assurance Boot, with the next embedded enhancements: SHA-256, 2048-bit RSA key, version control mechanism, warm boot, CSU and TZ initialization

Integrated memory system:

- Boot ROM, including High Assurance Boot (HAB) (64 KB)
- Internal multimedia / shared, fast access RAM (128 KB)
- Secure/non-secure RAM (16 KB)

The i.MX53 SoC is built around the following System buses:

- 64-bit AMBA AXI v1.0 (AXI). Provides high-bandwidth, low-latency connectivity for the ARM Platform, major multimedia accelerators (VPU, IPU, GPU3D, GPU2D) and the External Memory Controller (EXTMC)

- 32-bit AMBA AHB 2.0 (AHB). Provides connectivity for bus master peripherals, such as SDMA, RTIC, SCC, and SAHARA. (See block diagram for the complete list.)
- 32-bit Internal Peripheral (IP) Bus. Provides control and data communication for lower speed integrated peripheral devices.

The i.MX53 enables following interfaces to external devices (some of which are not available simultaneously):

- Hard Disk Drives
  - PATA, supports U-DMA mode 5 at transfer rates up to 100 MByte/sec.
  - SATA II protocol with a peak transfer rate of 3.0 Gbits per second.
- Displays
  - 5 interfaces available. Total rate of all interfaces is up to 180 Mpixels/sec, 24 bits per pixel. Up to two interfaces may be active at once.
  - Two Parallel 24-bit display ports supporting up to 165 Mpixels/sec (UXGA @ 60Hz)
  - LVDS serial ports: One port up to 165 Mpixels/sec or two ports up to 85 Mpixels/sec (WXGA @ 60Hz) each
  - One TV-out/VGA port up to 150 Mpixels/sec (1080p at 60 Hz)
- Camera sensors
  - Two parallel camera ports
    - Primary port only: up to 20 bits/pixel, up to 180 MHz peak pixel clock rate
    - Simultaneous use: 8-bit primary and 8-bit secondary.
- Expansion cards
  - Four SD/MMC card ports:
    - Two of which support 208 Mbps (4-bit)
    - One 416 Mbits/sec
    - One enhanced port - supports 832 Mbps, (8-bit, eMMC 4.4).
- External memory interfaces
  - 16/32-bit DDR2-800, LV-DDR2-800 and DDR3-800
  - 8/16-bit NAND SLC/MLC Flash, 4/8/14/16-bit ECC.
  - Supports Samsung OneNAND™ (in muxed I/O mode)
  - 8/16/32-bit NOR Flash (8-bit is not supported at byte D[23:16]). Interface is provided via the External Interface Module (EIM), all EIM pins are muxed with other interfaces (data with NAND Flash Controller (NFC) pins). I/O muxing logic selects the EIM port as primary muxing at system boot.
  - 8/16/32-bit Pseudostatic RAM (PSRAM), Cellular RAM.
- USB
  - High Speed (HS) USB 2.0 OTG (Up to 480 Mbps), with integrated HS USB Phy
  - Three USB 2.0 (480 Mbps) hosts:
    - HS host, with integrated High Speed Phy.

    - HS host for external HS/FS Transceivers via ULPI / Serial, supports IC-USB
    - HS host for external HS/FS Transceivers via ULPI / Serial, supports IC-USB
- Low Power modes
    - Supporting DVFS techniques for low power modes
    - Uses SRPG (State Retention Power Gating) for ARM and Neon
    - Support for various levels of system power modes
    - Flexible clock gating control scheme
- Other interfaces:
    - OWIRE
    - Three I2S/SSI/AC97, up to 1.4 Mbps each connected to Audio Multiplexer (AUDMUX) providing four external ports
    - Enhanced Serial Audio Interface (ESAI), up to 1.4 Mbps each channel
    - Five UART, up to 4.0 Mbps each
        - One of the five supports 8-wire (uart1) while others four supports 4-wire.
    - One CSPI; refer to the data sheet for performance parameters.
    - Two eCSPI (enhanced CSPI); refer to the data sheet for performance parameters.
    - Three I2C, supports 400 Kbps
    - Fast Ethernet Controller (FEC) IEEE1588 compliant, 10/100 Mbps
    - Two Pulse Width Modulators (PWM)
    - JTAG Controller (SJC)
    - GPIO with interrupt capabilities
    - Key Pad Port (KPP)
    - Sony Phillips Digital Interface (SPDIF), Rx and Tx
    - Two Controller Area Network (FLEXCAN), 1 Mbps each
    - Two Watchdog timers (WDOG)
    - Media Local Bus controller (MLB) provides interface to Media Oriented Systems Transport (MOST) Networks (50Mbps)

1.5 Architectural Overview

This section contains i.MX53 architectural details.

A simplified block diagram is provided in the following section.

1.5.1 Simplified Block Diagram

A high-level block diagram of the i.MX53 is shown in Figure 1-2. It provides a view of the major sub-systems (processor domains, shared peripherals domain, memories, etc.) and logical connectivity.

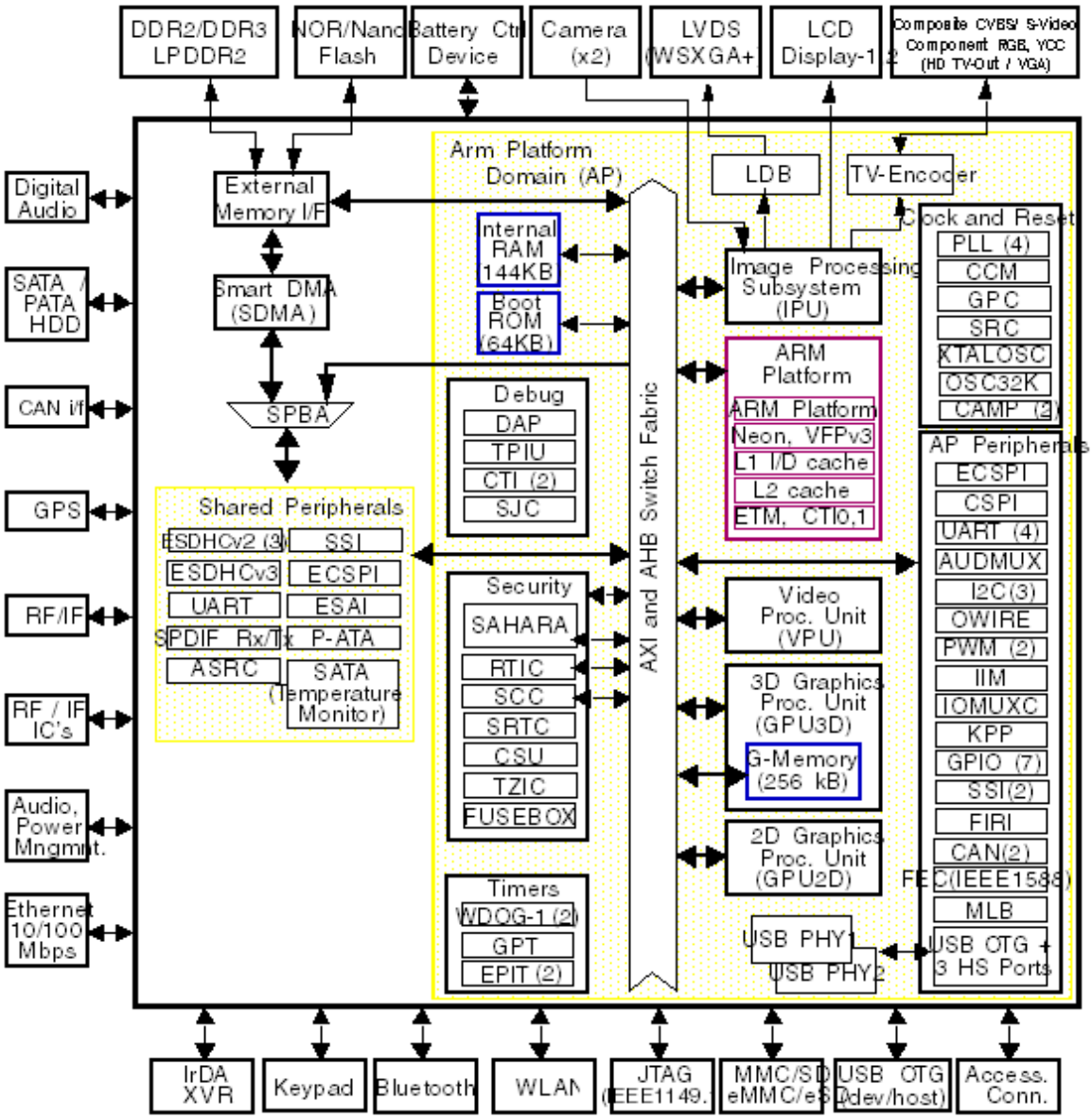


Figure 1-2. i.MX53 Simplified Block Diagram

1.5.2 Major Subsystems

i.MX53 consists of the following major subsystems:

- Core (ARM Cortex A8) Platform, L1/L2 memories
- SDMA controller and the shared peripheral domain
- System Control-Boot Flow control, Clocks distribution, "Reset" control and Low Power logic
- Multimedia
- Security

- Connectivity peripherals and timers
- External Memory Interface

1.5.3 Architectural Partitioning

The i.MX53 Architecture may be functionally organized in terms of five sub-systems. These are:

- The ARM Platform is the central processing unit for the SoC and runs the following software:
  - Power-on-Reset (POR) boot code
  - Boot-strap loader
  - Operating system and application program loader
  - User applications (including control over hardware accelerators and non-accelerated functions)
  - TrustZone applications
- Smart DMA enables data transfer between non-mastering (slave) peripherals and external or internal memories
- System Control
  - Clock Control Block (CCM)
  - Four PLLs
  - XTALOSC- 24MHz Crystal oscillator source support
  - XTALOSC32k- 32k Crystal oscillator support
  - System Reset Controller (SRC)
  - Global Power Controller (GPC)
  - Two Clock Amplifier (CAMP) blocks on CKIH and CKIH2 pins
  - Temperature Sensor, for monitoring and acting on high temperature situations.

NOTE

Refer to SATA Temperature Sensor application note (AN4380) for more details regarding programming and its usage.

- Multimedia
  - Image Processing Unit-IPU
    - Connectivity to displays, display controllers, cameras and auxiliary graphics coprocessors.
    - Display Processing: video/graphics combining, image enhancement
    - Image conversions: resizing, rotation/inversion, color conversion, deinterlacing
    - Synchronization and control capabilities, allowing autonomous operation
  - Video Processing Unit (VPU):

        - Various decoding/encoding formats in HW
        - Up to 1080i/p resolution (H.264, VC1, RV10, DivX)
        - Up to 720p encode (MPEG4, H.264)
        - TV-Encoder (TVE) for HD720p/1080p, PAL/NTSC or VGA output
    - Graphics Processing Unit (GPU3D), 3D graphics processing compliant with the following:
        - OpenGL ES Common Profile v1.0
        - OpenGL ES Common Profile v1.1/Direct3D Mobile
        - OpenGL ES Profile v2.0
    - Graphics Processing Unit (GPU2D), 2D graphics processing:
        - OpenVG 1.1
    - Audio
        - Audio codecs are provided by SW, which runs on ARM core, supporting (but not limited to) MP3, WMA, AAC, HE-AAC and Pro10
        - 3x SSIs
        - ESAI
        - SPDIF Tx/Rx
        - Audio Mux
        - ASRC (for sample rate conversion)
- Security
    - High Assurance Boot (HAB) System
    - ARM TrustZone (TZ) Trusted Execution environment
    - IC Identification Module (IIM) and Central Security Unit (CSU)
    - On-chip One-Time programmable electrical fuse array (FUSEBOX)
    - RTIC: Real-Time Integrity Checker
    - SAHARA Version 4 Lite (SAHARA) cryptographic acceleration engine
    - System JTAG controller (SJC)
    - Secure Real Time Clock (SRTC)
    - Security Controller version 2 (SCC) with 16KByte of secure/non-secure RAM
    - Tamper Detection
    - TrustZone Watchdog (WDOG-2)
- Connectivity peripherals and timers
    - Low level communication protocols
    - Embedded DMAs
    - 3.3V IO voltage for seamless integration
    - Four USB 2.0 ports, including two integrated PHYs
    - TV-Out Video codecs for HD, NTSC/PAL or VGA output
    - DDR2/3, LPDDR2 (PoP package), Nand (MLC 4/8/14/16-bit ECC) and NOR Flash memory interface via EMC

- Timers: 2xEPIT, GPT and Watch Dog timer (WDOG)
- Miscellaneous connectivity support-I2C, SPI, UART, PWM and Keypad interface

- External Memory Controller (EXTMC)
  - External Memory Interface
  - Integrated DRAM controller
    - Support for DDR2, DDR3, LV-DDR and LPDRAM
  - Other external storage devices
    - Support for NAND and NOR Flash
    - Support for PSRAM and Cellular RAM

1.5.4 Endianness Support

i.MX53 supports Little Endian mode only.

1.6 Block List

Table 1-1 provides listing of the blocks used by the various subsystems of the i.MX53.

Table 1-1. Digital and Analog Blocks

Block Mnemonic	Block Name	Subsystem	Brief Description
ARM Platform	ARM Cortex A8™ Platform	ARM	The ARM Platform consists of the ARM Cortex A8™ processor and its essential sub-blocks, the Level 2 Cache Controller and memory, event monitor, and debug blocks.
7x4 AHB MAX	7x4 AHB MAX	Data Path	7x4 AHB Cross-Bar Switch
AIPSTZ-1 AIPSTZ-2	AHB to IP Bridge	Data Path	AHB to IP Bridge (TrustZone)
ASRC	Asynchronous Sample Rate Converter	Multimedia Peripherals	The Asynchronous Sample Rate Converter (ASRC) converts the sampling rate of a signal associated to an input clock into a signal associated to a different output clock.The ASRC supports concurrent sample rate conversion of up to 10 channels. The sample rate conversion of each channel is associated to a pair of incoming and outgoing sampling rates.The ASRC supports up to 3 sampling rate pairs.
AUDMUX	Digital Audio Mux	Multimedia Peripherals	The AUDMUX is a programmable interconnect for voice, audio, and synchronous data routing between host serial interfaces (for example, SSI1, SSI2, and SSI3) and peripheral serial interfaces (audio and voice codecs). The AUDMUX has seven ports with identical functionality and programming models. A desired connectivity is achieved by configuring two or more AUDMUX ports.

Table continues on the next page...

Table 1-1. Digital and Analog Blocks (continued)

Block Mnemonic	Block Name	Subsystem	Brief Description
CAMP-1 CAMP-2	Clock Amplifier	Clocks, Resets, and Power Control	Clock Amplifier
CCM GPC SRC SRPGC	Clock Control Module, General Power Controller, System Reset Controller, State Retention Power Gate Control	Clocks, Resets, and Power Control	These blocks are responsible for clock and reset distribution in the system, and also for the system power management. The blocks include four PLLs.
CSPI ECSPI-1 ECSPI-2	Configurable SPI	Connectivity Peripherals	Full-duplex enhanced Synchronous Serial Interface (SSI), with data rate up to 26/52Mbit/s (CSPI/ECSPI). It is configurable to support Master/Slave modes, four chip selects to support multiple peripherals.
CSU	Central Security Unit	Security	The Central Security Unit (CSU) is responsible for setting comprehensive security policy within the i.MX53 platform and for sharing security information between the various security blocks. The Security Control Registers (SCR) of the CSU are set during boot time by the HAB and are locked to prevent further writing.
DPLLC-1 DPLLC-2 DPLLC-3 DPLLC-4	Digital Phase-Locked Loop Controller	System Control Peripherals	This chapter describes DPLLC. The chapter is intended for a block driver software developer. It describes block-level operation and programming.
DVFSC DVFSP	Dynamic Voltage and Frequency Scaling for Core and Peripherals	System Control Peripherals	The DVFS/DVFSP allows simple dynamic voltage frequency scaling. The frequency of the clock and the voltage of the power domain can be changed on the fly while all blocks continue their normal operation.
EIM	External Interface Module	System Control Peripherals	The EIM handles the interface to devices external to the chip, including generation of chip selects, clock and control for external peripherals and memory. It provides asynchronous access to devices with SRAM-like interface and synchronous access to devices with Nor-Flash like or PSRAM like interface.
EPIT-1 EPIT-2	Enhanced Periodic Interrupt Timer	Timer Peripherals	Each EPIT is a 32-bit "set and forget" timer that starts counting after the EPIT is enabled by software. It is capable of providing precise interrupts at regular intervals with minimal processor intervention. It has a 12-bit prescaler for division of input clock frequency to get the required time setting for the interrupts to occur, and counter value can be programmed on the fly.

Table continues on the next page...

Table 1-1. Digital and Analog Blocks (continued)

Block Mnemonic	Block Name	Subsystem	Brief Description
ESAI	Enhanced Serial Audio Interface	Connectivity Peripherals	The Enhanced Serial Audio Interface (ESAI) provides a full-duplex serial port for serial communication with a variety of serial devices, including industry-standard codecs, SPDIF transceivers, and other processors. The ESAI consists of independent transmitter and receiver sections, each section with its own clock generator. All serial transfers are synchronized to a clock. Additional synchronization signals are used to delineate the word frames. The normal mode of operation is used to transfer data at a periodic rate, one word per period. The network mode is also intended for periodic transfers; however, it supports up to 32 words (time slots) per period. This mode can be used to build time division multiplexed (TDM) networks. In contrast, the on-demand mode is intended for non-periodic transfers of data and to transfer data serially at high speed when the data becomes available. The ESAI has 12 pins for data and clocking connection to external devices.
ESDCTL	Enhanced SDRAM Controller	Memory Control	The ESDCTL is a configurable high performance and optimized SDRAM controller that supports DDR2, DDR3, and so on.

Table continues on the next page...

Table 1-1. Digital and Analog Blocks (continued)

Block Mnemonic	Block Name	Subsystem	Brief Description
ESDHCV2-1 ESDHCV2-2 ESDHCV2-4	Enhanced Multi-Media Card / Secure Digital Host Controller	Connectivity Peripheral	i.MX53 specific SoC characteristics: • Ports 1, 2, and 4 are Compatible with the "MMC System Specification" version 4.3, full support. • In i.MX53 SoC, ports 1 and 2 are limited to 4-bit data width interface. The generic features of the eSDHC block (ESDHCv2), when serving as SD/MMC host, include the following: • Conforms to "SD Host Controller Standard Specification" version 2.0, full support. • Compatible with the SD Memory Card Specification version 1.1 • Compatible with the SDIO Card Specification version 1.2 • Designed to work with SD Memory, miniSD Memory, SDIO, miniSDIO, SD Combo, MMC and MMC RS cards • Configurable to work in one of the following modes: - SD/SDIO 1-bit, 4-bit - MMC 1-bit, 4-bit, 8-bit (possibly restricted per SoC integration) • Full/High speed mode • Host clock frequency variable between 32kHz to 52 MHz • Up to 200 Mbps data transfer for SD/SDIO cards using 4 parallel data lines • Up to 416 Mbps data transfer for MMC cards using 8 parallel data lines for MMC 4.3, and 832Mbps for eMMC 4.4 cards. • eSDHC-2 is limited to bus width of 4-bits, so not recommended for HD use. See separate section below. • Can be configured either as SD/MMC controller • Support eSD and eMMC standard, for SD/MMC embedded type cards
ESDHCV3-3 (EMMC 4.4)	Ultra-High-Speed eMMC/SD host controller.	Connectivity Peripherals	Ultra High-Speed eSDHC, enhanced to support eMMC 4.4 standard specification, for 832 MBps. Block is backward compatible to eSDHCv2. See complete features listing in eSDHCv2 entry below. i.MX53 SoC specific characteristics: • Port 3 - enhanced, to support eMMC 4.4 specification, for double data rate (832Mbps, 8-bit port).

Table continues on the next page...

Table 1-1. Digital and Analog Blocks (continued)

Block Mnemonic	Block Name	Subsystem	Brief Description
EXTMC	External Memory Controller	Connectivity Peripherals	The EXTMC is an external and internal memory interface. It performs arbitration between multi-AXI masters to multi-memory controllers, divided into four major channels, fast memories (DDR2/DDR3/LPDDR2) channel, slow memories (NOR-FLASH / PSRAM / NAND-FLASH etc.) channel, Internal Memory (RAM, ROM) channel and Graphical Memory (GMEM) channel. In order to increase the bandwidth performance, the EXTMC separates the buffering and the arbitration between different channels, preventing interference between slow and fast accesses. EXTMC Features: • 64-bit and 32-bit AXI ports • Enhanced arbitration scheme for fast channel, including dynamic master priority, and taking into account which pages are open or closed and what type (Read or Write) was the last access • Flexible bank interleaving • Support 16/32-bit DDR2-800 or DDR3-800 or LPDDR2 • Support up to 2 GByte DDR memories • Support NFC, EIM signal muxing scheme • Support 8/16/32-bit Nor-Flash/PSRAM memories (sync and async operating modes), at slow frequency, (8bit is not supported on D[23]-D[16]) • Support 4/8/14/16-bit ECC, page sizes of 512B, 2KB and 4KB Nand-Flash (including MLC) • Multiple chip selects (up to 4) • Enhanced DDR memory controller, supporting access latency hiding • Support Watermark for security (Internal and External memories)
FEC	Fast Ethernet Controller	Connectivity Peripherals	The Ethernet Media Access Controller (MAC) is designed to support both 10 and 100 Mbps Ethernet/IEEE 802.3 networks. An external transceiver interface and transceiver function are required to complete the interface to the media. The i.MX53 also consists of HW assist for IEEE1588 standard. Refer to IEEE1588 section for more details.
FIRI	Fast Infra-Red Interface	Connectivity Peripherals	Fast Infra-Red Interface
FLEXCAN-1 FLEXCAN-2	Flexible Controller Area Network	Connectivity Peripherals	The CAN protocol was primarily, but not only, designed to be used as a vehicle serial data bus, meeting the specific requirements of this field: real-time processing, reliable operation in the EXTMC environment of a vehicle, cost-effectiveness and required bandwidth. The FLEXCAN is a full implementation of the CAN protocol specification, Version 2.0 B, which supports both standard and extended message frames.
FUSEBOX	Electrical Fuse Array	Security	Electrical Fuse Array (splitted to banks). Enables to setup Boot Modes, Security Levels, Security Keys and many other system parameters.

Table continues on the next page...

Table 1-1. Digital and Analog Blocks (continued)

Block Mnemonic	Block Name	Subsystem	Brief Description
GPIO-1 GPIO-2 GPIO-3 GPIO-4 GPIO-5 GPIO-6 GPIO-7	General Purpose I/O	System Control Peripherals	Used for general purpose input/output to external ICs. Each GPIO block supports 32 bits of I/O. The AP side has four GPIO blocks supporting up to 128 GPIO functions.
GPT	General Purpose Timer	Timer Peripherals	Each GPT is a 32-bit "free-running" or "set and forget" mode timer with programmable prescaler and compare and capture register. A timer counter value can be captured using an external event and can be configured to trigger a capture event on either the leading or trailing edges of an input pulse. When the timer is configured to operate in "set and forget" mode, it is capable of providing precise interrupts at regular intervals with minimal processor intervention. The counter has output compare logic to provide the status and interrupt at comparison. This timer can be configured to run either on an external clock or on an internal clock.
GPU2D	Graphics Processing Unit-2D, ver 1	Multimedia Peripherals	The GPU provides hardware acceleration for 2D graphics algorithms with sufficient processor power to run desk-top quality interactive graphics applications on displays up to HD1080 resolution.
GPU3D	Graphics Processing Unit, ver.3	Multimedia Peripherals	The GPU provides hardware acceleration for 2D and 3D graphics algorithms with sufficient processor power to run desk-top quality interactive graphics applications on displays up to HD1080 resolution. It supports color representation up to 32 bits per pixel. GPUv3 enables High Performance Mobile 3D and 2D Vector Graphics at rates up to 33 M triangles/sec, 200 Mpixels/sec, 800 MPixels/sec (Z) (as per IP vendor).
I2C-1 I2C-2 I2C-3	I^2C Interface	Connectivity Peripherals	I^2C provide serial interface for external devices. Data rates of up to 400 kbps are supported.

Table continues on the next page...

Table 1-1. Digital and Analog Blocks (continued)

Block Mnemonic	Block Name	Subsystem	Brief Description
IIM	IC Identification Module	Security	The IC Identification Module (IIM) provides an interface for reading, programming and/or overriding identification and control information stored in on-chip fuse elements. The block supports electrically-programmable poly fuses (e-Fuses) or Laser Fuses (L-fuses). The IIM also provides a set of volatile software-accessible signals which can be used for software control of hardware elements, not requiring non-volatility. The IIM provides the primary user-visible mechanism for interfacing with on-chip fuse elements. Among the uses for the fuses are unique chip identifiers, mask revision numbers, cryptographic keys, JTAG secure mode, boot characteristics and various control signals requiring permanent non-volatility. The IIM also provides up to 28 volatile control signals. The IIM consists of a master controller, a software fuse value shadow cache, and a set of registers to hold the values of signals visible outside the block
IOMUXC	IOMUX Controller	System Control Peripherals	This block enables flexible IO multiplexing. Each IO pad has default and several alternate functions. The alternate functions are software configurable.
IPTP	IEEE1588 HW Assist to Ethernet controller (FEC).	Connectivity Peripherals	The IEEE 1588-2002 standard defines a Precision Time Protocol (PTP) - which is a time-transfer protocol that enables synchronization of networks (e.g., Ethernet), to a high degree of accuracy and precision. The IEEE1588 hardware assist, is composed of the two IPs - TSU (Time Stamp Unit) and CE_RTC (Real Time Clock), which provides the time stamping protocol's functionality. (generating / reading) the needed timestamps. The hardware-assisted implementations deliver more precise clock synchronization, at significantly lower CPU load, than pure SW implementation.
IPU	Image Processing Unit, ver.3M	Multimedia Peripherals	IPUv3M enables connectivity to displays, relevant processing and synchronization. It supports two display ports and two camera ports, through the following interfaces. • Legacy Parallel Interfaces • Single/dual channel LVDS display interface • Analog TV or VGA interfaces The processing includes • Image enhancement: color adjustment and gamut mapping, gamma correction and contrast enhancement • Video/graphics combining • Support for display backlight reduction • Image conversion-resizing, rotation, inversion and color space conversion • Hardware de-interlacing support • Synchronization and control capabilities, allowing autonomous operation.

Table continues on the next page...

Table 1-1. Digital and Analog Blocks (continued)

Block Mnemonic	Block Name	Subsystem	Brief Description
KPP	Key Pad Port	Connectivity Peripherals	KPP Supports 8 x 8 external key pad matrix. KPP features are: • Open drain design • Glitch suppression circuit design • Multiple keys detection • Standby key press detection
LDB	LVDS Display Bridge	Connectivity Peripherals	LVDS Display Bridge is used to connect the IPU (Image Processing Unit) to External LVDS Display Interface. LDB supports two channels; each channel has following signals: • 1 clock pair • 4 data pairs Each signal pair contains - LVDS special differential pad (PadP, PadM).
LDO	Low-Dropout Regulator	Clocks, Resets and Power Control	LDO is an integrated 1.8 V/1.2 V linear regulator.
M4IF	Multi Master Multi Memory Interface	Memory Control	M4IF controls memory accesses from one or more masters through different port interfaces to different external memory controllers ESDCTL, NFC, and EIM, and to some 2 internal memories in the system as well.
MLB	Media Local Bus Controller	Connectivity / Multimedia Peripherals	The MLB interface block provides a link to a MOST® data network, using the standardized MediaLB protocol (up to 50 Mbps)
NFC	NAND Flash Controller	Memory Control	NFC is composed of various control logic units, a 4.5-Kbyte internal RAM and an internal ECC engine. The NFC can interface standard NAND Flash memory devices.
OCRAM	On-Chip Memory controller	Data Path	The On-Chip Memory controller (OCRAM) block, is designed as an interface between the system's AXI bus, to the internal (on-chip) SRAM memory block. In i.MX53, the OCRAM is used for controlling the 128KB multimedia RAM, via a 64-bit AXI bus.
OWIRE	One-Wire Interface	Connectivity Peripherals	OWIRE support provided for interfacing with an on board EEPROM, and smart battery interfaces, for example: Dallas DS2502
PATA	Parallel ATA	Connectivity Peripherals	The P-ATA block is a AT attachment host interface. Its main use is to interface with hard disc drives and optical disc drives. It interfaces with the ATA-6 compliant device over a number of ATA signals. It is possible to connect a bus buffer between the host side and the device side.
PFD	Power Fail Detector	Clocks, Resets, and Power Control	PFD circuit generates a reset signal till its supply vdd reaches a predetermined voltage level.The PFD circuit works only when the chip is in the Non-DSM mode
PLARB2 PLARB1	PL301 Arbiter (2x2) and (4x1)	Data Path	ARM (Ltd.) PL301 AXI arbiters to bridge between several AXI masters accessing several AXI slaves.

Table continues on the next page...

Table 1-1. Digital and Analog Blocks (continued)

Block Mnemonic	Block Name	Subsystem	Brief Description
POR	Power on Reset	Clocks, Resets, and Power Control	The POR is designed for use in SoC applications which require low voltage, low power consumption. POR circuit generates a reset signal till its supply VDD reaches a predetermined voltage level.
PWM-1 PWM-2	Pulse Width Modulation	Connectivity Peripherals	The pulse-width modulator (PWM) has a 16-bit counter and is optimized to generate sound from stored sample audio images and it can also generate tones. It uses 16-bit resolution and a 4x16 data FIFO to generate sound.
ROMC	ROM Controller with Patch	Data Path	ROM Controller with ROM Patch support
RTIC	Run Time Integrity Checker	Security	Protecting read only data from modification is one of the basic elements in trusted platforms. The Run-Time Integrity Checker (RTIC), is a data monitoring device responsible for ensuring that memory content is not corrupted during program execution. The RTICv3 mechanism periodically checks the integrity of code or data sections during normal OS run-time execution without interfering with normal operation. The purpose of the RTIC is to ensure the integrity of the peripheral memory contents, protect against unauthorized external memory elements replacement and assist with boot authentication.
SAHARA	SAHARA security accelerator	Security	SAHARA (Symmetric/Asymmetric Hashing and Random Accelerator) version 4 is a security coprocessor. It implements symmetric encryption algorithms, (AES, DES, 3DES, RC4 and C2), hashing algorithms (MD5, SHA-1, SHA-224 and SHA-256), and a hardware true random number generator. It has a slave IPBus interface for the host to write configuration and command information, and to read status information. It also has a DMA controller, with an AHB bus interface, to reduce the burden on the host to move the required data to and from memory.
SATA	Serial ATA	Connectivity Peripherals	The SATA controller and PHY is a complete mixed-signal block solution designed to implement SATA HDD connectivity in a i.MX53 design.
SCC	Security Controller	Security	The SCC-AES is the second generation of the Security Controller. It implements secure RAM that can be used either as general-purpose memory for storing data and software, or as special confidentiality-preserving memory that protects disclosure-sensitive data such as cryptographic keys, passwords, code, or PIN numbers. It also incorporates cryptographic logic and a DMA engine that can be used to safely export the data stored within a partition to external RAM or non-volatile memory

Table continues on the next page...

Table 1-1. Digital and Analog Blocks (continued)

Block Mnemonic	Block Name	Subsystem	Brief Description
SDMA	Smart Direct Memory Access	System Control Peripherals	The SDMA is multi-channel flexible DMA engine. It helps in maximizing system performance by off-loading the various cores in dynamic data routing. Features List: • Powered by a 16-bit Instruction-Set micro-RISC engine • Multi-channel DMA supporting up to 32 time-division multiplexed DMA channels • 48 events with total flexibility to trigger any combination of channels • Memory accesses including linear, FIFO, and 2D addressing • Shared peripherals betweenARM Cortex A8™ and SDMA • Very fast Context-Switching with 2-level priority based preemptive multi-tasking • DMA units with auto-flush and prefetch capability • Flexible address management for DMA transfers (increment, decrement, and no address changes on source and destination address) • DMA ports can handle unidirectional and bidirectional flows (copy mode) • Up to 8-word buffer for configurable burst transfers for EXTMC • Support of byte-swapping and CRC calculations • Library of Scripts and API is available
SJC	System JTAG Interface	System Control Peripherals	JTAG manipulation is one of the known hackers' ways of executing unauthorized program code, getting control over secure applications and running code in privileged modes. The JTAG port provides a debug access to several hardware blocks including the ARM processor and the system bus. The JTAG port must be accessible during platform initial laboratory bring-up, manufacturing tests and troubleshooting, as well as for software debugging by authorized entities. However, in order to properly secure the system, unauthorized JTAG usage should be strictly forbidden. In order to prevent JTAG manipulation while allowing access for manufacturing tests and software debugging, i.MX53 incorporates a mechanism for regulating JTAG access. i.MX53 System JTAG Controller provides four different JTAG security modes that can be selected via e-fuse configuration.
SPBA	Shared Peripheral Bus Arbiter	System Control Peripherals	SPBA (Shared Peripheral Bus Arbiter) is a three-to-one IP Bus interfaces (IP Bus) arbiter.
SPDIF	Sony Phillips Digital Interface	Multimedia Peripherals	A standard audio file transfer format. Developed jointly by the Sony and Phillips corporations. Transmitter and Receiver functionality.

Table continues on the next page...

Table 1-1. Digital and Analog Blocks (continued)

Block Mnemonic	Block Name	Subsystem	Brief Description
SRTC	Secure Real Time Clock	Security	The SRTC incorporates a special System State Retention Register (SSRR) that stores system parameters during system shut down modes. This register (as all SRTC counters) is power backed up by a coin-cell backup battery. This register is helpful for storing warm boot parameters. The SSRR also stores the system security state. In case of a security violation, the SSRR will mark the event (security violation indication).
SSI-1 SSI-2 SSI-3	I2S/SSI/AC97 Interface	Connectivity Peripherals	The SSI is a full-duplex synchronous interface which is used on the AP to provide connectivity with off-chip audio peripherals. The SSI supports a wide variety of protocols (SSI normal, SSI network, I2S, and AC-97), bit depths (up to 24 bits per word), and clock / frame sync options. The SSI has two pairs of 8x24 FIFOs and hardware support for an external DMA controller in order to minimize its impact on system performance. The second pair of FIFOs provides hardware interleaving of a second audio stream which reduces CPU overhead in use cases where two time slots are being used simultaneously.
TVE	TV-Encoder Ver 2.1	Multimedia	The TVEv2.1 which consists of the Digital Video Encoder (DVE) and a Triple Video Digital-to-Analog Converter (TVDAC); supports HD720p/1080p, PAL/NTSC or VGA output for direct connection to TV or LCD projector
TZIC	TrustZone Aware Interrupt Controller	ARM/Control	The TrustZone Interrupt Controller (TZIC) collects interrupt requests from all i.MX53 sources and routes them to the ARM core. Each interrupt can be configured as a normal or a secure interrupt. Software Force Registers and software Priority Masking are also supported.
UART-1 UART-2 UART-3 UART-4 UART-5	UART Interface	Connectivity Peripherals	Each of the UART supports the following serial data transmit/ receive protocols and configurations: • 7 or 8 bit data words, 1 or 2 stop bits, programmable parity (even, odd or none) • Programmable baud rates up to 4 MHz. This is a higher max baud rate relative to the 1.875 MHz which is stated by the TIA/EIA-232-F standard and the i.MX53 UART. • 32-byte FIFO on Tx and 32 half-word FIFO on Rx supporting auto-baud • IrDA 1.0 support (up to SIR speed of 115200 bps) • Option to operate as 8-pins full UART, DCE or DTE.

Table continues on the next page...

Table 1-1. Digital and Analog Blocks (continued)

Block Mnemonic	Block Name	Subsystem	Brief Description
USB	USB 2.0 High Speed OTG and 3x HS Hosts	Connectivity Peripherals	USBO HS (2.0 480MHz) contains one hi-speed OTG block, which is internally connected to the HS USB PHY, while still equipped with Transceiver-Less Logic to enable on-board USB connectivity without USB TransceiversUSBOH3 contains: • One high-speed OTG block with integrated HS USB PHY • One high-speed Host block with integrated HS USB PHY • Two identical high-speed Host blocks All the USB ports are equipped with standard digital interfaces (ULPI, HS IC-USB) and Transceiver-Less Logic to enable on-board USB connectivity without USB Transceivers.
VPU	Video Processing Unit	Multimedia Peripherals	A high-performing video processing unit (VPU), which covers many SD-level and HD-level video decoders and SD-level encoders as a multi-standard video codec engine as well as several important video processing such as rotation and mirroring. Refer to Table 9-2 for complete list of VPU's decoding/ encoding capabilities.
WDOG-1	Watch Dog	Timer Peripherals	The Watch Dog Timer supports 2 comparison points during each counting period. Each of the comparison points is configurable to evoke an interrupt to the ARM core, and a second point evokes an external event on the WDOG line.
XTALOSC XTALOSC_32K	Crystal Oscillator I/ F	Clocking	The XTALOSC is 24 MHz crystal oscillator and XTALOSC_32K is a 32.768 kHz crystal oscillator.
RAM 128 KB	Internal RAM	Internal Memory	Internal RAM, shared with VPU
RAM 16 KB	Secure/non-secure RAM	Secured Internal Memory	Secure/non-secure Internal RAM, controlled by SCC
ROM 64 KB	Boot ROM	Internal Memory	Supports secure and regular Boot Modes

1.7 Memory Interfaces

i.MX53 EXTMC supports the following memory interfaces:

- DDR2/LV-DDR2-800, 16/32 -bit, 400 MHz clock
- DDR3-800, 16/32 -bit, 400 MHz clock
- LPDDR2, 32bit on PoP package only, 400 MHz clock

- NAND (MLC/SLC) Flash, 8-bit/16-bit
- NOR Flash, SRAM and PSRAM, 8/16/32-bit (8-bit is not supported at byte D[23]-D[16])

Chapter 2
Memory Map

2.1 ARM Platform System Memory Map

The table below shows the system memory map.

Table 2-1. System Memory Map

ARM Platform		Size (bytes)	Region
Start Address	End Address		
On Chip Memories [HW connection via External Memory Interface (EXTMC)]			
0000_0000	0000_FFFF	64K	Boot ROM
0001_0000	00FF_FFFF	16M-64K	Boot ROM Aliasing
0100_0000	06FF_FFFF	96M	Reserved
0700_0000	0700_3FFF	16K	Security Controller RAM
0700_4000	07FF_FFFF	16M-16K	SCC RAM Aliasing
0800_0000	0FFF_BFFF	128M	Reserved
0FFF_C000	0FFF_FFFF	16K	Trust Zone Aware Interrupt Control (TZIC)
1000_0000	1000_3FFF	16K	Serial ATA (SATA)
1000_4000	13FF_FFFF	64M-16K	SATA Aliasing
1400_0000	17FF_FFFF	64M	Reserved
1800_0000	1FFF_FFFF	128M	Image Processing Unit (IPU)
2000_0000	2FFF_FFFF	256M	2D Graphics Processing Unit (GPU2D)
3000_0000	3FFF_FFFF	256M	3D Graphics Processing Unit (GPU3D)
On Chip AHB Accessed IPs-Debug APB			
4000_0000	4000_0FFF	4K	Debug ROM
4000_1000	4000_1FFF	4K	ETB
4000_2000	4000_2FFF	4K	ETM
4000_3000	4000_3FFF	4K	TPIU
4000_4000	4000_4FFF	4K	CTI0

Table continues on the next page...

Table 2-1. System Memory Map (continued)

ARM Platform		Size (bytes)	Region
Start Address	End Address		
4000_5000	4000_5FFF	4K	CTI1
4000_6000	4000_6FFF	4K	CTI2
4000_7000	4000_7FFF	4K	CTI3
4000_8000	4000_8FFF	4K	ARM Debug Unit
4000_9000	4FFF_FFFF	256M-36K	Reserved
AIPSTZ-1			
AIPSTZ-1- SPBA IPs, Mapped to global block enable 0			
5000_0000	5000_3FFF	16K	Reserved
5000_4000	5000_7FFF	16K	ESDHC1
5000_8000	5000_BFFF	16K	ESDHC2
5000_C000	5000_FFFF	16K	UART-3
5001_0000	5001_3FFF	16K	ECSPI-1
5001_4000	5001_7FFF	16K	SSI-2
5001_8000	5001_BFFF	16K	ESAI-1
5001_C000	5001_FFFF	16K	Reserved for Smart Direct Memory Access (SDMA) internal registers
5002_0000	5002_3FFF	16K	ESDHCV3-3
5002_4000	5002_7FFF	16K	ESDHCV2-4
5002_8000	5002_BFFF	16K	SPDIF
5002_C000	5002_FFFF	16K	Asynchronous Sample Rate Converter (ASRC)
5003_0000	5003_3FFF	16K	Parallel ATA (PATA) (PORT UDMA)
5003_4000	5003_7FFF	16K	Reserved
5003_8000	5003_BFFF	16K	Reserved
5003_C000	5003_FFFF	16K	SPBA
AIPSTZ-1- Global Module Enables			
5004_0000	51FF_FFFF	32M (minus 256K)	Reserved AIPSTZ-1 off platform global module enable #0
5200_0000	53EF_FFFF	31M	Reserved AIPSTZ-1 off platform global module enable #1
AIPSTZ-1- On Platform			
53F0_0000	53F7_FFFF	512K	Reserved AIPSTZ-1 on platform slots
AIPSTZ-1- Off Platform			
53F8_0000	53F8_3FFF	16K	USB 2.0 High Speed OTG and 3x HS Hosts (USB) (PORT USB)
53F8_4000	53F8_7FFF	16K	GPIO-1

Table continues on the next page...

Table 2-1. System Memory Map (continued)

ARM Platform		Size (bytes)	Region
Start Address	End Address		
53F8_8000	53F8_BFFF	16K	GPIO-2
53F8_C000	53F8_FFFF	16K	GPIO-3
53F9_0000	53F9_3FFF	16K	GPIO-4
53F9_4000	53F9_7FFF	16K	Key Pad Port (KPP)
53F9_8000	53F9_BFFF	16K	WDOG1
53F9_C000	53F9_FFFF	16K	WDOG2
53FA_0000	53FA_3FFF	16K	General Purpose Timer (GPT)
53FA_4000	53FA_7FFF	16K	Secure Real Time Clock (SRTC)
53FA_8000	53FA_BFFF	16K	IOMUX Control (IOMUXC)
53FA_C000	53FA_FFFF	16K	EPIT-1
53FB_0000	53FB_3FFF	16K	EPIT-2
53FB_4000	53FB_7FFF	16K	PWM-1
53FB_8000	53FB_BFFF	16K	PWM-2
53FB_C000	53FB_FFFF	16K	UART-1
53FC_0000	53FC_3FFF	16K	UART-2
53FC_4000	53FC_7FFF	16K	USB (PORT PL301)
53FC_8000	53FC_BFFF	16K	FLEXCAN-1
53FC_C000	53FC_FFFF	16K	FLEXCAN-2
53FD_0000	53FD_3FFF	16K	System Reset Controller (SRC)
53FD_4000	53FD_7FFF	16K	Clock Control Module (CCM)
53FD_8000	53FD_BFFF	16K	Global Power Controller (GPC)
53FD_C000	53FD_FFFF	16K	GPIO-5
53FE_0000	53FE_3FFF	16K	GPIO-6
53FE_4000	53FE_7FFF	16K	GPIO-7
53FE_8000	53FE_BFFF	16K	PATA (PORT PIO)
53FE_C000	53FE_FFFF	16K	I2C-3
53FF_0000	53FF_3FFF	16K	UART-4
53FF_4000	53FF_FFFF	48K	Reserved AIPSTZ-1 off platform space.
5400_0000	5FFF_FFFF	448M	Reserved (Aliased to AIPSTZ-1 slots)
AIPSTZ-2- Global Module Enables			
6000_0000	61FF_FFFF	32M	Reserved AIPSTZ-1 off platform global module enable #0
6200_0000	63EF_FFFF	31M	Reserved AIPSTZ-1 off platform global module enable #1
AIPSTZ-2- On Platform			

Table continues on the next page...

Table 2-1. System Memory Map (continued)

ARM Platform		Size (bytes)	Region
Start Address	End Address		
63F0_0000	63F7_FFFF	512K	Reserved AIPSTZ-2 on platform slots
AIPSTZ-2- Off Platform			
63F8_0000	63F8_3FFF	16K	DPLLC-1
63F8_4000	63F8_7FFF	16K	DPLLC-2
63F8_8000	63F8_BFFF	16K	DPLLC-3
63F8_C000	63F8_FFFF	16K	DPLLC-4
63F9_0000	63F9_3FFF	16K	UART-5
63F9_4000	63F9_7FFF	16K	AHBMAX
63F9_8000	63F9_BFFF	16K	IC Identification Module (IIM)
63F9_C000	63F9_FFFF	16K	Central Security Unit (CSU)
63FA_0000	63FA_3FFF	16K	ARM Platform
63FA_4000	63FA_7FFF	16K	One-Wire Interface (OWIRE)
63FA_8000	63FA_BFFF	16K	Fast Infrared Interface (FIRI)
63FA_C000	63FA_FFFF	16K	ECSPI-2
63FB_0000	63FB_3FFF	16K	SDMA (port IPS_HOST)
63FB_4000	63FB_7FFF	16K	SCC
63FB_8000	63FB_BFFF	16K	ROM Controller with Patch (ROMC)
63FB_C000	63FB_FFFF	16K	Real Time Integrity Checker, ver.2 (RTIC)
63FC_0000	63FC_3FFF	16K	Configurable SPI (CSPI)
63FC_4000	63FC_7FFF	16K	I2C-2
63FC_8000	63FC_BFFF	16K	I2C-1
63FC_C000	63FC_FFFF	16K	SSI-1
63FD_0000	63FD_3FFF	16K	Digital Audio Multiplexer (AUDMUX)
63FD_4000	63FD_7FFF	16K	RTC
63FD_8000	63FD_BFFF	16K	EXTMC (mapped block's register base address) 0x63FD_8000 - M4IF 0x63FD_9000 - ESDCTL 0x63FD_A000 - EIM 0x63FD_B000 - NFC 0x63FD_BF00 - EXTMC
63FD_C000	63FD_FFFF	16K	apb2ip_pl301_2x2
63FE_0000	63FE_3FFF	16K	apb2ip_pl301_4x1
63FE_4000	63FE_7FFF	16K	MLB
63FE_8000	63FE_BFFF	16K	SSI-3

Table continues on the next page...

Table 2-1. System Memory Map (continued)

ARM Platform		Size (bytes)	Region
Start Address	End Address		
63FE_C000	63FE_FFFF	16K	Fast Ethernet Controller (FEC)
63FF_0000	63FF_3FFF	16K	TV-Encoder Ver 2.1 (TVE)
63FF_4000	63FF_7FFF	16K	Video Processing Unit (VPU)
63FF_8000	63FF_BFFF	16K	SAHARA
63FF_C000	63FF_FFFF	16K	PTP
6400_0000	6FFF_BFFF	256M (minus 512K)	Reserved (aliased slots)
On Chip AHB Accessed IPs			
6FFF_C000	6FFF_FFFF	16K	Reserved
Off Chip Memories (HW connection via EXTMC)			
7000_0000	AFFF_FFFF	1G	CSD0 DDR
B000_0000	EFFF_FFFF	1G	CSD1 DDR
F000_0000 F000_0000 F400_0000	F7FE_FFFF F3FF_FFFF F7FE_FFFF	128M-64K 64M 64M-64k	CS0 (128M NOR/SRAM) - Default Configuration CS1, CS2, CS3 - Not Active. In case of 2 active CS. CS0(64M) CS1(64M) Via IOMUXC (GPRR1) register, 4CS of 32M are supported. The sum of all CS spaces must equal 128M, and can be splitted between maximum 4 CSs. The biggest region is CS0. No holes are supported. The supported configurations are: CS0(128M), CS1 (0M), CS2 (0M), CS3(0M) CS0(64M), CS1(64M), CS2(0M), CS3(0M) CS0(64M), CS1(32M), CS2(32M), CS3(0M) CS0(32M), CS1(32M), CS2(32M), CS3(32M)
On Chip Memories (HW connection via EXTMC)			
F7FF_0000	F7FF_FFFF	64K	NAND FLASH (internal buffer)
F800_0000	F801_FFFF	128K	iRAM (OCRAM)
F802_0000	F805_FFFF	256K	GPU3D GMEM
F806_0000	FFFF_FFFF	128M-384K	Reserved

NOTE

User should not address reserved memory regions. Access to reserved memory regions can cause unpredictable behavior.

2.2 DMA Memory Map

The Smart DMA memory map is defined in Table 2-2.

Table 2-2. SDMA Peripheral Memory Map

Peripheral	Base Address	Size	Comments
Reserved for SDMA internal memory	0x0000	4KB	Reserved
ESDHCv2-1	0x1000	4KB	
ESDHCv2-2	0x2000	4KB	
UART-3	0x3000	4KB	
ECSPI-1	0x4000	4KB	
SSI-2	0x5000	4KB	
Enhanced Serial Audio Interface (ESAI)	0x6000	4KB	
Reserved for SDMA internal registers	0x7000	4KB	Reserved
ESDHCv2-3 (CE-ATA)	0x8000	4KB	
ESDHCv2-4	0x9000	4KB	
SPDIF	0xA000	4KB	
ASRC	0xB000	4KB	
PATA	0xC000	4KB	
Reserved	0xD000	4KB	Reserved
Reserved	0xE000	4KB	Reserved
SPBA Registers	0xF000	4KB	

NOTE

User should not address reserved memory regions. Access to reserved memory regions can cause unpredictable behavior.

Chapter 3
Interrupts and SDMA Events

3.1 Overview

The Interrupts and SDMA Events chapter provides information on the assignments of the interrupts of the ARM platform domain in ARM Platform Interrupts and of the DMA events in SDMA Event Mapping.

3.2 ARM Platform Interrupts

The TrustZone Aware Interrupt Controller (TZIC) collects up to 128 interrupt requests from all MCIMX53 sources and provides an interface to the core. Each interrupt can be configured as a normal or a secure interrupt. Software force registers and software priority masking are also supported. Table 3-1 details the ARM Cortex A8^{tm} Platform (ARM) interrupt sources.

Table 3-1. ARM Domain Interrupt Summary

IRQ	Interrupt Source	Interrupt Description
0	Reserved	Reserved
1	ESDHCV2-1	Enhanced SDHC Interrupt Request
2	ESDHCV2-2	Enhanced SDHC Interrupt Request
3	ESDHCV3-3	CE-ATA Interrupt Request based on ESDHCV3-3
4	ESDHCV2-4	Enhanced SDHC Interrupt Request
5	DAP	
6	SDMA	AND of all 48 interrupts from all the channels
7	IOMUXC	POWER FAIL interrupt. This is a power fail indicator interrupt from on board power management IC via GPIO_16 PAD on ALT2, PWRFAIL_INT signal.

Table continues on the next page...

Table 3-1. ARM Domain Interrupt Summary (continued)

IRQ	Interrupt Source	Interrupt Description
8	EXTMC	NFC interrupt
9	VPU	VPU Interrupt Request
10	IPU	IPU Error Interrupt
11	IPU	IPU Sync Interrupt
12	GPU3D	GPU Interrupt Request
13	UART-4	UART-4 ORed interrupt
14	USB	USB Host 1
15	EXTMC	Consolidated EXTMC Interrupt
16	USB	USB Host 2
17	USB	USB Host 3
18	USB	USB OTG
19	SAHARA	SAHARA Interrupt for Host 0
20	SAHARA	SAHARA Intr for Host 1
21	SCC	Security Monitor High Priority Interrupt Request.
22	SCC	Secure (TrustZone) Interrupt Request.
23	SCC	Regular (Non-Secure) Interrupt Request.
24	SRTC	SRTC Consolidated Interrupt. Non TZ.
25	SRTC	SRTC Security Interrupt. TZ.
26	RTIC	RTIC (Trust Zone) Interrupt Request. Indicates that the RTIC has completed hashing the selected memory block(s) during single-hash/boot mode.
27	CSU	CSU Interrupt Request 1. Indicates to the processor that one or more alarm inputs were asserted
28	SATA	SATA interrupt request
29	SSI-1	SSI-1 Interrupt Request
30	SSI-2	SSI-2 Interrupt Request
31	UART-1	UART-1 ORed interrupt
32	UART-2	UART-2 ORed interrupt
33	UART-3	UART-3 ORed interrupt
34	IPTP	RTC (IEEE1588) interrupt request
35	IPTP	PTP (IEEE1588) interrupt request
36	ECSPI-1	ECSPI-1 interrupt request line to the core.
37	ECSPI-2	ECSPI-2 interrupt request line to the core.
38	CSPI	CSPI interrupt request line to the core.

Table continues on the next page...

Table 3-1. ARM Domain Interrupt Summary (continued)

IRQ	Interrupt Source	Interrupt Description
39	GPT	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1, 2 &3 Interrupt lines
40	EPIT-1	EPIT-1 output compare interrupt
41	EPIT-2	EPIT-2 output compare interrupt
42	GPIO-1	Active HIGH Interrupt from INT7 from GPIO
43	GPIO-1	Active HIGH Interrupt from INT6 from GPIO
44	GPIO-1	Active HIGH Interrupt from INT5 from GPIO
45	GPIO-1	Active HIGH Interrupt from INT4 from GPIO
46	GPIO-1	Active HIGH Interrupt from INT3 from GPIO
47	GPIO-1	Active HIGH Interrupt from INT2 from GPIO
48	GPIO-1	Active HIGH Interrupt from INT1 from GPIO
49	GPIO-1	Active HIGH Interrupt from INT0 from GPIO
50	GPIO-1	Combined interrupt indication for GPIO-1 signal 0 throughout 15
51	GPIO-1	Combined interrupt indication for GPIO-1 signal 16 throughout 31
52	GPIO-2	Combined interrupt indication for GPIO-2 signal 0 throughout 15
53	GPIO-2	Combined interrupt indication for GPIO-2 signal 16 throughout 31
54	GPIO-3	Combined interrupt indication for GPIO-3 signal 0 throughout 15
55	GPIO-3	Combined interrupt indication for GPIO-3 signal 16 throughout 31
56	GPIO-4	Combined interrupt indication for GPIO-4 signal 0 throughout 15
57	GPIO-4	Combined interrupt indication for GPIO-4 signal 16 throughout 31
58	WDOG-1	Watchdog Timer reset
59	WDOG-2	TrustZone Watchdog Timer reset
60	KPP	Keypad Interrupt
61	PWM-1	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line.
62	I2C-1	I2C-1 Interrupt
63	I2C-2	I2C-2 Interrupt
64	I2C-3	I2C-3 Interrupt
65	MLB	NOR of all interrupts, mlb_cint and mlb_sint

Table continues on the next page...

Table 3-1. ARM Domain Interrupt Summary (continued)

IRQ	Interrupt Source	Interrupt Description
66	ASRC	ASRC Interrupt for core 1
67	SPDIF	SPDIF Tx interrupt OR SPDIF Rx interrupt
68	Reserved	Reserved
69	IIM	Interrupt request to the processor. Indicates to the processor that program or explicit sense cycle is completed successfully or in case of error. This signal is low-asserted.
70	PATA	Parallel ATA host controller interrupt request
71	CCM	CCM, Interrupt Request 1
72	CCM	CCM, Interrupt Request 2
73	GPC	GPC, Interrupt Request 1
74	GPC	GPC, Interrupt Request 2
75	SRC	SRC interrupt request
76	P_PLATFORM_NE_32K_256K	Neon Monitor Interrupt
77	P_PLATFORM_NE_32K_256K	Performance Unit Interrupt (nPMUIRQ). This is an interrupt generated by the ARMCORE and used for system profiling and debug. GPIO_16 PAD in ALT3 mode acts as nPMUIRQ signal (also named PMU_IRQ_B signal).
78	P_PLATFORM_NE_32K_256K	CTI IRQ
79	P_PLATFORM_NE_32K_256K	Debug Interrupt, from Cross-Trigger 1 Interface 1
80	P_PLATFORM_NE_32K_256K	Debug Interrupt, from Cross-Trigger 1 Interface 0
81	ESAI	ESAI interrupt
82	FLEXCAN-1	NOR of all interrupts; ipi_int_mbor, ipi_int_wakein, ipi_int_busoff and ipi_int_error.
83	FLEXCAN-2	NOR of all interrupts; ipi_int_mbor, ipi_int_wakein, ipi_int_busoff and ipi_int_error.
84	OPENVG	General Interrupt
85	OPENVG	Busy signal (for S/W power gating feasibility)
86	UART-5	UART-5 ORed interrupt
87	FEC	Fast Interrupt Request (OR of 13 interrupt sources)
88	OWIRE	1-Wire Interrupt Request
89	P_PLATFORM_NE_32K_256K	Debug Interrupt, from Cross-Trigger 1 Interface 2
90	SJC	
91	Reserved	Reserved
92	TVE	
93	FIRI	FIRI Intr (OR of all 4 interrupt sources)

Table continues on the next page...

Table 3-1. ARM Domain Interrupt Summary (continued)

IRQ	Interrupt Source	Interrupt Description
94	PWM-2	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line.
95	Reserved	Reserved for SLM
96	SSI-3	SSI-3 Interrupt Request
97	Reserved	
98	P_PLATFORM_NE_32K_256K	Debug Interrupt, from Cross-Trigger 1 Interface 3
99	Reserved	Was belong to SLM
100	VPU	Idle interrupt from VPU (for S/W power gating)
101	EXTMC	Indicates all pages have been transferred to NFC during an auto_prog operation
102	GPU3D	Idle interrupt from GPU (for S/W power gating)
103	GPIO-5	Combined interrupt indication for GPIO-5 signal 0 throughout 15
104	GPIO-5	Combined interrupt indication for GPIO-5 signal 16 throughout 31
105	GPIO-6	Combined interrupt indication for GPIO-6 signal 0 throughout 15
106	GPIO-6	Combined interrupt indication for GPIO-6 signal 16 throughout 31
107	GPIO-7	Combined interrupt indication for GPIO-7 signal 0 throughout 15
108	GPIO-7	Combined interrupt indication for GPIO-7 signal 16 throughout 31
109_128	Reserved	Reserved

3.3 SDMA Event Mapping

Table 3-2 shows the DMA request signals for peripherals in i.MX53.

Table 3-2. SDMA Event Mapping

Event Number	DMA Source	Description
0	VPU	VPU DMA request
1	GPC	Will be used for power management.
2	UART-4 PATA	UART-4RX muxed with PATA RX (selector IOMUXC GPR0 register bit [7]).

Table continues on the next page...

Table 3-2. SDMA Event Mapping (continued)

Event Number	DMA Source	Description
3	UART-4 PATA	UART-4TX muxed with PATA TX (selector IOMUXC GPR0 register bit [8]).
4	PATA	PATA Transfer End
5	IPU	IPU DMA Event
6	ECSPI	DMA Rx request
7	ECSPI-1	DMA Tx request
8	ECSPI-2	DMA Rx request
9	ECSPI-2	DMA Tx request
10	I2C-3 ESDHCV3-3	I2C-3 muxed with ESDHCV3-3
11	ESDHCV2-4 CTI2	ESDHC4 muxed with CTI2 (SDMA_CTI) trigger_out[0] connected to SDMA event.
12	UART-2 FIRI	UART-2RX muxed with FIRI REQ[0] (selector IOMUXC GPR0 register bit [9]).
13	UART-2 FIRI	UART-2TX muxed with FIRI REQ[1] (selector IOMUXC GPR0 register bit [10]).
14	SPDIF IOMUXC	SPDIF RX DMA request Muxed with External DMA request #0 from PAD DISP0_DAT16 or GPIO_17 (using daisy chain selector). The event selector is in IOMUXC GPR0 register bit [4].
15	SPDIF	SPDIF TX DMA request
16	UART-5	Rx FIFO of UART-5
17	UART-5	Tx FIFO of UART-5
18	UART-1	Rx FIFO of UART-1
19	UART-1	Tx FIFO of UART-1
20	I2C-1 ESDHCV2-1	I2C-1 muxed with ESDHCV2-1
21	I2C-2 ESDHCV2-2	I2C-2 muxed with ESDHCV2-2
22	SSI-2	SSI-2 receive 2 DMA request
23	SSI-2	SSI-2 transmit 2 DMA request
24	SSI-2	SSI-2 receive 1 DMA request
25	SSI-2	SSI-2 transmit 1 DMA request
26	SSI-1	SSI-1 receive 2 DMA request
27	SSI-1	SSI-1 transmit 2 DMA request
28	SSI-1	SSI-1 receive 1 DMA request
29	SSI-1	SSI-1 transmit 1 DMA request
30	EXTMC	Asserts every time NFC finishes reading a page
31	EXTMC	Asserts at the beginning of auto-program sequence, and every time the NFC finishes transferring data from the RAM to the NAND (Meaning, the SDMA can write to the RAM the next page).

Table continues on the next page...

Table 3-2. SDMA Event Mapping (continued)

Event Number	DMA Source	Description
32	ASRC	ASRC dma1 request (Pair A input Request)
33	ASRC	ASRC dma2 request (Pair B input Request)
34	ASRC	ASRC dma3 request (Pair C input Request)
35	ASRC	ASRC dma4 request (Pair A output Request)
36	ASRC	ASRC dma5 request (Pair B output Request)
37	ASRC	ASRC dma6 request (Pair C output Request)
38	CSPI EPIT-2	CSPI DMA Rx request Muxed with EPIT-2 DMA request
39	CSPI IOMUXC	CSPI DMA Tx request Muxed with External DMA request #1 from PAD DISP0_DAT17 or GPIO_18 (using daisy chain selector). The event selector is in IOMUXC GPR0 register bit [6].
40	ESAI	ESAI Rx FIFO DMA request
41	ESAI	ESAI Tx FIFO DMA request
42	UART-3	Rx FIFO of UART-3
43	UART-3	Tx FIFO of UART-3
44	SSI-3	SSI-3 receive 2 DMA request
45	SSI-3	SSI-3 transmit 2 DMA request
46	SSI-3	SSI-3 receive 1 DMA request
47	SSI-3	SSI-3 transmit 1 DMA request

As shown in the table, some of the events are shared through a multiplexer. The select of shared DMA event sources is controlled by DMAREQ_MUX_SELn fields of the IOMUXC.IOMUXC_GPRO Register.

For other shared connectivity peripherals that do not have dedicated DMA request signals, the ARM platform interrupt service routines have the option to program the SDMA to move data between the peripheral and memory.

Chapter 4
External Signals and Pin Multiplexing

4.1 Overview

Internal signals are connected to external components via package-level electrical connections. In this document, these physical connections will be referred to as pins even though in specific products they may be implemented as solder balls or some other package specific means. Pins, in turn, are connected internally to SoC driver / receiver circuitry called pads.

The i.MX53 contains a number of functional blocks that present input / output signals (block I/Os) that are suitable for connection to components external to the SoC. There are many more block I/Os than there are package pins. To provide flexibility in routing block I/Os to external pins, the Input-Output Multiplexer Controller (IOMUXC) block provides a number of software configurable multiplexers that allow different internal block I/O signals to be routed to the available pins.

In addition to this function, IOMUXC allows software to configure pad electrical characteristics, such as output/output function, voltage level, drive strength, and hysteresis. Not all voltage and ground pins are considered here. For information on these pins, consult the appropriate data sheet for the specific product.

4.2 Controlling Pin Multiplexing

Some block I/Os are routed to dedicated package pins, but the majority of block I/O signals are routed through the IOMUXC. This allows internal blocks to share pins to drive or receive signals. Block I/Os are selected for routing to and from external package pins via the MUX_MODE field in the MUX_CTL registers. These selections are called ALT modes.

Some block input signals can be driven from alternate pins. This function is called a daisy chain. SELECT_INPUT registers allow software to chose between two to six different pins to provide input signals to some blocks.

Drive strength and direction (input or output) of many package pins may be selected by fields within PAD_CTL registers. Some pins have dedicated PAD_CTL_PAD registers. Most pins that are used in their default mode to interface to external DRAM are configured for drive strength and direction by PAD_CTL_GRP register although some have PAD_CTL_PAD registers as well.

Multiplexing and Pad Control below lists those registers used to control package pin drive characteristics and pin to internal block I/O connections.

Daisy Chain Control below lists registers involved in controlling daisy chaining.

See the IOMUXC chapter for more details.

4.2.1 Multiplexing and Pad Control

This section covers the multiplexing and pad control regitsters.

Table 4-1 shows the multiplexing control register and pad driver control register(s) for each package pin.

Table 4-1. Pin Control Registers

Pin Name	Mux Control	Pad / Group Control
GPIO_19	IOMUXC_SW_MUX_CTL_PAD_GPIO_19	IOMUXC_SW_PAD_CTL_PAD_GPIO_19
KEY_COL0	IOMUXC_SW_MUX_CTL_PAD_KEY_COL0	IOMUXC_SW_PAD_CTL_PAD_KEY_COL0
KEY_ ROW0	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW0	IOMUXC_SW_PAD_CTL_PAD_KEY_ROW0
KEY_COL1	IOMUXC_SW_MUX_CTL_PAD_KEY_COL1	IOMUXC_SW_PAD_CTL_PAD_KEY_COL1
KEY_ ROW1	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW1	IOMUXC_SW_PAD_CTL_PAD_KEY_ROW1
KEY_COL2	IOMUXC_SW_MUX_CTL_PAD_KEY_COL2	IOMUXC_SW_PAD_CTL_PAD_KEY_COL2
KEY_ ROW2	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW2	IOMUXC_SW_PAD_CTL_PAD_KEY_ROW2
KEY_COL3	IOMUXC_SW_MUX_CTL_PAD_KEY_COL3	IOMUXC_SW_PAD_CTL_PAD_KEY_COL3
KEY_ ROW3	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW3	IOMUXC_SW_PAD_CTL_PAD_KEY_ROW3
KEY_COL4	IOMUXC_SW_MUX_CTL_PAD_KEY_COL4	IOMUXC_SW_PAD_CTL_PAD_KEY_COL4
KEY_ ROW4	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW4	IOMUXC_SW_PAD_CTL_PAD_KEY_ROW4

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
NVCC_ KEYPAD		IOMUXC_SW_PAD_CTL_PAD_NVCC_KEYPAD
DI0_DISP_ CLK	IOMUXC_SW_MUX_CTL_PAD_DI0_DISP_CLK	IOMUXC_SW_PAD_CTL_PAD_DI0_DISP_CLK
DI0_PIN15	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN15	IOMUXC_SW_PAD_CTL_PAD_DI0_PIN15
DI0_PIN2	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN2	IOMUXC_SW_PAD_CTL_PAD_DI0_PIN2
DI0_PIN3	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN3	IOMUXC_SW_PAD_CTL_PAD_DI0_PIN3
DI0_PIN4	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN4	IOMUXC_SW_PAD_CTL_PAD_DI0_PIN4
DISP0_ DAT0	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT0	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT0
DISP0_ DAT1	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT1	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT1
DISP0_ DAT2	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT2	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT2
DISP0_ DAT3	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT3	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT3
DISP0_ DAT4	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT4	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT4
DISP0_ DAT5	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT5	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT5
DISP0_ DAT6	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT6	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT6
DISP0_ DAT7	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT7	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT7
DISP0_ DAT8	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT8	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT8
DISP0_ DAT9	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT9	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT9
DISP0_ DAT10	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT10	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT10
DISP0_ DAT11	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT11	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT11
DISP0_ DAT12	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT12	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT12
DISP0_ DAT13	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT13	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT13
DISP0_ DAT14	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT14	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT14
DISP0_ DAT15	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT15	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT15
DISP0_ DAT16	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT16	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT16

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
DISP0_ DAT17	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT17	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT17
DISP0_ DAT18	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT18	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT18
DISP0_ DAT19	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT19	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT19
DISP0_ DAT20	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT20	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT20
DISP0_ DAT21	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT21	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT21
DISP0_ DAT22	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT22	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT22
DISP0_ DAT23	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT23	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT23
CSI0_ PIXCLK	IOMUXC_SW_MUX_CTL_PAD_CSI0_PIXCLK	IOMUXC_SW_PAD_CTL_PAD_CSI0_PIXCLK
CSI0_ MCLK	IOMUXC_SW_MUX_CTL_PAD_CSI0_MCLK	IOMUXC_SW_PAD_CTL_PAD_CSI0_MCLK
CSI0_ DATA_EN	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA_EN	IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA_EN
CSI0_ VSYNC	IOMUXC_SW_MUX_CTL_PAD_CSI0_VSYNC	IOMUXC_SW_PAD_CTL_PAD_CSI0_VSYNC
CSI0_DAT4	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT4	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT4
CSI0_DAT5	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT5	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT5
CSI0_DAT6	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT6	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT6
CSI0_DAT7	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT7	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT7
CSI0_DAT8	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT8	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT8
CSI0_DAT9	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT9	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT9
CSI0_ DAT10	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT10	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT10
CSI0_ DAT11	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT11	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT11
CSI0_ DAT12	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT12	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT12
CSI0_ DAT13	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT13	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT13
CSI0_ DAT14	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT14	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT14
CSI0_ DAT15	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT15	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT15

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
CSI0_ DAT16	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT16	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT16
CSI0_ DAT17	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT17	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT17
CSI0_ DAT18	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT18	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT18
CSI0_ DAT19	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT19	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT19
NVCC_ CSI__0		IOMUXC_SW_PAD_CTL_PAD_NVCC_CSI__0
JTAG_TMS		IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS
JTAG_MOD		IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD
JTAG_ TRSTB		IOMUXC_SW_PAD_CTL_PAD_JTAG_TRSTB
JTAG_TDI		IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI
JTAG_TCK		IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK
JTAG_TDO		IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO
EIM_A25	IOMUXC_SW_MUX_CTL_PAD_EIM_A25	IOMUXC_SW_PAD_CTL_PAD_EIM_A25
EIM_EB2	IOMUXC_SW_MUX_CTL_PAD_EIM_EB2	IOMUXC_SW_PAD_CTL_PAD_EIM_EB2
EIM_D16	IOMUXC_SW_MUX_CTL_PAD_EIM_D16	IOMUXC_SW_PAD_CTL_PAD_EIM_D16
EIM_D17	IOMUXC_SW_MUX_CTL_PAD_EIM_D17	IOMUXC_SW_PAD_CTL_PAD_EIM_D17
NVCC_ EIM__0		
EIM_D18	IOMUXC_SW_MUX_CTL_PAD_EIM_D18	IOMUXC_SW_PAD_CTL_PAD_EIM_D18
EIM_D19	IOMUXC_SW_MUX_CTL_PAD_EIM_D19	IOMUXC_SW_PAD_CTL_PAD_EIM_D19
EIM_D20	IOMUXC_SW_MUX_CTL_PAD_EIM_D20	IOMUXC_SW_PAD_CTL_PAD_EIM_D20
EIM_D21	IOMUXC_SW_MUX_CTL_PAD_EIM_D21	IOMUXC_SW_PAD_CTL_PAD_EIM_D21
EIM_D22	IOMUXC_SW_MUX_CTL_PAD_EIM_D22	IOMUXC_SW_PAD_CTL_PAD_EIM_D22
EIM_D23	IOMUXC_SW_MUX_CTL_PAD_EIM_D23	IOMUXC_SW_PAD_CTL_PAD_EIM_D23
EIM_EB3	IOMUXC_SW_MUX_CTL_PAD_EIM_EB3	IOMUXC_SW_PAD_CTL_PAD_EIM_EB3
EIM_D24	IOMUXC_SW_MUX_CTL_PAD_EIM_D24	IOMUXC_SW_PAD_CTL_PAD_EIM_D24
EIM_D25	IOMUXC_SW_MUX_CTL_PAD_EIM_D25	IOMUXC_SW_PAD_CTL_PAD_EIM_D25
EIM_D26	IOMUXC_SW_MUX_CTL_PAD_EIM_D26	IOMUXC_SW_PAD_CTL_PAD_EIM_D26
EIM_D27	IOMUXC_SW_MUX_CTL_PAD_EIM_D27	IOMUXC_SW_PAD_CTL_PAD_EIM_D27
EIM_D28	IOMUXC_SW_MUX_CTL_PAD_EIM_D28	IOMUXC_SW_PAD_CTL_PAD_EIM_D28
EIM_D29	IOMUXC_SW_MUX_CTL_PAD_EIM_D29	IOMUXC_SW_PAD_CTL_PAD_EIM_D29
EIM_D30	IOMUXC_SW_MUX_CTL_PAD_EIM_D30	IOMUXC_SW_PAD_CTL_PAD_EIM_D30

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
EIM_D31	IOMUXC_SW_MUX_CTL_PAD_EIM_D31	IOMUXC_SW_PAD_CTL_PAD_EIM_D31
NVCC_ EIM__1		IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__1
EIM_A24	IOMUXC_SW_MUX_CTL_PAD_EIM_A24	IOMUXC_SW_PAD_CTL_PAD_EIM_A24
EIM_A23	IOMUXC_SW_MUX_CTL_PAD_EIM_A23	IOMUXC_SW_PAD_CTL_PAD_EIM_A23
EIM_A22	IOMUXC_SW_MUX_CTL_PAD_EIM_A22	IOMUXC_SW_PAD_CTL_PAD_EIM_A22
EIM_A21	IOMUXC_SW_MUX_CTL_PAD_EIM_A21	IOMUXC_SW_PAD_CTL_PAD_EIM_A21
EIM_A20	IOMUXC_SW_MUX_CTL_PAD_EIM_A20	IOMUXC_SW_PAD_CTL_PAD_EIM_A20
EIM_A19	IOMUXC_SW_MUX_CTL_PAD_EIM_A19	IOMUXC_SW_PAD_CTL_PAD_EIM_A19
EIM_A18	IOMUXC_SW_MUX_CTL_PAD_EIM_A18	IOMUXC_SW_PAD_CTL_PAD_EIM_A18
EIM_A17	IOMUXC_SW_MUX_CTL_PAD_EIM_A17	IOMUXC_SW_PAD_CTL_PAD_EIM_A17
EIM_A16	IOMUXC_SW_MUX_CTL_PAD_EIM_A16	IOMUXC_SW_PAD_CTL_PAD_EIM_A16
EIM_CS0	IOMUXC_SW_MUX_CTL_PAD_EIM_CS0	IOMUXC_SW_PAD_CTL_PAD_EIM_CS0
EIM_CS1	IOMUXC_SW_MUX_CTL_PAD_EIM_CS1	IOMUXC_SW_PAD_CTL_PAD_EIM_CS1
EIM_OE	IOMUXC_SW_MUX_CTL_PAD_EIM_OE	IOMUXC_SW_PAD_CTL_PAD_EIM_OE
EIM_RW	IOMUXC_SW_MUX_CTL_PAD_EIM_RW	IOMUXC_SW_PAD_CTL_PAD_EIM_RW
EIM_LBA	IOMUXC_SW_MUX_CTL_PAD_EIM_LBA	IOMUXC_SW_PAD_CTL_PAD_EIM_LBA
NVCC_ EIM__4		IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__4
EIM_EB0	IOMUXC_SW_MUX_CTL_PAD_EIM_EB0	IOMUXC_SW_PAD_CTL_PAD_EIM_EB0
EIM_EB1	IOMUXC_SW_MUX_CTL_PAD_EIM_EB1	IOMUXC_SW_PAD_CTL_PAD_EIM_EB1
EIM_DA0	IOMUXC_SW_MUX_CTL_PAD_EIM_DA0	IOMUXC_SW_PAD_CTL_PAD_EIM_DA0
EIM_DA1	IOMUXC_SW_MUX_CTL_PAD_EIM_DA1	IOMUXC_SW_PAD_CTL_PAD_EIM_DA1
EIM_DA2	IOMUXC_SW_MUX_CTL_PAD_EIM_DA2	IOMUXC_SW_PAD_CTL_PAD_EIM_DA2
EIM_DA3	IOMUXC_SW_MUX_CTL_PAD_EIM_DA3	IOMUXC_SW_PAD_CTL_PAD_EIM_DA3
EIM_DA4	IOMUXC_SW_MUX_CTL_PAD_EIM_DA4	IOMUXC_SW_PAD_CTL_PAD_EIM_DA4
EIM_DA5	IOMUXC_SW_MUX_CTL_PAD_EIM_DA5	IOMUXC_SW_PAD_CTL_PAD_EIM_DA5
EIM_DA6	IOMUXC_SW_MUX_CTL_PAD_EIM_DA6	IOMUXC_SW_PAD_CTL_PAD_EIM_DA6
EIM_DA7	IOMUXC_SW_MUX_CTL_PAD_EIM_DA7	IOMUXC_SW_PAD_CTL_PAD_EIM_DA7
EIM_DA8	IOMUXC_SW_MUX_CTL_PAD_EIM_DA8	IOMUXC_SW_PAD_CTL_PAD_EIM_DA8
EIM_DA9	IOMUXC_SW_MUX_CTL_PAD_EIM_DA9	IOMUXC_SW_PAD_CTL_PAD_EIM_DA9
EIM_DA10	IOMUXC_SW_MUX_CTL_PAD_EIM_DA10	IOMUXC_SW_PAD_CTL_PAD_EIM_DA10
EIM_DA11	IOMUXC_SW_MUX_CTL_PAD_EIM_DA11	IOMUXC_SW_PAD_CTL_PAD_EIM_DA11
EIM_DA12	IOMUXC_SW_MUX_CTL_PAD_EIM_DA12	IOMUXC_SW_PAD_CTL_PAD_EIM_DA12
EIM_DA13	IOMUXC_SW_MUX_CTL_PAD_EIM_DA13	IOMUXC_SW_PAD_CTL_PAD_EIM_DA13
EIM_DA14	IOMUXC_SW_MUX_CTL_PAD_EIM_DA14	IOMUXC_SW_PAD_CTL_PAD_EIM_DA14

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
EIM_DA15	IOMUXC_SW_MUX_CTL_PAD_EIM_DA15	IOMUXC_SW_PAD_CTL_PAD_EIM_DA15
NANDF_WE_B	IOMUXC_SW_MUX_CTL_PAD_NANDF_WE_B	IOMUXC_SW_PAD_CTL_PAD_NANDF_WE_B
NANDF_RE_B	IOMUXC_SW_MUX_CTL_PAD_NANDF_RE_B	IOMUXC_SW_PAD_CTL_PAD_NANDF_RE_B
EIM_WAIT	IOMUXC_SW_MUX_CTL_PAD_EIM_WAIT	IOMUXC_SW_PAD_CTL_PAD_EIM_WAIT
EIM_BCLK		IOMUXC_SW_PAD_CTL_PAD_EIM_BCLK
NVCC_EIM__7		IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__7
LVDS1_TX3_P	IOMUXC_SW_MUX_CTL_PAD_LVDS1_TX3_P	
LVDS1_TX3_N		
LVDS1_TX2_P	IOMUXC_SW_MUX_CTL_PAD_LVDS1_TX2_P	
LVDS1_TX2_N		
LVDS1_CLK_P	IOMUXC_SW_MUX_CTL_PAD_LVDS1_CLK_P	
LVDS1_CLK_N		
LVDS1_TX1_P	IOMUXC_SW_MUX_CTL_PAD_LVDS1_TX1_P	
LVDS1_TX1_N		
LVDS1_TX0_P	IOMUXC_SW_MUX_CTL_PAD_LVDS1_TX0_P	
LVDS1_TX0_N		
LVDS0_TX3_P	IOMUXC_SW_MUX_CTL_PAD_LVDS0_TX3_P	
LVDS0_TX3_N		
LVDS0_CLK_P	IOMUXC_SW_MUX_CTL_PAD_LVDS0_CLK_P	
LVDS0_CLK_N		
LVDS0_TX2_P	IOMUXC_SW_MUX_CTL_PAD_LVDS0_TX2_P	
LVDS0_TX2_N		

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
LVDS0_ TX1_P	IOMUXC_SW_MUX_CTL_PAD_LVDS0_TX1_P	
LVDS0_ TX1_N		
LVDS0_ TX0_P	IOMUXC_SW_MUX_CTL_PAD_LVDS0_TX0_P	
LVDS0_ TX0_N		
GPIO_10	IOMUXC_SW_MUX_CTL_PAD_GPIO_10	IOMUXC_SW_PAD_CTL_PAD_GPIO_10
GPIO_11		IOMUXC_SW_PAD_CTL_PAD_GPIO_11
GPIO_12		IOMUXC_SW_PAD_CTL_PAD_GPIO_12
GPIO_13		IOMUXC_SW_PAD_CTL_PAD_GPIO_13
GPIO_14		IOMUXC_SW_PAD_CTL_PAD_GPIO_14
DRAM_D24		IOMUXC_SW_PAD_CTL_GRP_B3DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL3
DRAM_D30		IOMUXC_SW_PAD_CTL_GRP_B3DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL3
DRAM_D26		IOMUXC_SW_PAD_CTL_GRP_B3DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL3
DRAM_ DQM3		IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM3

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
DRAM_D28		IOMUXC_SW_PAD_CTL_GRP_B3DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL3
DRAM_D25		IOMUXC_SW_PAD_CTL_GRP_B3DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL3
DRAM_ SDQS3_B		
DRAM_ SDQS3		IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS3
DRAM_D27		IOMUXC_SW_PAD_CTL_GRP_B3DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL3
DRAM_D31		IOMUXC_SW_PAD_CTL_GRP_B3DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL3

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
DRAM_D16		IOMUXC_SW_PAD_CTL_GRP_B2DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL2
DRAM_D29		IOMUXC_SW_PAD_CTL_GRP_B3DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL3
DRAM_D18		IOMUXC_SW_PAD_CTL_GRP_B2DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL2
DRAM_ SDCKE1		IOMUXC_SW_PAD_CTL_GRP_CTLDS IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE1
DRAM_D22		IOMUXC_SW_PAD_CTL_GRP_B2DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL2
DRAM_ DQM2		IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM2

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
DRAM_D20		IOMUXC_SW_PAD_CTL_GRP_B2DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL2
DRAM_ SDBA0		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_D17		IOMUXC_SW_PAD_CTL_GRP_B2DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL2
DRAM_ SDODT1		IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT1
DRAM_D19		IOMUXC_SW_PAD_CTL_GRP_B2DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL2
DRAM_ SDQS2_B		
DRAM_ SDQS2		IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS2

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
DRAM_D21		IOMUXC_SW_PAD_CTL_GRP_B2DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL2
DRAM_CS1		IOMUXC_SW_PAD_CTL_GRP_CTLDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_D23		IOMUXC_SW_PAD_CTL_GRP_B2DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL2
DRAM_ RESET		IOMUXC_SW_PAD_CTL_PAD_DRAM_RESET
DRAM_ SDBA1		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_ SDCLK_1_ B		
DRAM_ SDCLK_1		IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_1
DRAM_A8		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
DRAM_ SDBA2		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_A14		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_A3		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_A5		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_A7		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_A6		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_A9		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_A2		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_A0		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
DRAM_A15		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_A13		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_A11		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_A1		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_A12		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_ CAS		IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_PAD_DRAM_CAS
DRAM_ SDWE		IOMUXC_SW_PAD_CTL_GRP_CTLDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_CS0		IOMUXC_SW_PAD_CTL_GRP_CTLDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_A4		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
DRAM_ SDCLK_0_ B		
DRAM_ SDCLK_0		IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_0
DRAM_A10		IOMUXC_SW_PAD_CTL_GRP_ADDDS IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE
DRAM_D4		IOMUXC_SW_PAD_CTL_GRP_B0DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL0
DRAM_D6		IOMUXC_SW_PAD_CTL_GRP_B0DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL0
DRAM_D2		IOMUXC_SW_PAD_CTL_GRP_B0DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL0
DRAM_ SDQS0_B		
DRAM_ SDQS0		IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS0

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
DRAM_ SDODT0		IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT0
DRAM_ DQM0		IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM0
DRAM_ RAS		IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_PAD_DRAM_RAS
DRAM_D5		IOMUXC_SW_PAD_CTL_GRP_B0DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL0
DRAM_D0		IOMUXC_SW_PAD_CTL_GRP_B0DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL0
DRAM_D7		IOMUXC_SW_PAD_CTL_GRP_B0DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL0
DRAM_ SDCKE0		IOMUXC_SW_PAD_CTL_GRP_CTLDS IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE0

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
DRAM_D1		IOMUXC_SW_PAD_CTL_GRP_B0DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL0
DRAM_D14		IOMUXC_SW_PAD_CTL_GRP_B1DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL1
DRAM_D3		IOMUXC_SW_PAD_CTL_GRP_B0DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL0
DRAM_D12		IOMUXC_SW_PAD_CTL_GRP_B1DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL1
DRAM_D10		IOMUXC_SW_PAD_CTL_GRP_B1DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL1

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
DRAM_D8		IOMUXC_SW_PAD_CTL_GRP_B1DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL1
DRAM_D13		IOMUXC_SW_PAD_CTL_GRP_B1DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL1
DRAM_ SDQS1_B		
DRAM_ SDQS1		IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS1
DRAM_ DQM1		IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM1
DRAM_D9		IOMUXC_SW_PAD_CTL_GRP_B1DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL1

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
DRAM_D15		IOMUXC_SW_PAD_CTL_GRP_B1DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL1
DRAM_D11		IOMUXC_SW_PAD_CTL_GRP_B1DS IOMUXC_SW_PAD_CTL_GRP_DDRHYS IOMUXC_SW_PAD_CTL_GRP_DDRMODE IOMUXC_SW_PAD_CTL_GRP_DDRPK IOMUXC_SW_PAD_CTL_GRP_DDRPKE IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE IOMUXC_SW_PAD_CTL_GRP_TERM_CTL1
CKIH1		
CKIH2		
PMIC_ON_REQ		IOMUXC_SW_PAD_CTL_PAD_PMIC_ON_REQ
PMIC_STBY_REQ		IOMUXC_SW_PAD_CTL_PAD_PMIC_STBY_REQ
NANDF_CLE	IOMUXC_SW_MUX_CTL_PAD_NANDF_CLE	IOMUXC_SW_PAD_CTL_PAD_NANDF_CLE
NANDF_ALE	IOMUXC_SW_MUX_CTL_PAD_NANDF_ALE	IOMUXC_SW_PAD_CTL_PAD_NANDF_ALE
NANDF_WP_B	IOMUXC_SW_MUX_CTL_PAD_NANDF_WP_B	IOMUXC_SW_PAD_CTL_PAD_NANDF_WP_B
NANDF_RB0	IOMUXC_SW_MUX_CTL_PAD_NANDF_RB0	IOMUXC_SW_PAD_CTL_PAD_NANDF_RB0
NANDF_CS0	IOMUXC_SW_MUX_CTL_PAD_NANDF_CS0	IOMUXC_SW_PAD_CTL_PAD_NANDF_CS0
NANDF_CS1	IOMUXC_SW_MUX_CTL_PAD_NANDF_CS1	IOMUXC_SW_PAD_CTL_PAD_NANDF_CS1
NANDF_CS2	IOMUXC_SW_MUX_CTL_PAD_NANDF_CS2	IOMUXC_SW_PAD_CTL_PAD_NANDF_CS2
NANDF_CS3	IOMUXC_SW_MUX_CTL_PAD_NANDF_CS3	IOMUXC_SW_PAD_CTL_PAD_NANDF_CS3
NVCC_NANDF		IOMUXC_SW_PAD_CTL_PAD_NVCC_NANDF
FEC_MDIO	IOMUXC_SW_MUX_CTL_PAD_FEC_MDIO	IOMUXC_SW_PAD_CTL_PAD_FEC_MDIO

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
FEC_REF_CLK	IOMUXC_SW_MUX_CTL_PAD_FEC_REF_CLK	IOMUXC_SW_PAD_CTL_PAD_FEC_REF_CLK
FEC_RX_ER	IOMUXC_SW_MUX_CTL_PAD_FEC_RX_ER	IOMUXC_SW_PAD_CTL_PAD_FEC_RX_ER
FEC_CRS_DV	IOMUXC_SW_MUX_CTL_PAD_FEC_CRS_DV	IOMUXC_SW_PAD_CTL_PAD_FEC_CRS_DV
FEC_RXD1	IOMUXC_SW_MUX_CTL_PAD_FEC_RXD1	IOMUXC_SW_PAD_CTL_PAD_FEC_RXD1
FEC_RXD0	IOMUXC_SW_MUX_CTL_PAD_FEC_RXD0	IOMUXC_SW_PAD_CTL_PAD_FEC_RXD0
FEC_TX_EN	IOMUXC_SW_MUX_CTL_PAD_FEC_TX_EN	IOMUXC_SW_PAD_CTL_PAD_FEC_TX_EN
FEC_TXD1	IOMUXC_SW_MUX_CTL_PAD_FEC_TXD1	IOMUXC_SW_PAD_CTL_PAD_FEC_TXD1
FEC_TXD0	IOMUXC_SW_MUX_CTL_PAD_FEC_TXD0	IOMUXC_SW_PAD_CTL_PAD_FEC_TXD0
FEC_MDC	IOMUXC_SW_MUX_CTL_PAD_FEC_MDC	IOMUXC_SW_PAD_CTL_PAD_FEC_MDC
NVCC_FEC		IOMUXC_SW_PAD_CTL_PAD_NVCC_FEC
PATA_DIOW	IOMUXC_SW_MUX_CTL_PAD_PATA_DIOW	IOMUXC_SW_PAD_CTL_PAD_PATA_DIOW
PATA_DMACK	IOMUXC_SW_MUX_CTL_PAD_PATA_DMACK	IOMUXC_SW_PAD_CTL_PAD_PATA_DMACK
PATA_DMARQ	IOMUXC_SW_MUX_CTL_PAD_PATA_DMARQ	IOMUXC_SW_PAD_CTL_PAD_PATA_DMARQ
PATA_BUFFER_EN	IOMUXC_SW_MUX_CTL_PAD_PATA_BUFFER_EN	IOMUXC_SW_PAD_CTL_PAD_PATA_BUFFER_EN
PATA_INTRQ	IOMUXC_SW_MUX_CTL_PAD_PATA_INTRQ	IOMUXC_SW_PAD_CTL_PAD_PATA_INTRQ
PATA_DIOR	IOMUXC_SW_MUX_CTL_PAD_PATA_DIOR	IOMUXC_SW_PAD_CTL_PAD_PATA_DIOR
PATA_RESET_B	IOMUXC_SW_MUX_CTL_PAD_PATA_RESET_B	IOMUXC_SW_PAD_CTL_PAD_PATA_RESET_B
PATA_IORDY	IOMUXC_SW_MUX_CTL_PAD_PATA_IORDY	IOMUXC_SW_PAD_CTL_PAD_PATA_IORDY
PATA_DA_0	IOMUXC_SW_MUX_CTL_PAD_PATA_DA_0	IOMUXC_SW_PAD_CTL_PAD_PATA_DA_0
PATA_DA_1	IOMUXC_SW_MUX_CTL_PAD_PATA_DA_1	IOMUXC_SW_PAD_CTL_PAD_PATA_DA_1
PATA_DA_2	IOMUXC_SW_MUX_CTL_PAD_PATA_DA_2	IOMUXC_SW_PAD_CTL_PAD_PATA_DA_2
PATA_CS_0	IOMUXC_SW_MUX_CTL_PAD_PATA_CS_0	IOMUXC_SW_PAD_CTL_PAD_PATA_CS_0
PATA_CS_1	IOMUXC_SW_MUX_CTL_PAD_PATA_CS_1	IOMUXC_SW_PAD_CTL_PAD_PATA_CS_1

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
NVCC_ PATA__2		IOMUXC_SW_PAD_CTL_PAD_NVCC_PATA__2
PATA_ DATA0	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA0	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA0
PATA_ DATA1	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA1	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA1
PATA_ DATA2	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA2	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA2
PATA_ DATA3	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA3	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA3
PATA_ DATA4	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA4	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA4
PATA_ DATA5	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA5	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA5
PATA_ DATA6	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA6	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA6
PATA_ DATA7	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA7	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA7
PATA_ DATA8	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA8	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA8
PATA_ DATA9	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA9	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA9
PATA_ DATA10	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA10	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA10
PATA_ DATA11	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA11	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA11
PATA_ DATA12	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA12	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA12
PATA_ DATA13	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA13	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA13
PATA_ DATA14	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA14	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA14
PATA_ DATA15	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA15	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA15
NVCC_ PATA__0		IOMUXC_SW_PAD_CTL_PAD_NVCC_PATA__0
SD1_ DATA0	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0	IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0
SD1_ DATA1	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1	IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1
SD1_CMD	IOMUXC_SW_MUX_CTL_PAD_SD1_CMD	IOMUXC_SW_PAD_CTL_PAD_SD1_CMD
SD1_ DATA2	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2	IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2

Table continues on the next page...

Table 4-1. Pin Control Registers (continued)

Pin Name	Mux Control	Pad / Group Control
SD1_CLK	IOMUXC_SW_MUX_CTL_PAD_SD1_CLK	IOMUXC_SW_PAD_CTL_PAD_SD1_CLK
SD1_ DATA3	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3	IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3
NVCC_SD1		IOMUXC_SW_PAD_CTL_PAD_NVCC_SD1
SD2_CLK	IOMUXC_SW_MUX_CTL_PAD_SD2_CLK	IOMUXC_SW_PAD_CTL_PAD_SD2_CLK
SD2_CMD	IOMUXC_SW_MUX_CTL_PAD_SD2_CMD	IOMUXC_SW_PAD_CTL_PAD_SD2_CMD
SD2_ DATA3	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3	IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3
SD2_ DATA2	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2	IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2
SD2_ DATA1	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1	IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1
SD2_ DATA0	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0	IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0
NVCC_SD2		IOMUXC_SW_PAD_CTL_PAD_NVCC_SD2
GPIO_0	IOMUXC_SW_MUX_CTL_PAD_GPIO_0	IOMUXC_SW_PAD_CTL_PAD_GPIO_0
GPIO_1	IOMUXC_SW_MUX_CTL_PAD_GPIO_1	IOMUXC_SW_PAD_CTL_PAD_GPIO_1
GPIO_9	IOMUXC_SW_MUX_CTL_PAD_GPIO_9	IOMUXC_SW_PAD_CTL_PAD_GPIO_9
GPIO_3	IOMUXC_SW_MUX_CTL_PAD_GPIO_3	IOMUXC_SW_PAD_CTL_PAD_GPIO_3
GPIO_6	IOMUXC_SW_MUX_CTL_PAD_GPIO_6	IOMUXC_SW_PAD_CTL_PAD_GPIO_6
GPIO_2	IOMUXC_SW_MUX_CTL_PAD_GPIO_2	IOMUXC_SW_PAD_CTL_PAD_GPIO_2
GPIO_4	IOMUXC_SW_MUX_CTL_PAD_GPIO_4	IOMUXC_SW_PAD_CTL_PAD_GPIO_4
GPIO_5	IOMUXC_SW_MUX_CTL_PAD_GPIO_5	IOMUXC_SW_PAD_CTL_PAD_GPIO_5
GPIO_7	IOMUXC_SW_MUX_CTL_PAD_GPIO_7	IOMUXC_SW_PAD_CTL_PAD_GPIO_7
GPIO_8	IOMUXC_SW_MUX_CTL_PAD_GPIO_8	IOMUXC_SW_PAD_CTL_PAD_GPIO_8
GPIO_16	IOMUXC_SW_MUX_CTL_PAD_GPIO_16	IOMUXC_SW_PAD_CTL_PAD_GPIO_16
GPIO_17	IOMUXC_SW_MUX_CTL_PAD_GPIO_17	IOMUXC_SW_PAD_CTL_PAD_GPIO_17
GPIO_18	IOMUXC_SW_MUX_CTL_PAD_GPIO_18	IOMUXC_SW_PAD_CTL_PAD_GPIO_18
NVCC_ GPIO		IOMUXC_SW_PAD_CTL_PAD_NVCC_GPIO
POR_B		IOMUXC_SW_PAD_CTL_PAD_POR_B
BOOT_ MODE1		IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE1
RESET_IN_ B		IOMUXC_SW_PAD_CTL_PAD_RESET_IN_B
BOOT_ MODE0		IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE0
TEST_ MODE		IOMUXC_SW_PAD_CTL_PAD_TEST_MODE

Table 4-2 lists package pin multiplexing options by package pin.

Table 4-2. Pin Alternate Modes

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
GPIO_19	ALT0	KPP	COL[5]	IOMUXC_SW_PAD_CTL_PAD_GPIO_19 Drive Strength (DSE) = High Low/high output voltage (HVE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[5]	
	ALT2	CCM	CLKO	
	ALT3	SPDIF	OUT1	
	ALT4	RTC	CE_RTC_EXT_TRIG2	
	ALT5	ECSPI-1	RDY	
	ALT6	FEC	TDATA[3]	
	ALT7	SRC	INT_BOOT	
KEY_ COL0	ALT0	KPP	COL[0]	IOMUXC_SW_PAD_CTL_PAD_KEY_COL0 Drive Strength (DSE) = High Low/high output voltage (HVE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[6]	
	ALT2	AUDMUX	AUD5_TXC	
	ALT3	ARM Platform	CTI_TRIGIN7	
	ALT4	UART-4	TXD_MUX	
	ALT5	ECSPI-1	SCLK	
	ALT6	FEC	RDATA[3]	
	ALT7	SRC	ANY_PU_RST	
KEY_ ROW0	ALT0	KPP	ROW[0]	IOMUXC_SW_PAD_CTL_PAD_KEY_ROW0 Drive Strength (DSE) = High Low/high output voltage (HVE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[7]	
	ALT2	AUDMUX	AUD5_TXD	
	ALT3	ARM Platform	CTI_TRIGIN_ACK7	
	ALT4	UART-4	RXD_MUX	
	ALT5	ESCPI-1	MOSI	
	ALT6	FEC	TX_ER	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
KEY_ COL1	ALT0	KPP	COL[1]	IOMUXC_SW_PAD_CTL_PAD_KEY_COL1 Drive Strength (DSE) = High Low/high output voltage (HVE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[8]	
	ALT2	AUDMUX	AUD5_TXFS	
	ALT3	ARM Platform	CTI_TRIGOUT_ACK6	
	ALT4	UART-5	TXD_MUX	
	ALT5	ECSPI-1	MISO	
	ALT6	FEC	RX_CLK	
	ALT7	USBPHY-1	TXREADY	
KEY_ ROW1	ALT0	KPP	ROW[1]	IOMUXC_SW_PAD_CTL_PAD_KEY_ROW1 Drive Strength (DSE) = High Low/high output voltage (HVE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[9]	
	ALT2	AUDMUX	AUD5_RXD	
	ALT3	ARM Platform	CTI_TRIGOUT_ACK7	
	ALT4	UART-5	RXD_MUX	
	ALT5	ECSPI-1	SS0	
	ALT6	FEC	COL	
	ALT7	USBPHY-1	RXVALID	
KEY_ COL2	ALT0	KPP	COL[2]	IOMUXC_SW_PAD_CTL_PAD_KEY_COL2 Drive Strength (DSE) = High Low/high output voltage (HVE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[10]	
	ALT2	FLEXCAN-1	TXCAN	
	ALT3	ARM Platform	CTI_TRIGOUT6	
	ALT4	FEC	MDIO	
	ALT5	ECSPI-1	SS1	
	ALT6	FEC	RDATA[2]	
	ALT7	USBPHY-1	RXACTIVE	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
KEY_ ROW2	ALT0	KPP	ROW[2]	IOMUXC_SW_PAD_CTL_PAD_KEY_ROW2 Drive Strength (DSE) = High Low/high output voltage (HVE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[11]	
	ALT2	FLEXCAN-1	RXCAN	
	ALT3	ARM Platform	CTI_TRIGOUT7	
	ALT4	FEC	MDC	
	ALT5	ECSPI-1	SS2	
	ALT6	FEC	TDATA[2]	
	ALT7	USBPHY-1	RXERROR	
KEY_ COL3	ALT0	KPP	COL[3]	IOMUXC_SW_PAD_CTL_PAD_KEY_COL3 Drive Strength (DSE) = High Low/high output voltage (HVE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[12]	
	ALT2	USB	H2_DP	
	ALT3	SPDIF	IN1	
	ALT4	I2C-2	SCL	
	ALT5	ECSPI-1	SS3	
	ALT6	FEC	CRS	
	ALT7	USBPHY-1	SIECLOCK	
KEY_ ROW3	ALT0	KPP	ROW[3]	IOMUXC_SW_PAD_CTL_PAD_KEY_ROW3 Drive Strength (DSE) = High Low/high output voltage (HVE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[13]	
	ALT2	USB	H2_DM	
	ALT3	CCM	ASRC_EXT_CLK	
	ALT4	I2C-2	SDA	
	ALT5	XTALOSC32K	32K_OUT	
	ALT6	CCM	PLL4_BYP	
	ALT7	USBPHY-1	LINESTATE[0]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
KEY_ COL4	ALT0	KPP	COL[4]	IOMUXC_SW_PAD_CTL_PAD_KEY_COL4 Drive Strength (DSE) = High Low/high output voltage (HVE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[14]	
	ALT2	FLEXCAN-2	TXCAN	
	ALT3	IPU	SISG[4]	
	ALT4	UART-5	RTS	
	ALT5	USB	USBOTG_OC	
	ALT7	USBPHY-1	LINESTATE[1]	
KEY_ ROW4	ALT0	KPP	ROW[4]	IOMUXC_SW_PAD_CTL_PAD_KEY_ROW4 Drive Strength (DSE) = High Low/high output voltage (HVE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[15]	
	ALT2	FLEXCAN-2	RXCAN	
	ALT3	IPU	SISG[5]	
	ALT4	UART-5	CTS	
	ALT5	USB	USBOTG_PWR	
	ALT7	USBPHY-1	VBUSVALID	
DI0_ DISP_CLK	ALT0	IPU	DI0_DISP_CLK	IOMUXC_SW_PAD_CTL_PAD_DI0_DISP_CLK Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[16]	
	ALT2	USB	USBH2_DIR	
	ALT5	SDMA	DEBUG_CORE_STATE[0]	
	ALT6	EXTMC	EMI_DEBUG[0]	
	ALT7	USBPHY-1	AVALID	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DI0_PIN15	ALT0	IPU	DI0_PIN15	IOMUXC_SW_PAD_CTL_PAD_DI0_PIN15 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[17]	
	ALT2	AUDMUX	AUD6_TXC	
	ALT5	SDMA	DEBUG_CORE_STATE[1]	
	ALT6	EXTMC	EMI_DEBUG[1]	
	ALT7	USBPHY-1	BVALID	
DI0_PIN2	ALT0	IPU	DI0_PIN2	IOMUXC_SW_PAD_CTL_PAD_DI0_PIN2 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[18]	
	ALT2	AUDMUX	AUD6_TXD	
	ALT5	SDMA	DEBUG_CORE_STATE[2]	
	ALT6	EXTMC	EMI_DEBUG[2]	
	ALT7	USBPHY-1	ENDSESSION	
DI0_PIN3	ALT0	IPU	DI0_PIN3	IOMUXC_SW_PAD_CTL_PAD_DI0_PIN3 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[19]	
	ALT2	AUDMUX	AUD6_TXFS	
	ALT5	SDMA	DEBUG_CORE_STATE[3]	
	ALT6	EXTMC	EMI_DEBUG[3]	
	ALT7	USBPHY-1	IDDIG	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DI0_PIN4	ALT0	IPU	DI0_PIN4	IOMUXC_SW_PAD_CTL_PAD_DI0_PIN4 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[20]	
	ALT2	AUDMUX	AUD6_RXD	
	ALT3	ESDHV2-1	WP	
	ALT5	SDMA	DEBUG_YIELD	
	ALT6	EXTMC	EMI_DEBUG[4]	
	ALT7	USBPHY-1	HOSTDISCONNECT	
DISP0_ DAT0	ALT0	IPU	DISP0_DAT[0]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT0 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[21]	
	ALT2	CSPI	SCLK	
	ALT3	USB	USBH2_DATA[0]	
	ALT5	SDMA	DEBUG_CORE_RUN	
	ALT6	EXTMC	EMI_DEBUG[5]	
	ALT7	USBPHY-2	TXREADY	
DISP0_ DAT1	ALT0	IPU	DISP0_DAT[1]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT1 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[22]	
	ALT2	CSPI	MOSI	
	ALT3	USB	USBH2_DATA[1]	
	ALT5	SDMA	DEBUG_EVENT_ CHANNEL_SEL	
	ALT6	EXTMC	EMI_DEBUG[6]	
	ALT7	USBPHY-2	RXVALID	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DISP0_ DAT2	ALT0	IPU	DISP0_DAT[2]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT2 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[23]	
	ALT2	CSPI	MISO	
	ALT3	USB	USBH2_DATA[2]	
	ALT5	SDMA	DEBUG_MODE	
	ALT6	EXTMC	EMI_DEBUG[7]	
	ALT7	USBPHY-2	RXACTIVE	
DISP0_ DAT3	ALT0	IPU	DISP0_DAT[3]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT3 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[24]	
	ALT2	CSPI	SS0	
	ALT3	USB	USBH2_DATA[3]	
	ALT5	SDMA	DEBUG_BUS_ERROR	
	ALT6	EXTMC	EMI_DEBUG[8]	
	ALT7	USBPHY-2	RXERROR	
DISP0_ DAT4	ALT0	IPU	DISP0_DAT[4]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT4 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[25]	
	ALT2	CSPI	SS1	
	ALT3	USB	USBH2_DATA[4]	
	ALT5	SDMA	DEBUG_BUS_RWB	
	ALT6	EXTMC	EMI_DEBUG[9]	
	ALT7	USBPHY-2	SIECLOCK	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DISP0_ DAT5	ALT0	IPU	DISP0_DAT[5]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT5 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[26]	
	ALT2	CSPI	SS2	
	ALT3	USB	USBH2_DATA[5]	
	ALT5	SDMA	DEBUG_MATCHED_ DMBUS	
	ALT6	EXTMC	EMI_DEBUG[10]	
	ALT7	USBPHY-2	LINESTATE[0]	
DISP0_ DAT6	ALT0	IPU	DISP0_DAT[6]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT6 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[27]	
	ALT2	CSPI	SS3	
	ALT3	USB	USBH2_DATA[6]	
	ALT5	SDMA	DEBUG_RTBUFFER_ WRITE	
	ALT6	EXTMC	EMI_DEBUG[11]	
	ALT7	USBPHY-2	LINESTATE[1]	
DISP0_ DAT7	ALT0	IPU	DISP0_DAT[7]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT7 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[28]	
	ALT2	CSPI	RDY	
	ALT3	USB	USBH2_DATA[7]	
	ALT5	SDMA	DEBUG_EVENT_ CHANNEL[0]	
	ALT6	EXTMC	EMI_DEBUG[12]	
	ALT7	USBPHY-2	VBUSVALID	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DISP0_ DAT8	ALT0	IPU	DISP0_DAT[8]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT8 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[29]	
	ALT2	PWM-1	PWMO	
	ALT3	WDOG-1	WDOG_B	
	ALT5	SDMA	DEBUG_EVENT_ CHANNEL[1]	
	ALT6	EXTMC	EMI_DEBUG[13]	
	ALT7	USBPHY-2	AVALID	
DISP0_ DAT9	ALT0	IPU	DISP0_DAT[9]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT9 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[30]	
	ALT2	PWM-2	PWMO	
	ALT3	WDOG-2	WDOG_B	
	ALT5	SDMA	DEBUG_EVENT_ CHANNEL[2]	
	ALT6	EXTMC	EMI_DEBUG[14]	
	ALT7	USBPHY-2	VSTATUS[0]	
DISP0_ DAT10	ALT0	IPU	DISP0_DAT[10]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT10 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-4	GPIO[31]	
	ALT2	USB	USBH2_STP	
	ALT5	SDMA	DEBUG_EVENT_ CHANNEL[3]	
	ALT6	EXTMC	EMI_DEBUG[15]	
	ALT7	USBPHY-2	VSTATUS[1]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DISP0_ DAT11	ALT0	IPU	DISP0_DAT[11]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT11 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[5]	
	ALT2	USB	USBH2_NXT	
	ALT5	SDMA	DEBUG_EVENT_ CHANNEL[4]	
	ALT6	EXTMC	EMI_DEBUG[16]	
	ALT7	USBPHY-2	VSTATUS[2]	
DISP0_ DAT12	ALT0	IPU	DISP0_DAT[12]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT12 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[6]	
	ALT2	USB	USBH2_CLK	
	ALT5	SDMA	DEBUG_EVENT_ CHANNEL[5]	
	ALT6	EXTMC	EMI_DEBUG[17]	
	ALT7	USBPHY-2	VSTATUS[3]	
DISP0_ DAT13	ALT0	IPU	DISP0_DAT[13]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT13 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[7]	
	ALT3	AUDMUX	AUD5_RXFS	
	ALT5	SDMA	DEBUG_EVT_CHN_ LINES[0]	
	ALT6	EXTMC	EMI_DEBUG[18]	
	ALT7	USBPHY-2	VSTATUS[4]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DISP0_ DAT14	ALT0	IPU	DISP0_DAT[14]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT14 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[8]	
	ALT3	AUDMUX	AUD5_RXC	
	ALT5	SDMA	DEBUG_EVT_CHN_ LINES[1]	
	ALT6	EXTMC	EMI_DEBUG[19]	
	ALT7	USBPHY-2	VSTATUS[5]	
DISP0_ DAT15	ALT0	IPU	DISP0_DAT[15]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT15 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[9]	
	ALT2	ECSPI-1	SS1	
	ALT3	ECSPI-2	SS1	
	ALT5	SDMA	DEBUG_EVT_CHN_ LINES[2]	
	ALT6	EXTMC	EMI_DEBUG[20]	
	ALT7	USBPHY-2	VSTATUS[6]	
DISP0_ DAT16	ALT0	IPU	DISP0_DAT[16]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT16 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[10]	
	ALT2	ECSPI-2	MOSI	
	ALT3	AUDMUX	AUD5_TXC	
	ALT4	SDMA	SDMA_EXT_EVENT[0]	
	ALT5	SDMA	DEBUG_EVT_CHN_ LINES[3]	
	ALT6	EXTMC	EMI_DEBUG[21]	
	ALT7	USBPHY-2	VSTATUS[7]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DISP0_ DAT17	ALT0	IPU	DISP0_DAT[17]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT17 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[11]	
	ALT2	ECSPI-2	MISO	
	ALT3	AUDMUX	AUD5_TXD	
	ALT4	SDMA	SDMA_EXT_EVENT[1]	
	ALT5	SDMA	DEBUG_EVT_CHN_ LINES[4]	
	ALT6	EXTMC	EMI_DEBUG[22]	
DISP0_ DAT18	ALT0	IPU	DISP0_DAT[18]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT18 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[12]	
	ALT2	ECSPI-2	SS0	
	ALT3	AUDMUX	AUD5_TXFS	
	ALT4	AUDMUX	AUD4_RXFS	
	ALT5	SDMA	DEBUG_EVT_CHN_ LINES[5]	
	ALT6	EXTMC	EMI_DEBUG[23]	
	ALT7	EXTMC	WEIM_CS[2]	
DISP0_ DAT19	ALT0	IPU	DISP0_DAT[19]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT19 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[13]	
	ALT2	ECSPI-2	SCLK	
	ALT3	AUDMUX	AUD5_RXD	
	ALT4	AUDMUX	AUD4_RXC	
	ALT5	SDMA	DEBUG_EVT_CHN_ LINES[6]	
	ALT6	EXTMC	EMI_DEBUG[24]	
	ALT7	EXTMC	WEIM_CS[3]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DISP0_ DAT20	ALT0	IPU	DISP0_DAT[20]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT20 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[14]	
	ALT2	ECSPI-1	SCLK	
	ALT3	AUDMUX	AUD4_TXC	
	ALT5	SDMA	DEBUG_EVT_CHN_ LINES[7]	
	ALT6	EXTMC	EMI_DEBUG[25]	
	ALT7	SATA_PHY	TDI	
DISP0_ DAT21	ALT0	IPU	DISP0_DAT[21]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT21 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[15]	
	ALT2	ECSPI-1	MOSI	
	ALT3	AUDMUX	AUD4_TXD	
	ALT5	SDMA	DEBUG_BUS_DEVICE[0]	
	ALT6	EXTMC	EMI_DEBUG[26]	
	ALT7	SATA_PHY	TDO	
DISP0_ DAT22	ALT0	IPU	DISP0_DAT[22]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT22 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[16]	
	ALT2	ECSPI-1	MISO	
	ALT3	AUDMUX	AUD4_TXFS	
	ALT5	SDMA	DEBUG_BUS_DEVICE[1]	
	ALT6	EXTMC	EMI_DEBUG[27]	
	ALT7	SATA_PHY	TCK	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DISP0_ DAT23	ALT0	IPU	DISP0_DAT[23]	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT23 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[17]	
	ALT2	ECSPI-1	SS0	
	ALT3	AUDMUX	AUD4_RXD	
	ALT5	SDMA	DEBUG_BUS_DEVICE[2]	
	ALT6	EXTMC	EMI_DEBUG[28]	
	ALT7	SATA_PHY	TMS	
CSI0_ PIXCLK	ALT0	IPU	CSI0_PIXCLK	IOMUXC_SW_PAD_CTL_PAD_CSI0_PIXCLK Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[18]	
	ALT5	SDMA	DEBUG_PC[0]	
	ALT6	EXTMC	EMI_DEBUG[29]	
CSI0_ MCLK	ALT0	IPU	CSI0_HSYNC	IOMUXC_SW_PAD_CTL_PAD_CSI0_MCLK Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[19]	
	ALT2	CCM	CSI0_MCLK	
	ALT5	SDMA	DEBUG_PC[1]	
	ALT6	EXTMC	EMI_DEBUG[30]	
	ALT7	TPIU	TRCTL	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
CSI0_ DATA_EN	ALT0	IPU	CSI0_DATA_EN	IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA_EN Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[20]	
	ALT5	SDMA	DEBUG_PC[2]	
	ALT6	EXTMC	EMI_DEBUG[31]	
	ALT7	TPIU	TRCLK	
CSI0_ VSYNC	ALT0	IPU	CSI0_VSYNC	IOMUXC_SW_PAD_CTL_PAD_CSI0_VSYNC Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[21]	
	ALT5	SDMA	DEBUG_PC[3]	
	ALT6	EXTMC	EMI_DEBUG[32]	
	ALT7	TPIU	TRACE[0]	
CSI0_ DAT4	ALT0	IPU	CSI0_D[4]	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT4 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[22]	
	ALT2	KPP	COL[5]	
	ALT3	ECSPI-1	SCLK	
	ALT4	USB	USBH3_STP	
	ALT5	AUDMUX	AUD3_TXC	
	ALT6	EXTMC	EMI_DEBUG[33]	
	ALT7	TPIU	TRACE[1]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
CSI0_ DAT5	ALT0	IPU	CSI0_D[5]	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT5 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[23]	
	ALT2	KPP	ROW[5]	
	ALT3	ECSPI-1	MOSI	
	ALT4	USB	USBH3_NXT	
	ALT5	AUDMUX	AUD3_TXD	
	ALT6	EXTMC	EMI_DEBUG[34]	
	ALT7	TPIU	TRACE[2]	
CSI0_ DAT6	ALT0	IPU	CSI0_D[6]	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT6 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[24]	
	ALT2	KPP	COL[6]	
	ALT3	ECSPI-1	MISO	
	ALT4	USB	USBH3_CLK	
	ALT5	AUDMUX	AUD3_TXFS	
	ALT6	EXTMC	EMI_DEBUG[35]	
	ALT7	TPIU	TRACE[3]	
CSI0_ DAT7	ALT0	IPU	CSI0_D[7]	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT7 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[25]	
	ALT2	KPP	ROW[6]	
	ALT3	ECSPI-1	SS0	
	ALT4	USB	USBH3_DIR	
	ALT5	AUDMUX	AUD3_RXD	
	ALT6	EXTMC	EMI_DEBUG[36]	
	ALT7	TPIU	TRACE[4]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
CSI0_ DAT8	ALT0	IPU	CSI0_D[8]	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT8 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[26]	
	ALT2	KPP	COL[7]	
	ALT3	ECSPI-2	SCLK	
	ALT4	USB	USBH3_OC	
	ALT5	I2C-1	SDA	
	ALT6	EXTMC	EMI_DEBUG[37]	
	ALT7	TPIU	TRACE[5]	
CSI0_ DAT9	ALT0	IPU	CSI0_D[9]	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT9 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[27]	
	ALT2	KPP	ROW[7]	
	ALT3	ECSPI-2	MOSI	
	ALT4	USB	USBH3_PWR	
	ALT5	I2C-1	SCL	
	ALT6	EXTMC	EMI_DEBUG[38]	
	ALT7	TPIU	TRACE[6]	
CSI0_ DAT10	ALT0	IPU	CSI0_D[10]	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT10 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[28]	
	ALT2	UART-1	TXD_MUX	
	ALT3	ECSPI-2	MISO	
	ALT4	AUDMUX	AUD3_RXC	
	ALT5	SDMA	DEBUG_PC[4]	
	ALT6	EXTMC	EMI_DEBUG[39]	
	ALT7	TPIU	TRACE[7]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
CSI0_ DAT11	ALT0	IPU	CSI0_D[11]	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT11 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[29]	
	ALT2	UART-1	RXD_MUX	
	ALT3	ECSPI-2	SS0	
	ALT4	AUDMUX	AUD3_RXFS	
	ALT5	SDMA	DEBUG_PC[5]	
	ALT6	EXTMC	EMI_DEBUG[40]	
	ALT7	TPIU	TRACE[8]	
CSI0_ DAT12	ALT0	IPU	CSI0_D[12]	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT12 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[30]	
	ALT2	UART-4	TXD_MUX	
	ALT4	USB	USBH3_DATA[0]	
	ALT5	SDMA	DEBUG_PC[6]	
	ALT6	EXTMC	EMI_DEBUG[41]	
	ALT7	TPIU	TRACE[9]	
CSI0_ DAT13	ALT0	IPU	CSI0_D[13]	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT13 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[31]	
	ALT2	UART-4	RXD_MUX	
	ALT4	USB	USBH3_DATA[1]	
	ALT5	SDMA	DEBUG_PC[7]	
	ALT6	EXTMC	EMI_DEBUG[42]	
	ALT7	TPIU	TRACE[10]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
CSI0_ DAT14	ALT0	IPU	CSI0_D[14]	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT14 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[0]	
	ALT2	UART-5	TXD_MUX	
	ALT4	USB	USBH3_DATA[2]	
	ALT5	SDMA	DEBUG_PC[8]	
	ALT6	EXTMC	EMI_DEBUG[43]	
	ALT7	TPIU	TRACE[11]	
CSI0_ DAT15	ALT0	IPU	CSI0_D[15]	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT15 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[1]	
	ALT2	UART-5	RXD_MUX	
	ALT4	USB	USBH3_DATA[3]	
	ALT5	SDMA	DEBUG_PC[9]	
	ALT6	EXTMC	EMI_DEBUG[44]	
	ALT7	TPIU	TRACE[12]	
CSI0_ DAT16	ALT0	IPU	CSI0_D[16]	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT16 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[2]	
	ALT2	UART-4	RTS	
	ALT4	USB	USBH3_DATA[4]	
	ALT5	SDMA	DEBUG_PC[10]	
	ALT6	EXTMC	EMI_DEBUG[45]	
	ALT7	TPIU	TRACE[13]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
CSI0_ DAT17	ALT0	IPU	CSI0_D[17]	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT17 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[3]	
	ALT2	UART-4	CTS	
	ALT4	USB	USBH3_DATA[5]	
	ALT5	SDMA	DEBUG_PC[11]	
	ALT6	EXTMC	EMI_DEBUG[46]	
	ALT7	TPIU	TRACE[14]	
CSI0_ DAT18	ALT0	IPU	CSI0_D[18]	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT18 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[4]	
	ALT2	UART-5	RTS	
	ALT4	USB	USBH3_DATA[6]	
	ALT5	SDMA	DEBUG_PC[12]	
	ALT6	EXTMC	EMI_DEBUG[47]	
	ALT7	TPIU	TRACE[15]	
CSI0_ DAT19	ALT0	IPU	CSI0_D[19]	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT19 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[5]	
	ALT2	UART-5	CTS	
	ALT4	USB	USBH3_DATA[7]	
	ALT5	SDMA	DEBUG_PC[13]	
	ALT6	EXTMC	EMI_DEBUG[48]	
	ALT7	USBPHY-2	BISTOK	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
JTAG_ TMS	-	SJC	TMS	IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS Drive Strength (DSE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 47K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = N/A Pull / Keep Enable (PKE) =Enabled Slew Rate () = N/A TEST_TS = Disabled
JTAG_ MOD	-	SJC	MOD	IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD Drive Strength (DSE) = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = N/A Pull / Keep Enable (PKE) =Enabled Slew Rate () = N/A TEST_TS = Disabled
JTAG_ TRSTB	-	SJC	TRSTB	IOMUXC_SW_PAD_CTL_PAD_JTAG_TRSTB Drive Strength (DSE) = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 47K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = N/A Pull / Keep Enable (PKE) =Enabled Slew Rate () = N/A TEST_TS = Disabled

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
JTAG_TDI	-	SJC	TDI	IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI Drive Strength (DSE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 47K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = N/A Pull / Keep Enable (PKE) =Enabled Slew Rate () = N/A TEST_TS = Disabled
JTAG_ TCK	-	SJC	TCK	IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK Drive Strength (DSE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = N/A Pull / Keep Enable (PKE) =Enabled Slew Rate () = N/A TEST_TS = Disabled
JTAG_ TDO	-	SJC	TDO	IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO Drive Strength (DSE) = High Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Keep Pull Up / Down (PUS) = N/A Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_A25	ALT0	EXTMC	WEIM_A[25]	IOMUXC_SW_PAD_CTL_PAD_EIM_A25 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[2]	
	ALT2	ECSPI-2	RDY	
	ALT3	IPU	DI1_PIN12	
	ALT4	CSPI	SS1	
	ALT6	IPU	DI0_D1_CS	
	ALT7	USBPHY-1	BISTOK	
EIM_EB2	ALT0	EXTMC	WEIM_EB[2]	IOMUXC_SW_PAD_CTL_PAD_EIM_EB2 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[30]	
	ALT2	CCM	DI1_EXT_CLK	
	ALT3	IPU	SER_DISP1_CS	
	ALT4	ECSPI-1	SS0	
	ALT5	I2C-2	SCL	
EIM_D16	ALT0	EXTMC	WEIM_D[16]	IOMUXC_SW_PAD_CTL_PAD_EIM_D16 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[16]	
	ALT2	IPU	DI0_PIN5	
	ALT3	IPU	DISPB1_SER_CLK	
	ALT4	ECSPI-1	SCLK	
	ALT5	I2C-2	SDA	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_D17	ALT0	EXTMC	WEIM_D[17]	IOMUXC_SW_PAD_CTL_PAD_EIM_D17 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[17]	
	ALT2	IPU	DI0_PIN6	
	ALT3	IPU	DISPB1_SER_DIN	
	ALT4	ECSPI-1	MISO	
	ALT5	I2C-3	SCL	
EIM_D18	ALT0	EXTMC	WEIM_D[18]	IOMUXC_SW_PAD_CTL_PAD_EIM_D18 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[18]	
	ALT2	IPU	DI0_PIN7	
	ALT3	IPU	DISPB1_SER_DIO	
	ALT4	ECSPI-1	MOSI	
	ALT5	I2C-3	SDA	
	ALT6	IPU	DI1_D0_CS	
EIM_D19	ALT0	EXTMC	WEIM_D[19]	IOMUXC_SW_PAD_CTL_PAD_EIM_D19 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[19]	
	ALT2	IPU	DI0_PIN8	
	ALT3	IPU	DISPB1_SER_RS	
	ALT4	ECSPI-1	SS1	
	ALT5	EPIT-1	EPITO	
	ALT6	UART-1	CTS	
	ALT7	USB	USBH2_OC	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_D20	ALT0	EXTMC	WEIM_D[20]	IOMUXC_SW_PAD_CTL_PAD_EIM_D20 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[20]	
	ALT2	IPU	DI0_PIN16	
	ALT3	IPU	SER_DISP0_CS	
	ALT4	CSPI	SS0	
	ALT5	EPIT-2	EPITO	
	ALT6	UART-1	RTS	
	ALT7	USB	USBH2_PWR	
EIM_D21	ALT0	EXTMC	WEIM_D[21]	IOMUXC_SW_PAD_CTL_PAD_EIM_D21 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[21]	
	ALT2	IPU	DI0_PIN17	
	ALT3	IPU	DISPB0_SER_CLK	
	ALT4	CSPI	SCLK	
	ALT5	I2C-1	SCL	
	ALT6	USB	USBOTG_OC	
EIM_D22	ALT0	EXTMC	WEIM_D[22]	IOMUXC_SW_PAD_CTL_PAD_EIM_D22 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[22]	
	ALT2	IPU	DI0_PIN1	
	ALT3	IPU	DISPB0_SER_DIN	
	ALT4	CSPI	MISO	
	ALT6	USB	USBOTG_PWR	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_D23	ALT0	EXTMC	WEIM_D[23]	IOMUXC_SW_PAD_CTL_PAD_EIM_D23 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[23]	
	ALT2	UART-3	CTS	
	ALT3	UART-1	DCD	
	ALT4	IPU	DI0_D0_CS	
	ALT5	IPU	DI1_PIN2	
	ALT6	IPU	CSI1_DATA_EN	
	ALT7	IPU	DI1_PIN14	
EIM_EB3	ALT0	EXTMC	WEIM_EB[3]	IOMUXC_SW_PAD_CTL_PAD_EIM_EB3 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[31]	
	ALT2	UART-3	RTS	
	ALT3	UART-1	RI	
	ALT5	IPU	DI1_PIN3	
	ALT6	IPU	CSI1_HSYNC	
	ALT7	IPU	DI1_PIN16	
EIM_D24	ALT0	EXTMC	WEIM_D[24]	IOMUXC_SW_PAD_CTL_PAD_EIM_D24 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[24]	
	ALT2	UART-3	TXD_MUX	
	ALT3	ECSPI-1	SS2	
	ALT4	CSPI	SS2	
	ALT5	AUDMUX	AUD5_RXFS	
	ALT6	ECSPI-2	SS2	
	ALT7	UART-1	DTR	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_D25	ALT0	EXTMC	WEIM_D[25]	IOMUXC_SW_PAD_CTL_PAD_EIM_D25 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[25]	
	ALT2	UART-3	RXD_MUX	
	ALT3	ECSPI-1	SS3	
	ALT4	CSPI	SS3	
	ALT5	AUDMUX	AUD5_RXC	
	ALT6	ECSPI-2	SS3	
	ALT7	UART-1	DSR	
EIM_D26	ALT0	EXTMC	WEIM_D[26]	IOMUXC_SW_PAD_CTL_PAD_EIM_D26 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[26]	
	ALT2	UART-2	TXD_MUX	
	ALT3	FIRI	RXD	
	ALT4	IPU	CSI0_D[1]	
	ALT5	IPU	DI1_PIN11	
	ALT6	IPU	SISG[2]	
	ALT7	IPU	DISP1_DAT[22]	
EIM_D27	ALT0	EXTMC	WEIM_D[27]	IOMUXC_SW_PAD_CTL_PAD_EIM_D27 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[27]	
	ALT2	UART-2	RXD_MUX	
	ALT3	FIRI	TXD	
	ALT4	IPU	CSI0_D[0]	
	ALT5	IPU	DI1_PIN13	
	ALT6	IPU	SISG[3]	
	ALT7	IPU	DISP1_DAT[23]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_D28	ALT0	EXTMC	WEIM_D[28]	IOMUXC_SW_PAD_CTL_PAD_EIM_D28 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[28]	
	ALT2	UART-2	CTS	
	ALT3	IPU	DISPB0_SER_DIO	
	ALT4	CSPI	MOSI	
	ALT5	I2C-1	SDA	
	ALT6	IPU	EXT_TRIG	
	ALT7	IPU	DI0_PIN13	
EIM_D29	ALT0	EXTMC	WEIM_D[29]	IOMUXC_SW_PAD_CTL_PAD_EIM_D29 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[29]	
	ALT2	UART-2	RTS	
	ALT3	IPU	DISPB0_SER_RS	
	ALT4	CSPI	SS0	
	ALT5	IPU	DI1_PIN15	
	ALT6	IPU	CSI1_VSYNC	
	ALT7	IPU	DI0_PIN14	
EIM_D30	ALT0	EXTMC	WEIM_D[30]	IOMUXC_SW_PAD_CTL_PAD_EIM_D30 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[30]	
	ALT2	UART-3	CTS	
	ALT3	IPU	CSI0_D[3]	
	ALT4	IPU	DI0_PIN11	
	ALT5	IPU	DISP1_DAT[21]	
	ALT6	USB	USBH1_OC	
	ALT7	USB	USBH2_OC	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_D31	ALT0	EXTMC	WEIM_D[31]	IOMUXC_SW_PAD_CTL_PAD_EIM_D31 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[31]	
	ALT2	UART-3	RTS	
	ALT3	IPU	CSI0_D[2]	
	ALT4	IPU	DI0_PIN12	
	ALT5	IPU	DISP1_DAT[20]	
	ALT6	USB	USBH1_PWR	
	ALT7	USB	USBH2_PWR	
EIM_A24	ALT0	EXTMC	WEIM_A[24]	IOMUXC_SW_PAD_CTL_PAD_EIM_A24 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[4]	
	ALT2	IPU	DISP1_DAT[19]	
	ALT3	IPU	CSI1_D[19]	
	ALT6	IPU	SISG[2]	
	ALT7	USBPHY-2	BVALID	
EIM_A23	ALT0	EXTMC	WEIM_A[23]	IOMUXC_SW_PAD_CTL_PAD_EIM_A23 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[6]	
	ALT2	IPU	DISP1_DAT[18]	
	ALT3	IPU	CSI1_D[18]	
	ALT6	IPU	SISG[3]	
	ALT7	USBPHY-2	ENDSESSION	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_A22	ALT0	EXTMC	WEIM_A[22]	IOMUXC_SW_PAD_CTL_PAD_EIM_A22 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[16]	
	ALT2	IPU	DISP1_DAT[17]	
	ALT3	IPU	CSI1_D[17]	
	ALT7	SRC	BT_CFG1[7]	
EIM_A21	ALT0	EXTMC	WEIM_A[21]	IOMUXC_SW_PAD_CTL_PAD_EIM_A21 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[17]	
	ALT2	IPU	DISP1_DAT[16]	
	ALT3	IPU	CSI1_D[16]	
	ALT7	SRC	BT_CFG1[6]	
EIM_A20	ALT0	EXTMC	WEIM_A[20]	IOMUXC_SW_PAD_CTL_PAD_EIM_A20 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[18]	
	ALT2	IPU	DISP1_DAT[15]	
	ALT3	IPU	CSI1_D[15]	
	ALT7	SRC	BT_CFG1[5]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_A19	ALT0	EXTMC	WEIM_A[19]	IOMUXC_SW_PAD_CTL_PAD_EIM_A19 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[19]	
	ALT2	IPU	DISP1_DAT[14]	
	ALT3	IPU	CSI1_D[14]	
	ALT7	SRC	BT_CFG1[4]	
EIM_A18	ALT0	EXTMC	WEIM_A[18]	IOMUXC_SW_PAD_CTL_PAD_EIM_A18 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[20]	
	ALT2	IPU	DISP1_DAT[13]	
	ALT3	IPU	CSI1_D[13]	
	ALT7	SRC	BT_CFG1[3]	
EIM_A17	ALT0	EXTMC	WEIM_A[17]	IOMUXC_SW_PAD_CTL_PAD_EIM_A17 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[21]	
	ALT2	IPU	DISP1_DAT[12]	
	ALT3	IPU	CSI1_D[12]	
	ALT7	SRC	BT_CFG1[2]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_A16	ALT0	EXTMC	WEIM_A[16]	IOMUXC_SW_PAD_CTL_PAD_EIM_A16 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[22]	
	ALT2	IPU	DI1_DISP_CLK	
	ALT3	IPU	CSI1_PIXCLK	
	ALT7	SRC	BT_CFG1[1]	
EIM_CS0	ALT0	EXTMC	WEIM_CS[0]	IOMUXC_SW_PAD_CTL_PAD_EIM_CS0 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[23]	
	ALT2	ECSPI-2	SCLK	
	ALT3	IPU	DI1_PIN5	
EIM_CS1	ALT0	EXTMC	WEIM_CS[1]	IOMUXC_SW_PAD_CTL_PAD_EIM_CS1 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[24]	
	ALT2	ECSPI-2	MOSI	
	ALT3	IPU	DI1_PIN6	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_OE	ALT0	EXTMC	WEIM_OE	IOMUXC_SW_PAD_CTL_PAD_EIM_OE Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[25]	
	ALT2	ECSPI-2	MISO	
	ALT3	IPU	DI1_PIN7	
	ALT7	USBPHY-2	IDDIG	
EIM_RW	ALT0	EXTMC	WEIM_RW	IOMUXC_SW_PAD_CTL_PAD_EIM_RW Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[26]	
	ALT2	ECSPI-2	SS0	
	ALT3	IPU	DI1_PIN8	
	ALT7	USBPHY-2	HOSTDISCONNECT	
EIM_LBA	ALT0	EXTMC	WEIM_LBA	IOMUXC_SW_PAD_CTL_PAD_EIM_LBA Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[27]	
	ALT2	ECSPI-2	SS1	
	ALT3	IPU	DI1_PIN17	
	ALT7	SRC	BT_CFG1[0]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_EB0	ALT0	EXTMC	WEIM_EB[0]	IOMUXC_SW_PAD_CTL_PAD_EIM_EB0 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[28]	
	ALT3	IPU	DISP1_DAT[11]	
	ALT4	IPU	CSI1_D[11]	
	ALT5	GPC	PMIC_RDY	
	ALT7	SRC	BT_CFG2[7]	
EIM_EB1	ALT0	EXTMC	WEIM_EB[1]	IOMUXC_SW_PAD_CTL_PAD_EIM_EB1 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[29]	
	ALT3	IPU	DISP1_DAT[10]	
	ALT4	IPU	CSI1_D[10]	
	ALT7	SRC	BT_CFG2[6]	
EIM_DA0	ALT0	EXTMC	NAND_WEIM_DA[0]	IOMUXC_SW_PAD_CTL_PAD_EIM_DA0 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[0]	
	ALT3	IPU	DISP1_DAT[9]	
	ALT4	IPU	CSI1_D[9]	
	ALT7	SRC	BT_CFG2[5]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_DA1	ALT0	EXTMC	NAND_WEIM_DA[1]	IOMUXC_SW_PAD_CTL_PAD_EIM_DA1 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[1]	
	ALT3	IPU	DISP1_DAT[8]	
	ALT4	IPU	CSI1_D[8]	
	ALT7	SRC	BT_CFG2[4]	
EIM_DA2	ALT0	EXTMC	NAND_WEIM_DA[2]	IOMUXC_SW_PAD_CTL_PAD_EIM_DA2 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[2]	
	ALT3	IPU	DISP1_DAT[7]	
	ALT4	IPU	CSI1_D[7]	
	ALT7	SRC	BT_CFG2[3]	
EIM_DA3	ALT0	EXTMC	NAND_WEIM_DA[3]	IOMUXC_SW_PAD_CTL_PAD_EIM_DA3 Drive Strength (DSE) = High Low/high output voltage = NA Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[3]	
	ALT3	IPU	DISP1_DAT[6]	
	ALT4	IPU	CSI1_D[6]	
	ALT7	SRC	BT_CFG2[2]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_DA4	ALT0	EXTMC	NAND_WEIM_DA[4]	IOMUXC_SW_PAD_CTL_PAD_EIM_DA4 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[4]	
	ALT3	IPU	DISP1_DAT[5]	
	ALT4	IPU	CSI1_D[5]	
	ALT7	SRC	BT_CFG3[7]	
EIM_DA5	ALT0	EXTMC	NAND_WEIM_DA[5]	IOMUXC_SW_PAD_CTL_PAD_EIM_DA5 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[5]	
	ALT3	IPU	DISP1_DAT[4]	
	ALT4	IPU	CSI1_D[4]	
	ALT7	SRC	BT_CFG3[6]	
EIM_DA6	ALT0	EXTMC	NAND_WEIM_DA[6]	IOMUXC_SW_PAD_CTL_PAD_EIM_DA6 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[6]	
	ALT3	IPU	DISP1_DAT[3]	
	ALT4	IPU	CSI1_D[3]	
	ALT7	SRC	BT_CFG3[5]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_DA7	ALT0	EXTMC	NAND_WEIM_DA[7]	IOMUXC_SW_PAD_CTL_PAD_EIM_DA7 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[7]	
	ALT3	IPU	DISP1_DAT[2]	
	ALT4	IPU	CSI1_D[2]	
	ALT7	SRC	BT_CFG3[4]	
EIM_DA8	ALT0	EXTMC	NAND_WEIM_DA[8]	IOMUXC_SW_PAD_CTL_PAD_EIM_DA8 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[8]	
	ALT3	IPU	DISP1_DAT[1]	
	ALT4	IPU	CSI1_D[1]	
	ALT7	SRC	BT_CFG3[3]	
EIM_DA9	ALT0	EXTMC	NAND_WEIM_DA[9]	IOMUXC_SW_PAD_CTL_PAD_EIM_DA9 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[9]	
	ALT3	IPU	DISP1_DAT[0]	
	ALT4	IPU	CSI1_D[0]	
	ALT7	SRC	BT_CFG3[2]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_DA10	ALT0	EXTMC	NAND_WEIM_DA[10]	IOMUXC_SW_PAD_CTL_PAD_EIM_DA10 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[10]	
	ALT3	IPU	DI1_PIN15	
	ALT4	IPU	CSI1_DATA_EN	
	ALT7	SRC	BT_CFG3[1]	
EIM_DA11	ALT0	EXTMC	NAND_WEIM_DA[11]	IOMUXC_SW_PAD_CTL_PAD_EIM_DA11 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[11]	
	ALT3	IPU	DI1_PIN2	
	ALT4	IPU	CSI1_HSYNC	
EIM_DA12	ALT0	EXTMC	NAND_WEIM_DA[12]	IOMUXC_SW_PAD_CTL_PAD_EIM_DA12 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[12]	
	ALT3	IPU	DI1_PIN3	
	ALT4	IPU	CSI1_VSYNC	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_DA13	ALT0	EXTMC	NAND_WEIM_DA[13]	IOMUXC_SW_PAD_CTL_PAD_EIM_DA13 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[13]	
	ALT3	IPU	DI1_D0_CS	
	ALT4	CCM	DI1_EXT_CLK	
EIM_DA14	ALT0	EXTMC	NAND_WEIM_DA[14]	IOMUXC_SW_PAD_CTL_PAD_EIM_DA14 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[14]	
	ALT3	IPU	DI1_D1_CS	
	ALT4	CCM	DI0_EXT_CLK	
EIM_DA15	ALT0	EXTMC	NAND_WEIM_DA[15]	IOMUXC_SW_PAD_CTL_PAD_EIM_DA15 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-3	GPIO[15]	
	ALT3	IPU	DI1_PIN1	
	ALT4	IPU	DI1_PIN4	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
NANDF_WE_B	ALT0	EXTMC	NANDF_WE_B	IOMUXC_SW_PAD_CTL_PAD_NANDF_WE_B Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[12]	
NANDF_RE_B	ALT0	EXTMC	NANDF_RE_B	IOMUXC_SW_PAD_CTL_PAD_NANDF_RE_B Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[13]	
EIM_WAIT	ALT0	EXTMC	WEIM_WAIT	IOMUXC_SW_PAD_CTL_PAD_EIM_WAIT Drive Strength (DSE) = Low Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-5	GPIO[0]	
	ALT2	EXTMC	WEIM_DTACK_B	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
EIM_BCLK	-	EXTMC	WEIM_BCLK	IOMUXC_SW_PAD_CTL_PAD_EIM_BCLK Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = N/A DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
LVDS1_TX3_P	ALT0	GPIO-6	GPI[22]	
	ALT1	LDB	LVDS1_TX3	
LVDS1_TX3_N	-	GPIO-6	GPI[23]	
LVDS1_TX2_P	ALT0	GPIO-6	GPI[24]	
	ALT1	LDB	LVDS1_TX2	
LVDS1_TX2_N	-	GPIO-6	GPI[25]	
LVDS1_CLK_P	ALT0	GPIO-6	GPI[26]	
	ALT1	LDB	LVDS1_CLK	
LVDS1_CLK_N	-	GPIO-6	GPI[27]	
LVDS1_TX1_P	ALT0	GPIO-6	GPI[28]	
	ALT1	LDB	LVDS1_TX1	
LVDS1_TX1_N	-	GPIO-6	GPI[29]	
LVDS1_TX0_P	ALT0	GPIO-6	GPI[30]	
	ALT1	LDB	LVDS1_TX0	
LVDS1_TX0_N	-	GPIO-6	GPI[31]	
LVDS0_TX3_P	ALT0	GPIO-7	GPI[22]	
	ALT1	LDB	LVDS0_TX3	
LVDS0_TX3_N	-	GPIO-7	GPI[23]	
LVDS0_CLK_P	ALT0	GPIO-7	GPI[24]	
	ALT1	LDB	LVDS0_CLK	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
LVDS0_CLK_N	-	GPIO-7	GPI[25]	
LVDS0_TX2_P	ALT0	GPIO-7	GPI[26]	
	ALT1	LDB	LVDS0_TX2	
LVDS0_TX2_N	-	GPIO-7	GPI[27]	
LVDS0_TX1_P	ALT0	GPIO-7	GPI[28]	
	ALT1	LDB	LVDS0_TX1	
LVDS0_TX1_N	-	GPIO-7	GPI[29]	
LVDS0_TX0_P	ALT0	GPIO-7	GPI[30]	
	ALT1	LDB	LVDS0_TX0	
LVDS0_TX0_N	-	GPIO-7	GPI[31]	
GPIO_10	ALT0	GPIO-4	GPIO[0]	IOMUXC_SW_PAD_CTL_PAD_GPIO_10 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Enabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
	ALT1	XTALOSC32K	32K_OUT	
GPIO_11	-	GPIO-4	GPIO[1]	IOMUXC_SW_PAD_CTL_PAD_GPIO_11 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Enabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
GPIO_12	-	GPIO-4	GPIO[2]	IOMUXC_SW_PAD_CTL_PAD_GPIO_12/13
GPIO_13	-	GPIO-4	GPIO[3]	Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Enabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
GPIO_14	-	GPIO-4	GPIO[4]	IOMUXC_SW_PAD_CTL_PAD_GPIO_14 Drive Strength (DSE) = High Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Enabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
DRAM_D24	-	EXTMC	DRAM_D[24]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DRAM_D30	-	EXTMC	DRAM_D[30]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_D26	-	EXTMC	DRAM_D[26]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_DQM3	-	EXTMC	DRAM_DQM[3]	IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM3 Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Keep Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Disabled TEST_TS = Disabled On Die Termination = HiZ
DRAM_D28	-	EXTMC	DRAM_D[28]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DRAM_D25	-	EXTMC	DRAM_D[25]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_SDQS3_B	-	EXTMC	DRAM_SDQS_B[3]	See DRAM_SDQS3 settings.
DRAM_SDQS3	-	EXTMC	DRAM_SDQS[3]	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS3 Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Disabled TEST_TS = Disabled On Die Termination = Dependent on reference resistor
DRAM_D27	-	EXTMC	DRAM_D[27]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_D31	-	EXTMC	DRAM_D[31]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_D16	-	EXTMC	DRAM_D[16]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DRAM_D29	-	EXTMC	DRAM_D[29]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_D18	-	EXTMC	DRAM_D[18]	
DRAM_SDCKE1	-	EXTMC	DRAM_SDCKE[1]	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE1 Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled On Die Termination = HiZ
DRAM_D22	-	EXTMC	DRAM_D[22]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_DQM2	-	EXTMC	DRAM_DQM[2]	IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM2 Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Disabled TEST_TS = Disabled On Die Termination = HiZ

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DRAM_D20	-	EXTMC	DRAM_D[20]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_SDBA0	-	EXTMC	DRAM_SDBA[0]	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDBA0 Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_D17	-	EXTMC	DRAM_D[17]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_SDODT1	-	EXTMC	DRAM_ODT[1]	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT1 Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled On Die Termination = HiZ

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DRAM_D19	-	EXTMC	DRAM_D[19]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_SDQS2_B	-	EXTMC	DRAM_SDQS_B[2]	See DRAM_SDQS2 settings.
DRAM_SDQS2	-	EXTMC	DRAM_SDQS[2]	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS2 Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Keep Pull Up / Down (PUS) = 100K Ohm Pull Down DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Disabled TEST_TS = Disabled On Die Termination = Dependent on reference resistor
DRAM_D21	-	EXTMC	DRAM_D[21]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_CS1	-	EXTMC	DRAM_CS[1]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DRAM_D23	-	EXTMC	DRAM_D[23]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_RESET	-	EXTMC	DRAM_RESET	IOMUXC_SW_PAD_CTL_PAD_DRAM_RESET Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled On Die Termination = HiZ
DRAM_SDBA1	-	EXTMC	DRAM_SDBA[1]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_SDCLK_1_B	-	EXTMC	DRAM_SDCLK1_B	See DRAM_SDCL_K_1 settings.
DRAM_SDCLK_1	-	EXTMC	DRAM_SDCLK1	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_1 Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Keep Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = DDR2 input type Pull / Keep Enable (PKE) =Disabled TEST_TS = Disabled On Die Termination = HiZ

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DRAM_A8	-	EXTMC	DRAM_A[8]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_SDBA2	-	EXTMC	DRAM_SDBA[2]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_A14	-	EXTMC	DRAM_A[14]	
DRAM_A3	-	EXTMC	DRAM_A[3]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_A5	-	EXTMC	DRAM_A[5]	
DRAM_A7	-	EXTMC	DRAM_A[7]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_A6	-	EXTMC	DRAM_A[6]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DRAM_A9	-	EXTMC	DRAM_A[9]	IOMUXC_SW_PAD_CTL_GRP_*
DRAM_A2	-	EXTMC	DRAM_A[2]	Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_A0	-	EXTMC	DRAM_A[0]	IOMUXC_SW_PAD_CTL_GRP_*
DRAM_A15	-	EXTMC	DRAM_A[15]	Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_A13	-	EXTMC	DRAM_A[13]	IOMUXC_SW_PAD_CTL_GRP_*
DRAM_A11	-	EXTMC	DRAM_A[11]	Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_A1	-	EXTMC	DRAM_A[1]	IOMUXC_SW_PAD_CTL_GRP_*
DRAM_A12	-	EXTMC	DRAM_A[12]	Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DRAM_CAS	-	EXTMC	DRAM_CAS	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_SDWE	-	EXTMC	DRAM_SDWE	
DRAM_CS0	-	EXTMC	DRAM_CS[0]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_A4	-	EXTMC	DRAM_A[4]	
DRAM_SDCLK_0_B	-	EXTMC	DRAM_SDCLK0_B	See DRAM_SDCLK_0 settings
DRAM_SDCLK_0	-	EXTMC	DRAM_SDCLK0	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_0 Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Keeper Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = DDR2 input type Pull / Keep Enable (PKE) =Disabled TEST_TS = Disabled On Die Termination = HiZ
DRAM_A10	-	EXTMC	DRAM_A[10]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DRAM_D4	-	EXTMC	DRAM_D[4]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_D6	-	EXTMC	DRAM_D[6]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_D2	-	EXTMC	DRAM_D[2]	
DRAM_ SDQS0_B	-	EXTMC	DRAM_SDQS_B[0]	See DRAM_SDQS0 settings
DRAM_ SDQS0	-	EXTMC	DRAM_SDQS[0]	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS0 Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Disabled TEST_TS = Disabled On Die Termination = Dependent on Reference Resistor
DRAM_ SDODT0	-	EXTMC	DRAM_ODT[0]	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT0 Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled On Die Termination = HiZ

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DRAM_ DQM0	-	EXTMC	DRAM_DQM[0]	IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM0 Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Keep Pull Up / Down (PUS) = 100k Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Disabled TEST_TS = Disabled On Die Termination = HiZ
DRAM_ RAS	-	EXTMC	DRAM_RAS	IOMUXC_SW_PAD_CTL_PAD_DRAM_RAS Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Keep Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Disabled TEST_TS = Disabled On Die Termination = HiZ
DRAM_D5	-	EXTMC	DRAM_D[5]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_D0	-	EXTMC	DRAM_D[0]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_D7	-	EXTMC	DRAM_D[7]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DRAM_SDCKE0	-	EXTMC	DRAM_SDCKE[0]	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE0 Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled On Die Termination =HiZ
DRAM_D1	-	EXTMC	DRAM_D[1]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_D14	-	EXTMC	DRAM_D[14]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_D3	-	EXTMC	DRAM_D[3]	
DRAM_D12	-	EXTMC	DRAM_D[12]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_D10	-	EXTMC	DRAM_D[10]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DRAM_D8	-	EXTMC	DRAM_D[8]	IOMUXC_SW_PAD_CTL_GRP_*
DRAM_D13	-	EXTMC	DRAM_D[13]	Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_SDQS1_B	-	EXTMC	DRAM_SDQS_B[1]	See DRAM_SDQS1 settings
DRAM_SDQS1	-	EXTMC	DRAM_SDQS[1]	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQ_S1 Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled On Die Termination = Dependent on Reference Resistor
DRAM_DQM1	-	EXTMC	DRAM_DQM[1]	IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM1 Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Keep Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Disabled TEST_TS = Disabled On Die Termination =HiZ
DRAM_D9	-	EXTMC	DRAM_D[9]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
DRAM_D15	-	EXTMC	DRAM_D[15]	IOMUXC_SW_PAD_CTL_GRP_* Drive Strength (DSE) = DDR2/DDR3 Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DDR/CMOS Input Mode = CMOS Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
DRAM_D11	-	EXTMC	DRAM_D[11]	
CKIH1	-	CAMP-1	CKIH	
CKIH2	-	CAMP-2	CKIH	
PMIC_ON_REQ	-	SRTC	SRTCALARM	IOMUXC_SW_PAD_CTL_PAD_PMIC_ON_REQ Drive Strength (DSE) = High Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = N/A Pull Up / Down (PUS) = N/A Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
PMIC_STBY_REQ	-	CCM	PMIC_VSTBY_REQ	IOMUXC_SW_PAD_CTL_PAD_PMIC_STBY_REQ Drive Strength (DSE) = High Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = N/A Pull Up / Down (PUS) = N/A Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
NANDF_CLE	ALT0	EXTMC	NANDF_CLE	IOMUXC_SW_PAD_CTL_PAD_NANDF_CLE Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) =100 K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[7]	
	ALT7	USBPHY-1	VSTATUS[0]	
NANDF_ALE	ALT0	EXTMC	NANDF_ALE	IOMUXC_SW_PAD_CTL_PAD_NAND_ALE Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[8]	
	ALT7	USBPHY-1	VSTATUS[1]	
NANDF_WP_B	ALT0	EXTMC	NANDF_WP_B	IOMUXC_SW_PAD_CTL_PAD_NANDF_WP_B Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[9]	
	ALT7	USBPHY-1	VSTATUS[2]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
NANDF_RB0	ALT0	EXTMC	NANDF_RB[0]	IOMUXC_SW_PAD_CTL_PAD_NANDF_RB0 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[10]	
	ALT7	USBPHY-1	VSTATUS[3]	
NANDF_CS0	ALT0	EXTMC	NANDF_CS[0]	IOMUXC_SW_PAD_CTL_PAD_NANDF_CS0 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[11]	
	ALT7	USBPHY-1	VSTATUS[4]	
NANDF_CS1	ALT0	EXTMC	NANDF_CS[1]	IOMUXC_SW_PAD_CTL_PAD_NANDF_CS1 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[14]	
	ALT6	MLB	MLBCLK	
	ALT7	USBPHY-1	VSTATUS[5]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
NANDF_CS2	ALT0	EXTMC	NANDF_CS[2]	IOMUXC_SW_PAD_CTL_PAD_NANDF_CS2 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[15]	
	ALT2	IPU	SISG[0]	
	ALT3	ESAI-1	TX0	
	ALT4	EXTMC	WEIM_CRE	
	ALT5	CCM	CSI0_MCLK	
	ALT6	MLB	MLBSIG	
	ALT7	USBPHY-1	VSTATUS[6]	
NANDF_CS3	ALT0	EXTMC	NANDF_CS[3]	IOMUXC_SW_PAD_CTL_PAD_NANDF_CS3 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[16]	
	ALT2	IPU	SISG[1]	
	ALT3	ESAI-1	TX1	
	ALT4	EXTMC	WEIM_A[26]	
	ALT6	MLB	MLBDAT	
	ALT7	USBPHY-1	VSTATUS[7]	
FEC_MDIO	ALT0	FEC	MDIO	IOMUXC_SW_PAD_CTL_PAD_FEC_MDIO Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[22]	
	ALT2	ESAI-1	SCKR	
	ALT3	FEC	COL	
	ALT4	RTC	CE_RTC_PS2	
	ALT5	SDMA	DEBUG_BUS_DEVICE[3]	
	ALT6	EXTMC	EMI_DEBUG[49]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
FEC_ REF_CLK	ALT0	FEC	TX_CLK	IOMUXC_SW_PAD_CTL_PAD_FEC_REF_CLK Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[23]	
	ALT2	ESAI-1	FSR	
	ALT5	SDMA	DEBUG_BUS_DEVICE[4]	
	ALT6	EXTMC	EMI_DEBUG[50]	
FEC_RX_ ER	ALT0	FEC	RX_ER	IOMUXC_SW_PAD_CTL_PAD_FEC_RX_ER Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[24]	
	ALT2	ESAI-1	HCKR	
	ALT3	FEC	RX_CLK	
	ALT4	RTC	CE_RTC_PS3	
FEC_ CRS_DV	ALT0	FEC	RX_DV	IOMUXC_SW_PAD_CTL_PAD_FEC_CRS_DV Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[25]	
	ALT2	ESAI-1	SCKT	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
FEC_ RXD1	ALT0	FEC	RDATA[1]	IOMUXC_SW_PAD_CTL_PAD_FEC_RXD1 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[26]	
	ALT2	ESAI-1	FST	
	ALT3	MLB	MLBSIG	
	ALT4	RTC	CE_RTC_PS1	
FEC_ RXD0	ALT0	FEC	RDATA[0]	IOMUXC_SW_PAD_CTL_PAD_FEC_RXD0 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[27]	
	ALT2	ESAI-1	HCKT	
	ALT3	XTALOSC32K	32K_OUT	
FEC_TX_ EN	ALT0	FEC	TX_EN	IOMUXC_SW_PAD_CTL_PAD_FEC_TX_EN Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[28]	
	ALT2	ESAI-1	TX3_RX2	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
FEC_ TXD1	ALT0	FEC	TDATA[1]	IOMUXC_SW_PAD_CTL_PAD_FEC_TXD1 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[29]	
	ALT2	ESAI-1	TX2_RX3	
	ALT3	MLB	MLBCLK	
	ALT4	RTC	CE_RTC_PRSC_CLK	
FEC_ TXD0	ALT0	FEC	TDATA[0]	IOMUXC_SW_PAD_CTL_PAD_FEC_TXD0 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[30]	
	ALT2	ESAI-1	TX4_RX1	
	ALT7	USBPHY-2	DATAOUT[0]	
FEC_MDC	ALT0	FEC	MDC	IOMUXC_SW_PAD_CTL_PAD_FEC_MDC Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[31]	
	ALT2	ESAI-1	TX5_RX0	
	ALT3	MLB	MLBDAT	
	ALT4	RTC	CE_RTC_ALARM1_TRIG	
	ALT7	USBPHY-2	DATAOUT[1]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
PATA_DIOW	ALT0	PATA	DIOW	IOMUXC_SW_PAD_CTL_PAD_PATA_DIOW Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[17]	
	ALT3	UART-1	TXD_MUX	
	ALT7	USBPHY-2	DATAOUT[2]	
PATA_DMACK	ALT0	PATA	DMACK	IOMUXC_SW_PAD_CTL_PAD_PATA_DMACK Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-6	GPIO[18]	
	ALT3	UART-1	RXD_MUX	
	ALT7	USBPHY-2	DATAOUT[3]	
PATA_DMARQ	ALT0	PATA	DMARQ	IOMUXC_SW_PAD_CTL_PAD_PATA_DMARQ Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-7	GPIO[0]	
	ALT3	UART-2	TXD_MUX	
	ALT5	CCM	CCM_OUT_0	
	ALT7	USBPHY-2	DATAOUT[4]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
PATA_BUFFER_EN	ALT0	PATA	BUFFER_EN	IOMUXC_SW_PAD_CTL_PAD_PATA_BUFFER_EN Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-7	GPIO[1]	
	ALT3	UART-2	RXD_MUX	
	ALT5	CCM	CCM_OUT_1	
	ALT7	USBPHY-2	DATAOUT[5]	
PATA_INTRQ	ALT0	PATA	INTRQ	IOMUXC_SW_PAD_CTL_PAD_PATA_INTRQ Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-7	GPIO[2]	
	ALT3	UART-2	CTS	
	ALT4	FLEXCAN-1	TXCAN	
	ALT5	CCM	CCM_OUT_2	
	ALT7	USBPHY-2	DATAOUT[6]	
PATA_DIOR	ALT0	PATA	DIOR	IOMUXC_SW_PAD_CTL_PAD_PATA_DIOR Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-7	GPIO[3]	
	ALT3	UART-2	RTS	
	ALT4	FLEXCAN-1	RXCAN	
	ALT7	USBPHY-2	DATAOUT[7]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
PATA_ RESET_B	ALT0	PATA	PATA_RESET_B	IOMUXC_SW_PAD_CTL_PAD_PATA_RESET_B Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-7	GPIO[4]	
	ALT2	ESDHCV3-3	CMD	
	ALT3	UART-1	CTS	
	ALT4	FLEXCAN-2	TXCAN	
	ALT7	USBPHY-1	DATAOUT[0]	
PATA_ IORDY	ALT0	PATA	IORDY	IOMUXC_SW_PAD_CTL_PAD_PATA_IORDY Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-7	GPIO[5]	
	ALT2	ESDHCV3-3	CLK	
	ALT3	UART-1	RTS	
	ALT4	FLEXCAN-2	RXCAN	
	ALT7	USBPHY-1	DATAOUT[1]	
PATA_ DA_0	ALT0	PATA	DA_0	IOMUXC_SW_PAD_CTL_PAD_PATA_DA_0 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-7	GPIO[6]	
	ALT2	ESDHCV3-3	RST	
	ALT4	OWIRE	LINE	
	ALT7	USBPHY-1	DATAOUT[2]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
PATA_DA_1	ALT0	PATA	DA_1	IOMUXC_SW_PAD_CTL_PAD_PATA_DA_1 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-7	GPIO[7]	
	ALT2	ESDHCV2-4	CMD	
	ALT4	UARTV3-3	CTS	
	ALT7	USBPHY-1	DATAOUT[3]	
PATA_DA_2	ALT0	PATA	DA_2	IOMUXC_SW_PAD_CTL_PAD_PATA_DA_2 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-7	GPIO[8]	
	ALT2	ESDHCV2-4	CLK	
	ALT4	UART-3	RTS	
	ALT7	USBPHY-1	DATAOUT[4]	
PATA_CS_0	ALT0	PATA	CS_0	IOMUXC_SW_PAD_CTL_PAD_PATA_CS_0 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-7	GPIO[9]	
	ALT4	UART-3	TXD_MUX	
	ALT7	USBPHY-1	DATAOUT[5]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
PATA_CS_1	ALT0	PATA	CS_1	IOMUXC_SW_PAD_CTL_PAD_PATA_CS_1 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-7	GPIO[10]	
	ALT4	UART-3	RXD_MUX	
	ALT7	USBPHY-1	DATAOUT[6]	
PATA_DATA0	ALT0	PATA	PATA_DATA[0]	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA0 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[0]	
	ALT3	EXTMC	NANDF_D[0]	
	ALT4	ESDHCV3-3	DAT4	
	ALT5	GPU3D	GPU_DEBUG_OUT[0]	
	ALT6	IPU	IPU_DIAG_BUS[0]	
	ALT7	USBPHY-1	DATAOUT[7]	
PATA_DATA1	ALT0	PATA	PATA_DATA[1]	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA1 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[1]	
	ALT3	EXTMC	NANDF_D[1]	
	ALT4	ESDHCV3-3	DAT5	
	ALT5	GPU3D	GPU_DEBUG_OUT[1]	
	ALT6	IPU	IPU_DIAG_BUS[1]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
PATA_ DATA2	ALT0	PATA	PATA_DATA[2]	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA2 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[2]	
	ALT3	EXTMC	NANDF_D[2]	
	ALT4	ESDHCV3-3	DAT6	
	ALT5	GPU3D	GPU_DEBUG_OUT[2]	
	ALT6	IPU	IPU_DIAG_BUS[2]	
PATA_ DATA3	ALT0	PATA	PATA_DATA[3]	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA3 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[3]	
	ALT3	EXTMC	NANDF_D[3]	
	ALT4	ESDHCV3-3	DAT7	
	ALT5	GPU3D	GPU_DEBUG_OUT[3]	
	ALT6	IPU	IPU_DIAG_BUS[3]	
PATA_ DATA4	ALT0	PATA	PATA_DATA[4]	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA4 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[4]	
	ALT3	EXTMC	NANDF_D[4]	
	ALT4	ESDHCV2-4	DAT4	
	ALT5	GPU3D	GPU_DEBUG_OUT[4]	
	ALT6	IPU	IPU_DIAG_BUS[4]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
PATA_ DATA5	ALT0	PATA	PATA_DATA[5]	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA5 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[5]	
	ALT3	EXTMC	NANDF_D[5]	
	ALT4	ESDHCV2-4	DAT5	
	ALT5	GPU3D	GPU_DEBUG_OUT[5]	
	ALT6	IPU	IPU_DIAG_BUS[5]	
PATA_ DATA6	ALT0	PATA	PATA_DATA[6]	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA6 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[6]	
	ALT3	EXTMC	NANDF_D[6]	
	ALT4	ESDHCV2-4	DAT6	
	ALT5	GPU3D	GPU_DEBUG_OUT[6]	
	ALT6	IPU	IPU_DIAG_BUS[6]	
PATA_ DATA7	ALT0	PATA	PATA_DATA[7]	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA7 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[7]	
	ALT3	EXTMC	NANDF_D[7]	
	ALT4	ESDHCV2-4	DAT7	
	ALT5	GPU3D	GPU_DEBUG_OUT[7]	
	ALT6	IPU	IPU_DIAG_BUS[7]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
PATA_DATA8	ALT0	PATA	PATA_DATA[8]	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA8 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[8]	
	ALT2	ESDHCV2-1	DAT4	
	ALT3	EXTMC	NANDF_D[8]	
	ALT4	ESDHCV3-3	DAT0	
	ALT5	GPU3D	GPU_DEBUG_OUT[8]	
	ALT6	IPU	IPU_DIAG_BUS[8]	
PATA_DATA9	ALT0	PATA	PATA_DATA[9]	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA9 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[9]	
	ALT2	ESDHCV2-1	DAT5	
	ALT3	EXTMC	NANDF_D[9]	
	ALT4	ESDHCV3-3	DAT1	
	ALT5	GPU3D	GPU_DEBUG_OUT[9]	
	ALT6	IPU	IPU_DIAG_BUS[9]	
PATA_DATA10	ALT0	PATA	PATA_DATA[10]	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA10 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[10]	
	ALT2	ESDHCV2-1	DAT6	
	ALT3	EXTMC	NANDF_D[10]	
	ALT4	ESDHCV3-3	DAT2	
	ALT5	GPU3D	GPU_DEBUG_OUT[10]	
	ALT6	IPU	IPU_DIAG_BUS[10]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
PATA_ DATA11	ALT0	PATA	PATA_DATA[11]	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA11 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[11]	
	ALT2	ESDHCV2-1	DAT7	
	ALT3	EXTMC	NANDF_D[11]	
	ALT4	ESDHCV3-3	DAT3	
	ALT5	GPU3D	GPU_DEBUG_OUT[11]	
	ALT6	IPU	IPU_DIAG_BUS[11]	
PATA_ DATA12	ALT0	PATA	PATA_DATA[12]	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA12 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[12]	
	ALT2	ESDHCV2-2	DAT4	
	ALT3	EXTMC	NANDF_D[12]	
	ALT4	ESDHCV2-4	DAT0	
	ALT5	GPU3D	GPU_DEBUG_OUT[12]	
	ALT6	IPU	IPU_DIAG_BUS[12]	
PATA_ DATA13	ALT0	PATA	PATA_DATA[13]	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA13 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[13]	
	ALT2	ESDHCV2-2	DAT5	
	ALT3	EXTMC	NANDF_D[13]	
	ALT4	ESDHCV2-4	DAT1	
	ALT5	GPU3D	GPU_DEBUG_OUT[13]	
	ALT6	IPU	IPU_DIAG_BUS[13]	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
PATA_ DATA14	ALT0	PATA	PATA_DATA[14]	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA14 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[14]	
	ALT2	ESDHCV2-2	DAT6	
	ALT3	EXTMC	NANDF_D[14]	
	ALT4	ESDHCV2-4	DAT2	
	ALT5	GPU3D	GPU_DEBUG_OUT[14]	
	ALT6	IPU	IPU_DIAG_BUS[14]	
PATA_ DATA15	ALT0	PATA	PATA_DATA[15]	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA15 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-2	GPIO[15]	
	ALT2	ESDHCV2-2	DAT7	
	ALT3	EXTMC	NANDF_D[15]	
	ALT4	ESDHCV2-4	DAT3	
	ALT5	GPU3D	GPU_DEBUG_OUT[15]	
	ALT6	IPU	IPU_DIAG_BUS[15]	
SD1_ DATA0	ALT0	ESDHCV2-1	DAT0	IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[16]	
	ALT3	GPT	IND_CAPIN1	
	ALT5	CSPI	MISO	
	ALT7	CCM	PLL3_BYP	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
SD1_ DATA1	ALT0	ESDHCV2-1	DAT1	IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[17]	
	ALT3	GPT	IND_CAPIN2	
	ALT5	CSPI	SS0	
	ALT7	CCM	PLL4_BYP	
SD1_CMD	ALT0	ESDHCV2-1	CMD	IOMUXC_SW_PAD_CTL_PAD_SD1_CMD Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[18]	
	ALT3	GPT	DO_CMPOUT1	
	ALT5	CSPI	MOSI	
	ALT7	CCM	PLL1_BYP	
SD1_ DATA2	ALT0	ESDHCV2-1	DAT2	IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[19]	
	ALT2	GPT	DO_CMPOUT2	
	ALT3	PWM-2	PWMO	
	ALT4	WDOG-1	WDOG_B	
	ALT5	CSPI	SS1	
	ALT6	WDOG-1	WDOG_RST_B_DEB	
	ALT7	CCM	PLL2_BYP	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
SD1_CLK	ALT0	ESDHCV2-1	CLK	IOMUXC_SW_PAD_CTL_PAD_SD1_CLK Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[20]	
	ALT2	XTALOSC32K	32K_OUT	
	ALT3	GPT	IND_CLKIN	
	ALT5	CSPI	SCLK	
	ALT7	SATA_PHY	DTB[0]	
SD1_DATA3	ALT0	ESDHCV2-1	DAT3	OMUXC_SW_PAD_CTL_PAD_SD1_DATA3 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[21]	
	ALT2	GPT	IND_CMPOUT3	
	ALT3	PWM-1	PWMO	
	ALT4	WDOG-2	WDOG_B	
	ALT5	CSPI	SS2	
	ALT6	WDOG-2	WDOG_RST_B_DEB	
	ALT7	SATA_PHY	DTB[1]	
SD2_CLK	ALT0	ESDHCV2-2	CLK	IOMUXC_SW_PAD_CTL_PAD_SD2_CLK Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[10]	
	ALT2	KPP	COL[5]	
	ALT3	AUDMUX	AUD4_RXFS	
	ALT5	CSPI	SCLK	
	ALT7	SCC	RANDOM_V	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
SD2_CMD	ALT0	ESDHCV2-2	CMD	IOMUXC_SW_PAD_CTL_PAD_SD2_CMD Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[11]	
	ALT2	KPP	ROW[5]	
	ALT3	AUDMUX	AUD4_RXC	
	ALT5	CSPI	MOSI	
	ALT7	SCC	RANDOM	
SD2_ DATA3	ALT0	ESDHCV2-2	DAT3	IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[12]	
	ALT2	KPP	COL[6]	
	ALT3	AUDMUX	AUD4_TXC	
	ALT5	CSPI	SS2	
	ALT7	SJC	DONE	
SD2_ DATA2	ALT0	ESDHCV2-2	DAT2	IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[13]	
	ALT2	KPP	ROW[6]	
	ALT3	AUDMUX	AUD4_TXD	
	ALT5	CSPI	SS1	
	ALT7	SJC	FAIL	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
SD2_ DATA1	ALT0	ESDHCV2-2	DAT1	IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[14]	
	ALT2	KPP	COL[7]	
	ALT3	AUDMUX	AUD4_TXFS	
	ALT5	CSPI	SS0	
	ALT7	RTIC	RTIC_SEC_VIO	
SD2_ DATA0	ALT0	ESDHCV2-2	DAT0	IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[15]	
	ALT2	KPP	ROW[7]	
	ALT3	AUDMUX	AUD4_RXD	
	ALT5	CSPI	MISO	
	ALT7	RTIC	RTIC_DONE_INT	
GPIO_0	ALT0	CCM	CLKO	IOMUXC_SW_PAD_CTL_PAD_GPIO_0 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[0]	
	ALT2	KPP	COL[5]	
	ALT3	CCM	SSI_EXT1_CLK	
	ALT4	EPIT-1	EPITO	
	ALT5	SRTC	SRTC_ALARM_DEB	
	ALT6	USB	USBH1_PWR	
	ALT7	CSU	TD	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
GPIO_1	ALT0	ESAI-1	SCKR	IOMUXC_SW_PAD_CTL_PAD_GPIO_1 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[1]	
	ALT2	KPP	ROW[5]	
	ALT3	CCM	SSI_EXT2_CLK	
	ALT4	PWM-2	PWMO	
	ALT5	WDOG-2	WDOG_B	
	ALT6	ESDHCV2-1	CD	
	ALT7	SRC	TESTER_ACK	
GPIO_9	ALT0	ESAI-1	FSR	IOMUXC_SW_PAD_CTL_PAD_GPIO_9 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[9]	
	ALT2	KPP	COL[6]	
	ALT3	CCM	REF_EN_B	
	ALT4	PWM-1	PWMO	
	ALT5	WDOG-1	WDOG_B	
	ALT6	ESDHCV2-1	WP	
	ALT7	SCC	FAIL_STATE	
GPIO_3	ALT0	ESAI-1	HCKR	IOMUXC_SW_PAD_CTL_PAD_GPIO_3 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[3]	
	ALT2	I2C-3	SCL	
	ALT3	DPLLC-1	TOG_EN	
	ALT4	CCM	CLKO2	
	ALT5	OBSERVE_MUX	OBSRV_INT_OUT0	
	ALT6	USB	USBH1_OC	
	ALT7	MLB	MLBCLK	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
GPIO_6	ALT0	ESAI-1	SCKT	IOMUXC_SW_PAD_CTL_PAD_GPIO_6 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[6]	
	ALT2	I2C-3	SDA	
	ALT3	CCM	CCM_OUT_0	
	ALT4	CSU	CSU_INT_DEB	
	ALT5	OBSERVE_MUX	OBSRV_INT_OUT1	
	ALT6	ESDHCV2-2	LCTL	
	ALT7	MLB	MLBSIG	
GPIO_2	ALT0	ESAI-1	FST	IOMUXC_SW_PAD_CTL_PAD_GPIO_2 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[2]	
	ALT2	KPP	ROW[6]	
	ALT3	CCM	CCM_OUT_1	
	ALT4	CSU	CSU_ALARM_AUT[0]	
	ALT5	OBSERVE_MUX	OBSRV_INT_OUT2	
	ALT6	ESDHCV2-2	WP	
	ALT7	MLB	MLBDAT	
GPIO_4	ALT0	ESAI-1	HCKT	IOMUXC_SW_PAD_CTL_PAD_GPIO_4 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[4]	
	ALT2	KPP	COL[7]	
	ALT3	CCM	CCM_OUT_2	
	ALT4	CSU	CSU_ALARM_AUT[1]	
	ALT5	OBSERVE_MUX	OBSRV_INT_OUT3	
	ALT6	ESDHCV2-2	CD	
	ALT7	SCC	SEC_STATE	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
GPIO_5	ALT0	ESAI-1	TX2_RX3	IOMUXC_SW_PAD_CTL_PAD_GPIO_5 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[5]	
	ALT2	KPP	ROW[7]	
	ALT3	CCM	CLKO	
	ALT4	CSU	CSU_ALARM_AUT[2]	
	ALT5	OBSERVE_MUX	OBSRV_INT_OUT4	
	ALT6	I2C-3	SCL	
	ALT7	CCM	PLL1_BYP	
GPIO_7	ALT0	ESAI-1	TX4_RX1	IOMUXC_SW_PAD_CTL_PAD_GPIO_7 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[7]	
	ALT2	EPIT-1	EPITO	
	ALT3	FLEXCAN-1	TXCAN	
	ALT4	UART-2	TXD_MUX	
	ALT5	FIRI	RXD	
	ALT6	SPDIF	PLOCK	
	ALT7	CCM	PLL2_BYP	
GPIO_8	ALT0	ESAI-1	TX5_RX0	IOMUXC_SW_PAD_CTL_PAD_GPIO_8 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-1	GPIO[8]	
	ALT2	EPIT-2	EPITO	
	ALT3	FLEXCAN-1	RXCAN	
	ALT4	UART-2	RXD_MUX	
	ALT5	FIRI	TXD	
	ALT6	SPDIF	SRCLK	
	ALT7	CCM	PLL3_BYP	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
GPIO_16	ALT0	ESAI-1	TX3_RX2	IOMUXC_SW_PAD_CTL_PAD_GPIO_16 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-7	GPIO[11]	
	ALT2	TZIC	PWRFAIL_INT	
	ALT3	ARM Platform	PMU_IRQ_B	
	ALT4	RTC	CE_RTC_EXT_TRIG1	
	ALT5	SPDIF	IN1	
	ALT6	I2C-3	SDA	
	ALT7	SJC	DE_B	
GPIO_17	ALT0	ESAI-1	TX0	IOMUXC_SW_PAD_CTL_PAD_GPIO_17 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-7	GPIO[12]	
	ALT2	SDMA	SDMA_EXT_EVENT[0]	
	ALT3	GPC	PMIC_RDY	
	ALT4	RTC	CE_RTC_FSV_TRIG	
	ALT5	SPDIF	OUT1	
	ALT6	IPU	SNOOP2	
	ALT7	SJC	JTAG_ACT	
GPIO_18	ALT0	ESAI-1	TX1	IOMUXC_SW_PAD_CTL_PAD_GPIO_18 Drive Strength (DSE) = High Low/high output voltage = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 360K Ohm Pull Down DSE_TEST = Regular Open Drain Enable (ODE) = Disabled Pull / Keep Enable (PKE) =Enabled TEST_TS = Disabled
	ALT1	GPIO-7	GPIO[13]	
	ALT2	SDMA	SDMA_EXT_EVENT[1]	
	ALT3	OWIRE	LINE	
	ALT4	RTC	CE_RTC_ALARM2_TRIG	
	ALT5	CCM	ASRC_EXT_CLK	
	ALT6	ESDHCV2-1	LCTL	
	ALT7	SRC	SYSTEM_RST	

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
POR_B	-	SRC	POR_B	IOMUXC_SW_PAD_CTL_PAD_POR_B Drive Strength (DSE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = N/A Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
BOOT_ MODE1	-	SRC	BOOT_MODE[1]	IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE1 Drive Strength (DSE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = N/A Pull / Keep Enable (PKE) =Enabled Slew Rate () = Fast TEST_TS = Disabled
RESET_ IN_B	-	SRC	RESET_B	IOMUXC_SW_PAD_CTL_PAD_RESET_IN_B Drive Strength (DSE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Up Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = N/A Pull / Keep Enable (PKE) =Enabled Slew Rate () = N/A TEST_TS = Disabled

Table continues on the next page...

Table 4-2. Pin Alternate Modes (continued)

Package Pin Name	Mode	Instance	Block I/O	Pad Control Register / Package Pin Drive Default Settings
BOOT_ MODE0	-	SRC	BOOT_MODE[0]	IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE0 Drive Strength (DSE) = N/A Hysteresis Enable (HYS) = Enabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = N/A Pull / Keep Enable (PKE) =Enabled Slew Rate () = N/A TEST_TS = Disabled
TEST_ MODE	-	TCU	TEST_MODE	IOMUXC_SW_PAD_CTL_PAD_TEST_MODE Drive Strength (DSE) = N/A Hysteresis Enable (HYS) = Disabled Pull / Keep Select (PUE) = Pull Pull Up / Down (PUS) = 100K Ohm Pull Down Strength Mode (STRENGTH_MODE) = 4-LEVEL DSE_TEST = Regular Open Drain Enable (ODE) = N/A Pull / Keep Enable (PKE) =Enabled Slew Rate () = N/A TEST_TS = Disabled

Table 4-3 shows block I/O multiplexing options by block instance.

Table 4-3. Muxing Options sorted by IPs

Block Instance	Block I/O	Pin	Mode
TZIC	PWRFAIL_INT	GPIO_16	ALT2
RTIC	RTIC_DONE_INT	SD2_DATA0	ALT7
	RTIC_SEC_VIO	SD2_DATA1	ALT7
SRTC	SRTCALARM	PMIC_ON_REQ	No Muxing (ALT0)
	SRTC_ALARM_DEB	GPIO_0	ALT5
TCU	TEST_MODE	TEST_MODE	No Muxing (ALT0)
GPC	PMIC_RDY	EIM_EB0	ALT5
		GPIO_17	ALT3

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
MLB	MLBCLK	FEC_TXD1	ALT3
		GPIO_3	ALT7
		NANDF_CS1	ALT6
	MLBDAT	FEC_MDC	ALT3
		GPIO_2	ALT7
		NANDF_CS3	ALT6
	MLBSIG	FEC_RXD1	ALT3
		GPIO_6	ALT7
		NANDF_CS2	ALT6
ECSPI-1	MISO	CSI0_DAT6	ALT3
		DISP0_DAT22	ALT2
		EIM_D17	ALT4
		KEY_COL1	ALT5
	MOSI	CSI0_DAT5	ALT3
		DISP0_DAT21	ALT2
		EIM_D18	ALT4
ECSPI-1	MOSI	KEY_ROW0	ALT5
	RDY	GPIO_19	ALT5
	SCLK	CSI0_DAT4	ALT3
		DISP0_DAT20	ALT2
		EIM_D16	ALT4
		KEY_COL0	ALT5
	SS0	CSI0_DAT7	ALT3
		DISP0_DAT23	ALT2
		EIM_EB2	ALT4
		KEY_ROW1	ALT5
	SS1	DISP0_DAT15	ALT2
		EIM_D19	ALT4
		KEY_COL2	ALT5
	SS2	EIM_D24	ALT3
		KEY_ROW2	ALT5
	SS3	EIM_D25	ALT3
		KEY_COL3	ALT5

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
ECSPI-2	MISO	CSI0_DAT10	ALT3
		DISP0_DAT17	ALT2
		EIM_OE	ALT2
	MOSI	CSI0_DAT9	ALT3
		DISP0_DAT16	ALT2
		EIM_CS1	ALT2
	RDY	EIM_A25	ALT2
ECSPI-2	SCLK	CSI0_DAT8	ALT3
		DISP0_DAT19	ALT2
		EIM_CS0	ALT2
	SS0	CSI0_DAT11	ALT3
		DISP0_DAT18	ALT2
		EIM_RW	ALT2
	SS1	DISP0_DAT15	ALT3
		EIM_LBA	ALT2
	SS2	EIM_D24	ALT6
	SS3	EIM_D25	ALT6
UART-1	CTS	EIM_D19	ALT6
		PATA_RESET_B	ALT3
	DCD	EIM_D23	ALT3
	DSR	EIM_D25	ALT7
	DTR	EIM_D24	ALT7
	RI	EIM_EB3	ALT3
	RTS	EIM_D20	ALT6
		PATA_IORDY	ALT3
	RXD_MUX	CSI0_DAT11	ALT2
		PATA_DMACK	ALT3
	TXD_MUX	CSI0_DAT10	ALT2
		PATA_DIOW	ALT3
UART-2	CTS	EIM_D28	ALT2
		PATA_INTRQ	ALT3

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
UART-2	RTS	EIM_D29	ALT2
		PATA_DIOR	ALT3
	RXD_MUX	EIM_D27	ALT2
		GPIO_8	ALT4
		PATA_BUFFER_EN	ALT3
	TXD_MUX	EIM_D26	ALT2
		GPIO_7	ALT4
		PATA_DMARQ	ALT3
UART-3	CTS	EIM_D23	ALT2
		EIM_D30	ALT2
		PATA_DA_1	ALT4
	RTS	EIM_D31	ALT2
		EIM_EB3	ALT2
		PATA_DA_2	ALT4
	RXD_MUX	EIM_D25	ALT2
		PATA_CS_1	ALT4
	TXD_MUX	EIM_D24	ALT2
		PATA_CS_0	ALT4
UART-4	CTS	CSI0_DAT17	ALT2
	RTS	CSI0_DAT16	ALT2
	RXD_MUX	CSI0_DAT13	ALT2
		KEY_ROW0	ALT4
	TXD_MUX	CSI0_DAT12	ALT2
		KEY_COL0	ALT4
UART-5	CTS	CSI0_DAT19	ALT2
		KEY_ROW4	ALT4
	RTS	CSI0_DAT18	ALT2
		KEY_COL4	ALT4
	RXD_MUX	CSI0_DAT15	ALT2
		KEY_ROW1	ALT4
	TXD_MUX	CSI0_DAT14	ALT2
		KEY_COL1	ALT4
DPLLC-1	TOG_EN	GPIO_3	ALT3

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
ARM Platform	CTI_TRIGIN7	KEY_COL0	ALT3
	CTI_TRIGIN_ACK7	KEY_ROW0	ALT3
	CTI_TRIGOUT6	KEY_COL2	ALT3
	CTI_TRIGOUT7	KEY_ROW2	ALT3
	CTI_TRIGOUT_ACK6	KEY_COL1	ALT3
	CTI_TRIGOUT_ACK7	KEY_ROW1	ALT3
	PMU_IRQ_B	GPIO_16	ALT3
WDOG-1	WDOG_B	DISP0_DAT8	ALT3
		GPIO_9	ALT5
		SD1_DATA2	ALT4
	WDOG_RST_B_DEB		ALT6
WDOG-2	WDOG_B	DISP0_DAT9	ALT3
		GPIO_1	ALT5
		SD1_DATA3	ALT4
	WDOG_RST_B_DEB		ALT6

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
ESAI-1	FSR	FEC_REF_CLK	ALT2
		GPIO_9	ALT0
	FST	FEC_RXD1	ALT2
		GPIO_2	ALT0
	HCKR	FEC_RX_ER	ALT2
		GPIO_3	ALT0
	HCKT	FEC_RXD0	ALT2
		GPIO_4	ALT0
	SCKR	FEC_MDIO	ALT2
		GPIO_1	ALT0
	SCKT	FEC_CRS_DV	ALT2
		GPIO_6	ALT0
	TX0	GPIO_17	ALT0
		NANDF_CS2	ALT3
	TX1	GPIO_18	ALT0
		NANDF_CS3	ALT3
	TX2_RX3	FEC_TXD1	ALT2
		GPIO_5	ALT0
	TX3_RX2	FEC_TX_EN	ALT2
		GPIO_16	ALT0
	TX4_RX1	FEC_TXD0	ALT2
		GPIO_7	ALT0
	TX5_RX0	FEC_MDC	ALT2
		GPIO_8	ALT0

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
EXTMC	DRAM_A[0]	DRAM_A0	No Muxing (ALT0)
	DRAM_A[10]	DRAM_A10	No Muxing (ALT0)
	DRAM_A[11]	DRAM_A11	No Muxing (ALT0)
	DRAM_A[12]	DRAM_A12	No Muxing (ALT0)
	DRAM_A[13]	DRAM_A13	No Muxing (ALT0)
	DRAM_A[14]	DRAM_A14	No Muxing (ALT0)
	DRAM_A[15]	DRAM_A15	No Muxing (ALT0)
	DRAM_A[1]	DRAM_A1	No Muxing (ALT0)
	DRAM_A[2]	DRAM_A2	No Muxing (ALT0)
	DRAM_A[3]	DRAM_A3	No Muxing (ALT0)
	DRAM_A[4]	DRAM_A4	No Muxing (ALT0)
	DRAM_A[5]	DRAM_A5	No Muxing (ALT0)
	DRAM_A[6]	DRAM_A6	No Muxing (ALT0)
	DRAM_A[7]	DRAM_A7	No Muxing (ALT0)
	DRAM_A[8]	DRAM_A8	No Muxing (ALT0)
	DRAM_A[9]	DRAM_A9	No Muxing (ALT0)
	DRAM_CAS	DRAM_CAS	No Muxing (ALT0)
	DRAM_CS[0]	DRAM_CS0	No Muxing (ALT0)
	DRAM_CS[1]	DRAM_CS1	No Muxing (ALT0)
	DRAM_DQM[0]	DRAM_DQM0	No Muxing (ALT0)
	DRAM_DQM[1]	DRAM_DQM1	No Muxing (ALT0)
	DRAM_DQM[2]	DRAM_DQM2	No Muxing (ALT0)
	DRAM_DQM[3]	DRAM_DQM3	No Muxing (ALT0)
	DRAM_D[0]	DRAM_D0	No Muxing (ALT0)

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
EXTMC	DRAM_D[10]	DRAM_D10	No Muxing (ALT0)
	DRAM_D[11]	DRAM_D11	No Muxing (ALT0)
	DRAM_D[12]	DRAM_D12	No Muxing (ALT0)
	DRAM_D[13]	DRAM_D13	No Muxing (ALT0)
	DRAM_D[14]	DRAM_D14	No Muxing (ALT0)
	DRAM_D[15]	DRAM_D15	No Muxing (ALT0)
	DRAM_D[16]	DRAM_D16	No Muxing (ALT0)
	DRAM_D[17]	DRAM_D17	No Muxing (ALT0)
	DRAM_D[18]	DRAM_D18	No Muxing (ALT0)
	DRAM_D[19]	DRAM_D19	No Muxing (ALT0)
	DRAM_D[1]	DRAM_D1	No Muxing (ALT0)
	DRAM_D[20]	DRAM_D20	No Muxing (ALT0)
	DRAM_D[21]	DRAM_D21	No Muxing (ALT0)
	DRAM_D[22]	DRAM_D22	No Muxing (ALT0)
	DRAM_D[23]	DRAM_D23	No Muxing (ALT0)
	DRAM_D[24]	DRAM_D24	No Muxing (ALT0)
	DRAM_D[25]	DRAM_D25	No Muxing (ALT0)
	DRAM_D[26]	DRAM_D26	No Muxing (ALT0)
	DRAM_D[27]	DRAM_D27	No Muxing (ALT0)
	DRAM_D[28]	DRAM_D28	No Muxing (ALT0)
	DRAM_D[29]	DRAM_D29	No Muxing (ALT0)
	DRAM_D[2]	DRAM_D2	No Muxing (ALT0)
	DRAM_D[30]	DRAM_D30	No Muxing (ALT0)
	DRAM_D[31]	DRAM_D31	No Muxing (ALT0)

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
EXTMC	DRAM_D[3]	DRAM_D3	No Muxing (ALT0)
	DRAM_D[4]	DRAM_D4	No Muxing (ALT0)
	DRAM_D[5]	DRAM_D5	No Muxing (ALT0)
	DRAM_D[6]	DRAM_D6	No Muxing (ALT0)
	DRAM_D[7]	DRAM_D7	No Muxing (ALT0)
	DRAM_D[8]	DRAM_D8	No Muxing (ALT0)
	DRAM_D[9]	DRAM_D9	No Muxing (ALT0)
	DRAM_ODT[0]	DRAM_SDODT0	No Muxing (ALT0)
	DRAM_ODT[1]	DRAM_SDODT1	No Muxing (ALT0)
	DRAM_RAS	DRAM_RAS	No Muxing (ALT0)
	DRAM_RESET	DRAM_RESET	No Muxing (ALT0)
	DRAM_SDBA[0]	DRAM_SDBA0	No Muxing (ALT0)
	DRAM_SDBA[1]	DRAM_SDBA1	No Muxing (ALT0)
	DRAM_SDBA[2]	DRAM_SDBA2	No Muxing (ALT0)
	DRAM_SDCKE[0]	DRAM_SDCKE0	No Muxing (ALT0)
	DRAM_SDCKE[1]	DRAM_SDCKE1	No Muxing (ALT0)
	DRAM_SDCLK0	DRAM_SDCLK_0	No Muxing (ALT0)
	DRAM_SDCLK0_B	DRAM_SDCLK_0_B	No Muxing (ALT0)
	DRAM_SDCLK1	DRAM_SDCLK_1	No Muxing (ALT0)
	DRAM_SDCLK1_B	DRAM_SDCLK_1_B	No Muxing (ALT0)
	DRAM_SDQS[0]	DRAM_SDQS0	No Muxing (ALT0)
	DRAM_SDQS[1]	DRAM_SDQS1	No Muxing (ALT0)
	DRAM_SDQS[2]	DRAM_SDQS2	No Muxing (ALT0)
	DRAM_SDQS[3]	DRAM_SDQS3	No Muxing (ALT0)

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
EXTMC	DRAM_SDQS_B[0]	DRAM_SDQS0_B	No Muxing (ALT0)
	DRAM_SDQS_B[1]	DRAM_SDQS1_B	No Muxing (ALT0)
	DRAM_SDQS_B[2]	DRAM_SDQS2_B	No Muxing (ALT0)
	DRAM_SDQS_B[3]	DRAM_SDQS3_B	No Muxing (ALT0)
	DRAM_SDWE	DRAM_SDWE	No Muxing (ALT0)
	EMI_DEBUG[0]	DI0_DISP_CLK	ALT6
	EMI_DEBUG[10]	DISP0_DAT5	ALT6
	EMI_DEBUG[11]	DISP0_DAT6	ALT6
	EMI_DEBUG[12]	DISP0_DAT7	ALT6
	EMI_DEBUG[13]	DISP0_DAT8	ALT6
	EMI_DEBUG[14]	DISP0_DAT9	ALT6
	EMI_DEBUG[15]	DISP0_DAT10	ALT6
	EMI_DEBUG[16]	DISP0_DAT11	ALT6
	EMI_DEBUG[17]	DISP0_DAT12	ALT6
	EMI_DEBUG[18]	DISP0_DAT13	ALT6
	EMI_DEBUG[19]	DISP0_DAT14	ALT6
	EMI_DEBUG[1]	DI0_PIN15	ALT6
	EMI_DEBUG[20]	DISP0_DAT15	ALT6
	EMI_DEBUG[21]	DISP0_DAT16	ALT6
	EMI_DEBUG[22]	DISP0_DAT17	ALT6
	EMI_DEBUG[23]	DISP0_DAT18	ALT6
	EMI_DEBUG[24]	DISP0_DAT19	ALT6
	EMI_DEBUG[25]	DISP0_DAT20	ALT6
	EMI_DEBUG[26]	DISP0_DAT21	ALT6

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
EXTMC	EMI_DEBUG[27]	DISP0_DAT22	ALT6
	EMI_DEBUG[28]	DISP0_DAT23	ALT6
	EMI_DEBUG[29]	CSI0_PIXCLK	ALT6
	EMI_DEBUG[2]	DI0_PIN2	ALT6
	EMI_DEBUG[30]	CSI0_MCLK	ALT6
	EMI_DEBUG[31]	CSI0_DATA_EN	ALT6
	EMI_DEBUG[32]	CSI0_VSYNC	ALT6
	EMI_DEBUG[33]	CSI0_DAT4	ALT6
	EMI_DEBUG[34]	CSI0_DAT5	ALT6
	EMI_DEBUG[35]	CSI0_DAT6	ALT6
	EMI_DEBUG[36]	CSI0_DAT7	ALT6
	EMI_DEBUG[37]	CSI0_DAT8	ALT6
	EMI_DEBUG[38]	CSI0_DAT9	ALT6
	EMI_DEBUG[39]	CSI0_DAT10	ALT6
	EMI_DEBUG[3]	DI0_PIN3	ALT6
	EMI_DEBUG[40]	CSI0_DAT11	ALT6
	EMI_DEBUG[41]	CSI0_DAT12	ALT6
	EMI_DEBUG[42]	CSI0_DAT13	ALT6
	EMI_DEBUG[43]	CSI0_DAT14	ALT6
	EMI_DEBUG[44]	CSI0_DAT15	ALT6
	EMI_DEBUG[45]	CSI0_DAT16	ALT6
	EMI_DEBUG[46]	CSI0_DAT17	ALT6
	EMI_DEBUG[47]	CSI0_DAT18	ALT6
	EMI_DEBUG[48]	CSI0_DAT19	ALT6

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
EXTMC	EMI_DEBUG[49]	FEC_MDIO	ALT6
	EMI_DEBUG[4]	DI0_PIN4	ALT6
	EMI_DEBUG[50]	FEC_REF_CLK	ALT6
	EMI_DEBUG[5]	DISP0_DAT0	ALT6
	EMI_DEBUG[6]	DISP0_DAT1	ALT6
	EMI_DEBUG[7]	DISP0_DAT2	ALT6
	EMI_DEBUG[8]	DISP0_DAT3	ALT6
	EMI_DEBUG[9]	DISP0_DAT4	ALT6
	NANDF_ALE	NANDF_ALE	ALT0
	NANDF_CLE	NANDF_CLE	ALT0
	NANDF_CS[0]	NANDF_CS0	ALT0
	NANDF_CS[1]	NANDF_CS1	ALT0
	NANDF_CS[2]	NANDF_CS2	ALT0
	NANDF_CS[3]	NANDF_CS3	ALT0
	NANDF_D[0]	PATA_DATA0	ALT3
	NANDF_D[10]	PATA_DATA10	ALT3
	NANDF_D[11]	PATA_DATA11	ALT3
	NANDF_D[12]	PATA_DATA12	ALT3
	NANDF_D[13]	PATA_DATA13	ALT3
	NANDF_D[14]	PATA_DATA14	ALT3
	NANDF_D[15]	PATA_DATA15	ALT3
	NANDF_D[1]	PATA_DATA1	ALT3
	NANDF_D[2]	PATA_DATA2	ALT3
	NANDF_D[3]	PATA_DATA3	ALT3

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
EXTMC	NANDF_D[4]	PATA_DATA4	ALT3
	NANDF_D[5]	PATA_DATA5	ALT3
	NANDF_D[6]	PATA_DATA6	ALT3
	NANDF_D[7]	PATA_DATA7	ALT3
	NANDF_D[8]	PATA_DATA8	ALT3
	NANDF_D[9]	PATA_DATA9	ALT3
	NANDF_RB[0]	NANDF_RB0	ALT0
	NANDF_RE_B	NANDF_RE_B	ALT0
	NANDF_WE_B	NANDF_WE_B	ALT0
	NANDF_WP_B	NANDF_WP_B	ALT0
	NAND_WEIM_DA[0]	EIM_DA0	ALT0
	NAND_WEIM_DA[10]	EIM_DA10	ALT0
	NAND_WEIM_DA[11]	EIM_DA11	ALT0
	NAND_WEIM_DA[12]	EIM_DA12	ALT0
	NAND_WEIM_DA[13]	EIM_DA13	ALT0
	NAND_WEIM_DA[14]	EIM_DA14	ALT0
	NAND_WEIM_DA[15]	EIM_DA15	ALT0
	NAND_WEIM_DA[1]	EIM_DA1	ALT0
	NAND_WEIM_DA[2]	EIM_DA2	ALT0
	NAND_WEIM_DA[3]	EIM_DA3	ALT0
	NAND_WEIM_DA[4]	EIM_DA4	ALT0
	NAND_WEIM_DA[5]	EIM_DA5	ALT0
	NAND_WEIM_DA[6]	EIM_DA6	ALT0
	NAND_WEIM_DA[7]	EIM_DA7	ALT0

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
EXTMC	NAND_WEIM_DA[8]	EIM_DA8	ALT0
	NAND_WEIM_DA[9]	EIM_DA9	ALT0
	WEIM_A[16]	EIM_A16	ALT0
	WEIM_A[17]	EIM_A17	ALT0
	WEIM_A[18]	EIM_A18	ALT0
	WEIM_A[19]	EIM_A19	ALT0
	WEIM_A[20]	EIM_A20	ALT0
	WEIM_A[21]	EIM_A21	ALT0
	WEIM_A[22]	EIM_A22	ALT0
	WEIM_A[23]	EIM_A23	ALT0
	WEIM_A[24]	EIM_A24	ALT0
	WEIM_A[25]	EIM_A25	ALT0
	WEIM_A[26]	NANDF_CS3	ALT4
	WEIM_BCLK	EIM_BCLK	No Muxing (ALT0)
	WEIM_CRE	NANDF_CS2	ALT4
	WEIM_CS[0]	EIM_CS0	ALT0
	WEIM_CS[1]	EIM_CS1	ALT0
	WEIM_CS[2]	DISP0_DAT18	ALT7
	WEIM_CS[3]	DISP0_DAT19	ALT7
	WEIM_DTACK_B	EIM_WAIT	ALT2
	WEIM_D[16]	EIM_D16	ALT0
	WEIM_D[17]	EIM_D17	ALT0
	WEIM_D[18]	EIM_D18	ALT0
	WEIM_D[19]	EIM_D19	ALT0

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
EXTMC	WEIM_D[20]	EIM_D20	ALT0
	WEIM_D[21]	EIM_D21	ALT0
	WEIM_D[22]	EIM_D22	ALT0
	WEIM_D[23]	EIM_D23	ALT0
	WEIM_D[24]	EIM_D24	ALT0
	WEIM_D[25]	EIM_D25	ALT0
	WEIM_D[26]	EIM_D26	ALT0
	WEIM_D[27]	EIM_D27	ALT0
	WEIM_D[28]	EIM_D28	ALT0
	WEIM_D[29]	EIM_D29	ALT0
	WEIM_D[30]	EIM_D30	ALT0
	WEIM_D[31]	EIM_D31	ALT0
	WEIM_EB[0]	EIM_EB0	ALT0
	WEIM_EB[1]	EIM_EB1	ALT0
	WEIM_EB[2]	EIM_EB2	ALT0
	WEIM_EB[3]	EIM_EB3	ALT0
	WEIM_LBA	EIM_LBA	ALT0
	WEIM_OE	EIM_OE	ALT0
	WEIM_RW	EIM_RW	ALT0
	WEIM_WAIT	EIM_WAIT	ALT0
IPU	CSI0_DATA_EN	CSI0_DATA_EN	ALT0
	CSI0_D[0]	EIM_D27	ALT4
	CSI0_D[10]	CSI0_DAT10	ALT0
	CSI0_D[11]	CSI0_DAT11	ALT0

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
IPU	CSI0_D[12]	CSI0_DAT12	ALT0
	CSI0_D[13]	CSI0_DAT13	ALT0
	CSI0_D[14]	CSI0_DAT14	ALT0
	CSI0_D[15]	CSI0_DAT15	ALT0
	CSI0_D[16]	CSI0_DAT16	ALT0
	CSI0_D[17]	CSI0_DAT17	ALT0
	CSI0_D[18]	CSI0_DAT18	ALT0
	CSI0_D[19]	CSI0_DAT19	ALT0
	CSI0_D[1]	EIM_D26	ALT4
	CSI0_D[2]	EIM_D31	ALT3
	CSI0_D[3]	EIM_D30	ALT3
	CSI0_D[4]	CSI0_DAT4	ALT0
	CSI0_D[5]	CSI0_DAT5	ALT0
	CSI0_D[6]	CSI0_DAT6	ALT0
	CSI0_D[7]	CSI0_DAT7	ALT0
	CSI0_D[8]	CSI0_DAT8	ALT0
	CSI0_D[9]	CSI0_DAT9	ALT0
	CSI0_HSYNC	CSI0_MCLK	ALT0
	CSI0_PIXCLK	CSI0_PIXCLK	ALT0
	CSI0_VSYNC	CSI0_VSYNC	ALT0
	CSI1_DATA_EN	EIM_D23	ALT6
		EIM_DA10	ALT4
	CSI1_D[0]	EIM_DA9	ALT4
	CSI1_D[10]	EIM_EB1	ALT4

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
IPU	CSI1_D[11]	EIM_EB0	ALT4
	CSI1_D[12]	EIM_A17	ALT3
	CSI1_D[13]	EIM_A18	ALT3
	CSI1_D[14]	EIM_A19	ALT3
	CSI1_D[15]	EIM_A20	ALT3
	CSI1_D[16]	EIM_A21	ALT3
	CSI1_D[17]	EIM_A22	ALT3
	CSI1_D[18]	EIM_A23	ALT3
	CSI1_D[19]	EIM_A24	ALT3
	CSI1_D[1]	EIM_DA8	ALT4
	CSI1_D[2]	EIM_DA7	ALT4
	CSI1_D[3]	EIM_DA6	ALT4
	CSI1_D[4]	EIM_DA5	ALT4
	CSI1_D[5]	EIM_DA4	ALT4
	CSI1_D[6]	EIM_DA3	ALT4
	CSI1_D[7]	EIM_DA2	ALT4
	CSI1_D[8]	EIM_DA1	ALT4
	CSI1_D[9]	EIM_DA0	ALT4
	CSI1_HSYNC	EIM_DA11	ALT4
		EIM_EB3	ALT6
	CSI1_PIXCLK	EIM_A16	ALT3
	CSI1_VSYNC	EIM_D29	ALT6
		EIM_DA12	ALT4
	DI0_D0_CS	EIM_D23	ALT4

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
IPU	DI0_D1_CS	EIM_A25	ALT6
	DI0_DISP_CLK	DI0_DISP_CLK	ALT0
	DI0_PIN1	EIM_D22	ALT2
	DI0_PIN11	EIM_D30	ALT4
	DI0_PIN12	EIM_D31	ALT4
	DI0_PIN13	EIM_D28	ALT7
	DI0_PIN14	EIM_D29	ALT7
	DI0_PIN15	DI0_PIN15	ALT0
	DI0_PIN16	EIM_D20	ALT2
	DI0_PIN17	EIM_D21	ALT2
	DI0_PIN2	DI0_PIN2	ALT0
	DI0_PIN3	DI0_PIN3	ALT0
	DI0_PIN4	DI0_PIN4	ALT0
	DI0_PIN5	EIM_D16	ALT2
	DI0_PIN6	EIM_D17	ALT2
	DI0_PIN7	EIM_D18	ALT2
	DI0_PIN8	EIM_D19	ALT2
	DI1_D0_CS	EIM_D18	ALT6
		EIM_DA13	ALT3
	DI1_D1_CS	EIM_DA14	ALT3
	DI1_DISP_CLK	EIM_A16	ALT2
	DI1_PIN1	EIM_DA15	ALT3
	DI1_PIN11	EIM_D26	ALT5
	DI1_PIN12	EIM_A25	ALT3

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
IPU	DI1_PIN13	EIM_D27	ALT5
	DI1_PIN14	EIM_D23	ALT7
	DI1_PIN15	EIM_D29	ALT5
		EIM_DA10	ALT3
	DI1_PIN16	EIM_EB3	ALT7
	DI1_PIN17	EIM_LBA	ALT3
	DI1_PIN2	EIM_D23	ALT5
		EIM_DA11	ALT3
	DI1_PIN3	EIM_DA12	ALT3
		EIM_EB3	ALT5
	DI1_PIN4	EIM_DA15	ALT4
	DI1_PIN5	EIM_CS0	ALT3
	DI1_PIN6	EIM_CS1	ALT3
	DI1_PIN7	EIM_OE	ALT3
	DI1_PIN8	EIM_RW	ALT3
	DISP0_DAT[0]	DISP0_DAT0	ALT0
	DISP0_DAT[10]	DISP0_DAT10	ALT0
	DISP0_DAT[11]	DISP0_DAT11	ALT0
	DISP0_DAT[12]	DISP0_DAT12	ALT0
	DISP0_DAT[13]	DISP0_DAT13	ALT0
	DISP0_DAT[14]	DISP0_DAT14	ALT0
	DISP0_DAT[15]	DISP0_DAT15	ALT0
	DISP0_DAT[16]	DISP0_DAT16	ALT0
	DISP0_DAT[17]	DISP0_DAT17	ALT0

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
IPU	DISP0_DAT[18]	DISP0_DAT18	ALT0
	DISP0_DAT[19]	DISP0_DAT19	ALT0
	DISP0_DAT[1]	DISP0_DAT1	ALT0
	DISP0_DAT[20]	DISP0_DAT20	ALT0
	DISP0_DAT[21]	DISP0_DAT21	ALT0
	DISP0_DAT[22]	DISP0_DAT22	ALT0
	DISP0_DAT[23]	DISP0_DAT23	ALT0
	DISP0_DAT[2]	DISP0_DAT2	ALT0
	DISP0_DAT[3]	DISP0_DAT3	ALT0
	DISP0_DAT[4]	DISP0_DAT4	ALT0
	DISP0_DAT[5]	DISP0_DAT5	ALT0
	DISP0_DAT[6]	DISP0_DAT6	ALT0
	DISP0_DAT[7]	DISP0_DAT7	ALT0
	DISP0_DAT[8]	DISP0_DAT8	ALT0
	DISP0_DAT[9]	DISP0_DAT9	ALT0
	DISP1_DAT[0]	EIM_DA9	ALT3
	DISP1_DAT[10]	EIM_EB1	ALT3
	DISP1_DAT[11]	EIM_EB0	ALT3
	DISP1_DAT[12]	EIM_A17	ALT2
	DISP1_DAT[13]	EIM_A18	ALT2
	DISP1_DAT[14]	EIM_A19	ALT2
	DISP1_DAT[15]	EIM_A20	ALT2
	DISP1_DAT[16]	EIM_A21	ALT2
	DISP1_DAT[17]	EIM_A22	ALT2

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
IPU	DISP1_DAT[18]	EIM_A23	ALT2
	DISP1_DAT[19]	EIM_A24	ALT2
	DISP1_DAT[1]	EIM_DA8	ALT3
	DISP1_DAT[20]	EIM_D31	ALT5
	DISP1_DAT[21]	EIM_D30	ALT5
	DISP1_DAT[22]	EIM_D26	ALT7
	DISP1_DAT[23]	EIM_D27	ALT7
	DISP1_DAT[2]	EIM_DA7	ALT3
	DISP1_DAT[3]	EIM_DA6	ALT3
	DISP1_DAT[4]	EIM_DA5	ALT3
	DISP1_DAT[5]	EIM_DA4	ALT3
	DISP1_DAT[6]	EIM_DA3	ALT3
	DISP1_DAT[7]	EIM_DA2	ALT3
	DISP1_DAT[8]	EIM_DA1	ALT3
	DISP1_DAT[9]	EIM_DA0	ALT3
	DISPB0_SER_CLK	EIM_D21	ALT3
	DISPB0_SER_DIN	EIM_D22	ALT3
	DISPB0_SER_DIO	EIM_D28	ALT3
	DISPB0_SER_RS	EIM_D29	ALT3
	DISPB1_SER_CLK	EIM_D16	ALT3
	DISPB1_SER_DIN	EIM_D17	ALT3
	DISPB1_SER_DIO	EIM_D18	ALT3
	DISPB1_SER_RS	EIM_D19	ALT3
	EXT_TRIG	EIM_D28	ALT6

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
IPU	IPU_DIAG_BUS[0]	PATA_DATA0	ALT6
	IPU_DIAG_BUS[10]	PATA_DATA10	ALT6
	IPU_DIAG_BUS[11]	PATA_DATA11	ALT6
	IPU_DIAG_BUS[12]	PATA_DATA12	ALT6
	IPU_DIAG_BUS[13]	PATA_DATA13	ALT6
	IPU_DIAG_BUS[14]	PATA_DATA14	ALT6
	IPU_DIAG_BUS[15]	PATA_DATA15	ALT6
	IPU_DIAG_BUS[1]	PATA_DATA1	ALT6
	IPU_DIAG_BUS[2]	PATA_DATA2	ALT6
	IPU_DIAG_BUS[3]	PATA_DATA3	ALT6
	IPU_DIAG_BUS[4]	PATA_DATA4	ALT6
	IPU_DIAG_BUS[5]	PATA_DATA5	ALT6
	IPU_DIAG_BUS[6]	PATA_DATA6	ALT6
	IPU_DIAG_BUS[7]	PATA_DATA7	ALT6
	IPU_DIAG_BUS[8]	PATA_DATA8	ALT6
	IPU_DIAG_BUS[9]	PATA_DATA9	ALT6
	SER_DISP0_CS	EIM_D20	ALT3
	SER_DISP1_CS	EIM_EB2	ALT3
	SISG[0]	NANDF_CS2	ALT2
	SISG[1]	NANDF_CS3	ALT2
	SISG[2]	EIM_A24	ALT6
		EIM_D26	ALT6
	SISG[3]	EIM_A23	ALT6
		EIM_D27	ALT6
	SISG[4]	KEY_COL4	ALT3
	SISG[5]	KEY_ROW4	ALT3
	SNOOP2	GPIO_17	ALT6
XTALOSC32K	32K_OUT	FEC_RXD0	ALT3
		GPIO_10	ALT1
		KEY_ROW3	ALT5
		SD1_CLK	ALT2

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
SDMA	DEBUG_BUS_DEVICE[0]	DISP0_DAT21	ALT5
	DEBUG_BUS_DEVICE[1]	DISP0_DAT22	ALT5
	DEBUG_BUS_DEVICE[2]	DISP0_DAT23	ALT5
	DEBUG_BUS_DEVICE[3]	FEC_MDIO	ALT5
	DEBUG_BUS_DEVICE[4]	FEC_REF_CLK	ALT5
	DEBUG_BUS_ERROR	DISP0_DAT3	ALT5
	DEBUG_BUS_RWB	DISP0_DAT4	ALT5
	DEBUG_CORE_RUN	DISP0_DAT0	ALT5
	DEBUG_CORE_STATE[0]	DI0_DISP_CLK	ALT5
	DEBUG_CORE_STATE[1]	DI0_PIN15	ALT5
	DEBUG_CORE_STATE[2]	DI0_PIN2	ALT5
	DEBUG_CORE_STATE[3]	DI0_PIN3	ALT5
	DEBUG_EVENT_CHANNEL[0]	DISP0_DAT7	ALT5
	DEBUG_EVENT_CHANNEL[1]	DISP0_DAT8	ALT5
	DEBUG_EVENT_CHANNEL[2]	DISP0_DAT9	ALT5
	DEBUG_EVENT_CHANNEL[3]	DISP0_DAT10	ALT5
	DEBUG_EVENT_CHANNEL[4]	DISP0_DAT11	ALT5

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
SDMA	DEBUG_EVENT_CHANNEL[5]	DISP0_DAT12	ALT5
	DEBUG_EVENT_CHANNEL_SEL	DISP0_DAT1	ALT5
	DEBUG_EVT_CHN_LINES[0]	DISP0_DAT13	ALT5
	DEBUG_EVT_CHN_LINES[1]	DISP0_DAT14	ALT5
	DEBUG_EVT_CHN_LINES[2]	DISP0_DAT15	ALT5
	DEBUG_EVT_CHN_LINES[3]	DISP0_DAT16	ALT5
	DEBUG_EVT_CHN_LINES[4]	DISP0_DAT17	ALT5
	DEBUG_EVT_CHN_LINES[5]	DISP0_DAT18	ALT5
	DEBUG_EVT_CHN_LINES[6]	DISP0_DAT19	ALT5
	DEBUG_EVT_CHN_LINES[7]	DISP0_DAT20	ALT5
	DEBUG_MATCHED_DMBUS	DISP0_DAT5	ALT5
	DEBUG_MODE	DISP0_DAT2	ALT5
	DEBUG_PC[0]	CSI0_PIXCLK	ALT5
	DEBUG_PC[10]	CSI0_DAT16	ALT5
	DEBUG_PC[11]	CSI0_DAT17	ALT5
	DEBUG_PC[12]	CSI0_DAT18	ALT5
	DEBUG_PC[13]	CSI0_DAT19	ALT5
	DEBUG_PC[1]	CSI0_MCLK	ALT5
	DEBUG_PC[2]	CSI0_DATA_EN	ALT5
	DEBUG_PC[3]	CSI0_VSYNC	ALT5
	DEBUG_PC[4]	CSI0_DAT10	ALT5
	DEBUG_PC[5]	CSI0_DAT11	ALT5
	DEBUG_PC[6]	CSI0_DAT12	ALT5
	DEBUG_PC[7]	CSI0_DAT13	ALT5
SDMA	DEBUG_PC[8]	CSI0_DAT14	ALT5
	DEBUG_PC[9]	CSI0_DAT15	ALT5
	DEBUG_RTBUFFER_WRITE	DISP0_DAT6	ALT5
	DEBUG_YIELD	DI0_PIN4	ALT5
	SDMA_EXT_EVENT[0]	DISP0_DAT16	ALT4
		GPIO_17	ALT2
	SDMA_EXT_EVENT[1]	DISP0_DAT17	ALT4
		GPIO_18	ALT2

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
FEC	COL	FEC_MDIO	ALT3
		KEY_ROW1	ALT6
	CRS	KEY_COL3	ALT6
	MDC	FEC_MDC	ALT0
		KEY_ROW2	ALT4
	MDIO	FEC_MDIO	ALT0
		KEY_COL2	ALT4
	RDATA[0]	FEC_RXD0	ALT0
	RDATA[1]	FEC_RXD1	ALT0
	RDATA[2]	KEY_COL2	ALT6
	RDATA[3]	KEY_COL0	ALT6
	RX_CLK	FEC_RX_ER	ALT3
		KEY_COL1	ALT6
	RX_DV	FEC_CRS_DV	ALT0
	RX_ER	FEC_RX_ER	ALT0
	TDATA[0]	FEC_TXD0	ALT0
	TDATA[1]	FEC_TXD1	ALT0
	TDATA[2]	KEY_ROW2	ALT6
	TDATA[3]	GPIO_19	ALT6
	TX_CLK	FEC_REF_CLK	ALT0
	TX_EN	FEC_TX_EN	ALT0
	TX_ER	KEY_ROW0	ALT6

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
PATA	BUFFER_EN	PATA_BUFFER_EN	ALT0
	CS_0	PATA_CS_0	ALT0
	CS_1	PATA_CS_1	ALT0
	DA_0	PATA_DA_0	ALT0
	DA_1	PATA_DA_1	ALT0
	DA_2	PATA_DA_2	ALT0
	DIOR	PATA_DIOR	ALT0
	DIOW	PATA_DIOW	ALT0
	DMACK	PATA_DMACK	ALT0
	DMARQ	PATA_DMARQ	ALT0
	INTRQ	PATA_INTRQ	ALT0
	IORDY	PATA_IORDY	ALT0
	PATA_DATA[0]	PATA_DATA0	ALT0
	PATA_DATA[10]	PATA_DATA10	ALT0
	PATA_DATA[11]	PATA_DATA11	ALT0
	PATA_DATA[12]	PATA_DATA12	ALT0
	PATA_DATA[13]	PATA_DATA13	ALT0
	PATA_DATA[14]	PATA_DATA14	ALT0
	PATA_DATA[15]	PATA_DATA15	ALT0
	PATA_DATA[1]	PATA_DATA1	ALT0
	PATA_DATA[2]	PATA_DATA2	ALT0
	PATA_DATA[3]	PATA_DATA3	ALT0
	PATA_DATA[4]	PATA_DATA4	ALT0
	PATA_DATA[5]	PATA_DATA5	ALT0
	PATA_DATA[6]	PATA_DATA6	ALT0
	PATA_DATA[7]	PATA_DATA7	ALT0
	PATA_DATA[8]	PATA_DATA8	ALT0
	PATA_DATA[9]	PATA_DATA9	ALT0
	PATA_RESET_B	PATA_RESET_B	ALT0
OBSERVE_ MUX	OBSRV_INT_OUT0	GPIO_3	ALT5
	OBSRV_INT_OUT1	GPIO_6	ALT5
	OBSRV_INT_OUT2	GPIO_2	ALT5
	OBSRV_INT_OUT3	GPIO_4	ALT5
	OBSRV_INT_OUT4	GPIO_5	ALT5

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
FIRI	RXD	EIM_D26	ALT3
		GPIO_7	ALT5
	TXD	EIM_D27	ALT3
		GPIO_8	ALT5
CCM	ASRC_EXT_CLK	GPIO_18	ALT5
		KEY_ROW3	ALT3
	CCM_OUT_0	GPIO_6	ALT3
		PATA_DMARQ	ALT5
CCM	CCM_OUT_1	GPIO_2	ALT3
		PATA_BUFFER_EN	ALT5
	CCM_OUT_2	GPIO_4	ALT3
		PATA_INTRQ	ALT5
	CLKO	GPIO_0	ALT0
		GPIO_19	ALT2
		GPIO_5	ALT3
	CLKO2	GPIO_3	ALT4
	CSI0_MCLK	CSI0_MCLK	ALT2
		NANDF_CS2	ALT5
	DI0_EXT_CLK	EIM_DA14	ALT4
	DI1_EXT_CLK	EIM_DA13	ALT4
		EIM_EB2	ALT2
	PLL1_BYP	GPIO_5	ALT7
		SD1_CMD	ALT7
	PLL2_BYP	GPIO_7	ALT7
		SD1_DATA2	ALT7
	PLL3_BYP	GPIO_8	ALT7
		SD1_DATA0	ALT7
	PLL4_BYP	KEY_ROW3	ALT6
		SD1_DATA1	ALT7
	PMIC_VSTBY_REQ	PMIC_STBY_REQ	No Muxing (ALT0)
	REF_EN_B	GPIO_9	ALT3
	SSI_EXT1_CLK	GPIO_0	ALT3
	SSI_EXT2_CLK	GPIO_1	ALT3

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
GPT	IND_CAPIN1	SD1_DATA0	ALT3
	IND_CAPIN2	SD1_DATA1	ALT3
	IND_CLKIN	SD1_CLK	ALT3
	DO_CMPOUT1	SD1_CMD	ALT3
	DO_CMPOUT2	SD1_DATA2	ALT2
	DO_CMPOUT3	SD1_DATA3	ALT2
KPP	COL[0]	KEY_COL0	ALT0
	COL[1]	KEY_COL1	ALT0
	COL[2]	KEY_COL2	ALT0
	COL[3]	KEY_COL3	ALT0
	COL[4]	KEY_COL4	ALT0
	COL[5]	CSI0_DAT4	ALT2
		GPIO_0	ALT2
		GPIO_19	ALT0
		SD2_CLK	ALT2
	COL[6]	CSI0_DAT6	ALT2
		GPIO_9	ALT2
		SD2_DATA3	ALT2
	COL[7]	CSI0_DAT8	ALT2
		GPIO_4	ALT2
		SD2_DATA1	ALT2
	ROW[0]	KEY_ROW0	ALT0
	ROW[1]	KEY_ROW1	ALT0
	ROW[2]	KEY_ROW2	ALT0
	ROW[3]	KEY_ROW3	ALT0
	ROW[4]	KEY_ROW4	ALT0
	ROW[5]	CSI0_DAT5	ALT2
		GPIO_1	ALT2
		SD2_CMD	ALT2
	ROW[6]	CSI0_DAT7	ALT2
		GPIO_2	ALT2
		SD2_DATA2	ALT2
	ROW[7]	CSI0_DAT9	ALT2
		GPIO_5	ALT2
		SD2_DATA0	ALT2

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
I2C-1	SCL	CSI0_DAT9	ALT5
		EIM_D21	ALT5
	SDA	CSI0_DAT8	ALT5
		EIM_D28	ALT5
I2C-2	SCL	EIM_EB2	ALT5
		KEY_COL3	ALT4
	SDA	EIM_D16	ALT5
		KEY_ROW3	ALT4
I2C-3	SCL	EIM_D17	ALT5
		GPIO_3	ALT2
		GPIO_5	ALT6
	SDA	EIM_D18	ALT5
	SDA	GPIO_16	ALT6
		GPIO_6	ALT2
SPDIF	IN1	GPIO_16	ALT5
		KEY_COL3	ALT3
	OUT1	GPIO_17	ALT5
		GPIO_19	ALT3
	PLOCK	GPIO_7	ALT6
	SRCLK	GPIO_8	ALT6

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
GPIO-1	GPIO[0]	GPIO_0	ALT1
	GPIO[10]	SD2_CLK	ALT1
	GPIO[11]	SD2_CMD	ALT1
	GPIO[12]	SD2_DATA3	ALT1
	GPIO[13]	SD2_DATA2	ALT1
	GPIO[14]	SD2_DATA1	ALT1
	GPIO[15]	SD2_DATA0	ALT1
	GPIO[16]	SD1_DATA0	ALT1
	GPIO[17]	SD1_DATA1	ALT1
	GPIO[18]	SD1_CMD	ALT1
	GPIO[19]	SD1_DATA2	ALT1
	GPIO[1]	GPIO_1	ALT1
	GPIO[20]	SD1_CLK	ALT1
	GPIO[21]	SD1_DATA3	ALT1
	GPIO[22]	FEC_MDIO	ALT1
	GPIO[23]	FEC_REF_CLK	ALT1
GPIO-1	GPIO[24]	FEC_RX_ER	ALT1
	GPIO[25]	FEC_CRS_DV	ALT1
	GPIO[26]	FEC_RXD1	ALT1
	GPIO[27]	FEC_RXD0	ALT1
	GPIO[28]	FEC_TX_EN	ALT1
	GPIO[29]	FEC_TXD1	ALT1
	GPIO[2]	GPIO_2	ALT1
	GPIO[30]	FEC_TXD0	ALT1
	GPIO[31]	FEC_MDC	ALT1
	GPIO[3]	GPIO_3	ALT1
	GPIO[4]	GPIO_4	ALT1
	GPIO[5]	GPIO_5	ALT1
	GPIO[6]	GPIO_6	ALT1
	GPIO[7]	GPIO_7	ALT1
	GPIO[8]	GPIO_8	ALT1
	GPIO[9]	GPIO_9	ALT1

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
GPIO-2	GPIO[0]	PATA_DATA0	ALT1
	GPIO[10]	PATA_DATA10	ALT1
	GPIO[11]	PATA_DATA11	ALT1
	GPIO[12]	PATA_DATA12	ALT1
	GPIO[13]	PATA_DATA13	ALT1
	GPIO[14]	PATA_DATA14	ALT1
	GPIO[15]	PATA_DATA15	ALT1
	GPIO[16]	EIM_A22	ALT1
GPIO-2	GPIO[17]	EIM_A21	ALT1
	GPIO[18]	EIM_A20	ALT1
	GPIO[19]	EIM_A19	ALT1
	GPIO[1]	PATA_DATA1	ALT1
	GPIO[20]	EIM_A18	ALT1
	GPIO[21]	EIM_A17	ALT1
	GPIO[22]	EIM_A16	ALT1
	GPIO[23]	EIM_CS0	ALT1
	GPIO[24]	EIM_CS1	ALT1
	GPIO[25]	EIM_OE	ALT1
	GPIO[26]	EIM_RW	ALT1
	GPIO[27]	EIM_LBA	ALT1
	GPIO[28]	EIM_EB0	ALT1
	GPIO[29]	EIM_EB1	ALT1
	GPIO[2]	PATA_DATA2	ALT1
	GPIO[30]	EIM_EB2	ALT1
	GPIO[31]	EIM_EB3	ALT1
	GPIO[3]	PATA_DATA3	ALT1
	GPIO[4]	PATA_DATA4	ALT1
	GPIO[5]	PATA_DATA5	ALT1
	GPIO[6]	PATA_DATA6	ALT1
	GPIO[7]	PATA_DATA7	ALT1
	GPIO[8]	PATA_DATA8	ALT1
	GPIO[9]	PATA_DATA9	ALT1

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
GPIO-3	GPIO[0]	EIM_DA0	ALT1
	GPIO[10]	EIM_DA10	ALT1
	GPIO[11]	EIM_DA11	ALT1
	GPIO[12]	EIM_DA12	ALT1
	GPIO[13]	EIM_DA13	ALT1
	GPIO[14]	EIM_DA14	ALT1
	GPIO[15]	EIM_DA15	ALT1
	GPIO[16]	EIM_D16	ALT1
	GPIO[17]	EIM_D17	ALT1
	GPIO[18]	EIM_D18	ALT1
	GPIO[19]	EIM_D19	ALT1
	GPIO[1]	EIM_DA1	ALT1
	GPIO[20]	EIM_D20	ALT1
	GPIO[21]	EIM_D21	ALT1
	GPIO[22]	EIM_D22	ALT1
	GPIO[23]	EIM_D23	ALT1
	GPIO[24]	EIM_D24	ALT1
	GPIO[25]	EIM_D25	ALT1
	GPIO[26]	EIM_D26	ALT1
	GPIO[27]	EIM_D27	ALT1
	GPIO[28]	EIM_D28	ALT1
	GPIO[29]	EIM_D29	ALT1
	GPIO[2]	EIM_DA2	ALT1
	GPIO[30]	EIM_D30	ALT1
GPIO-3	GPIO[31]	EIM_D31	ALT1
	GPIO[3]	EIM_DA3	ALT1
	GPIO[4]	EIM_DA4	ALT1
	GPIO[5]	EIM_DA5	ALT1
	GPIO[6]	EIM_DA6	ALT1
	GPIO[7]	EIM_DA7	ALT1
	GPIO[8]	EIM_DA8	ALT1
	GPIO[9]	EIM_DA9	ALT1

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
GPIO-4	GPIO[0]	GPIO_10	ALT0
	GPIO[10]	KEY_COL2	ALT1
	GPIO[11]	KEY_ROW2	ALT1
	GPIO[12]	KEY_COL3	ALT1
	GPIO[13]	KEY_ROW3	ALT1
	GPIO[14]	KEY_COL4	ALT1
	GPIO[15]	KEY_ROW4	ALT1
	GPIO[16]	DI0_DISP_CLK	ALT1
	GPIO[17]	DI0_PIN15	ALT1
	GPIO[18]	DI0_PIN2	ALT1
	GPIO[19]	DI0_PIN3	ALT1
	GPIO[1]	GPIO_11	No Muxing (ALT0)
	GPIO[20]	DI0_PIN4	ALT1
	GPIO[21]	DISP0_DAT0	ALT1
	GPIO[22]	DISP0_DAT1	ALT1
	GPIO[23]	DISP0_DAT2	ALT1
GPIO-4	GPIO[24]	DISP0_DAT3	ALT1
	GPIO[25]	DISP0_DAT4	ALT1
	GPIO[26]	DISP0_DAT5	ALT1
	GPIO[27]	DISP0_DAT6	ALT1
	GPIO[28]	DISP0_DAT7	ALT1
	GPIO[29]	DISP0_DAT8	ALT1
	GPIO[2]	GPIO_12	No Muxing (ALT0)
	GPIO[30]	DISP0_DAT9	ALT1
	GPIO[31]	DISP0_DAT10	ALT1
	GPIO[3]	GPIO_13	No Muxing (ALT0)
	GPIO[4]	GPIO_14	No Muxing (ALT0)
	GPIO[5]	GPIO_19	ALT1
	GPIO[6]	KEY_COL0	ALT1
	GPIO[7]	KEY_ROW0	ALT1
	GPIO[8]	KEY_COL1	ALT1
	GPIO[9]	KEY_ROW1	ALT1

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
GPIO-5	GPIO[0]	EIM_WAIT	ALT1
	GPIO[10]	DISP0_DAT16	ALT1
	GPIO[11]	DISP0_DAT17	ALT1
	GPIO[12]	DISP0_DAT18	ALT1
	GPIO[13]	DISP0_DAT19	ALT1
	GPIO[14]	DISP0_DAT20	ALT1
	GPIO[15]	DISP0_DAT21	ALT1
	GPIO[16]	DISP0_DAT22	ALT1
GPIO-5	GPIO[17]	DISP0_DAT23	ALT1
	GPIO[18]	CSI0_PIXCLK	ALT1
	GPIO[19]	CSI0_MCLK	ALT1
	GPIO[20]	CSI0_DATA_EN	ALT1
	GPIO[21]	CSI0_VSYNC	ALT1
	GPIO[22]	CSI0_DAT4	ALT1
	GPIO[23]	CSI0_DAT5	ALT1
	GPIO[24]	CSI0_DAT6	ALT1
	GPIO[25]	CSI0_DAT7	ALT1
	GPIO[26]	CSI0_DAT8	ALT1
	GPIO[27]	CSI0_DAT9	ALT1
	GPIO[28]	CSI0_DAT10	ALT1
	GPIO[29]	CSI0_DAT11	ALT1
	GPIO[2]	EIM_A25	ALT1
	GPIO[30]	CSI0_DAT12	ALT1
	GPIO[31]	CSI0_DAT13	ALT1
	GPIO[4]	EIM_A24	ALT1
	GPIO[5]	DISP0_DAT11	ALT1
	GPIO[6]	DISP0_DAT12	ALT1
	GPIO[7]	DISP0_DAT13	ALT1
	GPIO[8]	DISP0_DAT14	ALT1
	GPIO[9]	DISP0_DAT15	ALT1
GPIO-6	GPIO[0]	CSI0_DAT14	ALT1
	GPIO[10]	NANDF_RB0	ALT1

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
GPIO-6	GPIO[11]	NANDF_CS0	ALT1
	GPIO[12]	NANDF_WE_B	ALT1
	GPIO[13]	NANDF_RE_B	ALT1
	GPIO[14]	NANDF_CS1	ALT1
	GPIO[15]	NANDF_CS2	ALT1
	GPIO[16]	NANDF_CS3	ALT1
	GPIO[17]	PATA_DIOW	ALT1
	GPIO[18]	PATA_DMACK	ALT1
	GPIO[1]	CSI0_DAT15	ALT1
	GPIO[2]	CSI0_DAT16	ALT1
	GPIO[3]	CSI0_DAT17	ALT1
	GPIO[4]	CSI0_DAT18	ALT1
	GPIO[5]	CSI0_DAT19	ALT1
	GPIO[6]	EIM_A23	ALT1
	GPIO[7]	NANDF_CLE	ALT1
	GPIO[8]	NANDF_ALE	ALT1
	GPIO[9]	NANDF_WP_B	ALT1
	GPI[22]	LVDS1_TX3_P	ALT0
	GPI[23]	LVDS1_TX3_N	No Muxing (ALT0)
	GPI[24]	LVDS1_TX2_P	ALT0
	GPI[25]	LVDS1_TX2_N	No Muxing (ALT0)
	GPI[26]	LVDS1_CLK_P	ALT0
	GPI[27]	LVDS1_CLK_N	No Muxing (ALT0)
	GPI[28]	LVDS1_TX1_P	ALT0
	GPI[29]	LVDS1_TX1_N	No Muxing (ALT0)
	GPI[30]	LVDS1_TX0_P	ALT0
	GPI[31]	LVDS1_TX0_N	No Muxing (ALT0)

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
GPIO-7	GPIO[0]	PATA_DMARQ	ALT1
	GPIO[10]	PATA_CS_1	ALT1
	GPIO[11]	GPIO_16	ALT1
	GPIO[12]	GPIO_17	ALT1
	GPIO[13]	GPIO_18	ALT1
	GPIO[1]	PATA_BUFFER_EN	ALT1
	GPIO[2]	PATA_INTRQ	ALT1
	GPIO[3]	PATA_DIOR	ALT1
	GPIO[4]	PATA_RESET_B	ALT1
	GPIO[5]	PATA_IORDY	ALT1
	GPIO[6]	PATA_DA_0	ALT1
	GPIO[7]	PATA_DA_1	ALT1
	GPIO[8]	PATA_DA_2	ALT1
	GPIO[9]	PATA_CS_0	ALT1
	GPI[22]	LVDS0_TX3_P	ALT0
	GPI[23]	LVDS0_TX3_N	No Muxing (ALT0)
	GPI[24]	LVDS0_CLK_P	ALT0
	GPI[25]	LVDS0_CLK_N	No Muxing (ALT0)
	GPI[26]	LVDS0_TX2_P	ALT0
	GPI[27]	LVDS0_TX2_N	No Muxing (ALT0)
	GPI[28]	LVDS0_TX1_P	ALT0
	GPI[29]	LVDS0_TX1_N	No Muxing (ALT0)
	GPI[30]	LVDS0_TX0_P	ALT0
	GPI[31]	LVDS0_TX0_N	No Muxing (ALT0)
SJC	DE_B	GPIO_16	ALT7
	DONE	SD2_DATA3	ALT7
	FAIL	SD2_DATA2	ALT7
	JTAG_ACT	GPIO_17	ALT7
	MOD	JTAG_MOD	No Muxing (ALT0)
	TCK	JTAG_TCK	No Muxing (ALT0)
	TDI	JTAG_TDI	No Muxing (ALT0)
	TDO	JTAG_TDO	No Muxing (ALT0)
	TMS	JTAG_TMS	No Muxing (ALT0)
	TRSTB	JTAG_TRSTB	No Muxing (ALT0)

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
CAMP-1	CKIH	CKIH1	No Muxing (ALT0)
CAMP-2		CKIH2	No Muxing (ALT0)
SCC	FAIL_STATE	GPIO_9	ALT7
	RANDOM	SD2_CMD	ALT7
	RANDOM_V	SD2_CLK	ALT7
	SEC_STATE	GPIO_4	ALT7
CSPI	MISO	DISP0_DAT2	ALT2
		EIM_D22	ALT4
		SD1_DATA0	ALT5
		SD2_DATA0	ALT5
	MOSI	DISP0_DAT1	ALT2
CSPI	MOSI	EIM_D28	ALT4
		SD1_CMD	ALT5
		SD2_CMD	ALT5
	RDY	DISP0_DAT7	ALT2
	SCLK	DISP0_DAT0	ALT2
		EIM_D21	ALT4
		SD1_CLK	ALT5
		SD2_CLK	ALT5
	SS0	DISP0_DAT3	ALT2
		EIM_D20	ALT4
		EIM_D29	ALT4
		SD1_DATA1	ALT5
		SD2_DATA1	ALT5
	SS1	DISP0_DAT4	ALT2
		EIM_A25	ALT4
		SD1_DATA2	ALT5
		SD2_DATA2	ALT5
	SS2	DISP0_DAT5	ALT2
		EIM_D24	ALT4
		SD1_DATA3	ALT5
		SD2_DATA3	ALT5
	SS3	DISP0_DAT6	ALT2
		EIM_D25	ALT4
SRC	ANY_PU_RST	KEY_COL0	ALT7

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
SRC	BOOT_MODE[0]	BOOT_MODE0	No Muxing (ALT0)
	BOOT_MODE[1]	BOOT_MODE1	No Muxing (ALT0)
	BT_CFG1[0]	EIM_LBA	ALT7
	BT_CFG1[1]	EIM_A16	ALT7
	BT_CFG1[2]	EIM_A17	ALT7
	BT_CFG1[3]	EIM_A18	ALT7
	BT_CFG1[4]	EIM_A19	ALT7
	BT_CFG1[5]	EIM_A20	ALT7
	BT_CFG1[6]	EIM_A21	ALT7
	BT_CFG1[7]	EIM_A22	ALT7
	BT_CFG2[2]	EIM_DA3	ALT7
	BT_CFG2[3]	EIM_DA2	ALT7
	BT_CFG2[4]	EIM_DA1	ALT7
	BT_CFG2[5]	EIM_DA0	ALT7
	BT_CFG2[6]	EIM_EB1	ALT7
	BT_CFG2[7]	EIM_EB0	ALT7
	BT_CFG3[1]	EIM_DA10	ALT7
	BT_CFG3[2]	EIM_DA9	ALT7
	BT_CFG3[3]	EIM_DA8	ALT7
	BT_CFG3[4]	EIM_DA7	ALT7
	BT_CFG3[5]	EIM_DA6	ALT7
	BT_CFG3[6]	EIM_DA5	ALT7
	BT_CFG3[7]	EIM_DA4	ALT7
	INT_BOOT	GPIO_19	ALT7
	POR_B	POR_B	No Muxing (ALT0)
	RESET_B	RESET_IN_B	No Muxing (ALT0)
	SYSTEM_RST	GPIO_18	ALT7
	TESTER_ACK	GPIO_1	ALT7

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
RTC	CE_RTC_ALARM1_TRIG	FEC_MDC	ALT4
	CE_RTC_ALARM2_TRIG	GPIO_18	ALT4
	CE_RTC_EXT_TRIG1	GPIO_16	ALT4
	CE_RTC_EXT_TRIG2	GPIO_19	ALT4
	CE_RTC_FSV_TRIG	GPIO_17	ALT4
	CE_RTC_PRSC_CLK	FEC_TXD1	ALT4
	CE_RTC_PS1	FEC_RXD1	ALT4
	CE_RTC_PS2	FEC_MDIO	ALT4
	CE_RTC_PS3	FEC_RX_ER	ALT4
TPIU	TRACE[0]	CSI0_VSYNC	ALT7
	TRACE[10]	CSI0_DAT13	ALT7
	TRACE[11]	CSI0_DAT14	ALT7
	TRACE[12]	CSI0_DAT15	ALT7
	TRACE[13]	CSI0_DAT16	ALT7
	TRACE[14]	CSI0_DAT17	ALT7
	TRACE[15]	CSI0_DAT18	ALT7
	TRACE[1]	CSI0_DAT4	ALT7
	TRACE[2]	CSI0_DAT5	ALT7
	TRACE[3]	CSI0_DAT6	ALT7
	TRACE[4]	CSI0_DAT7	ALT7
	TRACE[5]	CSI0_DAT8	ALT7
	TRACE[6]	CSI0_DAT9	ALT7
	TRACE[7]	CSI0_DAT10	ALT7
	TRACE[8]	CSI0_DAT11	ALT7
	TRACE[9]	CSI0_DAT12	ALT7
	TRCLK	CSI0_DATA_EN	ALT7
	TRCTL	CSI0_MCLK	ALT7

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
GPU3D	GPU_DEBUG_OUT[0]	PATA_DATA0	ALT5
	GPU_DEBUG_OUT[10]	PATA_DATA10	ALT5
	GPU_DEBUG_OUT[11]	PATA_DATA11	ALT5
	GPU_DEBUG_OUT[12]	PATA_DATA12	ALT5
	GPU_DEBUG_OUT[13]	PATA_DATA13	ALT5
	GPU_DEBUG_OUT[14]	PATA_DATA14	ALT5
	GPU_DEBUG_OUT[15]	PATA_DATA15	ALT5
	GPU_DEBUG_OUT[1]	PATA_DATA1	ALT5
	GPU_DEBUG_OUT[2]	PATA_DATA2	ALT5
	GPU_DEBUG_OUT[3]	PATA_DATA3	ALT5
	GPU_DEBUG_OUT[4]	PATA_DATA4	ALT5
	GPU_DEBUG_OUT[5]	PATA_DATA5	ALT5
	GPU_DEBUG_OUT[6]	PATA_DATA6	ALT5
	GPU_DEBUG_OUT[7]	PATA_DATA7	ALT5
	GPU_DEBUG_OUT[8]	PATA_DATA8	ALT5
	GPU_DEBUG_OUT[9]	PATA_DATA9	ALT5
FLEXCAN-1	RXCAN	GPIO_8	ALT3
	RXCAN	KEY_ROW2	ALT2
		PATA_DIOR	ALT4
	TXCAN	GPIO_7	ALT3
		KEY_COL2	ALT2
		PATA_INTRQ	ALT4
FLEXCAN-2	RXCAN	KEY_ROW4	ALT2
		PATA_IORDY	ALT4
	TXCAN	KEY_COL4	ALT2
		PATA_RESET_B	ALT4
PWM-1	PWMO	DISP0_DAT8	ALT2
		GPIO_9	ALT4
		SD1_DATA3	ALT3
PWM-2		DISP0_DAT9	ALT2
		GPIO_1	ALT4
		SD1_DATA2	ALT3
OWIRE	LINE	GPIO_18	ALT3
		PATA_DA_0	ALT4

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
LDB[1]	LVDS0_CLK	LVDS0_CLK_P	ALT1
	LVDS0_TX0	LVDS0_TX0_P	ALT1
	LVDS0_TX1	LVDS0_TX1_P	ALT1
	LVDS0_TX2	LVDS0_TX2_P	ALT1
	LVDS0_TX3	LVDS0_TX3_P	ALT1
	LVDS1_CLK	LVDS1_CLK_P	ALT1
	LVDS1_TX0	LVDS1_TX0_P	ALT1
	LVDS1_TX1	LVDS1_TX1_P	ALT1
	LVDS1_TX2	LVDS1_TX2_P	ALT1
	LVDS1_TX3	LVDS1_TX3_P	ALT1
EPIT-1	EPITO	EIM_D19	ALT5
		GPIO_0	ALT4
		GPIO_7	ALT2
EPIT-2		EIM_D20	ALT5
		GPIO_8	ALT2
CSU	CSU_ALARM_AUT[0]	GPIO_2	ALT4
	CSU_ALARM_AUT[1]	GPIO_4	ALT4
	CSU_ALARM_AUT[2]	GPIO_5	ALT4
	CSU_INT_DEB	GPIO_6	ALT4
	TD	GPIO_0	ALT7

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
USBPHY-1	AVALID	DI0_DISP_CLK	ALT7
	BISTOK	EIM_A25	ALT7
	BVALID	DI0_PIN15	ALT7
	DATAOUT[0]	PATA_RESET_B	ALT7
	DATAOUT[1]	PATA_IORDY	ALT7
	DATAOUT[2]	PATA_DA_0	ALT7
	DATAOUT[3]	PATA_DA_1	ALT7
	DATAOUT[4]	PATA_DA_2	ALT7
	DATAOUT[5]	PATA_CS_0	ALT7
	DATAOUT[6]	PATA_CS_1	ALT7
	DATAOUT[7]	PATA_DATA0	ALT7
	ENDSESSION	DI0_PIN2	ALT7
	HOSTDISCONNECT	DI0_PIN4	ALT7
	IDDIG	DI0_PIN3	ALT7
	LINESTATE[0]	KEY_ROW3	ALT7
	LINESTATE[1]	KEY_COL4	ALT7
	RXACTIVE	KEY_COL2	ALT7
	RXERROR	KEY_ROW2	ALT7
	RXVALID	KEY_ROW1	ALT7
	SIECLOCK	KEY_COL3	ALT7
	TXREADY	KEY_COL1	ALT7
	VBUSVALID	KEY_ROW4	ALT7
	VSTATUS[0]	NANDF_CLE	ALT7
	VSTATUS[1]	NANDF_ALE	ALT7
	VSTATUS[2]	NANDF_WP_B	ALT7
	VSTATUS[3]	NANDF_RB0	ALT7
	VSTATUS[4]	NANDF_CS0	ALT7
	VSTATUS[5]	NANDF_CS1	ALT7
	VSTATUS[6]	NANDF_CS2	ALT7
	VSTATUS[7]	NANDF_CS3	ALT7

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
USBPHY-2	AVALID	DISP0_DAT8	ALT7
	BISTOK	CSI0_DAT19	ALT7
	BVALID	EIM_A24	ALT7
	DATAOUT[0]	FEC_TXD0	ALT7
	DATAOUT[1]	FEC_MDC	ALT7
	DATAOUT[2]	PATA_DIOW	ALT7
	DATAOUT[3]	PATA_DMACK	ALT7
	DATAOUT[4]	PATA_DMARQ	ALT7
	DATAOUT[5]	PATA_BUFFER_EN	ALT7
	DATAOUT[6]	PATA_INTRQ	ALT7
	DATAOUT[7]	PATA_DIOR	ALT7
	ENDSESSION	EIM_A23	ALT7
	HOSTDISCONNECT	EIM_RW	ALT7
	IDDIG	EIM_OE	ALT7
	LINESTATE[0]	DISP0_DAT5	ALT7
	LINESTATE[1]	DISP0_DAT6	ALT7
	RXACTIVE	DISP0_DAT2	ALT7
	RXERROR	DISP0_DAT3	ALT7
	RXVALID	DISP0_DAT1	ALT7
	SIECLOCK	DISP0_DAT4	ALT7
	TXREADY	DISP0_DAT0	ALT7
	VBUSVALID	DISP0_DAT7	ALT7
	VSTATUS[0]	DISP0_DAT9	ALT7
	VSTATUS[1]	DISP0_DAT10	ALT7
	VSTATUS[2]	DISP0_DAT11	ALT7
	VSTATUS[3]	DISP0_DAT12	ALT7
	VSTATUS[4]	DISP0_DAT13	ALT7
	VSTATUS[5]	DISP0_DAT14	ALT7
	VSTATUS[6]	DISP0_DAT15	ALT7
	VSTATUS[7]	DISP0_DAT16	ALT7

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
AUDMUX	AUD3_RXC	CSI0_DAT10	ALT4
	AUD3_RXD	CSI0_DAT7	ALT5
	AUD3_RXFS	CSI0_DAT11	ALT4
	AUD3_TXC	CSI0_DAT4	ALT5
	AUD3_TXD	CSI0_DAT5	ALT5
	AUD3_TXFS	CSI0_DAT6	ALT5
	AUD4_RXC	DISP0_DAT19	ALT4
		SD2_CMD	ALT3
	AUD4_RXD	DISP0_DAT23	ALT3
		SD2_DATA0	ALT3
	AUD4_RXFS	DISP0_DAT18	ALT4
		SD2_CLK	ALT3
	AUD4_TXC	DISP0_DAT20	ALT3
		SD2_DATA3	ALT3
	AUD4_TXD	DISP0_DAT21	ALT3
		SD2_DATA2	ALT3
	AUD4_TXFS	DISP0_DAT22	ALT3
		SD2_DATA1	ALT3
	AUD5_RXC	DISP0_DAT14	ALT3
		EIM_D25	ALT5
	AUD5_RXD	DISP0_DAT19	ALT3
		KEY_ROW1	ALT2
	AUD5_RXFS	DISP0_DAT13	ALT3
AUDMUX	AUD5_RXFS	EIM_D24	ALT5
	AUD5_TXC	DISP0_DAT16	ALT3
		KEY_COL0	ALT2
	AUD5_TXD	DISP0_DAT17	ALT3
		KEY_ROW0	ALT2
	AUD5_TXFS	DISP0_DAT18	ALT3
		KEY_COL1	ALT2
	AUD6_RXD	DI0_PIN4	ALT2
	AUD6_TXC	DI0_PIN15	ALT2
	AUD6_TXD	DI0_PIN2	ALT2
	AUD6_TXFS	DI0_PIN3	ALT2

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
ESDHCV2-1	CD	GPIO_1	ALT6
	CLK	SD1_CLK	ALT0
	CMD	SD1_CMD	ALT0
	DAT0	SD1_DATA0	ALT0
	DAT1	SD1_DATA1	ALT0
	DAT2	SD1_DATA2	ALT0
	DAT3	SD1_DATA3	ALT0
	DAT4	PATA_DATA8	ALT2
	DAT5	PATA_DATA9	ALT2
	DAT6	PATA_DATA10	ALT2
	DAT7	PATA_DATA11	ALT2
	LCTL	GPIO_18	ALT6
	WP	DI0_PIN4	ALT3
	WP	GPIO_9	ALT6
ESDHCV2-2	CD	GPIO_4	ALT6
	CLK	SD2_CLK	ALT0
	CMD	SD2_CMD	ALT0
	DAT0	SD2_DATA0	ALT0
	DAT1	SD2_DATA1	ALT0
	DAT2	SD2_DATA2	ALT0
	DAT3	SD2_DATA3	ALT0
	DAT4	PATA_DATA12	ALT2
	DAT5	PATA_DATA13	ALT2
	DAT6	PATA_DATA14	ALT2
	DAT7	PATA_DATA15	ALT2
	LCTL	GPIO_6	ALT6
	WP	GPIO_2	ALT6

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
ESDHCV3-3	CLK	PATA_IORDY	ALT2
	CMD	PATA_RESET_B	ALT2
	DAT0	PATA_DATA8	ALT4
	DAT1	PATA_DATA9	ALT4
	DAT2	PATA_DATA10	ALT4
	DAT3	PATA_DATA11	ALT4
	DAT4	PATA_DATA0	ALT4
	DAT5	PATA_DATA1	ALT4
	DAT6	PATA_DATA2	ALT4
	DAT7	PATA_DATA3	ALT4
	RST	PATA_DA_0	ALT2
ESDHCV2-4	CLK	PATA_DA_2	ALT2
	CMD	PATA_DA_1	ALT2
	DAT0	PATA_DATA12	ALT4
	DAT1	PATA_DATA13	ALT4
	DAT2	PATA_DATA14	ALT4
	DAT3	PATA_DATA15	ALT4
	DAT4	PATA_DATA4	ALT4
	DAT5	PATA_DATA5	ALT4
	DAT6	PATA_DATA6	ALT4
	DAT7	PATA_DATA7	ALT4
SATA_PHY	DTB[0]	SD1_CLK	ALT7
	DTB[1]	SD1_DATA3	ALT7
	TCK	DISP0_DAT22	ALT7
	TDI	DISP0_DAT20	ALT7
	TDO	DISP0_DAT21	ALT7
	TMS	DISP0_DAT23	ALT7
USB	H2_DM	KEY_ROW3	ALT2
	H2_DP	KEY_COL3	ALT2
	USBH1_OC	EIM_D30	ALT6
		GPIO_3	ALT6
	USBH1_PWR	EIM_D31	ALT6
		GPIO_0	ALT6
	USBH2_CLK	DISP0_DAT12	ALT2

Table continues on the next page...

Table 4-3. Muxing Options sorted by IPs (continued)

Block Instance	Block I/O	Pin	Mode
USB	USBH2_DATA[0]	DISP0_DAT0	ALT3
	USBH2_DATA[1]	DISP0_DAT1	ALT3
	USBH2_DATA[2]	DISP0_DAT2	ALT3
	USBH2_DATA[3]	DISP0_DAT3	ALT3
	USBH2_DATA[4]	DISP0_DAT4	ALT3
	USBH2_DATA[5]	DISP0_DAT5	ALT3
	USBH2_DATA[6]	DISP0_DAT6	ALT3
	USBH2_DATA[7]	DISP0_DAT7	ALT3
	USBH2_DIR	DI0_DISP_CLK	ALT2
	USBH2_NXT	DISP0_DAT11	ALT2
	USBH2_OC	EIM_D19	ALT7
		EIM_D30	ALT7
	USBH2_PWR	EIM_D20	ALT7
		EIM_D31	ALT7
	USBH2_STP	DISP0_DAT10	ALT2
	USBH3_CLK	CSI0_DAT6	ALT4
	USBH3_DATA[0]	CSI0_DAT12	ALT4
	USBH3_DATA[1]	CSI0_DAT13	ALT4
	USBH3_DATA[2]	CSI0_DAT14	ALT4
	USBH3_DATA[3]	CSI0_DAT15	ALT4
	USBH3_DATA[4]	CSI0_DAT16	ALT4
	USBH3_DATA[5]	CSI0_DAT17	ALT4
	USBH3_DATA[6]	CSI0_DAT18	ALT4
	USBH3_DATA[7]	CSI0_DAT19	ALT4
USB	USBH3_DIR	CSI0_DAT7	ALT4
	USBH3_NXT	CSI0_DAT5	ALT4
	USBH3_OC	CSI0_DAT8	ALT4
	USBH3_PWR	CSI0_DAT9	ALT4
	USBH3_STP	CSI0_DAT4	ALT4
	USBOTG_OC	EIM_D21	ALT6
		KEY_COL4	ALT5
	USBOTG_PWR	EIM_D22	ALT6
		KEY_ROW4	ALT5

1. If the LVDS Display Bridge (LDB) block is disabled, then each LVDS output pin pair is available to be used as a pair of general purpose CMOS inputs (controlled by the GPIO blocks).

4.2.2 Daisy Chain Control

To increase the flexibility of routing signals from package pins to certain block I/Os, the IOMUXC provides multiplexers that can be programmed to receive inputs from a number of different package pins. This is called daisy chaining. This selection is controlled by daisy chain control registers. These are described below.

Table 4-4 shows the daisy chain control registers sorted by package pin.

Table 4-4. Daisy Chain Multiplexer Control Registers (by pin)

Pin Name	Daisy Chain Control Registers
GPIO_19	IOMUXC_KPP_IPP_IND_COL_5_SELECT_INPUT
KEY_COL0	IOMUXC_AUDMUX_P5_INPUT_TXCLK_AMX_SELECT_INPUT IOMUXC_ECSPI1_IPP_CSPI_CLK_IN_SELECT_INPUT IOMUXC_UART4_IPP_UART_RXD_MUX_SELECT_INPUT
KEY_ROW0	IOMUXC_AUDMUX_P5_INPUT_DB_AMX_SELECT_INPUT IOMUXC_ECSPI1_IPP_IND_MOSI_SELECT_INPUT IOMUXC_UART4_IPP_UART_RXD_MUX_SELECT_INPUT
KEY_COL1	IOMUXC_AUDMUX_P5_INPUT_TXFS_AMX_SELECT_INPUT IOMUXC_ECSPI1_IPP_IND_MISO_SELECT_INPUT IOMUXC_FEC_FEC_RX_CLK_SELECT_INPUT IOMUXC_UART5_IPP_UART_RXD_MUX_SELECT_INPUT
KEY_ROW1	IOMUXC_AUDMUX_P5_INPUT_DA_AMX_SELECT_INPUT IOMUXC_ECSPI1_IPP_IND_SS_B_0_SELECT_INPUT IOMUXC_FEC_FEC_COL_SELECT_INPUT IOMUXC_UART5_IPP_UART_RXD_MUX_SELECT_INPUT
KEY_COL2	IOMUXC_ECSPI1_IPP_IND_SS_B_1_SELECT_INPUT IOMUXC_FEC_FEC_MDI_SELECT_INPUT
KEY_ROW2	IOMUXC_CAN1_IPP_IND_CANRX_SELECT_INPUT IOMUXC_ECSPI1_IPP_IND_SS_B_2_SELECT_INPUT
KEY_COL3	IOMUXC_ECSPI1_IPP_IND_SS_B_3_SELECT_INPUT IOMUXC_I2C2_IPP_SCL_IN_SELECT_INPUT IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT
KEY_ROW3	IOMUXC_CCM_IPP_ASRC_EXT_SELECT_INPUT IOMUXC_CCM_PLL4_BYPASS_CLK_SELECT_INPUT IOMUXC_I2C2_IPP_SDA_IN_SELECT_INPUT
KEY_COL4	IOMUXC_UART5_IPP_UART_RTS_B_SELECT_INPUT IOMUXC_USBOH3_IPP_IND_OTG_OC_SELECT_INPUT

Table continues on the next page...

Table 4-4. Daisy Chain Multiplexer Control Registers (by pin) (continued)

Pin Name	Daisy Chain Control Registers
KEY_ROW4	IOMUXC_CAN2_IPP_IND_CANRX_SELECT_INPUT IOMUXC_UART5_IPP_UART_RTS_B_SELECT_INPUT
NVCC_KEYPAD	
DI0_DISP_CLK	
DI0_PIN15	
DI0_PIN2	
DI0_PIN3	
DI0_PIN4	IOMUXC_ESDHC1_IPP_WP_ON_SELECT_INPUT
DISP0_DAT0	IOMUXC_CSPI_IPP_CSPI_CLK_IN_SELECT_INPUT
DISP0_DAT1	IOMUXC_CSPI_IPP_IND_MOSI_SELECT_INPUT
DISP0_DAT2	IOMUXC_CSPI_IPP_IND_MISO_SELECT_INPUT
DISP0_DAT3	IOMUXC_CSPI_IPP_IND_SS0_B_SELECT_INPUT
DISP0_DAT4	IOMUXC_CSPI_IPP_IND_SS1_B_SELECT_INPUT
DISP0_DAT5	IOMUXC_CSPI_IPP_IND_SS2_B_SELECT_INPUT
DISP0_DAT6	IOMUXC_CSPI_IPP_IND_SS3_B_SELECT_INPUT
DISP0_DAT7	
DISP0_DAT8	
DISP0_DAT9	
DISP0_DAT10	
DISP0_DAT11	
DISP0_DAT12	
DISP0_DAT13	IOMUXC_AUDMUX_P5_INPUT_RXFS_AMX_SELECT_INPUT
DISP0_DAT14	IOMUXC_AUDMUX_P5_INPUT_RXCLK_AMX_SELECT_INPUT
DISP0_DAT15	IOMUXC_ECSPI1_IPP_IND_SS_B_1_SELECT_INPUT IOMUXC_ECSPI2_IPP_IND_SS_B_1_SELECT_INPUT
DISP0_DAT16	IOMUXC_AUDMUX_P5_INPUT_TXCLK_AMX_SELECT_INPUT IOMUXC_ECSPI2_IPP_IND_MOSI_SELECT_INPUT IOMUXC_SDMA_EVENTS_14_SELECT_INPUT
DISP0_DAT17	IOMUXC_AUDMUX_P5_INPUT_DB_AMX_SELECT_INPUT IOMUXC_ECSPI2_IPP_IND_MISO_SELECT_INPUT IOMUXC_SDMA_EVENTS_15_SELECT_INPUT
DISP0_DAT18	IOMUXC_AUDMUX_P4_INPUT_RXFS_AMX_SELECT_INPUT IOMUXC_AUDMUX_P5_INPUT_TXFS_AMX_SELECT_INPUT IOMUXC_ECSPI2_IPP_IND_SS_B_0_SELECT_INPUT

Table continues on the next page...

Table 4-4. Daisy Chain Multiplexer Control Registers (by pin) (continued)

Pin Name	Daisy Chain Control Registers
DISP0_DAT19	IOMUXC_AUDMUX_P4_INPUT_RXCLK_AMX_SELECT_INPUT IOMUXC_AUDMUX_P5_INPUT_DA_AMX_SELECT_INPUT IOMUXC_ECSPI2_IPP_CSPI_CLK_IN_SELECT_INPUT
DISP0_DAT20	IOMUXC_AUDMUX_P4_INPUT_TXCLK_AMX_SELECT_INPUT IOMUXC_ECSPI1_IPP_CSPI_CLK_IN_SELECT_INPUT
DISP0_DAT21	IOMUXC_AUDMUX_P4_INPUT_DB_AMX_SELECT_INPUT IOMUXC_ECSPI1_IPP_IND_MOSI_SELECT_INPUT
DISP0_DAT22	IOMUXC_AUDMUX_P4_INPUT_TXFS_AMX_SELECT_INPUT IOMUXC_ECSPI1_IPP_IND_MISO_SELECT_INPUT
DISP0_DAT23	IOMUXC_AUDMUX_P4_INPUT_DA_AMX_SELECT_INPUT IOMUXC_ECSPI1_IPP_IND_SS_B_0_SELECT_INPUT
CSI0_PIXCLK	
CSI0_MCLK	
CSI0_DATA_EN	
CSI0_VSYNC	
CSI0_DAT4	IOMUXC_ECSPI1_IPP_CSPI_CLK_IN_SELECT_INPUT IOMUXC_KPP_IPP_IND_COL_5_SELECT_INPUT
CSI0_DAT5	IOMUXC_ECSPI1_IPP_IND_MOSI_SELECT_INPUT IOMUXC_KPP_IPP_IND_ROW_5_SELECT_INPUT
CSI0_DAT6	IOMUXC_ECSPI1_IPP_IND_MISO_SELECT_INPUT IOMUXC_KPP_IPP_IND_COL_6_SELECT_INPUT
CSI0_DAT7	IOMUXC_ECSPI1_IPP_IND_SS_B_0_SELECT_INPUT IOMUXC_KPP_IPP_IND_ROW_6_SELECT_INPUT
CSI0_DAT8	IOMUXC_ECSPI2_IPP_CSPI_CLK_IN_SELECT_INPUT IOMUXC_I2C1_IPP_SDA_IN_SELECT_INPUT IOMUXC_KPP_IPP_IND_COL_7_SELECT_INPUT
CSI0_DAT9	IOMUXC_ECSPI2_IPP_IND_MOSI_SELECT_INPUT IOMUXC_I2C1_IPP_SCL_IN_SELECT_INPUT IOMUXC_KPP_IPP_IND_ROW_7_SELECT_INPUT
CSI0_DAT10	IOMUXC_ECSPI2_IPP_IND_MISO_SELECT_INPUT IOMUXC_UART1_IPP_UART_RXD_MUX_SELECT_INPUT
CSI0_DAT11	IOMUXC_ECSPI2_IPP_IND_SS_B_0_SELECT_INPUT IOMUXC_UART1_IPP_UART_RXD_MUX_SELECT_INPUT
CSI0_DAT12	IOMUXC_UART4_IPP_UART_RXD_MUX_SELECT_INPUT
CSI0_DAT13	IOMUXC_UART4_IPP_UART_RXD_MUX_SELECT_INPUT
CSI0_DAT14	IOMUXC_UART5_IPP_UART_RXD_MUX_SELECT_INPUT

Table continues on the next page...

Table 4-4. Daisy Chain Multiplexer Control Registers (by pin) (continued)

Pin Name	Daisy Chain Control Registers
CSI0_DAT15	IOMUXC_UART5_IPP_UART_RXD_MUX_SELECT_INPUT
CSI0_DAT16	IOMUXC_UART4_IPP_UART_RTS_B_SELECT_INPUT
CSI0_DAT17	IOMUXC_UART4_IPP_UART_RTS_B_SELECT_INPUT
CSI0_DAT18	IOMUXC_UART5_IPP_UART_RTS_B_SELECT_INPUT
CSI0_DAT19	IOMUXC_UART5_IPP_UART_RTS_B_SELECT_INPUT
NVCC_CSI__0	
JTAG_TMS	
JTAG_MOD	
JTAG_TRSTB	
JTAG_TDI	
JTAG_TCK	
JTAG_TDO	
EIM_A25	IOMUXC_CSPI_IPP_IND_SS1_B_SELECT_INPUT
EIM_EB2	IOMUXC_CCM_IPP_DI1_CLK_SELECT_INPUT IOMUXC_ECSPI1_IPP_IND_SS_B_0_SELECT_INPUT IOMUXC_I2C2_IPP_SCL_IN_SELECT_INPUT
EIM_D16	IOMUXC_ECSPI1_IPP_CSPI_CLK_IN_SELECT_INPUT IOMUXC_I2C2_IPP_SDA_IN_SELECT_INPUT
EIM_D17	IOMUXC_ECSPI1_IPP_IND_MISO_SELECT_INPUT IOMUXC_I2C3_IPP_SCL_IN_SELECT_INPUT IOMUXC_IPU_IPP_DI_1_IND_DISPB_SD_D_SELECT_INPUT
NVCC_EIM__0	
EIM_D18	IOMUXC_ECSPI1_IPP_IND_MOSI_SELECT_INPUT IOMUXC_I2C3_IPP_SDA_IN_SELECT_INPUT IOMUXC_IPU_IPP_DI_1_IND_DISPB_SD_D_SELECT_INPUT
EIM_D19	IOMUXC_ECSPI1_IPP_IND_SS_B_1_SELECT_INPUT IOMUXC_UART1_IPP_UART_RTS_B_SELECT_INPUT IOMUXC_USBOH3_IPP_IND_UH2_OC_SELECT_INPUT
EIM_D20	IOMUXC_CSPI_IPP_IND_SS0_B_SELECT_INPUT IOMUXC_UART1_IPP_UART_RTS_B_SELECT_INPUT
EIM_D21	IOMUXC_CSPI_IPP_CSPI_CLK_IN_SELECT_INPUT IOMUXC_I2C1_IPP_SCL_IN_SELECT_INPUT IOMUXC_USBOH3_IPP_IND_OTG_OC_SELECT_INPUT
EIM_D22	IOMUXC_CSPI_IPP_IND_MISO_SELECT_INPUT IOMUXC_IPU_IPP_DI_0_IND_DISPB_SD_D_SELECT_INPUT

Table continues on the next page...

Table 4-4. Daisy Chain Multiplexer Control Registers (by pin) (continued)

Pin Name	Daisy Chain Control Registers
EIM_D23	IOMUXC_IPU_IPP_IND_SENS1_DATA_EN_SELECT_INPUT IOMUXC_UART3_IPP_UART_RTS_B_SELECT_INPUT
EIM_EB3	IOMUXC_IPU_IPP_IND_SENS1_HSYNC_SELECT_INPUT IOMUXC_UART3_IPP_UART_RTS_B_SELECT_INPUT
EIM_D24	IOMUXC_AUDMUX_P5_INPUT_RXFS_AMX_SELECT_INPUT IOMUXC_CSPI_IPP_IND_SS2_B_SELECT_INPUT IOMUXC_ECSPI1_IPP_IND_SS_B_2_SELECT_INPUT IOMUXC_UART3_IPP_UART_RXD_MUX_SELECT_INPUT
EIM_D25	IOMUXC_AUDMUX_P5_INPUT_RXCLK_AMX_SELECT_INPUT IOMUXC_CSPI_IPP_IND_SS3_B_SELECT_INPUT IOMUXC_ECSPI1_IPP_IND_SS_B_3_SELECT_INPUT IOMUXC_UART3_IPP_UART_RXD_MUX_SELECT_INPUT
EIM_D26	IOMUXC_FIRI_IPP_IND_RXD_SELECT_INPUT IOMUXC_UART2_IPP_UART_RXD_MUX_SELECT_INPUT
EIM_D27	IOMUXC_UART2_IPP_UART_RXD_MUX_SELECT_INPUT
EIM_D28	IOMUXC_CSPI_IPP_IND_MOSI_SELECT_INPUT IOMUXC_I2C1_IPP_SDA_IN_SELECT_INPUT IOMUXC_IPU_IPP_DI_0_IND_DISPB_SD_D_SELECT_INPUT IOMUXC_UART2_IPP_UART_RTS_B_SELECT_INPUT
EIM_D29	IOMUXC_CSPI_IPP_IND_SS0_B_SELECT_INPUT IOMUXC_IPU_IPP_IND_SENS1_VSYNC_SELECT_INPUT IOMUXC_UART2_IPP_UART_RTS_B_SELECT_INPUT
EIM_D30	IOMUXC_UART3_IPP_UART_RTS_B_SELECT_INPUT IOMUXC_USBOH3_IPP_IND_UH1_OC_SELECT_INPUT IOMUXC_USBOH3_IPP_IND_UH2_OC_SELECT_INPUT
EIM_D31	IOMUXC_UART3_IPP_UART_RTS_B_SELECT_INPUT
NVCC_EIM__1	
EIM_A24	
EIM_A23	
EIM_A22	
EIM_A21	
EIM_A20	
EIM_A19	
EIM_A18	
EIM_A17	
EIM_A16	

Table continues on the next page...

Table 4-4. Daisy Chain Multiplexer Control Registers (by pin) (continued)

Pin Name	Daisy Chain Control Registers
EIM_CS0	IOMUXC_ECSPI2_IPP_CSPI_CLK_IN_SELECT_INPUT
EIM_CS1	IOMUXC_ECSPI2_IPP_IND_MOSI_SELECT_INPUT
EIM_OE	IOMUXC_ECSPI2_IPP_IND_MISO_SELECT_INPUT
EIM_RW	IOMUXC_ECSPI2_IPP_IND_SS_B_0_SELECT_INPUT
EIM_LBA	IOMUXC_ECSPI2_IPP_IND_SS_B_1_SELECT_INPUT
NVCC_EIM__4	
EIM_EB0	IOMUXC_GPC_PMIC_RDY_SELECT_INPUT
EIM_EB1	
EIM_DA0	
EIM_DA1	
EIM_DA2	
EIM_DA3	
EIM_DA4	
EIM_DA5	
EIM_DA6	
EIM_DA7	
EIM_DA8	
EIM_DA9	
EIM_DA10	IOMUXC_IPU_IPP_IND_SENS1_DATA_EN_SELECT_INPUT
EIM_DA11	IOMUXC_IPU_IPP_IND_SENS1_HSYNC_SELECT_INPUT
EIM_DA12	IOMUXC_IPU_IPP_IND_SENS1_VSYNC_SELECT_INPUT
EIM_DA13	IOMUXC_CCM_IPP_DI1_CLK_SELECT_INPUT
EIM_DA14	
EIM_DA15	
NANDF_WE_B	
NANDF_RE_B	
EIM_WAIT	
EIM_BCLK	
NVCC_EIM__7	
LVDS1_TX3_P	
LVDS1_TX3_N	
LVDS1_TX2_P	
LVDS1_TX2_N	
LVDS1_CLK_P	
LVDS1_CLK_N	

Table continues on the next page...

Table 4-4. Daisy Chain Multiplexer Control Registers (by pin) (continued)

Pin Name	Daisy Chain Control Registers
LVDS1_TX1_P	
LVDS1_TX1_N	
LVDS1_TX0_P	
LVDS1_TX0_N	
LVDS0_TX3_P	
LVDS0_TX3_N	
LVDS0_CLK_P	
LVDS0_CLK_N	
LVDS0_TX2_P	
LVDS0_TX2_N	
LVDS0_TX1_P	
LVDS0_TX1_N	
LVDS0_TX0_P	
LVDS0_TX0_N	
GPIO_10	
GPIO_11	
GPIO_12	
GPIO_13	
GPIO_14	
DRAM_D24	
DRAM_D30	
DRAM_D26	
DRAM_DQM3	
DRAM_D28	
DRAM_D25	
DRAM_SDQS3_B	
DRAM_SDQS3	
DRAM_D27	
DRAM_D31	
DRAM_D16	
DRAM_D29	
DRAM_D18	
DRAM_SDCKE1	
DRAM_D22	
DRAM_DQM2	

Table continues on the next page...

Table 4-4. Daisy Chain Multiplexer Control Registers (by pin) (continued)

Pin Name	Daisy Chain Control Registers
DRAM_D20	
DRAM_SDBA0	
DRAM_D17	
DRAM_SDODT1	
DRAM_D19	
DRAM_SDQS2_B	
DRAM_SDQS2	
DRAM_D21	
DRAM_CS1	
DRAM_D23	
DRAM_RESET	
DRAM_SDBA1	
DRAM_SDCLK_1_B	
DRAM_SDCLK_1	
DRAM_A8	
DRAM_SDBA2	
DRAM_A14	
DRAM_A3	
DRAM_A5	
DRAM_A7	
DRAM_A6	
DRAM_A9	
DRAM_A2	
DRAM_A0	
DRAM_A15	
DRAM_A13	
DRAM_A11	
DRAM_A1	
DRAM_A12	
DRAM_CAS	
DRAM_SDWE	
DRAM_CS0	
DRAM_A4	
DRAM_SDCLK_0_B	

Table continues on the next page...

Table 4-4. Daisy Chain Multiplexer Control Registers (by pin) (continued)

Pin Name	Daisy Chain Control Registers
DRAM_SDCLK_0	
DRAM_A10	
DRAM_D4	
DRAM_D6	
DRAM_D2	
DRAM_SDQS0_B	
DRAM_SDQS0	
DRAM_SDODT0	
DRAM_DQM0	
DRAM_RAS	
DRAM_D5	
DRAM_D0	
DRAM_D7	
DRAM_SDCKE0	
DRAM_D1	
DRAM_D14	
DRAM_D3	
DRAM_D12	
DRAM_D10	
DRAM_D8	
DRAM_D13	
DRAM_SDQS1_B	
DRAM_SDQS1	
DRAM_DQM1	
DRAM_D9	
DRAM_D15	
DRAM_D11	
CKIH1	
CKIH2	
PMIC_ON_REQ	
PMIC_STBY_REQ	
NANDF_CLE	
NANDF_ALE	
NANDF_WP_B	
NANDF_RB0	

Table continues on the next page...

Table 4-4. Daisy Chain Multiplexer Control Registers (by pin) (continued)

Pin Name	Daisy Chain Control Registers
NANDF_CS0	
NANDF_CS1	IOMUXC_MLB_MLBCLK_IN_SELECT_INPUT
NANDF_CS2	IOMUXC_ESAI1_IPP_IND_SDO0_SELECT_INPUT IOMUXC_MLB_MLBSIG_IN_SELECT_INPUT
NANDF_CS3	IOMUXC_ESAI1_IPP_IND_SDO1_SELECT_INPUT IOMUXC_MLB_MLBDAT_IN_SELECT_INPUT
NVCC_NANDF	
FEC_MDIO	IOMUXC_ESAI1_IPP_IND_SCKR_SELECT_INPUT IOMUXC_FEC_FEC_COL_SELECT_INPUT IOMUXC_FEC_FEC_MDI_SELECT_INPUT
FEC_REF_CLK	IOMUXC_ESAI1_IPP_IND_FSR_SELECT_INPUT
FEC_RX_ER	IOMUXC_ESAI1_IPP_IND_HCKR_SELECT_INPUT IOMUXC_FEC_FEC_RX_CLK_SELECT_INPUT
FEC_CRS_DV	IOMUXC_ESAI1_IPP_IND_SCKT_SELECT_INPUT
FEC_RXD1	IOMUXC_ESAI1_IPP_IND_FST_SELECT_INPUT IOMUXC_MLB_MLBSIG_IN_SELECT_INPUT
FEC_RXD0	IOMUXC_ESAI1_IPP_IND_HCKT_SELECT_INPUT
FEC_TX_EN	IOMUXC_ESAI1_IPP_IND_SDO3_SDI2_SELECT_INPUT
FEC_TXD1	IOMUXC_ESAI1_IPP_IND_SDO2_SDI3_SELECT_INPUT IOMUXC_MLB_MLBCLK_IN_SELECT_INPUT
FEC_TXD0	IOMUXC_ESAI1_IPP_IND_SDO4_SDI1_SELECT_INPUT
FEC_MDC	IOMUXC_ESAI1_IPP_IND_SDO5_SDI0_SELECT_INPUT IOMUXC_MLB_MLBDAT_IN_SELECT_INPUT
NVCC_FEC	
PATA_DIOW	IOMUXC_UART1_IPP_UART_RXD_MUX_SELECT_INPUT
PATA_DMACK	IOMUXC_UART1_IPP_UART_RXD_MUX_SELECT_INPUT
PATA_DMARQ	IOMUXC_UART2_IPP_UART_RXD_MUX_SELECT_INPUT
PATA_BUFFER_EN	IOMUXC_UART2_IPP_UART_RXD_MUX_SELECT_INPUT
PATA_INTRQ	IOMUXC_UART2_IPP_UART_RTS_B_SELECT_INPUT
PATA_DIOR	IOMUXC_CAN1_IPP_IND_CANRX_SELECT_INPUT IOMUXC_UART2_IPP_UART_RTS_B_SELECT_INPUT
PATA_RESET_B	IOMUXC_UART1_IPP_UART_RTS_B_SELECT_INPUT
PATA_IORDY	IOMUXC_CAN2_IPP_IND_CANRX_SELECT_INPUT IOMUXC_UART1_IPP_UART_RTS_B_SELECT_INPUT
PATA_DA_0	IOMUXC_OWIRE_BATTERY_LINE_IN_SELECT_INPUT

Table continues on the next page...

Table 4-4. Daisy Chain Multiplexer Control Registers (by pin) (continued)

Pin Name	Daisy Chain Control Registers
PATA_DA_1	IOMUXC_UART3_IPP_UART_RTS_B_SELECT_INPUT
PATA_DA_2	IOMUXC_UART3_IPP_UART_RTS_B_SELECT_INPUT
PATA_CS_0	IOMUXC_UART3_IPP_UART_RXD_MUX_SELECT_INPUT
PATA_CS_1	IOMUXC_UART3_IPP_UART_RXD_MUX_SELECT_INPUT
NVCC_PATA__2	
PATA_DATA0	
PATA_DATA1	
PATA_DATA2	
PATA_DATA3	
PATA_DATA4	
PATA_DATA5	
PATA_DATA6	
PATA_DATA7	
PATA_DATA8	
PATA_DATA9	
PATA_DATA10	
PATA_DATA11	
PATA_DATA12	
PATA_DATA13	
PATA_DATA14	
PATA_DATA15	
NVCC_PATA__0	
SD1_DATA0	IOMUXC_CCM_PLL3_BYPASS_CLK_SELECT_INPUT IOMUXC_CSPI_IPP_IND_MISO_SELECT_INPUT
SD1_DATA1	IOMUXC_CCM_PLL4_BYPASS_CLK_SELECT_INPUT IOMUXC_CSPI_IPP_IND_SS0_B_SELECT_INPUT
SD1_CMD	IOMUXC_CCM_PLL1_BYPASS_CLK_SELECT_INPUT IOMUXC_CSPI_IPP_IND_MOSI_SELECT_INPUT
SD1_DATA2	IOMUXC_CCM_PLL2_BYPASS_CLK_SELECT_INPUT IOMUXC_CSPI_IPP_IND_SS1_B_SELECT_INPUT
SD1_CLK	IOMUXC_CSPI_IPP_CSPI_CLK_IN_SELECT_INPUT
SD1_DATA3	IOMUXC_CSPI_IPP_IND_SS2_B_SELECT_INPUT
NVCC_SD1	

Table continues on the next page...

Table 4-4. Daisy Chain Multiplexer Control Registers (by pin) (continued)

Pin Name	Daisy Chain Control Registers
SD2_CLK	IOMUXC_AUDMUX_P4_INPUT_RXFS_AMX_SELECT_INPUT IOMUXC_CSPI_IPP_CSPI_CLK_IN_SELECT_INPUT IOMUXC_KPP_IPP_IND_COL_5_SELECT_INPUT
SD2_CMD	IOMUXC_AUDMUX_P4_INPUT_RXCLK_AMX_SELECT_INPUT IOMUXC_CSPI_IPP_IND_MOSI_SELECT_INPUT IOMUXC_KPP_IPP_IND_ROW_5_SELECT_INPUT
SD2_DATA3	IOMUXC_AUDMUX_P4_INPUT_TXCLK_AMX_SELECT_INPUT IOMUXC_CSPI_IPP_IND_SS2_B_SELECT_INPUT IOMUXC_KPP_IPP_IND_COL_6_SELECT_INPUT
SD2_DATA2	IOMUXC_AUDMUX_P4_INPUT_DB_AMX_SELECT_INPUT IOMUXC_CSPI_IPP_IND_SS1_B_SELECT_INPUT IOMUXC_KPP_IPP_IND_ROW_6_SELECT_INPUT
SD2_DATA1	IOMUXC_AUDMUX_P4_INPUT_TXFS_AMX_SELECT_INPUT IOMUXC_CSPI_IPP_IND_SS0_B_SELECT_INPUT IOMUXC_KPP_IPP_IND_COL_7_SELECT_INPUT
SD2_DATA0	IOMUXC_AUDMUX_P4_INPUT_DA_AMX_SELECT_INPUT IOMUXC_CSPI_IPP_IND_MISO_SELECT_INPUT IOMUXC_KPP_IPP_IND_ROW_7_SELECT_INPUT
NVCC_SD2	
GPIO_0	IOMUXC_KPP_IPP_IND_COL_5_SELECT_INPUT
GPIO_1	IOMUXC_ESAI1_IPP_IND_SCKR_SELECT_INPUT IOMUXC_KPP_IPP_IND_ROW_5_SELECT_INPUT
GPIO_9	IOMUXC_ESAI1_IPP_IND_FSR_SELECT_INPUT IOMUXC_ESDHC1_IPP_WP_ON_SELECT_INPUT IOMUXC_KPP_IPP_IND_COL_6_SELECT_INPUT
GPIO_3	IOMUXC_ESAI1_IPP_IND_HCKR_SELECT_INPUT IOMUXC_I2C3_IPP_SCL_IN_SELECT_INPUT IOMUXC_MLB_MLBCLK_IN_SELECT_INPUT IOMUXC_USBOH3_IPP_IND_UH1_OC_SELECT_INPUT
GPIO_6	IOMUXC_ESAI1_IPP_IND_SCKT_SELECT_INPUT IOMUXC_I2C3_IPP_SDA_IN_SELECT_INPUT IOMUXC_MLB_MLBSIG_IN_SELECT_INPUT
GPIO_2	IOMUXC_ESAI1_IPP_IND_FST_SELECT_INPUT IOMUXC_KPP_IPP_IND_ROW_6_SELECT_INPUT IOMUXC_MLB_MLBDAT_IN_SELECT_INPUT

Table continues on the next page...

Table 4-4. Daisy Chain Multiplexer Control Registers (by pin) (continued)

Pin Name	Daisy Chain Control Registers
GPIO_4	IOMUXC_ESAI1_IPP_IND_HCKT_SELECT_INPUT IOMUXC_KPP_IPP_IND_COL_7_SELECT_INPUT
GPIO_5	IOMUXC_CCM_PLL1_BYPASS_CLK_SELECT_INPUT IOMUXC_ESAI1_IPP_IND_SDO2_SDI3_SELECT_INPUT IOMUXC_I2C3_IPP_SCL_IN_SELECT_INPUT IOMUXC_KPP_IPP_IND_ROW_7_SELECT_INPUT
GPIO_7	IOMUXC_CCM_PLL2_BYPASS_CLK_SELECT_INPUT IOMUXC_ESAI1_IPP_IND_SDO4_SDI1_SELECT_INPUT IOMUXC_FIRI_IPP_IND_RXD_SELECT_INPUT IOMUXC_UART2_IPP_UART_RXD_MUX_SELECT_INPUT
GPIO_8	IOMUXC_CAN1_IPP_IND_CANRX_SELECT_INPUT IOMUXC_CCM_PLL3_BYPASS_CLK_SELECT_INPUT IOMUXC_ESAI1_IPP_IND_SDO5_SDI0_SELECT_INPUT IOMUXC_UART2_IPP_UART_RXD_MUX_SELECT_INPUT
GPIO_16	IOMUXC_ESAI1_IPP_IND_SDO3_SDI2_SELECT_INPUT IOMUXC_I2C3_IPP_SDA_IN_SELECT_INPUT IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT
GPIO_17	IOMUXC_ESAI1_IPP_IND_SDO0_SELECT_INPUT IOMUXC_GPC_PMIC_RDY_SELECT_INPUT IOMUXC_SDMA_EVENTS_14_SELECT_INPUT
GPIO_18	IOMUXC_CCM_IPP_ASRC_EXT_SELECT_INPUT IOMUXC_ESAI1_IPP_IND_SDO1_SELECT_INPUT IOMUXC_OWIRE_BATTERY_LINE_IN_SELECT_INPUT IOMUXC_SDMA_EVENTS_15_SELECT_INPUT
NVCC_GPIO	
POR_B	
BOOT_MODE1	
RESET_IN_B	
BOOT_MODE0	
TEST_MODE	

Table 4-5 shows the daisy chain control registers sorted by block instance and block I/O.

Table 4-5. Daisy Chain Multiplexer Control Registers (by block instance / block I/O)

Block Instance	Block I/O	Select Register	DAISY	Package Pin	Mode
AUDMUX	AUD4_RXD	IOMUXC_AUDMUX_P4_INPUT_DA_AMX_SELECT_INPUT	0	DISP0_DAT23	ALT3
			1	SD2_DATA0	ALT3
	AUD4_TXD	IOMUXC_AUDMUX_P4_INPUT_DB_AMX_SELECT_INPUT	0	DISP0_DAT21	ALT3
			1	SD2_DATA2	ALT3
	AUD4_RXC	IOMUXC_AUDMUX_P4_INPUT_RXCLK_AMX_SELECT_INPUT	0	DISP0_DAT19	ALT4
			1	SD2_CMD	ALT3
	AUD4_RXFS	IOMUXC_AUDMUX_P4_INPUT_RXFS_AMX_SELECT_INPUT	0	DISP0_DAT18	ALT4
			1	SD2_CLK	ALT3
AUDMUX	AUD4_TXC	IOMUXC_AUDMUX_P4_INPUT_TXCLK_AMX_SELECT_INPUT	0	DISP0_DAT20	ALT3
			1	SD2_DATA3	ALT3
	AUD4_TXFS	IOMUXC_AUDMUX_P4_INPUT_TXFS_AMX_SELECT_INPUT	0	DISP0_DAT22	ALT3
			1	SD2_DATA1	ALT3
	AUD5_RXD	IOMUXC_AUDMUX_P5_INPUT_DA_AMX_SELECT_INPUT	0	KEY_ROW1	ALT2
			1	DISP0_DAT19	ALT3
	AUD5_TXD	IOMUXC_AUDMUX_P5_INPUT_DB_AMX_SELECT_INPUT	0	KEY_ROW0	ALT2
			1	DISP0_DAT17	ALT3

Table continues on the next page...

Table 4-5. Daisy Chain Multiplexer Control Registers (by block instance / block I/O) (continued)

Block Instance	Block I/O	Select Register	DAISY	Package Pin	Mode
AUDMUX	AUD5_RXC	IOMUXC_AUDMUX_P5_INPUT_RXCLK_AMX_SELECT_INPUT	0	DISP0_DAT14	ALT3
			1	EIM_D25	ALT5
	AUD5_RXFS	IOMUXC_AUDMUX_P5_INPUT_RXFS_AMX_SELECT_INPUT	0	DISP0_DAT13	ALT3
			1	EIM_D24	ALT5
	AUD5_TXC	IOMUXC_AUDMUX_P5_INPUT_TXCLK_AMX_SELECT_INPUT	0	KEY_COL0	ALT2
			1	DISP0_DAT16	ALT3
	AUD5_TXFS	IOMUXC_AUDMUX_P5_INPUT_TXFS_AMX_SELECT_INPUT	0	KEY_COL1	ALT2
			1	DISP0_DAT18	ALT3
CAN-1	RXCAN	IOMUXC_CAN1_IPP_IND_CANRX_SELECT_INPUT	0b00	KEY_ROW2	ALT2
			0b01	PATA_DIOR	ALT4
			0b10	GPIO_8	ALT3
CAN-2	RXCAN	IOMUXC_CAN2_IPP_IND_CANRX_SELECT_INPUT	0	KEY_ROW4	ALT2
			1	PATA_IORDY	ALT4

Table continues on the next page...

Table 4-5. Daisy Chain Multiplexer Control Registers (by block instance / block I/O) (continued)

Block Instance	Block I/O	Select Register	DAISY	Package Pin	Mode
CCM	ASRC_EXT_CLK	IOMUXC_CCM_IPP_ASRC_EXT_ SELECT_INPUT	0	KEY_ ROW3	ALT3
			1	GPIO_18	ALT5
	DI1_EXT_CLK	IOMUXC_CCM_IPP_DI1_CLK_SELECT_ INPUT	0	EIM_EB2	ALT2
			1	EIM_ DA13	ALT4
	PLL1_BYP	IOMUXC_CCM_PLL1_BYPASS_CLK_ SELECT_INPUT	0	SD1_ CMD	ALT7
			1	GPIO_5	ALT7
	PLL2_BYP	IOMUXC_CCM_PLL2_BYPASS_CLK_ SELECT_INPUT	0	SD1_ DATA2	ALT7
			1	GPIO_7	ALT7
	PLL3_BYP	IOMUXC_CCM_PLL3_BYPASS_CLK_ SELECT_INPUT	0	SD1_ DATA0	ALT7
			1	GPIO_8	ALT7
	PLL4_BYP	IOMUXC_CCM_PLL4_BYPASS_CLK_ SELECT_INPUT	0	KEY_ ROW3	ALT6
			1	SD1_ DATA1	ALT7

Table continues on the next page...

Table 4-5. Daisy Chain Multiplexer Control Registers (by block instance / block I/O) (continued)

Block Instance	Block I/O	Select Register	DAISY	Package Pin	Mode
CSPI	SCLK	IOMUXC_CSPI_IPP_CSPI_CLK_IN_SELECT_INPUT	0b00	DISP0_DAT0	ALT2
			0b01	EIM_D21	ALT4
			0b10	SD1_CLK	ALT5
			0b11	SD2_CLK	ALT5
	MISO	IOMUXC_CSPI_IPP_IND_MISO_SELECT_INPUT	0b00	DISP0_DAT2	ALT2
			0b01	EIM_D22	ALT4
			0b10	SD1_DATA0	ALT5
			0b11	SD2_DATA0	ALT5
	MOSI	IOMUXC_CSPI_IPP_IND_MOSI_SELECT_INPUT	0b00	DISP0_DAT1	ALT2
			0b01	EIM_D28	ALT4
			0b10	SD1_CMD	ALT5
			0b11	SD2_CMD	ALT5
CSPI	SS0	IOMUXC_CSPI_IPP_IND_SS0_B_SELECT_INPUT	0b000	DISP0_DAT3	ALT2
			0b001	EIM_D20	ALT4
			0b010	EIM_D29	ALT4
			0b011	SD1_DATA1	ALT5
			0b100	SD2_DATA1	ALT5

Table continues on the next page...

Table 4-5. Daisy Chain Multiplexer Control Registers (by block instance / block I/O) (continued)

Block Instance	Block I/O	Select Register	DAISY	Package Pin	Mode
CSPI	SS1	IOMUXC_CSPI_IPP_IND_SS1_B_SELECT_INPUT	0b00	DISP0_DAT4	ALT2
			0b01	EIM_A25	ALT4
			0b10	SD1_DATA2	ALT5
			0b11	SD2_DATA2	ALT5
	SS2	IOMUXC_CSPI_IPP_IND_SS2_B_SELECT_INPUT	0b00	DISP0_DAT5	ALT2
			0b01	EIM_D24	ALT4
			0b10	SD1_DATA3	ALT5
			0b11	SD2_DATA3	ALT5
	SS3	IOMUXC_CSPI_IPP_IND_SS3_B_SELECT_INPUT	0	DISP0_DAT6	ALT2
			1	EIM_D25	ALT4
ECSPI-1	SCLK	IOMUXC_ECSPI1_IPP_CSPI_CLK_IN_SELECT_INPUT	0b00	KEY_COL0	ALT5
			0b01	DISP0_DAT20	ALT2
			0b10	CSI0_DAT4	ALT3
			0b11	EIM_D16	ALT4
	MISO	IOMUXC_ECSPI1_IPP_IND_MISO_SELECT_INPUT	0b00	KEY_COL1	ALT5
			0b01	DISP0_DAT22	ALT2
			0b10	CSI0_DAT6	ALT3
			0b11	EIM_D17	ALT4
	MOSI	IOMUXC_ECSPI1_IPP_IND_MOSI_SELECT_INPUT	0b00	KEY_ROW0	ALT5
			0b01	DISP0_DAT21	ALT2
			0b10	CSI0_DAT5	ALT3
			0b11	EIM_D18	ALT4

Table continues on the next page...

Table 4-5. Daisy Chain Multiplexer Control Registers (by block instance / block I/O) (continued)

Block Instance	Block I/O	Select Register	DAISY	Package Pin	Mode
ECSPI-1	SS0	IOMUXC_ECSPI1_IPP_IND_SS_B_0_SELECT_INPUT	0b00	KEY_ROW1	ALT5
			0b01	DISP0_DAT23	ALT2
			0b10	CSI0_DAT7	ALT3
			0b11	EIM_EB2	ALT4
	SS1	IOMUXC_ECSPI1_IPP_IND_SS_B_1_SELECT_INPUT	0b00	KEY_COL2	ALT5
			0b01	DISP0_DAT15	ALT2
			0b10	EIM_D19	ALT4
ECSPI-1	SS2	IOMUXC_ECSPI1_IPP_IND_SS_B_2_SELECT_INPUT	0	KEY_ROW2	ALT5
			1	EIM_D24	ALT3
	SS3	IOMUXC_ECSPI1_IPP_IND_SS_B_3_SELECT_INPUT	0	KEY_COL3	ALT5
			1	EIM_D25	ALT3

Table continues on the next page...

Table 4-5. Daisy Chain Multiplexer Control Registers (by block instance / block I/O) (continued)

Block Instance	Block I/O	Select Register	DAISY	Package Pin	Mode
ECSPI-2	SCLK	IOMUXC_ECSPI2_IPP_CSPI_CLK_IN_SELECT_INPUT	0b00	DISP0_DAT19	ALT2
			0b01	CSI0_DAT8	ALT3
			0b10	EIM_CS0	ALT2
	MISO	IOMUXC_ECSPI2_IPP_IND_MISO_SELECT_INPUT	0b00	DISP0_DAT17	ALT2
			0b01	CSI0_DAT10	ALT3
			0b10	EIM_OE	ALT2
	MOSI	IOMUXC_ECSPI2_IPP_IND_MOSI_SELECT_INPUT	0b00	DISP0_DAT16	ALT2
			0b01	CSI0_DAT9	ALT3
			0b10	EIM_CS1	ALT2
	SS0	IOMUXC_ECSPI2_IPP_IND_SS_B_0_SELECT_INPUT	0b00	DISP0_DAT18	ALT2
			0b01	CSI0_DAT11	ALT3
			0b10	EIM_RW	ALT2
	SS1	IOMUXC_ECSPI2_IPP_IND_SS_B_1_SELECT_INPUT	0	DISP0_DAT15	ALT3
			1	EIM_LBA	ALT2

Table continues on the next page...

Table 4-5. Daisy Chain Multiplexer Control Registers (by block instance / block I/O) (continued)

Block Instance	Block I/O	Select Register	DAISY	Package Pin	Mode
ESAI-1	FSR	IOMUXC_ESAI1_IPP_IND_FSR_SELECT_INPUT	0	FEC_REF_CLK	ALT2
			1	GPIO_9	ALT0
	FST	IOMUXC_ESAI1_IPP_IND_FST_SELECT_INPUT	0	FEC_RXD1	ALT2
			1	GPIO_2	ALT0
	HCKR	IOMUXC_ESAI1_IPP_IND_HCKR_SELECT_INPUT	0	FEC_RX_ER	ALT2
			1	GPIO_3	ALT0
	HCKT	IOMUXC_ESAI1_IPP_IND_HCKT_SELECT_INPUT	0	FEC_RXD0	ALT2
			1	GPIO_4	ALT0
	SCKR	IOMUXC_ESAI1_IPP_IND_SCKR_SELECT_INPUT	0	FEC_MDIO	ALT2
			1	GPIO_1	ALT0
	SCKT	IOMUXC_ESAI1_IPP_IND_SCKT_SELECT_INPUT	0	FEC_CRS_DV	ALT2
			1	GPIO_6	ALT0
	TX0	IOMUXC_ESAI1_IPP_IND_SDO0_SELECT_INPUT	0	NANDF_CS2	ALT3
			1	GPIO_17	ALT0
	TX1	IOMUXC_ESAI1_IPP_IND_SDO1_SELECT_INPUT	0	NANDF_CS3	ALT3
			1	GPIO_18	ALT0
	TX2_RX3	IOMUXC_ESAI1_IPP_IND_SDO2_SDI3_SELECT_INPUT	0	FEC_TXD1	ALT2
			1	GPIO_5	ALT0
	TX3_RX2	IOMUXC_ESAI1_IPP_IND_SDO3_SDI2_SELECT_INPUT	0	FEC_TX_EN	ALT2
			1	GPIO_16	ALT0
	TX4_RX1	IOMUXC_ESAI1_IPP_IND_SDO4_SDI1_SELECT_INPUT	0	FEC_TXD0	ALT2
			1	GPIO_7	ALT0
	TX5_RX0	IOMUXC_ESAI1_IPP_IND_SDO5_SDI0_SELECT_INPUT	0	FEC_MDC	ALT2
			1	GPIO_8	ALT0

Table continues on the next page...

Table 4-5. Daisy Chain Multiplexer Control Registers (by block instance / block I/O) (continued)

Block Instance	Block I/O	Select Register	DAISY	Package Pin	Mode
ESDHCV2-1	WP	IOMUXC_ESDHC1_IPP_WP_ON_SELECT_INPUT	0	DI0_PIN4	ALT3
			1	GPIO_9	ALT6
FEC	COL	IOMUXC_FEC_FEC_COL_SELECT_INPUT	0	KEY_ROW1	ALT6
			1	FEC_MDIO	ALT3
	MDIO	IOMUXC_FEC_FEC_MDI_SELECT_INPUT	0	KEY_COL2	ALT4
			1	FEC_MDIO	ALT0
	RX_CLK	IOMUXC_FEC_FEC_RX_CLK_SELECT_INPUT	0	KEY_COL1	ALT6
			1	FEC_RX_ER	ALT3
FIRI	RXD	IOMUXC_FIRI_IPP_IND_RXD_SELECT_INPUT	0	EIM_D26	ALT3
			1	GPIO_7	ALT5
GPC	PMIC_RDY	IOMUXC_GPC_PMIC_RDY_SELECT_INPUT	0	EIM_EB0	ALT5
			1	GPIO_17	ALT3
I2C-1	SCL	IOMUXC_I2C1_IPP_SCL_IN_SELECT_INPUT	0	CSI0_DAT9	ALT5
			1	EIM_D21	ALT5
	SDA	IOMUXC_I2C1_IPP_SDA_IN_SELECT_INPUT	0	CSI0_DAT8	ALT5
			1	EIM_D28	ALT5
I2C-2	SCL	IOMUXC_I2C2_IPP_SCL_IN_SELECT_INPUT	0	KEY_COL3	ALT4
			1	EIM_EB2	ALT5
	SDA	IOMUXC_I2C2_IPP_SDA_IN_SELECT_INPUT	0	KEY_ROW3	ALT4
			1	EIM_D16	ALT5
I2C-3	SCL	IOMUXC_I2C3_IPP_SCL_IN_SELECT_INPUT	0b00	EIM_D17	ALT5
			0b01	GPIO_3	ALT2
			0b10	GPIO_5	ALT6
	SDA	IOMUXC_I2C3_IPP_SDA_IN_SELECT_INPUT	0b00	EIM_D18	ALT5
			0b01	GPIO_6	ALT2
			0b10	GPIO_16	ALT6

Table continues on the next page...

Table 4-5. Daisy Chain Multiplexer Control Registers (by block instance / block I/O) (continued)

Block Instance	Block I/O	Select Register	DAISY	Package Pin	Mode
IPU	DISPB0_SER_DIN	IOMUXC_IPU_IPP_DI_0_IND_DISPB_SD_D_SELECT_INPUT	0	EIM_D22	ALT3
			1	EIM_D28	ALT3
	DISPB1_SER_DIN	IOMUXC_IPU_IPP_DI_1_IND_DISPB_SD_D_SELECT_INPUT	0	EIM_D17	ALT3
			1	EIM_D18	ALT3
	CSI1_DATA_EN	IOMUXC_IPU_IPP_IND_SENS1_DATA_EN_SELECT_INPUT	0	EIM_D23	ALT6
			1	EIM_DA10	ALT4
	CSI1_HSYNC	IOMUXC_IPU_IPP_IND_SENS1_HSYNC_SELECT_INPUT	0	EIM_EB3	ALT6
			1	EIM_DA11	ALT4
	CSI1_VSYNC	IOMUXC_IPU_IPP_IND_SENS1_VSYNC_SELECT_INPUT	0	EIM_D29	ALT6
			1	EIM_DA12	ALT4

Table continues on the next page...

Table 4-5. Daisy Chain Multiplexer Control Registers (by block instance / block I/O) (continued)

Block Instance	Block I/O	Select Register	DAISY	Package Pin	Mode
KPP	COL[5]	IOMUXC_KPP_IPP_IND_COL_5_ SELECT_INPUT	0b00	GPIO_19	ALT0
			0b01	CSI0_ DAT4	ALT2
			0b10	SD2_ CLK	ALT2
			0b11	GPIO_0	ALT2
	COL[6]	IOMUXC_KPP_IPP_IND_COL_6_ SELECT_INPUT	0b00	CSI0_ DAT6	ALT2
			0b01	SD2_ DATA3	ALT2
			0b10	GPIO_9	ALT2
	COL[7]	IOMUXC_KPP_IPP_IND_COL_7_ SELECT_INPUT	0b00	CSI0_ DAT8	ALT2
			0b01	SD2_ DATA1	ALT2
			0b10	GPIO_4	ALT2
	ROW[5]	IOMUXC_KPP_IPP_IND_ROW_5_ SELECT_INPUT	0b00	CSI0_ DAT5	ALT2
			0b01	SD2_ CMD	ALT2
			0b10	GPIO_1	ALT2
	ROW[6]	IOMUXC_KPP_IPP_IND_ROW_6_ SELECT_INPUT	0b00	CSI0_ DAT7	ALT2
			0b01	SD2_ DATA2	ALT2
			0b10	GPIO_2	ALT2
	ROW[7]	IOMUXC_KPP_IPP_IND_ROW_7_ SELECT_INPUT	0b00	CSI0_ DAT9	ALT2
			0b01	SD2_ DATA0	ALT2
			0b10	GPIO_5	ALT2

Table continues on the next page...

Table 4-5. Daisy Chain Multiplexer Control Registers (by block instance / block I/O) (continued)

Block Instance	Block I/O	Select Register	DAISY	Package Pin	Mode
MLB	MLBCLK	IOMUXC_MLB_MLBCLK_IN_SELECT_INPUT	0b00	NANDF_CS1	ALT6
			0b01	FEC_TXD1	ALT3
			0b10	GPIO_3	ALT7
	MLBDAT	IOMUXC_MLB_MLBDAT_IN_SELECT_INPUT	0b00	NANDF_CS3	ALT6
			0b01	FEC_MDC	ALT3
			0b10	GPIO_2	ALT7
	MLBSIG	IOMUXC_MLB_MLBSIG_IN_SELECT_INPUT	0b00	NANDF_CS2	ALT6
			0b01	FEC_RXD1	ALT3
			0b10	GPIO_6	ALT7
OWIRE	LINE	IOMUXC_OWIRE_BATTERY_LINE_IN_SELECT_INPUT	0	PATA_DA_0	ALT4
			1	GPIO_18	ALT3
SDMA	SDMA_EXT_EVENT[0]	IOMUXC_SDMA_EVENTS_14_SELECT_INPUT	0	DISP0_DAT16	ALT4
			1	GPIO_17	ALT2
	SDMA_EXT_EVENT[1]	IOMUXC_SDMA_EVENTS_15_SELECT_INPUT	0	DISP0_DAT17	ALT4
			1	GPIO_18	ALT2
SPDIF	IN1	IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT	0	KEY_COL3	ALT3
			1	GPIO_16	ALT5

Table continues on the next page...

Table 4-5. Daisy Chain Multiplexer Control Registers (by block instance / block I/O) (continued)

Block Instance	Block I/O	Select Register	DAISY	Package Pin	Mode
UART-1	RTS	IOMUXC_UART1_IPP_UART_RTS_B_SELECT_INPUT	0b00	EIM_D19	ALT6
			0b01	EIM_D20	ALT6
			0b10	PATA_RESET_B	ALT3
			0b11	PATA_IORDY	ALT3
	RXD_MUX	IOMUXC_UART1_IPP_UART_RXD_MUX_SELECT_INPUT	0b00	CSI0_DAT10	ALT2
			0b01	CSI0_DAT11	ALT2
			0b10	PATA_DIOW	ALT3
			0b11	PATA_DMACK	ALT3
UART-2	RTS	IOMUXC_UART2_IPP_UART_RTS_B_SELECT_INPUT	0b00	EIM_D28	ALT2
			0b01	EIM_D29	ALT2
			0b10	PATA_INTRQ	ALT3
			0b11	PATA_DIOR	ALT3
	RXD_MUX	IOMUXC_UART2_IPP_UART_RXD_MUX_SELECT_INPUT	0b000	EIM_D26	ALT2
			0b001	EIM_D27	ALT2
			0b010	PATA_DMARQ	ALT3
			0b011	PATA_BUFFER_EN	ALT3
			0b100	GPIO_7	ALT4
			0b101	GPIO_8	ALT4

Table continues on the next page...

Table 4-5. Daisy Chain Multiplexer Control Registers (by block instance / block I/O) (continued)

Block Instance	Block I/O	Select Register	DAISY	Package Pin	Mode
UART-3	RTS	IOMUXC_UART3_IPP_UART_RTS_B_SELECT_INPUT	0b000	EIM_D23	ALT2
			0b001	EIM_EB3	ALT2
			0b010	EIM_D30	ALT2
			0b011	EIM_D31	ALT2
			0b100	PATA_DA_1	ALT4
			0b101	PATA_DA_2	ALT4
	RXD_MUX	IOMUXC_UART3_IPP_UART_RXD_MUX_SELECT_INPUT	0b00	EIM_D24	ALT2
			0b01	EIM_D25	ALT2
			0b10	PATA_CS_0	ALT4
			0b11	PATA_CS_1	ALT4
UART-4	RTS	IOMUXC_UART4_IPP_UART_RTS_B_SELECT_INPUT	0	CSI0_DAT16	ALT2
			1	CSI0_DAT17	ALT2
	RXD_MUX	IOMUXC_UART4_IPP_UART_RXD_MUX_SELECT_INPUT	0b00	KEY_COL0	ALT4
			0b01	KEY_ROW0	ALT4
			0b10	CSI0_DAT12	ALT2
			0b11	CSI0_DAT13	ALT2

Table continues on the next page...

Table 4-5. Daisy Chain Multiplexer Control Registers (by block instance / block I/O) (continued)

Block Instance	Block I/O	Select Register	DAISY	Package Pin	Mode
UART-5	RTS	IOMUXC_UART5_IPP_UART_RTS_B_SELECT_INPUT	0b00	KEY_COL4	ALT4
			0b01	KEY_ROW4	ALT4
			0b10	CSI0_DAT18	ALT2
			0b11	CSI0_DAT19	ALT2
	RXD_MUX	IOMUXC_UART5_IPP_UART_RXD_MUX_SELECT_INPUT	0b00	KEY_COL1	ALT4
			0b01	KEY_ROW1	ALT4
			0b10	CSI0_DAT14	ALT2
			0b11	CSI0_DAT15	ALT2
USB	USBOTG_OC	IOMUXC_USBOH3_IPP_IND_OTG_OC_SELECT_INPUT	0	KEY_COL4	ALT5
			1	EIM_D21	ALT6
	USBH1_OC	IOMUXC_USBOH3_IPP_IND_UH1_OC_SELECT_INPUT	0	EIM_D30	ALT6
			1	GPIO_3	ALT6
	USBH2_OC USBH2_OC	IOMUXC_USBOH3_IPP_IND_UH2_OC_SELECT_INPUT	0	EIM_D19	ALT7
			1	EIM_D30	ALT7

4.3 Special Package Pins

In addition to the package pins discussed above, i.MX53 has other dedicated pins that serve special purposes. These are generally pins that convey signals with special voltage or drive characteristics or are used for test purposes.

Table 4-6 lists package analog and power pins and their characteristics.

Table 4-6. Special Package Pins

Pin Name	Internal Signal Name	Buffer Type	Signal Use
CKIL	CKIL	Analog	Pervasive
ECKIL	ECKIL	Analog	Pervasive
EXTAL	EXTAL	Analog	Pervasive

Table continues on the next page...

Table 4-6. Special Package Pins (continued)

Pin Name	Internal Signal Name	Buffer Type	Signal Use
FASTR_ANA	FASTR_ANA	Analog	Freescale use only
FASTR_DIG	FASTR_DIG	Analog	Freescale use only
DRAM_CALIBRATION	DRAM_CALIBRATION	Calibration	External Memory Controller
LVDS_BG_RES	LVDS_BG_RES	Analog	Pervasive
SATA_REFCLKM	SATA_REFCLKM	Analog	SATA PHY
SATA_REFCLKP	SATA_REFCLKP	Analog	SATA PHY
SATA_REXT	SATA_REXT	Analog	SATA PHY
SATA_RXM	SATA_RXM	Analog	SATA PHY
SATA_RXP	SATA_RXP	Analog	SATA PHY
SATA_TXM	SATA_TXM	Analog	SATA PHY
SATA_TXP	SATA_TXP	Analog	SATA PHY
TVCDC_IOB_BACK	TVCDC_IOB_BACK	Analog	TV Encoder
TVCDC_IOG_BACK	TVCDC_IOG_BACK	Analog	TV Encoder
TVCDC_IOR_BACK	TVCDC_IOR_BACK	Analog	TV Encoder
TVDAC_COMP	TVDAC_COMP	Analog	TV Encoder
TVDAC_IOB	TVDAC_IOB	Analog	TV Encoder
TVDAC_IOG	TVDAC_IOG	Analog	TV Encoder
TVDAC_IOR	TVDAC_IOR	Analog	TV Encoder
TVDAC_VREF	TVDAC_VREF	Analog	TV Encoder
USB_H1_DN	USB_H1_DN	Analog50	USB H1 PHY
USB_H1_DP	USB_H1_DP	Analog50	USB H1 PHY
USB_H1_GPANAIO	USB_H1_GPANAIO	Analog25	USB H1 PHY
USB_H1_RREFEXT	USB_H1_RREFEXT	Analog25	USB H1 PHY
USB_H1_VBUS	USB_H1_VBUS	Analog50	USB H1 PHY
USB_OTG_DN	USB_OTG_DN	Analog50	USB OTG PHY
USB_OTG_DP	USB_OTG_DP	Analog50	USB OTG PHY
USB_OTG_GPANAIO	USB_OTG_GPANAIO	Analog25	USB OTG PHY
USB_OTG_ID	USB_OTG_ID	Analog25	USB OTG PHY
USB_OTG_RREFEXT	USB_OTG_RREFEXT	Analog25	USB OTG PHY
USB_OTG_VBUS	USB_OTG_VBUS	Analog50	USB OTG PHY
XTAL	XTAL	Analog	Pervasive

Chapter 5
External Memory

5.1 Overview

The External Memory Controller (EXTMC) is the block that services all i.MX53 external memory accesses requests (read/write/erase/program) from all the AXI bus masters in the system. All accesses are arbitrated by the Multi Master Multi Memory Interface (M4IF) block and controlled by the respective memory controller. The high level block diagram is presented in Figure 5-1.

Each AXI bus master interface port is an AXI interface with separated input bus for read and write access, so masters that handle separated buses for read and write access can access the EXTMC. The data width can be 32 or 64 bits. All ports support 32 or 64-bit IF.

NOTE

The following is a summary of the EXTMC used in i.MX53. This summary takes precedence if any discrepancy between the information here and the information in the EXTMC (and related sub-blocks) block guide arises. Please note that in case of conflict, the information related to frequencies of operation supersedes that of the block chapters and is superseded by that of the data sheet.

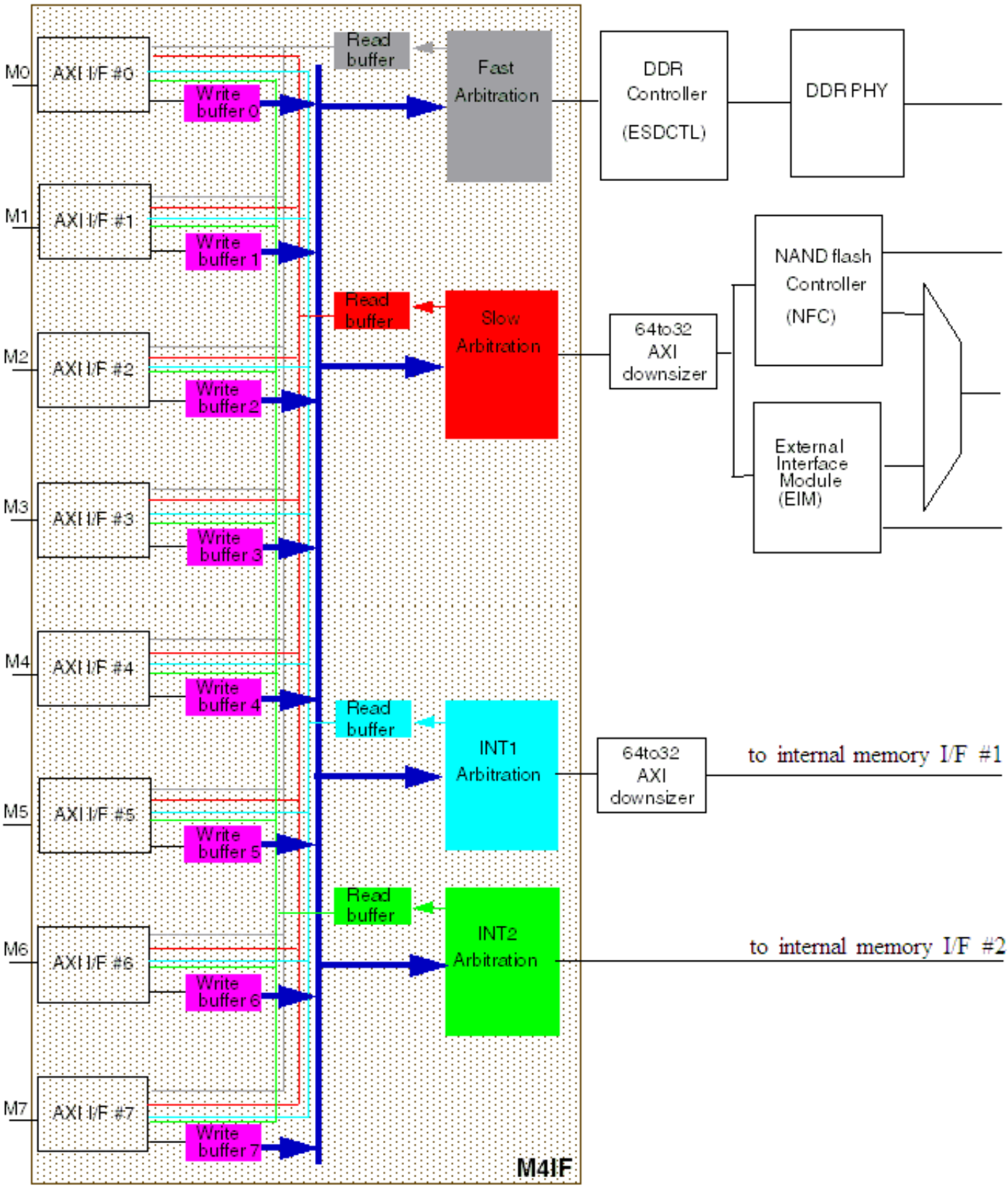


Figure 5-1. EXTMC High Level Block Diagram

5.2 External Memory Interface - i.MX53 Specific Configuration

The EXTMC provides the ability to connect to a wide variety of memory devices. This chapter contains the technical information about the operation and configuration of the EXTMC in the chip to allow the designer to quickly integrate external memory devices into new and existing designs. Pin sharing is done between the data bus of the external interface module (EIM) and the NAND Flash Controller (NFC) in order to reduce the total number of pins needed for the EXTMC.

The EXTMC is an External Memory Interface and arbitration between multi AXI masters to multi memory controllers, divided into three major channels: fast memories (DDR2, DDR3, LPDDR2) channel, slow memories (NOR-FLASH/PSRAM/NAND-FLASH etc.) channel, Internal Memory (RAM, ROM) channel, and Internal GMEM memory.

In order to increase the bandwidth performance, the EXTMC separates the buffering and the arbitration between accesses to fast channel slow channel and Internal Memory channels, so parallel accesses can occur. By separating the three channels slow accesses don't interfere with fast accesses.

The EXTMC contains the arbitration interface and different external memory controllers in order to support several memory devices:

- M4IF - Multi Master Multi Memory Interface.
- ESDRAMC - Enhanced DDR2/DDR3/LPDDR2 memory controller.
- NFC - NANDFlash memory controller.
- EIM - SRAM/PSRAM/NOR FLASH memory controller

5.2.1 EXTMC - AXI Bus Masters

The table below provides details on AXI bus masters and the AXI bus master interface port associated with each master. Please refer to the M4IF chapter for buffer sizes for each master.

Table 5-1. AXI Bus Masters

Block	Direct Bus	Master Port	Buffered	Boundary Crossing	Max frequency Mhz
IPU	AXI	0	y	4K	200MHz
VPU	AXI	1	y	4K	200MHz

Table continues on the next page...

Table 5-1. AXI Bus Masters (continued)

Block	Direct Bus	Master Port	Buffered	Boundary Crossing	Max frequency Mhz
DAP	AHB via AHBMAX	2	n	1K	133MHz
SAHARA	AHB via AHBMAX	2	n	1K	133MHz
SCC	AHB via AHBMAX	2	n	1K	133MHz
RTIC	AHB via AHBMAX	2	n	1K	133MHz
SDMA (non-burst)	AHB via AHBMAX	2	n	1K	133MHz
ESDHCV2-2	AHB via AHBMAX	2	n	1K	133MHz
ESDHCV2-1	AHB via AHBMAX	2	n	1K	133MHz
ECSPI-1	AHB via AHBMAX	2	n	1K	133MHz
ESDHCV2-3	AHB via AHBMAX	2	n	1K	133MHz
PATA	AHB via AHBMAX	2	n	1K	133MHz
EXTMC ARB3	AHB via AHBMAX	2	n	1K	133MHz
ARM Cortex-A8	AXI	3	y	4K	200MHz
SDMA (burst)	AHB via PLARB1	4	n	1K	133MHz
FEC	AHB via PLARB1	4	n	1K	66MHz
SATA	AHB via PLARB1	4	n	1K	133MHz
GPU3D	AXI	5	y	4K	200MHz
GPU2D (OpenVG)	AXI	6	y	4K	200MHz
USB	AXI	7	n	4K	133MHz

5.2.2 Features

Each of the EXTMC memory controller block guide specify detailed information on the supported features, programming model. However, in i.MX53 some of those feature are disabled or not supported.

The EXTMC in the i.MX53 includes these distinctive features:

- Multi Master Multi Memory Interface (M4IF)
  - Supports multiple accesses from 8 masters through different input ports interfaces. Each port can support either of the following two data width options:
    - x32 AXI port.
    - x64 AXI port.
  - Supports different clock domain for each AXI port master.
  - Configurable memory watermark protection per DDR2/DDR3/LPDDR2 CS.
  - Enhanced arbitration scheme for fast channel, consider page hit/miss, last access details (read/write) and fixed priority configuration.

- Enhanced DDR Controller (ESDRAMC)
  - Up to 2 chip selects support up to 1GByte each.
  - x64 AXI port.
  - Supports x16/x32 DDR2 memories up to 400MHz (800 MHz data rate)
  - Supports x16/x32 DDR3 memories up to 400MHz (800 MHz data rate)
  - Supports x32 LPDDR2 memory (Please refer to the i.MX53 Data Sheet for target frequencies)
  - Supports latency hiding logic.
- NANDFlash Controller - (NFC)
  - x8/x16 NAND interface.
  - Up to 8 chip selects support up to 8Gbit in 1/2K page mode, 64Gbit in 2K page mode and 256Gbit in 4K page mode each.
  - 4.5K RAM Internal Buffer
  - MLC and SLC memory support.
  - Configurable operation mode - symmetric and asymmetric.
  - Configurable page mode, 1/2K, 2K or 4K.
  - ECC support up to 16 bit.
  - Supports up to 8 mutually exclusive, yet interleaved nand devices.
  - Automatic Common NAND Operations
- External Interface Memory Controller - (EIM)
  - Up to 4 chip selects, programmable using the General Purpose Register in IOMUXC.
  - Supports x32/x16/x8 PSRAM/NOR (at slow frequency).
- Supports DVFS, voltage and frequency change.
- Supports automatic shut-down (power saving features)
- Enhanced debug capabilities (trace mode).

Table 5-2. EIM multiplexing

Setup	Non Multiplexed Address/Data Mode						Multiplexed Address/ Data mode	
	8 Bit			16 Bit		32 Bit	16 Bit	32 Bit
	MUM = 0, DSZ = 100	MUM = 0, DSZ = 101	MUM = 0, DSZ = 111	MUM = 0, DSZ = 001	MUM = 0, DSZ = 010	MUM = 0, DSZ = 011	MUM = 1, DSZ = 001	MUM = 1, DSZ = 011
A[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]
A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[24:16][1]	EIM_A[25:16]	EIM_A[8:0]
D[7:0], EIM_EB0	NANDF_D[7:0]	-	-	NANDF_D[7:0][2]	-	NANDF_D[7:0]	EIM_DA[7:0]	EIM_DA[7:0]
D[15:8], EIM_EB1	-	NANDF_D[15:8]	-	NANDF_D[15:8][3]	-	NANDF_D[15:8]	EIM_DA[15:8]	EIM_DA[15:8]

Table continues on the next page...

Table 5-2. EIM multiplexing (continued)

Setup	Non Multiplexed Address/Data Mode						Multiplexed Address/ Data mode	
	8 Bit			16 Bit		32 Bit	16 Bit	32 Bit
	MUM = 0, DSZ = 100	MUM = 0, DSZ = 101	MUM = 0, DSZ = 111	MUM = 0, DSZ = 001	MUM = 0, DSZ = 010	MUM = 0, DSZ = 011	MUM = 1, DSZ = 001	MUM = 1, DSZ = 011
D[23:16], EIM_EB2	-	-	-	-	EIM_D[23:16]	EIM_D[23:16]	-	NANDF_D[7:0]
D[31:24], EIM_EB3	-	-	EIM_D[31:24]	-	EIM_D[31:24]	EIM_D[31:24]	-	NANDF_D[15:8]

1. For 32-bit, the address range is A[24:0], due to address space allocation in memory map.
2. NANDF_D[7:0] multiplexed on ALT3 mode of PATA_DATA[7:0]
3. NANDF_D[15:8] multiplexed on ALT3 mode of PATA_DATA[15:8]

5.3 EXTMC Setup

The following section describes i.MX53 specific requirements in order to use the EXTMC.

5.3.1 Clock Domains

EXTMC contains the following clock domains:

- EXTMC IP Bus clock.
- ESDCTLV2 main clock (up to 400MHz)
- Slow arbitration clock. (up to 133MHz)
- Internal memory arbitration 1 & 2 clocks. (up to 133MHz)
- 8 x master clocks, can be asynchronous to EXTMC arbitration clocks (up to 200MHz) see Table 5-3.
- NFC main clock - should be integer divided from slow arbitration clock. (up to 50MHz)
- SDCLK - SDR/DDR clock to SDR/DDR SDRAM device.
- BCLK - NOR Flash/PSRAM clock WEIMv2 in synchronous mode.

5.3.2 Boot Scenarios

The EXTMC memory controllers allow booting from the following memories: NOR Flash devices (through the EIM) and NAND Flash devices (through the NFC) special signals coming from the SoC define the different booting options to those memories like

the memory data width, memory page size and other specific parameters for the initial access of the boot. For more details refer to the memory controllers detailed chapters and boot chapter.

5.3.3 Watermark Ports

Watermark regions are supported in the EXTMC while the configuration is handled by the CSU. The external memory spaces will have trust zone area's that only trust zone accesses will be able to reach. Non trust zone accesses will be blocked. The central security unit (CSU) has the registers that define those regions and send the indication of the region and if specific access is trust zone or not to the EXTMC.

The watermark indication is dynamic and can change for each access.

For details on the watermark configuration refer to the CSU chapter.

5.3.4 EXTMC I/O Multiplexing

The EXTMC gives the system the ability to multiplex some of its signals in the SoC to allow pin sharing between the different memory controllers. Multiplexing is possible on the "slow" channel only because of timing limitations.

Only shared i.MX53 pins signals/busses are routed through the IOMUXC toward the external devices. Signals (mainly controls) that have dedicated pins are directly routed from the memory controllers to the external devices.

Table 5-3. EXTMC I/O Multiplexing in i.MX53

EXTMC port	i.MX53 pin	Mode
DRAM_A[0]	DRAM_A0	No Muxing (ALT0)
DRAM_A[10]	DRAM_A10	No Muxing (ALT0)
DRAM_A[11]	DRAM_A11	No Muxing (ALT0)
DRAM_A[12]	DRAM_A12	No Muxing (ALT0)
DRAM_A[13]	DRAM_A13	No Muxing (ALT0)
DRAM_A[14]	DRAM_A14	No Muxing (ALT0)
DRAM_A[15]	DRAM_A15	No Muxing (ALT0)
DRAM_A[1]	DRAM_A1	No Muxing (ALT0)
DRAM_A[2]	DRAM_A2	No Muxing (ALT0)
DRAM_A[3]	DRAM_A3	No Muxing (ALT0)
DRAM_A[4]	DRAM_A4	No Muxing (ALT0)

Table continues on the next page...

Table 5-3. EXTMC I/O Multiplexing in i.MX53 (continued)

EXTMC port	i.MX53 pin	Mode
DRAM_A[5]	DRAM_A5	No Muxing (ALT0)
DRAM_A[6]	DRAM_A6	No Muxing (ALT0)
DRAM_A[7]	DRAM_A7	No Muxing (ALT0)
DRAM_A[8]	DRAM_A8	No Muxing (ALT0)
DRAM_A[9]	DRAM_A9	No Muxing (ALT0)
DRAM_CAS	DRAM_CAS	No Muxing (ALT0)
DRAM_CS[0]	DRAM_CS0	No Muxing (ALT0)
DRAM_CS[1]	DRAM_CS1	No Muxing (ALT0)
DRAM_DQM[0]	DRAM_DQM0	No Muxing (ALT0)
DRAM_DQM[1]	DRAM_DQM1	No Muxing (ALT0)
DRAM_DQM[2]	DRAM_DQM2	No Muxing (ALT0)
DRAM_DQM[3]	DRAM_DQM3	No Muxing (ALT0)
DRAM_D[0]	DRAM_D0	No Muxing (ALT0)
DRAM_D[10]	DRAM_D10	No Muxing (ALT0)
DRAM_D[11]	DRAM_D11	No Muxing (ALT0)
DRAM_D[12]	DRAM_D12	No Muxing (ALT0)
DRAM_D[13]	DRAM_D13	No Muxing (ALT0)
DRAM_D[14]	DRAM_D14	No Muxing (ALT0)
DRAM_D[15]	DRAM_D15	No Muxing (ALT0)
DRAM_D[16]	DRAM_D16	No Muxing (ALT0)
DRAM_D[17]	DRAM_D17	No Muxing (ALT0)
DRAM_D[18]	DRAM_D18	No Muxing (ALT0)
DRAM_D[19]	DRAM_D19	No Muxing (ALT0)
DRAM_D[1]	DRAM_D1	No Muxing (ALT0)
DRAM_D[20]	DRAM_D20	No Muxing (ALT0)
DRAM_D[21]	DRAM_D21	No Muxing (ALT0)
DRAM_D[22]	DRAM_D22	No Muxing (ALT0)
DRAM_D[23]	DRAM_D23	No Muxing (ALT0)
DRAM_D[24]	DRAM_D24	No Muxing (ALT0)
DRAM_D[25]	DRAM_D25	No Muxing (ALT0)
DRAM_D[26]	DRAM_D26	No Muxing (ALT0)
DRAM_D[27]	DRAM_D27	No Muxing (ALT0)
DRAM_D[28]	DRAM_D28	No Muxing (ALT0)
DRAM_D[29]	DRAM_D29	No Muxing (ALT0)
DRAM_D[2]	DRAM_D2	No Muxing (ALT0)

Table continues on the next page...

Table 5-3. EXTMC I/O Multiplexing in i.MX53 (continued)

EXTMC port	i.MX53 pin	Mode
DRAM_D[30]	DRAM_D30	No Muxing (ALT0)
DRAM_D[31]	DRAM_D31	No Muxing (ALT0)
DRAM_D[3]	DRAM_D3	No Muxing (ALT0)
DRAM_D[4]	DRAM_D4	No Muxing (ALT0)
DRAM_D[5]	DRAM_D5	No Muxing (ALT0)
DRAM_D[6]	DRAM_D6	No Muxing (ALT0)
DRAM_D[7]	DRAM_D7	No Muxing (ALT0)
DRAM_D[8]	DRAM_D8	No Muxing (ALT0)
DRAM_D[9]	DRAM_D9	No Muxing (ALT0)
DRAM_ODT[0]	DRAM_SDODT0	No Muxing (ALT0)
DRAM_ODT[1]	DRAM_SDODT1	No Muxing (ALT0)
DRAM_RAS	DRAM_RAS	No Muxing (ALT0)
DRAM_RESET	DRAM_RESET	No Muxing (ALT0)
DRAM_SDBA[0]	DRAM_SDBA0	No Muxing (ALT0)
DRAM_SDBA[1]	DRAM_SDBA1	No Muxing (ALT0)
DRAM_SDBA[2]	DRAM_SDBA2	No Muxing (ALT0)
DRAM_SDCKE[0]	DRAM_SDCKE0	No Muxing (ALT0)
DRAM_SDCKE[1]	DRAM_SDCKE1	No Muxing (ALT0)
DRAM_SDCLK0	DRAM_SDCLK_0	No Muxing (ALT0)
DRAM_SDCLK0_B	DRAM_SDCLK_0_B	No Muxing (ALT0)
DRAM_SDCLK1	DRAM_SDCLK_1	No Muxing (ALT0)
DRAM_SDCLK1_B	DRAM_SDCLK_1_B	No Muxing (ALT0)
DRAM_SDQS[0]	DRAM_SDQS0	No Muxing (ALT0)
DRAM_SDQS[1]	DRAM_SDQS1	No Muxing (ALT0)
DRAM_SDQS[2]	DRAM_SDQS2	No Muxing (ALT0)
DRAM_SDQS[3]	DRAM_SDQS3	No Muxing (ALT0)
DRAM_SDQS_B[0]	DRAM_SDQS0_B	No Muxing (ALT0)
DRAM_SDQS_B[1]	DRAM_SDQS1_B	No Muxing (ALT0)
DRAM_SDQS_B[2]	DRAM_SDQS2_B	No Muxing (ALT0)
DRAM_SDQS_B[3]	DRAM_SDQS3_B	No Muxing (ALT0)
DRAM_SDWE	DRAM_SDWE	No Muxing (ALT0)
EMI_DEBUG[0]	DI0_DISP_CLK	ALT6
EMI_DEBUG[10]	DISP0_DAT5	ALT6
EMI_DEBUG[11]	DISP0_DAT6	ALT6
EMI_DEBUG[12]	DISP0_DAT7	ALT6

Table continues on the next page...

Table 5-3. EXTMC I/O Multiplexing in i.MX53 (continued)

EXTMC port	i.MX53 pin	Mode
EMI_DEBUG[13]	DISP0_DAT8	ALT6
EMI_DEBUG[14]	DISP0_DAT9	ALT6
EMI_DEBUG[15]	DISP0_DAT10	ALT6
EMI_DEBUG[16]	DISP0_DAT11	ALT6
EMI_DEBUG[17]	DISP0_DAT12	ALT6
EMI_DEBUG[18]	DISP0_DAT13	ALT6
EMI_DEBUG[19]	DISP0_DAT14	ALT6
EMI_DEBUG[1]	DI0_PIN15	ALT6
EMI_DEBUG[20]	DISP0_DAT15	ALT6
EMI_DEBUG[21]	DISP0_DAT16	ALT6
EMI_DEBUG[22]	DISP0_DAT17	ALT6
EMI_DEBUG[23]	DISP0_DAT18	ALT6
EMI_DEBUG[24]	DISP0_DAT19	ALT6
EMI_DEBUG[25]	DISP0_DAT20	ALT6
EMI_DEBUG[26]	DISP0_DAT21	ALT6
EMI_DEBUG[27]	DISP0_DAT22	ALT6
EMI_DEBUG[28]	DISP0_DAT23	ALT6
EMI_DEBUG[29]	CSI0_PIXCLK	ALT6
EMI_DEBUG[2]	DI0_PIN2	ALT6
EMI_DEBUG[30]	CSI0_MCLK	ALT6
EMI_DEBUG[31]	CSI0_DATA_EN	ALT6
EMI_DEBUG[32]	CSI0_VSYNC	ALT6
EMI_DEBUG[33]	CSI0_DAT4	ALT6
EMI_DEBUG[34]	CSI0_DAT5	ALT6
EMI_DEBUG[35]	CSI0_DAT6	ALT6
EMI_DEBUG[36]	CSI0_DAT7	ALT6
EMI_DEBUG[37]	CSI0_DAT8	ALT6
EMI_DEBUG[38]	CSI0_DAT9	ALT6
EMI_DEBUG[39]	CSI0_DAT10	ALT6
EMI_DEBUG[3]	DI0_PIN3	ALT6
EMI_DEBUG[40]	CSI0_DAT11	ALT6
EMI_DEBUG[41]	CSI0_DAT12	ALT6
EMI_DEBUG[42]	CSI0_DAT13	ALT6
EMI_DEBUG[43]	CSI0_DAT14	ALT6
EMI_DEBUG[44]	CSI0_DAT15	ALT6

Table continues on the next page...

Table 5-3. EXTMC I/O Multiplexing in i.MX53 (continued)

EXTMC port	i.MX53 pin	Mode
EMI_DEBUG[45]	CSI0_DAT16	ALT6
EMI_DEBUG[46]	CSI0_DAT17	ALT6
EMI_DEBUG[47]	CSI0_DAT18	ALT6
EMI_DEBUG[48]	CSI0_DAT19	ALT6
EMI_DEBUG[49]	FEC_MDIO	ALT6
EMI_DEBUG[4]	DI0_PIN4	ALT6
EMI_DEBUG[50]	FEC_REF_CLK	ALT6
EMI_DEBUG[5]	DISP0_DAT0	ALT6
EMI_DEBUG[6]	DISP0_DAT1	ALT6
EMI_DEBUG[7]	DISP0_DAT2	ALT6
EMI_DEBUG[8]	DISP0_DAT3	ALT6
EMI_DEBUG[9]	DISP0_DAT4	ALT6
NANDF_ALE	NANDF_ALE	ALT0
NANDF_CLE	NANDF_CLE	ALT0
NANDF_CS[0]	NANDF_CS0	ALT0
NANDF_CS[1]	NANDF_CS1	ALT0
NANDF_CS[2]	NANDF_CS2	ALT0
NANDF_CS[3]	NANDF_CS3	ALT0
NANDF_D[0]	PATA_DATA0	ALT3
NANDF_D[10]	PATA_DATA10	ALT3
NANDF_D[11]	PATA_DATA11	ALT3
NANDF_D[12]	PATA_DATA12	ALT3
NANDF_D[13]	PATA_DATA13	ALT3
NANDF_D[14]	PATA_DATA14	ALT3
NANDF_D[15]	PATA_DATA15	ALT3
NANDF_D[1]	PATA_DATA1	ALT3
NANDF_D[2]	PATA_DATA2	ALT3
NANDF_D[3]	PATA_DATA3	ALT3
NANDF_D[4]	PATA_DATA4	ALT3
NANDF_D[5]	PATA_DATA5	ALT3
NANDF_D[6]	PATA_DATA6	ALT3
NANDF_D[7]	PATA_DATA7	ALT3
NANDF_D[8]	PATA_DATA8	ALT3
NANDF_D[9]	PATA_DATA9	ALT3
NANDF_RB[0]	NANDF_RB0	ALT0

Table continues on the next page...

Table 5-3. EXTMC I/O Multiplexing in i.MX53 (continued)

EXTMC port	i.MX53 pin	Mode
NANDF_RE_B	NANDF_RE_B	ALT0
NANDF_WE_B	NANDF_WE_B	ALT0
NANDF_WP_B	NANDF_WP_B	ALT0
NAND_WEIM_DA[0]	EIM_DA0	ALT0
NAND_WEIM_DA[10]	EIM_DA10	ALT0
NAND_WEIM_DA[11]	EIM_DA11	ALT0
NAND_WEIM_DA[12]	EIM_DA12	ALT0
NAND_WEIM_DA[13]	EIM_DA13	ALT0
NAND_WEIM_DA[14]	EIM_DA14	ALT0
NAND_WEIM_DA[15]	EIM_DA15	ALT0
NAND_WEIM_DA[1]	EIM_DA1	ALT0
NAND_WEIM_DA[2]	EIM_DA2	ALT0
NAND_WEIM_DA[3]	EIM_DA3	ALT0
NAND_WEIM_DA[4]	EIM_DA4	ALT0
NAND_WEIM_DA[5]	EIM_DA5	ALT0
NAND_WEIM_DA[6]	EIM_DA6	ALT0
NAND_WEIM_DA[7]	EIM_DA7	ALT0
NAND_WEIM_DA[8]	EIM_DA8	ALT0
NAND_WEIM_DA[9]	EIM_DA9	ALT0
WEIM_A[16]	EIM_A16	ALT0
WEIM_A[17]	EIM_A17	ALT0
WEIM_A[18]	EIM_A18	ALT0
WEIM_A[19]	EIM_A19	ALT0
WEIM_A[20]	EIM_A20	ALT0
WEIM_A[21]	EIM_A21	ALT0
WEIM_A[22]	EIM_A22	ALT0
WEIM_A[23]	EIM_A23	ALT0
WEIM_A[24]	EIM_A24	ALT0
WEIM_A[25]	EIM_A25	ALT0
WEIM_A[26]	NANDF_CS3	ALT4
WEIM_BCLK	EIM_BCLK	No Muxing (ALT0)
WEIM_CRE	NANDF_CS2	ALT4
WEIM_CS[0]	EIM_CS0	ALT0
WEIM_CS[1]	EIM_CS1	ALT0
WEIM_CS[2]	DISP0_DAT18	ALT7

Table continues on the next page...

Table 5-3. EXTMC I/O Multiplexing in i.MX53 (continued)

EXTMC port	i.MX53 pin	Mode
WEIM_CS[3]	DISP0_DAT19	ALT7
WEIM_DTACK_B	EIM_WAIT	ALT2
WEIM_D[16]	EIM_D16	ALT0
WEIM_D[17]	EIM_D17	ALT0
WEIM_D[18]	EIM_D18	ALT0
WEIM_D[19]	EIM_D19	ALT0
WEIM_D[20]	EIM_D20	ALT0
WEIM_D[21]	EIM_D21	ALT0
WEIM_D[22]	EIM_D22	ALT0
WEIM_D[23]	EIM_D23	ALT0
WEIM_D[24]	EIM_D24	ALT0
WEIM_D[25]	EIM_D25	ALT0
WEIM_D[26]	EIM_D26	ALT0
WEIM_D[27]	EIM_D27	ALT0
WEIM_D[28]	EIM_D28	ALT0
WEIM_D[29]	EIM_D29	ALT0
WEIM_D[30]	EIM_D30	ALT0
WEIM_D[31]	EIM_D31	ALT0
WEIM_EB[0]	EIM_EB0	ALT0
WEIM_EB[1]	EIM_EB1	ALT0
WEIM_EB[2]	EIM_EB2	ALT0
WEIM_EB[3]	EIM_EB3	ALT0
WEIM_LBA	EIM_LBA	ALT0
WEIM_OE	EIM_OE	ALT0
WEIM_RW	EIM_RW	ALT0
WEIM_WAIT	EIM_WAIT	ALT0

5.3.5 External Interface Module (EIM) boot configuration

Table 5-4. EIM Boot configuration

EIM_BOOT bits	EIM affected bits	EIM Register	Boot Value
12	NUM16_BYP_GRANT	EIM_CS0GCR2	Constant 1
11	DZS[2]	EIM_CS0GCR1	Constant 0

Table continues on the next page...

Table 5-4. EIM Boot configuration (continued)

EIM_BOOT bits	EIM affected bits	EIM Register	Boot Value
10	AUS	EIM_CS0GCR1	Constant 0
[9:8]	CSREC[2:1]	EIM_CS0GCR1	Constant 11
[7:5]	RWSC[4:2] WWSC[4:2]	EIM_CS0GCR1 EIM_CS0WCR	Constant 111
4	ERRST	EIM_WCR	Constant 0
3	RAL WAL	EIM_CS0RCR1 EIM_CS0WCR	Constant 0
2	MUM OEA	EIM_CS0GCR1 EIM_CS0RCR1	Configurable by fuses; If SRC.SBMR[BOOT_CFG2[7:6]] = 0b00, value = 0x100 If SRC.SBMR[BOOT_CFG2[7:6]] = 0b00, value = 0x010
[1:0]	DSZ[1:0]	EIM_CS0GCR1	

5.4 External Memory Controller (EXTMC) Restrictions

5.4.1 Exclusive Access Support

EXTMC does not include an exclusive access monitor. Master 3 (M3) connected to ARM platform at i.MX53 returns EXOKAY for any exclusive access coming from ARM platform. For write accesses this means that the memory location is updated but exclusivity is not guaranteed.

5.4.2 Software LPMD

On i.MX53 Software LPMD done by using control bits on M4IF is not supported.

5.4.3 Data Paths

Table 5-5. EXTMC Valid Data Paths Summary

Masters\Slaves	ARB1 [FAST CHANNEL - DDR]	ARB2 [SLOW CHANNEL - EIM, NFC]	ARB3 [INTERNAL CHANNEL0, ROMC, SCC_RAM and IP Bus]	ARB4 [INTERNAL CHANNEL1 - OCRAM]
M0 [IPU]	+	-	+	+

Table continues on the next page...

Table 5-5. EXTMC Valid Data Paths Summary (continued)

Masters\Slaves	ARB1 [FAST CHANNEL - DDR]	ARB2 [SLOW CHANNEL - EIM, NFC]	ARB3 [INTERNAL CHANNEL0, ROMC, SCC_RAM and IP Bus]	ARB4 [INTERNAL CHANNEL1 - OCRAM]
M1 [VPU]	+	-	+	+
M2 [AHBMAX]	+	+	+	+
M3 [ARM]	+	+	+	+
M4 [PLARB1]	+	+	+	+
M5 [GPU3D]	+	-	+	+
M6 [GPU2D]	+	-	+	+
M7 [USB]	+	-	+	+

Table 5-6. PLARB2 Valid Data Paths Summary

Masters\Slaves	S0 [GMEM]	S1 [OCRAM]
M0 [VPU]	-	+
M1 [EXTMC ARB4]	+	+

5.4.4 NAND Flash Restrictions/Limitations

- NFC does not support WRAP access type, so for example FEC accesses to NAND Flash are not supported.
- Using 'Read-Status' Command (that is, not using R/B signals at all, RBB_MODE=0)
  - To work with interleaved (and non-interleaved) schemes, this is the preferred operating mode.
  - In this mode, the status of each device is tested using the 'read-status' command. In interleaved mode, each device status is checked prior to accessing it.
  - This mode, enables use of "Number of Iterations" > "Number of devices", with no impact on performance.
- Using wired OR of device's R/B signals. (RBB_MODE=1)
  - It implies use of R/B signal, with all NAND devices R/B signals are connected in Wired-OR scheme.
  - In non-interleave mode - this mode works just as well as with 'read-status' command.
  - In interleave mode the performance degrades for "Number of Iterations" > "Number of devices" because each new round of accesses (that is, returning to first device), implies that all other devices return to their 'ready' state. This occurs because the NFC checks for 'ready' on the 'OR' R/B signal before re-accessing the first device in the next round. (For example, if using two devices and Number of Iterations = 6, the R/B checks on accesses 1,3 and 5, and the

controller waits for the other device to finish before allowing access to the first device.)

5.4.5 OneNAND Restrictions/Limitations

- For OneNAND devices the following limitation exists (since int_rdy is not in use); The modes described in the EXTMC spec allowing faster wakeup, or getting the interrupt from the EXTMC are not supported.
  - Poll the device and see that it is ready (SW performs read from the device).
  - Connect the RDY signal of the device to any GPIO pin and use it as an interrupt indicating on ready state.

Chapter 6
System Debug

6.1 Overview

This chapter describes the debug architecture of the i.MX53.

6.1.1 Introduction

This chapter describes the hardware and software debug and application development features and resources of i.MX53. There are core/platform-specific resources, resources associated with some of the more complex blocks, and chip-wide resources. Also discussed is the interface to external debug and development tools.

The debug architecture of i.MX53 is based on that of previous members of the i.MX application processor family. A small number of changes and extensions were necessary to accommodate new blocks.

An overview i.MX53's primary debug capabilities and features include:

- JTAG-based access (control, monitoring), built around the System JTAG Controller (SJC)
- Trace Port
- Real-time and Halt-mode debug and profiling capabilities of the ARM platform
- Real-time and Halt-mode debug capabilities of the SDMA core
- An SoC-wide cross-trigger system built around ARM's Embedded Cross Trigger (ECT)
- Visibility of pre-selected critical internal signals via pin muxing
- External memory interface/controller arbitration profiling

Each of these features are described in detail in the following sections of this chapter. i.MX53 also supports ARM CoreSight architecture for system debug and trace capabilities

6.1.2 Debug Strategy

- Software debug - (MSFT kernel)
- Trace the ARM platform activity via ETM trace port or ETB (ARM only)
- Control and monitor via JTAG
- Monitor pre-selected critical internal signals via pin multiplexing - scope

6.2 System JTAG Controller - SJC

The System JTAG Controller (SJC) block is the bridge between external development and test instrumentation and the internal JTAG-accessible debug and test resources. It implements and manages the daisy-chained topology consisting of its own TAP and those of the SDMA, a DSP core (not implemented on i.MX53), the ARM Core, and the ARM DAP. The ARM Core is not part of a daisy-chain. The DSP TAP is excluded by the SJC configuration tie-off, so there is no need to use the BYPASS IR.

6.2.1 JTAG Topology

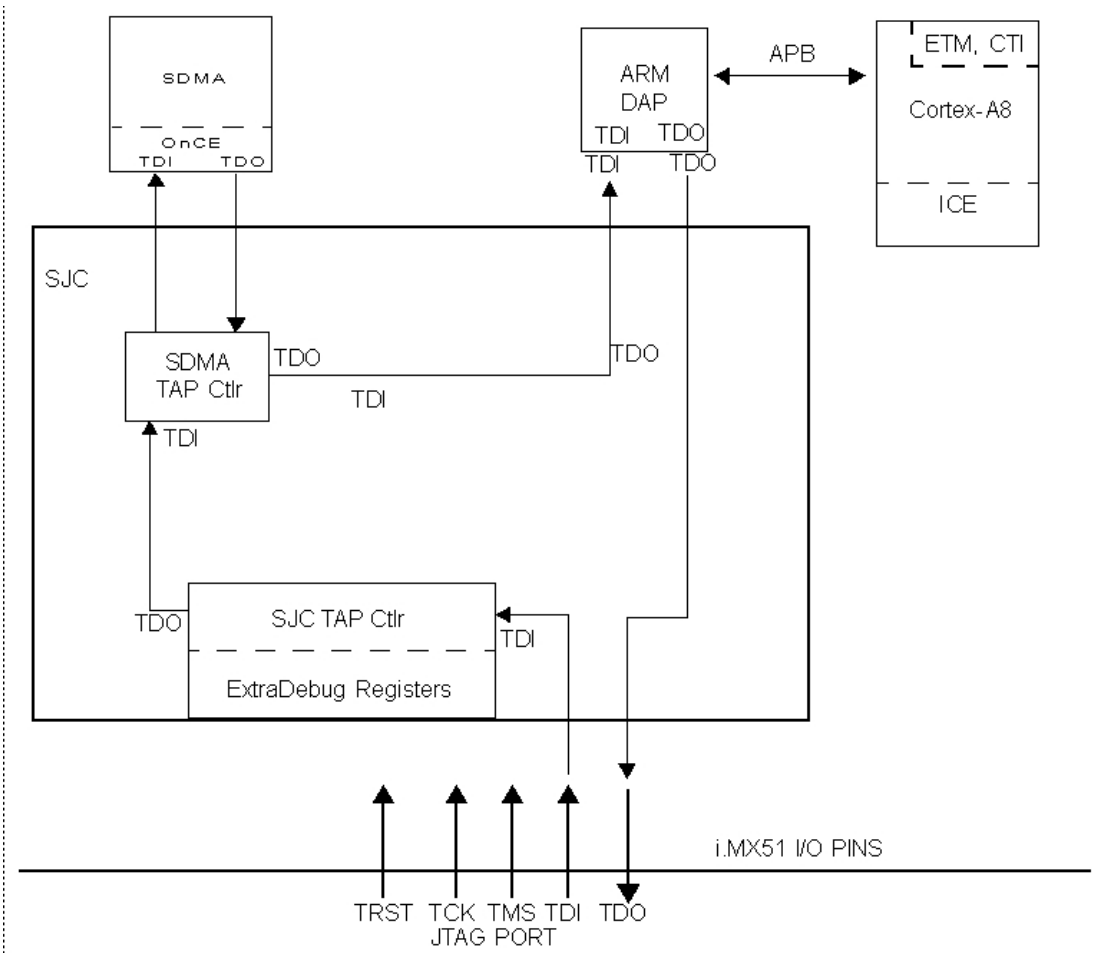


Figure 6-1. JTAG Connectivity

6.2.2 System JTAG Controller Main Feature

- IEEE P1149.1 (standard JTAG) interface to off-chip test and development equipment
  - includes an SJC-only mode for true IEEE 1149.1 compliance, used primarily for board-level implementation of boundary scan.
- Debug-related control and status, such as putting selected cores into reset and/or debug mode and the ability to monitor individual core status signals via JTAG.
- System status, such as the state of the PLLs (locked or not locked)
- Generic status and control, defined on a per-SoC basis by the architecture team
- Factory test related control/status such as PLL bypass and memory BIST
- Four levels of security, ranging from no security to no JTAG accessibility to the chip

6.2.3 SJC TAP Port

The SJC in i.MX53 supports the following standard JTAG pins: TRSTB, TDI, TDO, TCK, and TMS.

6.2.4 SJC Main Blocks

- Interface to the outside world via the standard JTAG pins
- Interface to the external Debug_Event pin
- A master TAP controller, which implements the standard JTAG state machine.
- Implementation of the mandatory and optional IEEE P1149.1 (JTAG) instructions
  - Mandatory: "EXTEST", "SAMPLE/PRELOAD", and "BYPASS"
  - Optional: "ID_CODE" (SoC JTAG ID register), "HIGHZ"
- Supports the SDMA's DR-path-only JTAG architecture by implementing the controller portion of it's TAP (including "BYPASS" as the default state) within the SJC
- Provides the serial interfaces to various DSP-domain JTAG resources (not implemented on i.MX53)
- The ExtraDebug registers, which implement a variety of control and status features
  - Three 32-bit insecure general purpose status registers
  - Two 32-bit secure status registers - one predefined, one general purpose
  - Control and status registers for debug, core, charge pump, PLL, and BIST related functions
  - Control bits for memory timing control (not used on i.MX53)
- Four levels of fuse-defined security, ranging from no security to no access.

Both predefined and user-defined (SoC integration team) Control and Status functions are supported by the SJC. The user-defined functions are defined and documented by the SoC integration team.

6.2.5 i.MX53 Specific SJC Features

6.2.5.1 JTAG Disable Mode

In addition to four different JTAG security modes that are implemented internally in the System JTAG Controller (SJC), there is an option to disable the SJC functionality by e-fuse configuration. This creates additional JTAG mode "JTAG Disabled" with highest

level of JTAG protection. In this mode all JTAG features are disabled. Specifically, the following debug features are disabled in addition to the features that were already disabled in "No Debug" JTAG mode:

- Memory BIST
- Boundary scan register (BSR)
- Non-Secure JTAG control registers (PLL configuration, Deterministic Reset, PLL bypass)
- Non-Secure JTAG status registers (Core status)
- Chip Identification Code (IDCODE)

Please see System Debug SJC Memory Map/Register Definition for i.MX53 specific SJC registers.

6.2.5.2 PROD ID and JTAG ID

i.MX53 PROD_ID=11111010 and JTAG_ID=0x1190_D01D.

Product revision (hw_rev[7:4] in IC Identification Module (IIM)) is = 4'b0000 accordingly.

6.3 CoreSight Design Kit

i.MX53 include ARM CoreSight component for multi core debug and trace solution. CoreSight component can be found in few hierarchy level: ARM Core, ARM platform and top level.

- ARM Core: include the following CoreSight components: ETM, CTI0
- ARM platform: include the following CoreSight components: ETB, ATB Replicator, CTI1, CTM
- Top level: include the following CoreSight components: debug access port (DAP), CTI2, CTI3, TPIU

6.3.1 Memory Map and Register Definition

Each CSDK component has a 4KB location block in the CoreSight memory map. The base address of the CoreSight location block is not fixed but the address offsets are fixed. Each CSDK component has a 4KB memory map, that is, 1K words.

Components connected to the DAP internal bus appear as part of a memory mapped structure with parallel address and two data busses, read data and write data. The bus master, JTAG-DP, controls the ARM platform in this structure the slaves, using normal bus transactions. The JTAG-DP reads and writes registers within this bus.

6.3.2 CoreSight Clock Enable

By default the CoreSight component clocks are enabled, the system get out from reset with all clocks enabled. The CoreSight clocks gating are controlled by programmable Clock Control Module (CCM) registers and DBGEN. The CCM allows control of each CoreSight component separately (by default allowed) and DBGEN that allow and disable all debug clocks. DBGEN is raised when detecting activity on JTAG ports (TMS) and descend on reset (by dap_sys) or controlled by ARM. If both CCM registers bits and DBGEN are high then clock is propagate to CoreSight component.

6.3.3 CoreSight DAP and DAP_SYS

ARM's DAP has several functions. DAP_SYS is a wrapper around DAP and is the FSL block that is actually instantiated in a design (DAP is contained within DAP_SYS). The following list is a summary of the features provided by DAP_SYS.

- Enables debug-related communication between various parts of the system
  - It is a modular block
  - Slave-side support for JTAG and APB (ARM Peripheral Bus) protocols.
  - Debug master can access many debug resources in real time without having to halt the core
  - Enabling of system access to anything connected to the Debug-APB via the APB multiplexer.
- ROM Table - provides a list of memory locations of CoreSight components connected to the Debug APB. Visible from both tools and system access.
- AHB to APB gasket to translate System AHB accesses to APB (input as system APB to the DAP).
- APB Decode to issue select signals to CoreSight components on APB, and handle the PSLVERR and PREADY signals from the components to the DAP.
- AHB-Lite to AHB converter, to translate DAP's AHB-Lite accesses to system's AHB protocol.
- Debug and Reset logic to generate DBGEN signal and reset signals.

The CoreSight components that reside on the Debug-APB include all Embedded Cross Trigger blocks, the ETM, and the ETB.

6.3.4 Embedded Cross Trigger (ECT)

System events, such as a debug request, a core entering or exiting debug mode, the occurrence of a breakpoint or watch point, the occurrence of a particular error, the state of a buffer, etc., can be useful or even essential for debugging or profiling system performance. Whatever the source of the event, the embedded cross trigger implements a flexible, programmable mechanism to transport these events from a source to one or multiple destinations. This includes handling handshake requirements with both the source and the destination as needed and synchronization of signals from different clock domains. The end result is that events of interest that occur in one domain, such as in the ARM Cortex A8 domain, can be recognized and responded to be a destination domain, such as the Smart Direct Memory Access (SDMA).

i.MX53 uses the CoreSight ECT (provided by ARM as part of their full CoreSight Debug Support package) and a custom wrapper to accommodate the various necessary interface scenarios. The ECT (as delivered by ARM) consists of two major blocks - a central Cross Trigger Matrix (CTM) and a Cross Trigger Interface (CTI) block. A Freescale-custom wrapper is added to the CTI to accommodate a variety of different input and output interface scenarios. The wrapped CTI is called the Extended CTI and is configured on a signal by signal basis with port tie-offs.

i.MX53's ECT architecture consists of a single CTM and three wrapped CTIs covering 3 basic groups of functionality as follows:

- CTI0 - is in the Cortex A8 core
- CTI1 - is in the ARM Cortex-A8 core platform, 2 trigger in and 2 trigger out is used by the platform and the others by the device peripherals
- CTI2 - is in the SoC level, is by SDMA
- CTI3 - is in the SoC level, is used by ARM platform peripherals (IPU, GPU, VPU)

Detail CTI pin assignment can be found in CoreSight CTI.

Figure 6-2 is a simple illustration of the relationship between the two processor domains, the SJC and peripherals, and the ARM CoreSight DAP. APB is an AMBA bus used for debugging. It defines what debug and trace components are required and how they are connected. The DAP is the Debug-APB master of the following debug units: 4 CTI blocks, ETM, ETB, TPIU and Cortex A8 debug unit. The Debug-APB bus differs from system-level APB's in that it can be directly accessed via JTAG as well as from the ARM Core. Block selects for each slave module on the Debug-APB are generated by the DAP_SYS.

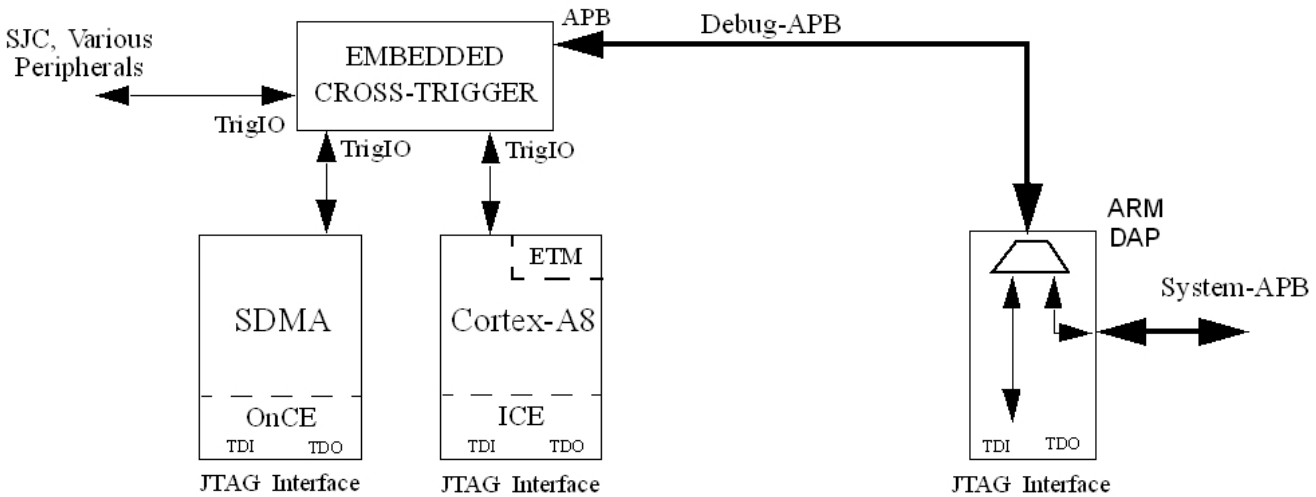


Figure 6-2. Embedded Cross Trigger Topology

Figure 6-3 illustrates the connectivity between the CTM and CTIO components of the CoreSight ECT.

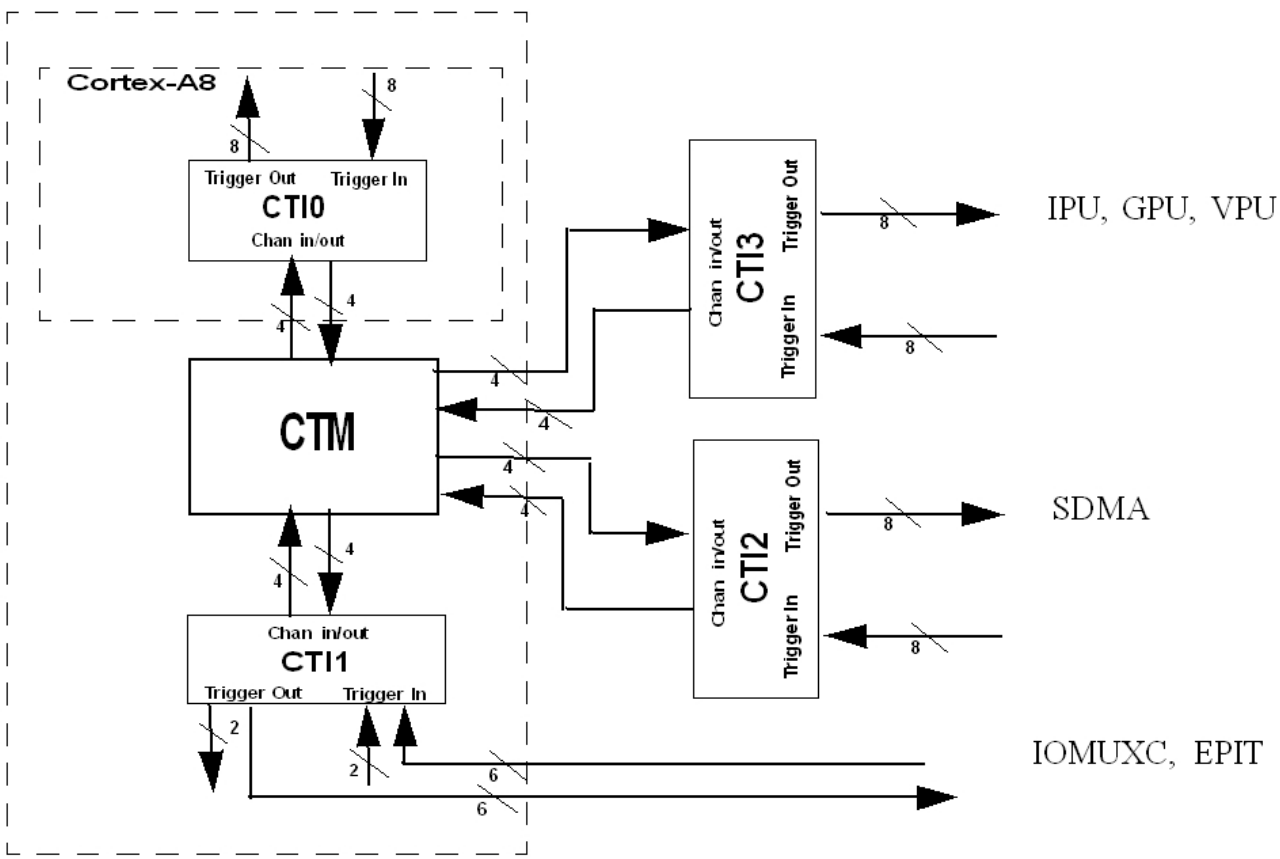


Figure 6-3. Embedded Cross Trigger - Basic Architecture

6.3.4.1 CoreSight CTM

The Cross Trigger Matrix (CTM) is provided by ARM. The CTM is a relatively simple block with no programmability. On i.MX53, it's configured to have 4 input and 4 output channels. Whatever comes in on an input channel is simply routed to all corresponding output channels. For instance, if a trigger input comes in on input channel 0 from CTI0, it is routed to output channel 0 to CTI1, CTI2 & CTI3. All selection of drivers and receivers for an event is done in the individual CTI blocks.

The final feature provided by the CTM is handshake with each CTI. This insures that as trigger signals cross different clock domain boundaries, they won't be lost due to slower downstream sampling frequencies. It is a simple mechanism in which a CTI holds an output until it receives the acknowledgement from the CTM. Handshake is also implemented in the reverse direction, insuring that a trigger signal propagating from the CTM to a CTI won't be missed by the CTI.

6.3.4.2 CoreSight CTI

The CTI (Cross Trigger Interface) is also provided by ARM. A summary description of the CTI is provided here, but please refer to ARM documentation for detailed information. As mentioned earlier, there are 3 CTI's in i.MX53. Each of these has 8 trigger inputs and 8 trigger outputs that connect to logic in the domain to be debugged or profiled. Each CTI also includes a 4 channel interface to the CTM (4 inputs and 4 outputs). The CTIs are wrapped with additional logic to from the Extended CTI, which is described in the next section.

Table 6-1. CTI 1 Input Assignments

Trigger input	Debug resource name	Clock[1]	Channel function	SYNC (dbg_atclk)	Active	ack	Comment
0	observe_mux_0	-	Debug	No	High	No	see IOMUX Control (IOMUXC) observe mux_0 register
1	observe_mux_1	-	Debug	No	High	No	see IOMUXC observe mux_1 register
2-3	-	-	TRACING				ARM
4	observe_mux_2	-	Debug	No	High	No	see IOMUXC observe mux_2 register
5	observe_mux_3	-	Debug	No	High	No	see IOMUXC observe mux_3 register

Table continues on the next page...

Table 6-1. CTI 1 Input Assignments (continued)

Trigger input	Debug resource name	Clock[1]	Channel function	SYNC (dbg_atclk)	Active	ack	Comment
6	observe_mux_4	-	Debug	No	High	No	see IOMUXC observe mux_4 register
7	iomux	-	Debug	-	-	Yes	IOMUX (key_col0 alt3, ack to KEY_ROW0 alt3)

1. The "Clocks" column specifies the associated clock which need to be running for the event to propagate.

Table 6-2. CTI 1 Output Assignments

Trigger output	Debug resource name	Clock[1]	Channel function	SYNC (dbg_atclk)	Active	ack	Comment
0	intin_b	lpcg_ahb_clk_tzic	General Purpose	No	-	No	will allow creating interrupts - int 79
1	intin_b	lpcg_ahb_clk_tzic	General Purpose	No	-	No	will allow creating interrupts - int 80
2	intin_b	lpcg_ahb_clk_tzic	General Purpose	No	-	No	will allow creating interrupts - int 89
3	initin_b	lpcg_ahb_clk_tzic	General Purpose	No	-	No	will allow creating interrupts - int 98
4-5	-	arm_clk	-	-	-		ARM
6	cti_out[6]	arm_clk	Debug	No	-	YES	IOMUX (key col2 alt3, ack on key_col1 alt3)
7	cti_out[7]	arm_clk	Debug	No	-	YES	IOMUX (key_row2 alt3, ack on key row1 alt3)

1. The "Clocks" column specifies the associated clock which need to be running for the event to propagate.

Table 6-3. CTI 2 Input Assignments

Trigger input	Debug resource name	Clock[1]	Channel function	SYNC (debug_apb_clk)	Active	ack	Comment
0	debug_mode	ipg_clk_sdma	Debug Acknowledge	Yes	High	No	SDMA
1	debug_core_run	ipg_clk_sdma	debug	Yes	High	No	SDMA
2	evt_chn_lines[0]	ipg_clk_sdma	debug	Yes	High	No	SDMA
3	evt_chn_lines[1]	ipg_clk_sdma	debug	Yes	High	No	SDMA
4	evt_chn_lines[2]	ipg_clk_sdma	debug	Yes	High	No	SDMA
5	req_ma	ipg_clk_spba	debug	Yes	High	No	SPBA
6	req_mb	ipg_clk_spba	debug	Yes	High	No	SPBA
7	req_mc	ipg_clk_spba	debug	Yes	High	No	SPBA

1. The "Clocks" column specifies the associated clock which need to be running for the event to pı

Table 6-4. CTI 2 Output Assignments

Trigger input	Debug resource name	Clock[1]	Channel function	SYNC (debug_apb_clk)	Active	ack	Comment
0	events[11]	ipg_clk_sdma		Yes	High	No	SDMA. muxed with esdhc4.
1	sdma_dreq_i	ipg_clk_sdma		Yes	High	No	SDMA
2	FLUSHIN	lpcg_debug_clk_tpiu		No	High	Yes	TPIU, FLUSHINACK
3	TRIGIN	lpcg_debug_clk_tpiu		No	High	Yes	TPIU, TRIGINACK
4	system_debug	GND/tck		NO	High	No	System JTAG Controller (SJC)
5	ect_out	emi_enfc_clk		Yes	High	No	External Memory Controller (EXTMC)
6	-	-	-	-	-	-	Not used
7	-	-	-	-	-	-	Not used

1. The "Clocks" column specifies the associated clock which need to be running for the event to propagate.

Table 6-5. CTI 3 Input Assignments

Trigger input	Debug resource name	Clock[1]	Channel function	SYNC (debug_apb_clk)	Active	ack	Comment
0	ipi_int_epit_oc	ipg_clk_epit1	TRACING	Yes	High	No	Enhanced Periodic Interrupt Timer (EPIT)
1	ipi_int_ipu_func	hsp_clk	debug	Yes	High	No	IPU general interrupt
2	ipi_int_ipu_err	hsp_clk	debug	Yes	High	No	Image Processing Unit (IPU) error interrupt
3	ipu_sdma_event	hsp_clk	debug	Yes	High	No	IPU sdma event
4	gpu_use_bufid	ack_gpu	debug	Yes	High	No	GPU - OR output of the bus 3 bits
5	gpu_int_b	ack_gpu	debug	Yes	Low	No	GPU genral purpose interrupt
6	vpu_idle	cclk	Trace	Yes	High	No	VPU Idle
7	vpu_underrun	cclk	debug	Yes	High	No	VPU Underrun

1. The "Clocks" column specifies the associated clock which need to be running for the event to propagate.

Table 6-6. CTI 3 Output Assignments

Trigger input	Debug resource name	Clock	Channel function	SYNC (debug_apb_clk)	Active	ack	Comment
0	-	-	-	-	-	-	Not used
1	-	-	-	-	-	-	Not used

Table continues on the next page...

Table 6-6. CTI 3 Output Assignments (continued)

Trigger input	Debug resource name	Clock	Channel function	SYNC (debug_apb_clk)	Active	ack	Comment
2	-	-	-	-	-	-	Not used
3	-	-	-	-	-	-	Not used
4	-	-	-	-	-	-	Not used
6	-	-	-	-	-	-	Not used
7	-	-	-	-	-	-	Not used

6.3.4.3 Extended CTI (CTI Wrapper)

This wrapper consists of additional logic for each input and each output trigger to implement several features that can be selected or enabled with hard tie-off's on the block's boundary. It is combined with ARM's CTI to form the CTI_Extended block. A functional diagram of the input wrapper is shown in Figure 6-3. A summary of the extended input features implemented by logic in the wrapper is as follows:

- Invert the incoming signal
  - The ARM CTI assumes all inputs are active-high. An active-low input must first be inverted before going into the ARM CTI
- Sample the incoming trigger signal with a clock synchronous to the incoming signal
  - Samples the incoming trigger with it's own clock
- Convert the incoming signal to a pulse
  - Used when an input trigger has a long assertion time. Destinations may be sensitive to a long assertion time and may see it as multiple input triggers.
- Implement holding logic to insure that the ARM CTI captures and acknowledges the incoming signal. The holding logic clock must be synchronous to the clock that generated the input trigger.
- Provides a synchronized or asynchronous acknowledge signal back to the origin of the trigger.
  - If the asynchronous trigger source requires an ACK signal back from the CTI, but a clock from that source is not available to the CTI, the source must first synchronize the received ACK before using it in the trigger source logic.

6.3.5 CoreSight Trace Port Interface (TPIU)

The TPIU is one of the CoreSight trace sink components it acts as a bridge between the on-chip trace data to a data stream that is then driven out the trace port. ATB interface is used by the TPIU accepts trace data from a trace source, either direct from a trace source or using a Trace Funnel. The TPIU has 32 bit connected to the Chip pad. The APB interface is the programming interface for the TPIU configuration.

6.4 Cortex A8 Core and Platform

Cortex A8 Debug architecture includes support for TrustZone and CoreSight. The memory-mapped external debug interface replaces the coprocessor interface defined in the previous version of the Debug architecture. A full access to the processor debug capability available by Cortex A8 debug register map through the *Advanced Peripheral Bus* (APB) slave port. The core include Processor Debug Unit allow: stop program execution, examine and alter processor and coprocessor state, examine and alter memory and input/output peripheral state and restart the processor core.

6.4.1 Cortex A8 Core Debug Support Features

- CoreSight Embedded Trace Macro (ETM11) - trace generator for the Cortex A8 core
- Support for a TrustZone-related 3-level debug scheme:
  - Debug everywhere
  - Debug in Non-Secure privileged and user, and Secure user
  - Debug in Non-Secure only
- Embedded ICE-RT logic
  - Support for both monitor-mode and halt-mode debugging.
  - Core run/halt control, debug status/control
  - Breakpoint/watch point control
  - Core- and memory-mapped resource examination/modification
- Data communication channel between ARM Core and host debugger via JTAG
- PMU - Performance Metrics Unit used for system profiling and debug.
- CP15 register for debugging the MMU, I & D L1 cache, and TLB
- EVTMON - L2 Event Monitor - Supports debug and profiling of the ARM's L2 cache activity

6.4.2 Embedded Cross Trigger Interface

Please refer to the Embedded Cross Trigger section of this chapter for detailed information about the Core and platform interface to that subsystem.

6.4.3 Additional Platform Debug Functionality

- CoreSight Embedded Trace Buffer (ETB11) - 4 Kbyte RAM array to be used for on-chip capture of trace data output from the ETM11
- ATB Replicator to connect the trace data to TPIU (Trace Port Interface) and ETB (Embedded Trace Buffer).
- Debug Visibility - select critical signals routed to the I/O pads as alternate outputs for external visibility

6.5 Smart Direct Memory Access (SDMA) Core

The Smart Direct Memory Access (SDMA) is a dedicated, programmable DMA engine. It is an integration of a 32-bit RISC core and DMA-specific hardware, and includes ports for the ARM platform domain and a peripheral domain, along with a burst-capable port for direct external memory access. The SDMA and it's integration in i.MX53 is unchanged from previous SoCs.

The main SDMA debug features:

- OnCE - On Chip Emulator, provides the following capabilities:
  - SDMA core control - run/halt/single-step
  - SDMA core register/memory-map access
  - Event detection, watch points, and hardware breakpoints
  - Real time buffer and PC trace buffer capability
- Trace buffer
  - Contains information to identify the 32 last changes of flow detected during a program execution
- Context dump
  - include information about all the channel dump activity
  - Current contents of SDMA RAM
- ROMPATCH

6.5.1 SDMA On Chip Emulation Module (OnCE) Feature Summary

The SDMA debug features are primarily defined by the OnCE portion of it's design, which are summarized as follows:

- Memory And Register Access - dedicated logic enables user-access to SDMA memory and register locations. These accesses are supported only when the processor is in debug mode.
- Event Detection Unit - watches signals from the data memory bus (DMBus) which is used by the RISC core to access its RAM, ROM, and memory-mapped registers
- Watch points - one output signal is available to watch event matching conditions at the chip level. Match conditions are defined by programming memory-mapped registers.
- Hardware Breakpoint - a counter is decremented after an event detection. A debug request is sent to the SDMA core only when the counter reaches the value of zero. It is possible to program the initial value of the counter or to disable the use of the counter if a debug request must be generated after each event detection.
- Real Time Buffer - The Real Time Buffer Register (RTB) is a single 32-bit memory-mapped register which can be accessed as a regular memory location during program execution. It is used to store and retrieve run time information without putting the SDMA in debug mode. Each write to this register causes an event. This register is, in fact, located in the OnCE. Executing through JTAG, a buffer command exports the content of this register through the JTAG port.
- Core Control (Core Status / Single Stepping) - Commands are provided to monitor and control processor activity. The commands can halt the core, rerun the core from another address location, and get processor status.
- Trace Buffer - a 32x32 buffer which records the last 32 changes of flow during program execution. The buffer stores data in a modulo fashion (that is, the 33rd instruction change replaces the 1st). Captured trace information is retrieved via reads to the Trace Buffer Register.

6.5.2 Other SDMA Debug Functionality

- Core Trace - basic core trace capability is available through debug visibility functionality only. ETM/Nexus trace capability does not exist.
- ROM Patch - can be accomplished by manipulating the CHN0ADDR register through JTAG or via the ARM platform's ability to write to SDMA OnCE registers. This must be done right after reset and before the SDMA core is enabled to begin processing events.
- Additional debug control/status interaction with the SJC

- SJC-controlled Debug Request
- SJC-readable Debug Acknowledge (in debug mode)
- Debug clock control - allows SJC to force clocks on for debug purposes
- Debug core state (SDMA RISC Core State) - 4 bits accessible from the SJC via JTAG

- Debug Visibility - observable outputs as alternate (programmable) output functions of I/O pins
  - Debug Request, Debug Mode
  - Debug Yield
  - Debug Event Channel[5:0] (indicates requesting event or channel being processed)
  - Debug PC [13:0] (for SDMA core trace)
  - Debug core state [3:0]
  - Debug Real-time Buffer write
  - Debug Addr/Data match
  - Debug bus error
  - Debug bus device
  - Debug bus r/w
  - Debug Event Channels[7:0]
  - Debug Core Run (active when core is running)

6.5.3 Embedded Cross Trigger Interface (SDMA)

Please refer to the Embedded Cross Trigger section of this chapter for detailed information about the SDMA interface to that subsystem.

6.6 External Memory Controller (EXTMC)

The second-generation External Memory Controller (EXTMC) contains an arbitration profiling unit. This can be used to monitor and profile the dynamic arbitration behavior during operation. Several internal signals would be routed out by EXTMC debug unit to give SoC the capability to track the internal logic mainly for the arbitration mechanism and memory served accesses. In addition, the debug unit gives the ability to profile a master connected to EXTMC and get the bus performance of the arbitration.

Signals routed to 51-bit ipp_do_emi_debug bus are routed to the PAD.

Profiling unit include

- Register that saves the maximal time that a selected master was pending without getting the bus - MDSR0

- The maximum value of the dynamic priority which was selected - MDSR0
- Counters to record the number of access that each master performed through each channel - MDSR2-5
- Counter of the total number of accesses a specific request (or type of request) has accessed the bus can be fast, slow or intr - MDSR6
- Register that sums of "time for bus" the time it took a specified request (or type of request) to get the bus - MDSR7
- Register that sums the time it took each access to get the bus (pending time) - MDSR8
- Ability to reset all the counters

The EXTMC profiling logic includes a START signal, which comes from the ECT. Thus, any definable ECT condition can be used to initiate profiling.

6.7 Debug Visibility - IOMUX

Certain predefined, internal signals can be viewed at the package pins. The pad logic for most pins includes an IOMUX, allowing that pad to be shared across multiple functions. Each pad has a primary function that is selected at reset. The alternate functions, such as displaying the state of an internal signal for debug purposes, is selected by reprogramming the IOMUX.

Table 6-7. Debug Visibility

Block	Signals
SDMA	debug_mode, debug_bus_error, debug_bus_device[4:0], debug_bus_rwb, debug_matched_dmbus, debug_rtbuffer_write, debug_evt_chn_lines[7:0]
EXTMC	ipp_do_emi_debug[50:0]
IPU	ipu_diagbus[15:0]
GPU3D	SYS_GC_debug_out[16:0]
TPIU	TRACEDATA[31:0]

The ARM Platform Chapter of the i.MX53 Block Guide contains a complete listing of debug visibility signals and their assigned pins.

6.8 ARM Platform Peripherals

6.8.1 Image Processing Unit (IPU)

The Image Processing Unit (IPU) block includes a debug unit that allows routing of its signals to the SoC level. The signals muxing to be routed to top are controlled by DP Debug Control register (DP_DEBUG_CNT) and are connected to 16 bit diagnostic bus (ipu_diagbus). The IPU also include status register (DP_DEBUG_STAT) that can be used for debug.

The IPU can enter the system into debug mode by the cross trigger system.Three of the IPU signals are input to ETC: IPU general interrupt, IPU error interrupt and IPU end of frame; those signals can trigger the system to enter debug mode.

More details on signal muxing and IPU debug registers can be found in the IPU chapter.

6.8.2 Video Processing Unit (VPU)

The Video Processing Unit (VPU) can enter the system into debug mode by the cross trigger system. Three of the VPU signals are input to ETC: VPU idle, VPU interrupt and VPU underrun; those signals can trigger the system to enter debug mode.

The VPU internal register can be approach the AHB bus for further analysis.

6.8.3 GPU3D

The GPU3D support propagating debug information from the core over a 32b wide debug bus. The debug outputs are directed to only 16 visible ports. GPU3D include debug control bus SYS_GC_gpio[15:0] that selects which debug signals are exposed on the debug out bus GC_SYS_debug_out[31:0].

GPU3D can enter the system into debug mode by the cross trigger system. 2 of GPU3D signals are input to ETC: GPU3D error interrupt and GPU3D end of frame; those signals can trigger the system to enter debug mode.

6.9 Supported Tools

i.MX53 support RealView ARM Debugger, the debugger should be connect to i.MX53 from host by RealView ICE protocol converter.

6.10 Interrupt Visibility

Similar to debug visibility, i.MX53 includes multiplexrs that allow users to view selected internal interrupts. Up to five pads support this as alternate pad output functions, allowing the user to view up to five different interrupt sources simultaneously. These five signals also route to ECT inputs, allowing them to generate ECT trigger events as desired.

6.11 Miscellaneous

6.11.1 SoC-level Bus Trace

There is no SoC-level bus trace capability on i.MX53.

6.11.2 Clock / Reset / Power

The interaction between the debug system and the system clock, reset, and power control blocks includes enhancements to insure that status of critical system components can be monitored at all times, and that a debugger can insure that all critical clocks and power are enabled when needed. Specifically, a master signal coming from an SJC control bit and going to the Clock/PLL logic is implemented for each major clock domain. The same is done for each power domain. Historically, JTAG based operations have not been possible under certain system conditions. For i.MX53, the debugger must have the ability to determine the status of all critical system elements and to override any condition that would prevent the desired debug resources from operating properly.

6.11.3 RVI connection Settings

The RVI connection settings are shown in the figures below.

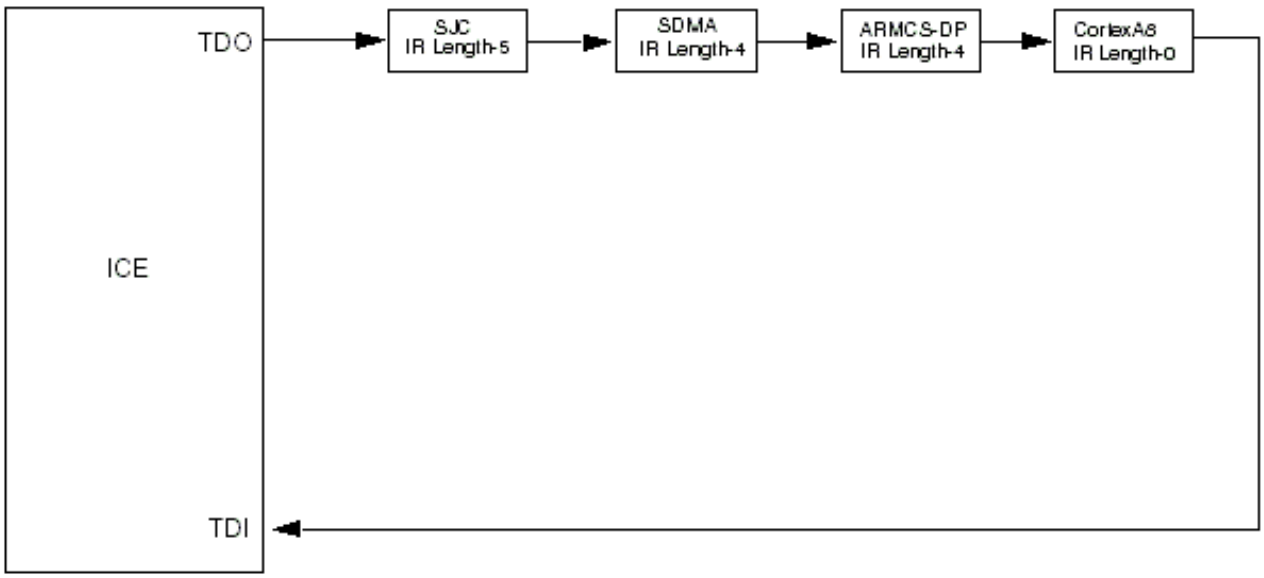


Figure 6-4. JTAG chain configuration

Table 6-8. CoreSight-A8 Base Address Configuration

Item	Value
CoreSight AP index	0x00000001
CoreSight base address	0xC0008000
Code Sequences Enabled	--
Code Sequence Address	0x0007FF80
Code Sequence Size (bytes)	0x00000080
Code Sequence Timeout (ms)	100
Bypass memory protection when in debug	True/False
Clear breakpoint hardware on connect	TrueFalse
Unwind vector when halt on SWI	True/False
Ignore debug priviledge errors when starting core	--
JTAG timeouts enabled	True/False

6.12 System Debug SJC Memory Map/Register Definition

The following a summary of the SJC registers used in i.MX53. This summary takes precedence is any discrepancy between the information found here, and the information in the SJC register summary arises.

SJC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0000_0000	General Purpose Unsecured Status Register 1 (SJC_GPUSR1)	32	R	0000_0000h	6.12.1/438
0000_0001	General Purpose Unsecured Status Register 2 (SJC_GPUSR2)	32	R	0000_0000h	6.12.2/440
0000_0002	General Purpose Unsecured Status Register 3 (SJC_GPUSR3)	32	R	0000_0000h	6.12.3/441
0000_0003	General Purpose Secured Status Register (SJC_GPSSR)	32	R	0000_0000h	6.12.4/442
0000_0004	Debug Control Register (Secured) (SJC_DCR)	32	R/W	0000_0000h	6.12.5/442
0000_0005	Security Status Register (SJC_SSR)	32	R	0000_0000h	6.12.6/444
0000_0006	Charge Pump Configuration Register (SJC_CPCR)	32	R	0000_0000h	6.12.7/446
0000_0007	General Purpose Clocks Control Register (SJC_GPCCR)	32	R/W	0000_0000h	6.12.8/446
0000_0008	PLL Bypass Register (SJC_PLLBR)	32	R/W	0000_0000h	6.12.9/448
0000_0009	General Purpose Unsecured Control Register 1 n (SJC_GPUCR1)	32	R/W	0002_0000h	6.12.10/ 450
0000_000A	General Purpose Unsecured Control Register 2 n (SJC_GPUCR2)	32	R/W	0511_0981h	6.12.11/ 452
0000_000B	General Purpose Unsecured Control Register 3 n (SJC_GPUCR3)	32	R/W	0000_0000h	6.12.12/ 453
0000_000C	General Purpose Secured Control Register (SJC_GPSCR)	32	R/W	0000_0000h	6.12.13/ 456
0000_000D	Test Register (SJC_TESTREG)	16	R/W	0000h	6.12.14/ 456
0000_000E	Serial Access Select Register (SJC_SASR)	32	R/W	0000_0000h	6.12.15/ 457
0000_000F	BIST Configuration Register 1 (SJC_BISTCR1)	32	R/W	0000_0000h	6.12.16/ 457
0000_0010	BIST Configuration Register 2 (SJC_BISTCR2)	32	R/W	0000_0000h	6.12.17/ 459
0000_0011	BIST Configuration Register 3 (SJC_BISTCR3)	32	R/W	0000_0000h	6.12.18/ 460
0000_0012	BIST Configuration Register 4 n (SJC_BISTCR4)	32	R/W	0000_0000h	6.12.19/ 461
0000_0013	BIST Configuration Register 5 (SJC_BISTCR5)	32	R/W	0000_0000h	6.12.20/ 463
0000_0014	Bist Configuration Register 6 (SJC_BISTCR6)	32	R/W	0000_0000h	6.12.21/ 464
0000_0015	Bist Configuration Register 7 (SJC_BISTCR7)	32	R/W	0000_0000h	6.12.22/ 464
0000_0016	Memory BIST Pass-Fail Register 1 (reserved for Test) (SJC_MBISTPASSR1)	32	R	0000_0000h	6.12.23/ 465

Table continues on the next page...

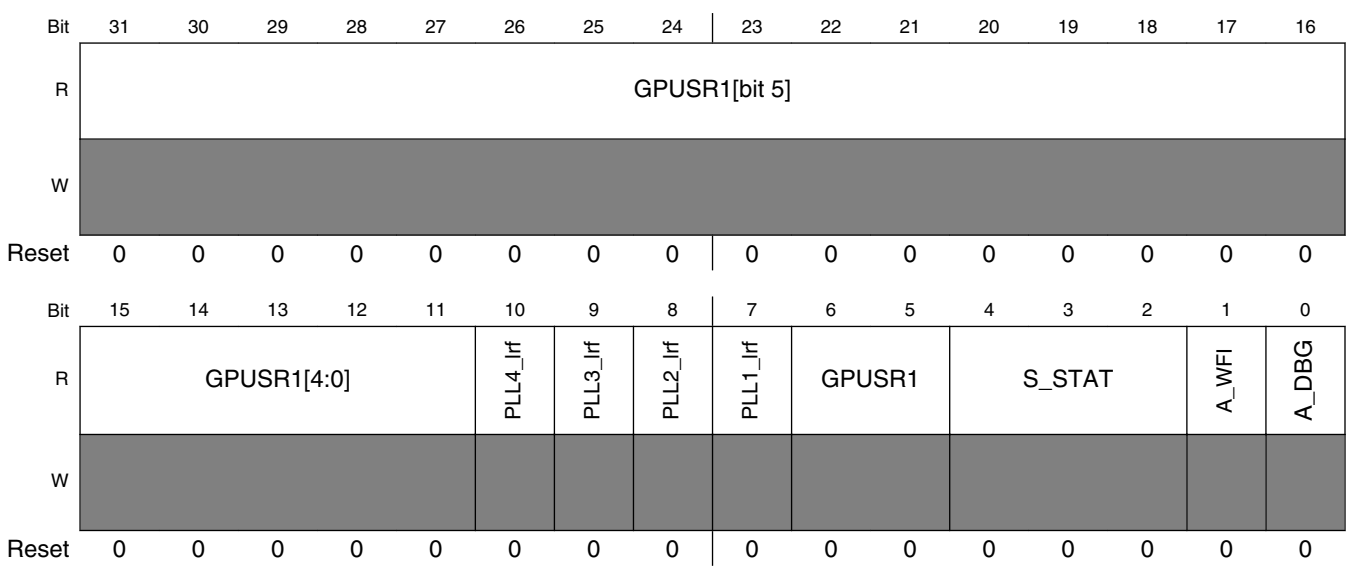
SJC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0000_0017	Memory BIST Pass-Fail Register 2 (SJC_MBISTPASSR2)	32	R	0000_0000h	6.12.24/ 467
0000_0018	Memory BIST Done Register 1 (SJC_MBISTDONER1)	32	R	0000_0000h	6.12.25/ 468
0000_0019	Memory BIST Done Register 2 (SJC_MBISTDONER2)	32	R	0000_0000h	6.12.26/ 470
0000_001B	Memory BIST Mask Register 2 (SJC_MBISTMASKR2)	16	R/W	0000h	6.12.27/ 470
0000_001C	BIST Pass-Fail Register (SJC_BISTPASSR)	32	R	0000_0000h	6.12.28/ 471
0000_001D	BIST Done Register (SJC_BISTDONER)	32	R	0000_0000h	6.12.29/ 471
0000_001E	Monitor BIST Select Register (SJC_MONBISTSELR)	32	R/W	0000_0000h	6.12.30/ 472
0000_001F	RVAL/WVAL Control Register (SJC_RWVALCR)	32	R/W	0000_0000h	6.12.31/ 472

6.12.1 General Purpose Unsecured Status Register 1 (SJC_GPUSR1)

The General Purpose Unsecured Status Register 1 is a read only register used to check the status of the different Cores and of the DPLL. The rest of its bits are for general purpose use.

Address: SJC_GPUSR1 is 0h base + 0h offset = 0000_0000h

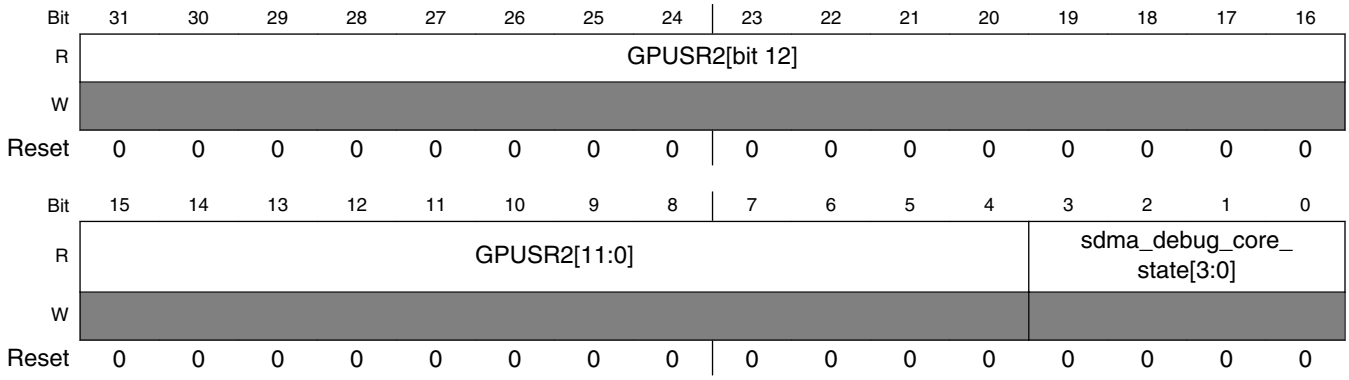


SJC_GPUSR1 field descriptions

Field	Description
31–11 GPUSR1	General Purpose Unsecured Status Register Register is used for testing and debug.
10 PLL4_lrf	Status bit indicating if dpll4 is locked 0 LL not locked 1 LL locked
9 PLL3_lrf	Status bit indicating if dpll3 is locked 0 LL not locked 1 LL locked
8 PLL2_lrf	Status bit indicating if dpll2 is locked 0 LL not locked 1 LL locked
7 PLL1_lrf	Status bit indicating if dpll1 is locked 0 LL not locked 1 LL locked
6–5 GPUSR1	General Purpose Unsecured Status Register Register is used for testing and debug.
4–2 S_STAT	3 LSBits of SDMA core statusH.
1 A_WFI	ARM core wait-for interrupt bit Bit 1 is the ARM core standbywfi (stand by wait-for interrupt). When this bit is HIGH, ARM core is in wait for interrupt mode.
0 A_DBG	ARM core debug status bit Bit 0 is the ARM core DBGACK (debug acknowledge) DBGACK can be overwritten in the ARM core DCR to force a particular DBGACK value. Consequently interpretation of the DBGACK value is highly dependent on the debug sequence. When this bit is HIGH, ARM core is in debug.

6.12.2 General Purpose Unsecured Status Register 2 (SJC_GPUSR2)

Address: SJC_GPUSR2 is 0h base + 1h offset = 0000_0001h

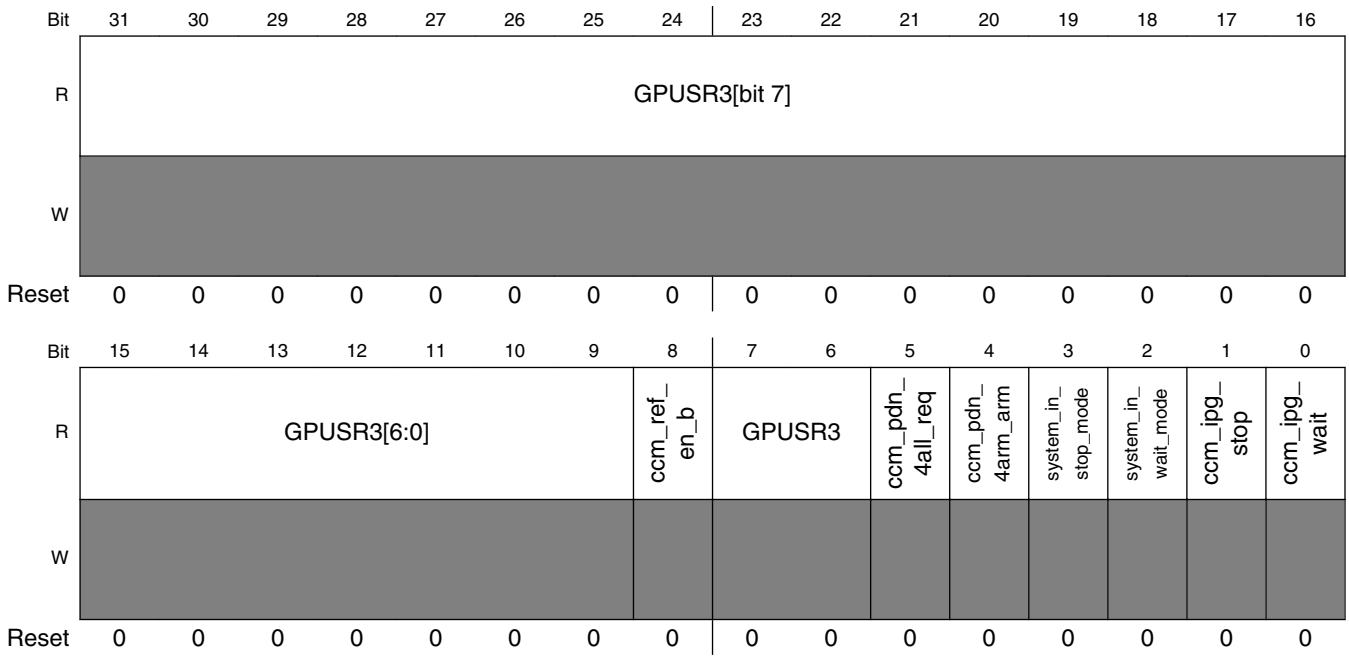


SJC_GPUSR2 field descriptions

Field	Description
31–4 GPUSR2	General Purpose Unsecured Status Register Register is used for testing and debug.
3–0 sdma_debug_ core_state[3:0]	SDMA core status (ipc_cstatus[3:0]). NOTE-The bits are read '0' unless visibility of SDMA Debug signals is enabled (SDMA Config Register, RTDOBS bit). Settings: 0000 PGM 0001 DATA 0010 Change Flow 0011 Set Wakeup 0100 DEBUG 0101 FU Bus 0110 SLEEP 0111 Save 1000 Program in Sleep 1001 Data in Sleep 1010 Change Of Flow in Sleep 1011 Error in Loop in Sleep 1100 Debug in Sleep 1101 Functional Unit in Sleep 1110 Sleep After Reset 1111 Restore See SDMA spec for details.

6.12.3 General Purpose Unsecured Status Register 3 (SJC_GPUSR3)

Address: SJC_GPUSR3 is 0h base + 2h offset = 0000_0002h



SJC_GPUSR3 field descriptions

Field	Description
31–9 GPUSR3	General Purpose Unsecured Status Register Register is used for testing and debug.
8 ccm_ref_en_b	Indicates if external oscillator is enabled. Settings 0 external oscillator disabled 1 external oscillator enabled
7–6 GPUSR3	
5 ccm_pdn_4all_req	Indicates a request to Global Power Controller (GPC) to power down all units Settings 0 no power down request to GPC 1 power down request sent to GPC
4 ccm_pdn_4arm_arm	Indicates a request to GPC to power down ARM Settings 0 no power down request to GPC 1 power down request sent to GPC
3 system_in_stop_mode	Indicates system is in stop mode Settings 0 system not in stop mode 1 system in stop mode
2 system_in_wait_mode	Indicates system is in wait mode Settings 0 system not in stop mode 1 system in stop mode
1 ccm_ipg_stop	Indicates CCM started stop entrance procedure Settings 0 CCM did not start stop entrance procedure 1 CCM started stop entrance procedure

Table continues on the next page...

SJC_GPUSR3 field descriptions (continued)

Field	Description
0 ccm_ipg_wait	Indicates CCM started wait entrance procedure Settings 0 CCM did not start wait entrance procedure 1 CCM started wait entrance procedure

6.12.4 General Purpose Secured Status Register (SJC_GPSSR)

The General Purpose Secured Status Register is a read only register used to check the status of the different critical information in the SoC. This register cannot be accessed in insecure modes.

Address: SJC_GPSSR is 0h base + 3h offset = 0000_0003h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPSSR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_GPSSR field descriptions

Field	Description
31–0 GPSSR	General Purpose Secured Status Register Register is used for testing and debug.

6.12.5 Debug Control Register (Secured) (SJC_DCR)

This register is used to control propagation of the debug request from the DE_B pad to the cores and debug signals from the embedded cross-trigger block to the DE_B pad. For more details see the SJC block guide.

Address: SJC_DCR is 0h base + 4h offset = 0000_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DCR[bit 9]															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DCR[8:0]									direct_arm_req_en	direct_sdma_req_en	direct_dsp_req_en	debug_obs	de_to_DSP	de_to_SDMA	de_to_ARM
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0				

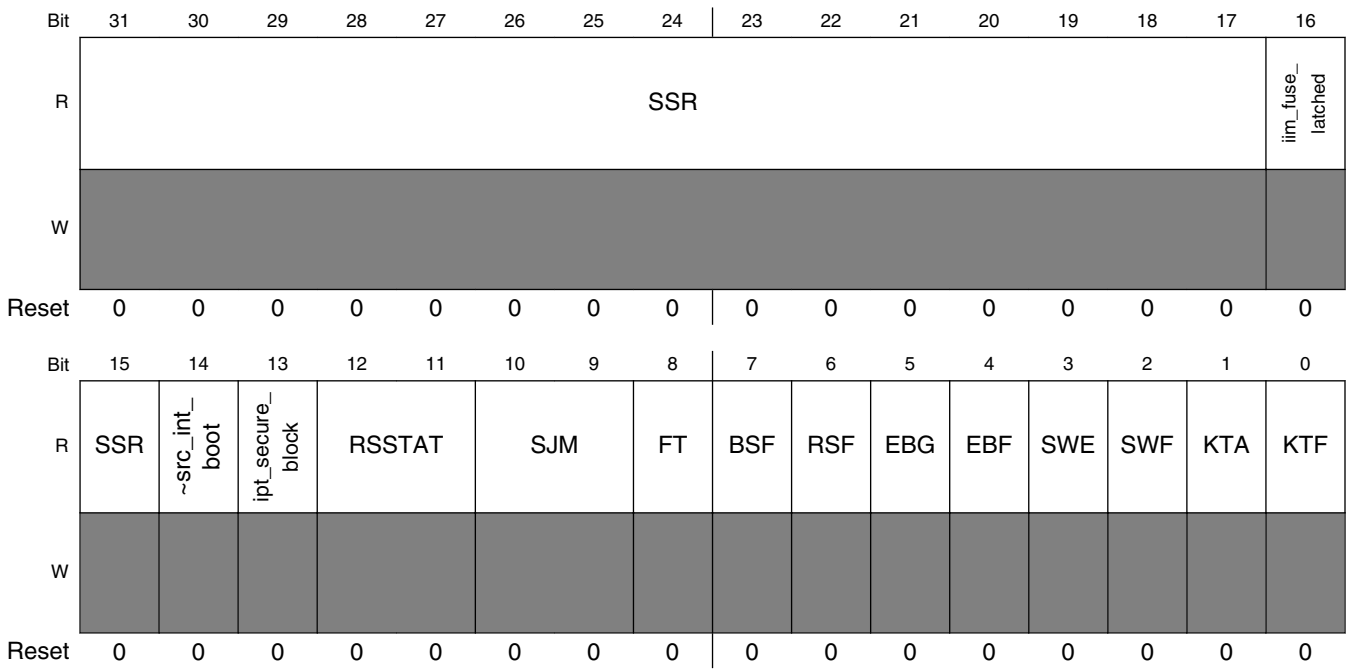
SJC_DCR field descriptions

Field	Description
31–7 DCR	General Purpose Unsecured Status Register Register is used for testing and debug.
6 direct_arm_req_en	Debug enable of the arm debug request This bit controls the propagation of debug request DE_B to the arm. Settings 0 - Disable propagation of system debug to (DE pin) to arm. 1 - Enable propagation of system debug to (DE pin) to arm. 0 Disable propagation of system debug to (DE pin) to arm. 1 Enable propagation of system debug to (DE pin) to arm.
5 direct_sdma_req_en	Debug enable of the SDMA debug request This bit controls the propagation of debug request DE_B to the SDMA. Settings 0 - Disable propagation of system debug to (DE pin) to SDMA. 1 - Enable propagation of system debug to (DE pin) to scam. 0 Disable propagation of system debug to (DE pin) to sdma. 1 Enable propagation of system debug to (DE pin) to sdma.
4 direct_dsp_req_en	Debug enable of the dsp debug request - This bit controls the propagation of debug request DE_B to the dsp. - No DSP in i.MX53 - so this bit is not connected outside. Settings 0 - Disable propagation of system debug to (DE pin) to dsp 1 - Enable propagation of system debug to (DE pin) to dsp
3 debug_obs	Debug observability This bit controls propagation of the system debug signal from Embedded Cross Trigger Block (ECT) to DE_B pad (can propagate debug acknowledge for example). Settings 0 - disable sjc_debug_b_oe 1 - enable sjc_debug_b_oe (when system is in debug) 0 Disable propagation of system debug to DE pin 1 Enable propagation of system debug to DE pin
2 de_to_DSP	DSP debug request input propagation - This bit controls propagation of the debug request to the DSP . - No DSP in i.MX53 - so this bit is not connected outside. Settings 0 - Disable propagation of debug request to DSP 1- Enable propagation of debug request to DSP
1 de_to_SDMA	SDMA debug request input propagation This bit controls propagation of the debug request to the SDMA. Settings 0 - Disable propagation of debug request to SDMA 1- Enable propagation of debug request to SDMA 0 Disable propagation of debug request to SDMA 1 Enable propagation of debug request to SDMA
0 de_to_ARM	ARM platform debug request input propagation This bit controls propagation of the debug request to the ARM platform. Settings 0 - Disable propagation of debug request to ARM platform 1- Enable propagation of debug request to ARM platform 0 Disable propagation of debug request to ARM 1 Enable propagation of debug request to ARM

6.12.6 Security Status Register (SJC_SSR)

This register is used to reflect IC security status and is accessible in all the security modes. The assumption is that the information contained in this register does not pose any security breach for the system.

Address: SJC_SSR is 0h base + 5h offset = 0000_0005h



SJC_SSR field descriptions

Field	Description
31–17 SSR	Security Status Register Register is used for testing and debug.
16 iim_fuse_latched	Indicates that fuse values have been latched. Settings 0 = fuses have not yet been latched 1 = fuses have been latched
15 SSR	Reserved
14 ~src_int_boot	System Reset Controller (SRC) internal boot
13 ipt_secure_block	Indicate if invasive/non-invasive debug can be done to the ARM Settings 0 =invasive and non-invasive debug cannot be done 1= invasive and non-invasive debug can be done
12–11 RSSTAT	SJC internal registers reserved - see SJC block guide 00 Response wasn't entered 01 Response was entered but not verified

Table continues on the next page...

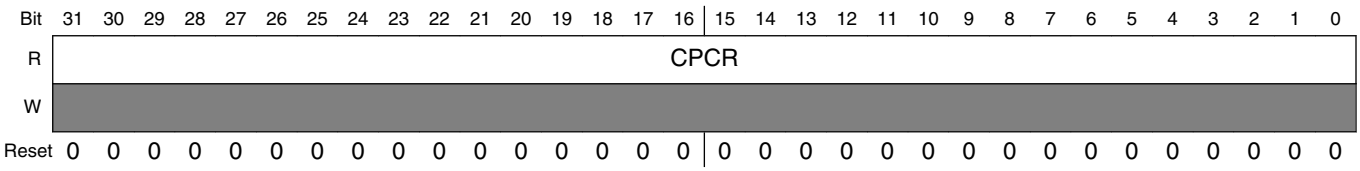
SJC_SSR field descriptions (continued)

Field	Description
	10 Response was entered and is incorrect 11 Response is correct
10–9 SJM	SJC internal registers reserved - see SJC block guide 00 No debug (#1) 01 Secure JTAG (#2) 10 JTAG enabled (#3) 11 Reserved
8 FT	SJC internal registers reserved - see SJC block guide 0 E-fuse technology 1 Laser fuse technology
7 BSF	SJC internal registers reserved - see SJC block guide 0 (intact) - no bypass 1 (burned) - bypass security
6 RSF	SJC internal registers reserved - see SJC block guide 0 (intact) - no re-enable 1 (burned) - secure JTAG is re-enabled
5 EBG	SJC internal registers reserved - see SJC block guide 1 granted 0 not granted
4 EBF	SJC internal registers reserved - see SJC block guide 0 (intact) - external boot is allowed 1 (burned) - external boot is disabled
3 SWE	SJC internal registers reserved - see SJC block guide 1 enabled 0 disabled
2 SWF	SJC internal registers reserved - see SJC block guide 0 (intact) - SW enable possible 1 (intact) - no SW enable possible
1 KTA	SJC internal registers reserved - see SJC block guide 1 active 0 not active
0 KTF	SJC internal registers reserved - see SJC block guide 0 (intact) - kill trace is never active 1 (burned) - kill trace functionality enabled

6.12.7 Charge Pump Configuration Register (SJC_CPCR)

This register is used to enable controlling charge pump configuration for Manufacturing Test via SJC. Note that the multiplexing is handled externally to SJC.

Address: SJC_CPCR is 0h base + 6h offset = 0000_0006h



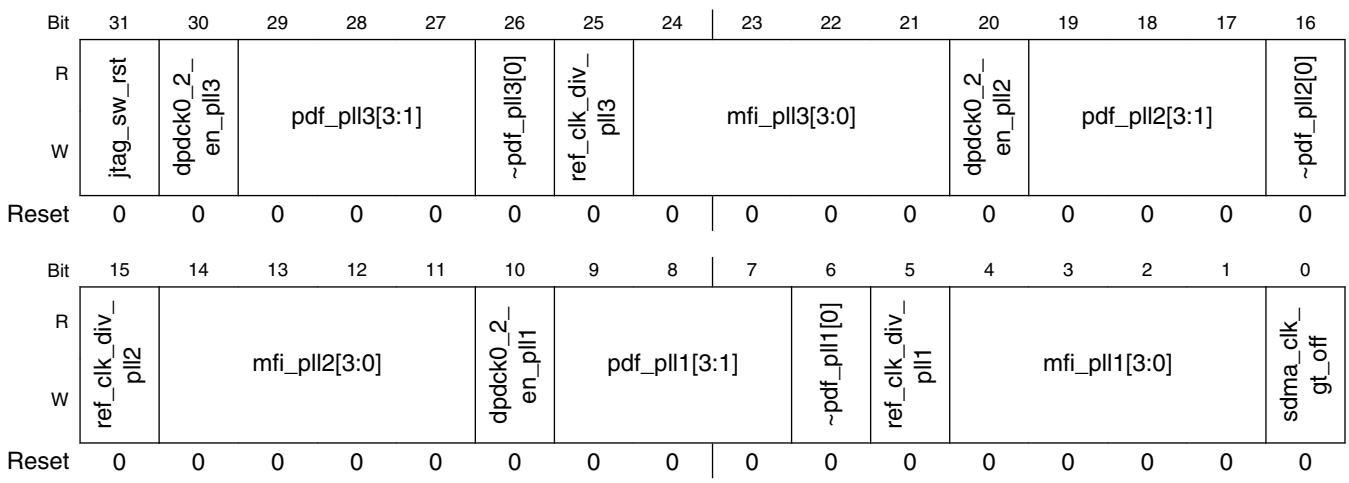
SJC_CPCR field descriptions

Field	Description
31–0 CPCR	Charge Pump Configuration Register-Register is used for testing and debug.

6.12.8 General Purpose Clocks Control Register (SJC_GPCCR)

This register is used to configure clock related modes in SoC. Those bits are directly connected to JTAG outputs.

Address: SJC_GPCCR is 0h base + 7h offset = 0000_0007h



SJC_GPCCR field descriptions

Field	Description
31 jtag_sw_rst	Provides software reset to SRC

Table continues on the next page...

SJC_GPCCR field descriptions (continued)

Field	Description
30 dpdck0_2_en_pll3	Enables double frequency dock for dpllip3 Settings: 0 double frequency clock disabled 1 double frequency clock enabled
29–27 pdf_pll3[3:1]	Pre division factor for dpllip3 Note: This is the actual value of the pre-division factor. (after the PLL adds 1) See DPLLC config spec for more details Settings: 0001 PDF = 1 0000 PDF = 2 0011 PDF = 3 0010 PDF = 4 0101 PDF = 5 0100 PDF = 6 0111 PDF = 7 0110 PDF = 8 1001 PDF = 9 1000 PDF = 10 1011 PDF = 11 1010 PDF = 12 1101 PDF = 13 1100 PDF = 14 1111 PDF = 15 1110 PDF = 16
26 ~pdf_pll3[0]	Pre division factor for dpllip3 Note: This is the actual value of the pre-division factor. (after the PLL adds 1) See DPLLC config spec for more details Settings: 0001 PDF = 1 0000 PDF = 2 0011 PDF = 3 0010 PDF = 4 0101 PDF = 5 0100 PDF = 6 0111 PDF = 7 0110 PDF = 8 1001 PDF = 9 1000 PDF = 10 1011 PDF = 11 1010 PDF = 12 1101 PDF = 13 1100 PDF = 14 1111 PDF = 15 1110 PDF = 16
25 ref_clk_div_pll3	Divides pll3 reference clock for dpllc3 Settings: 0 reference clock not divided by 2 1 reference clock divided by 2
24–21 mfi_pll3[3:0]	Multiplication factor for dpllc3 If MFI is written a value less than 5, it is defaulted to 5 This is also valid for ac scan mode Settings: 0000 MFI is 5 0001 MFI is 5 ... 0101 MFI is 5 0110 MFI is 6 0111 MFI is 7 ...
20 dpdck0_2_en_pll2	Ena ble double frequency clock for dpllc2 Settings: 0 double frequency clock disabled 1 double frequency clock enabled
19–17 pdf_pll2[3:1]	Pre division factor for dpllc2 Settings: See above bits 29:26
16 ~pdf_pll2[0]	Pre division factor for dpllc2 Settings: See above bits 29:26
15 ref_clk_div_pll2	Divides pll2 reference clock for dpll2 Settings: 0 reference clock not divided by 2 1 reference clock divided by 2
14–11 mfi_pll2[3:0]	Multiplication factor for dpllc2 If MFI is written a value less than 5, it is defaulted to 5 This is also valid for ac scan mode Settings: 0000 MFI is 5 0001 MFI is 5 ... 0101 MFI is 5 0110 MFI is 6 0111 MFI is 7 ...
10 dpdck0_2_en_pll1	Enables double frequency clock for dpllc1 Settings: 0 double frequency clock disabled 1 double frequency clock enabled
9–7 pdf_pll1[3:1]	Pre division factor for dpllc1 Settings: See above bits 29:26
6 ~pdf_pll1[0]	Pre division factor for dpllc1 Settings: See above bits 29:26
5 ref_clk_div_pll1	Divides pll1 reference clock for dpllc1 Settings: 0 reference clock not divided by 2 1 reference clock divided by 2
4–1 mfi_pll1[3:0]	Multiplication factor for dpllc1 If MFI is written a value less than 5, it is defaulted to 5 This is also valid for ac scan mode

Table continues on the next page...

SJC_GPCCR field descriptions (continued)

Field	Description
	Settings: 0000 MFI is 5 0001 MFI is 5 ... 0101 MFI is 5 0110 MFI is 6 0111 MFI is 7 ...
0 sdma_clk_gt_off	This bit will force the clock on the SDMA - see SJC guide for more details. Settings: 0 SDMA clock is not forced ON. 1 Force SDMA clock ON

6.12.9 PLL Bypass Register (SJC_PLLBR)

This register is used to enable the bypass functionality on internal clock signals for Manufacturing Test.

Address: SJC_PLLBR is 0h base + 8h offset = 0000_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	sjc_ac_scan_cpd_pll3[3:0]				sjc_ac_scan_cpd_pll2[3:0]				sjc_ac_scan_cpd_pll1[3:0]				pll4_bypass_en_io	pll4_bypass_alternative	pll3_bypass_alternative	pll2_bypass_alternative
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	pll1_bypass_alternative	sjc_crstrt_pll3	sjc_crstrt_pll2	sjc_crstrt_pll1	sjc_cpres_pll3	sjc_cpres_pll2	sjc_cpres_pll1	sjc_cpen_pll3	sjc_cpen_pll2	sjc_cpen_pll1	sjc_control_sel_pll3	sjc_control_sel_pll2	sjc_control_sel_pll1	pll3_bypass_en_io	pll2_bypass_en_io	pll1_bypass_en_io
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_PLLBR field descriptions

Field	Description
31–28 sjc_ac_scan_cpd_pll3[3:0]	Pre-division factor - 1 for PLL3 in ac_scan mode
27–24 sjc_ac_scan_cpd_pll2[3:0]	Pre-division factor - 1 for PLL2 in ac_scan mode
23–20 sjc_ac_scan_cpd_pll1[3:0]	Pre-division factor - 1 for PLL1 in ac_scan mode
19 pll4_bypass_en_io	Enables PLL4 bypass clock in IORING Settings: 0 bypass clock not enabled 1 bypass clock enabled
18 pll4_bypass_alternative	Selects the pad for the PLL4 bypass clock Settings: 0 main PAD 1 alternative PAD

Table continues on the next page...

SJC_PLLBR field descriptions (continued)

Field	Description
17 pll3_bypass_alternative	Selects the pad for the PLL3 bypass clock Settings: 0 main PAD 1 alternative PAD
16 pll2_bypass_alternative	Selects the pad for the PLL2 bypass clock Settings: 0 main PAD 1 alternative PAD
15 pll1_bypass_alternative	Selects the pad for the PLL1 bypass clock Settings: 0 main PAD 1 alternative PAD
14 sjc_crstrt_pll3	PLL3 receives crstrt if SJC control select =1
13 sjc_crstrt_pll2	PLL2 receives crstrt if SJC control select =1
12 sjc_crstrt_pll1	PLL1 receives crstrt if SJC control select =1
11 sjc_cpres_pll3	PLL3 receives CPRES if SJC control select = 1
10 sjc_cpres_pll2	PLL2 receives CPRES if SJC control select = 1
9 sjc_cpres_pll1	PLL1 receives CPRES if SJC control select = 1
8 sjc_cpen_pll3	PLL3 receives CPEN if SJC control select = 1
7 sjc_cpen_pll2	PLL2 receives CPEN if SJC control select = 1
6 sjc_cpen_pll1	PLL1 receives CPEN if SJC control select = 1
5 sjc_control_sel_pll3	SJC controls the PLL3 parameters Settings: 0 SJC does not control PLL parameters 1 SJC controls PLL parameters
4 sjc_control_sel_pll2	SJC controls the PLL2 parameters Settings: 0 SJC does not control PLL parameters 1 SJC controls PLL parameters
3 sjc_control_sel_pll1	SJC controls the PLL1 parameters Settings: 0 SJC does not control PLL parameters 1 SJC controls PLL parameters
2 pll3_bypass_en_io	Enables PLL3 bypass clock in IORING Settings: 0 bypass clock not enabled 1 bypass clock enabled
1 pll2_bypass_en_io	Enables PLL2 bypass clock in IORING Settings: 0 bypass clock not enabled 1 bypass clock enabled
0 pll1_bypass_en_io	Enables PLL1 bypass clock in IORING Settings: 0 bypass clock not enabled 1 bypass clock enabled

6.12.10 General Purpose Unsecured Control Register 1 n (SJC_GPUCR1)

These registers are used to configure WMSG IEEE1149.1 JTAG for special Test or debug modes. This register is not secured (accessible in all JTAG security modes). The bits of this register are directly connected to SJC outputs.

Address: SJC_GPUCR1 is 0h base + 9h offset = 0000_0009h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	efuse_prog_supply_gate	de_bhard_en	sjc_crstrt_pll3	sjc_cpres_pll4	sjc_cpen_pll4	ipt_sjc_test_mode[1:0]			ipt_sjc_lch_mode[2:0]		dpdck0_2en_pll4	pdf_pll4[3:1]			~pdf_pll4[0]	ref_clk_div_pll4
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	mfi_pll4[3:0]				GPUCR[11]	pll4_bypass_en_ccm	pll3_bypass_en_ccm	pll2_bypass_en_ccm	pll1_bypass_en_ccm	ipt_bipg_tdien	sjc_ac_scan_cpd_pll3[3:0]				usbphy_source_sel	sjc_control_sel_pll4
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_GPUCR1 field descriptions

Field	Description
31 efuse_prog_supply_gate	Supply gating for FUSEBOX programing Settings: 0 = fuse programing supply voltage is gated off to the FUSEBOX 1 = allow fuse programing
30 de_bhard_en	If set the PAD allocated for de_b is routed to SJC (without IOMUX programming). If clear the PAD is not allocated for SJC.
29 sjc_crstrt_pll3	PLL4 receives crstrt if SJC control select =1
28 sjc_cpres_pll4	PLL4 receives CPRES if SJC control select =1
27 sjc_cpen_pll4	PLL4 receives CPEN if SJC control select =1
26–25 ipt_sjc_test_mode[1:0]	These bits determine the type of burn_in mode Settings: 00 = BI_BIST 01 = BI_LCH_ALL 10 = BI_EDT_ALL
24–22 ipt_sjc_lch_mode[2:0]	Long Chain configuration Settings: 000 = ALL_LONCHAIN 001 = ARM_LONG_CHAIN 010 = APU_LONG_CHAIN 100 = MIXES_LONG_CHAIN 101 = BIPG_ONLY_LONG_CHAIN 110 = NO_LONG_CHAIN
21 dpdck0_2en_pll4	Enables double frequency clock for dpllc4 Settings: 0 = double frequency clock disabled 1= double frequency clock enabled

Table continues on the next page...

SJC_GPUCR1 field descriptions (continued)

Field	Description
20–18 pdf_pll4[3:1]	Pre division factor for dpllc4 Settings: See GPCCR reg
17 ~pdf_pll4[0]	Pre division factor for dpllc4 Settings: See GPCCR reg
16 ref_clk_div_pll4	Divides pll1 reference clock for dpllc4 Settings: 0 = reference clock not divided by 2 1 = reference clock divided by 2
15–12 mfi_pll4[3:0]	Multiplication factor for dpllc4 If MFI is written a value less than 5, it is defaulted to 5 This is also valid for ac scan mode Settings: 0000 = MFI is 5 0001 = MFI is 5 ... 0101 = MFI is 5 0110 = MFI is 6 0111 = MFI is 7 ...
11 GPUCR[11]	Reserved
10 pll4_bypass_en_ccm	Enables PLL4 bypass clock in CCM Settings: 0 = bypass clock not enabled 1 = bypass clock enabled
9 pll3_bypass_en_ccm	Enables PLL3 bypass clock in CCM Settings: 0 = bypass clock not enabled 1 = bypass clock enabled
8 pll2_bypass_en_ccm	Enables PLL2 bypass clock in CCM Settings: 0 = bypass clock not enabled 1 = bypass clock enabled
7 pll1_bypass_en_ccm	Enables PLL1 bypass clock in CCM Settings: 0 = bypass clock not enabled 1 = bypass clock enabled
6 ipt_bipg_tdien	This bit enables tdi for BURN IN mode Settings: 0 = tdi not enabled in BURN IN mode 1 = tdi enabled in BURN IN mode
5–2 sjc_ac_scan_cpd_pll3[3:0]	Pre divsion factor - 1 for PLL4 in ac_scan mode
1 usbphy_source_sel	Selects USB PHY input source Settings: 0 = USB 1 = external source through general purpose I/O modules (GPIO)
0 sjc_control_sel_pll4	SJC controls the PLL4 parameters Settings: 0 = SJC does not control PLL parameters 1 = SJC controls PLL parameters

6.12.11 General Purpose Unsecured Control Register 2 n (SJC_GPUCR2)

Address: SJC_GPUCR2 is 0h base + Ah offset = 0000_000Ah

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	sata_phyjtagalt7_hard_en	ddr_clk_mux_select	init_ssi_apm_clk_sel		init_periph_clk_sel	~init_periph_apm_sel[1:0]		init_nfc_podf[2:0]			init_emi_slow_podf_[2:0]			init_axi_b_podf[2:0]		
Reset	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	init_axi_a_podf[2:0]			init_ahb_podf[2:0]			init_ipg_podf[1:0]		init_perclk_podf[2:0]			init_perclk_pred2[2:0]			init_perclk_pred1[1:0]	
Reset	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1

SJC_GPUCR2 field descriptions

Field	Description
31 sata_phyjtagalt7_hard_en	This bit is used as an ALT7 hard enable for a separate JTAG port in SATA_PHY. Pads used for this port: DISP0_DAT20 - TDI DISP0_DAT21 - TDO DISP0_DAT22 - TCK DISP0_DAT23 - TMS Settings: 1 = SATA_PHY JTAG port is connected to I/O pads 0 = SATA_PHY JTAG port is connected to DAP
30 ddr_clk_mux_select	This select bit is used to collaborate en external clock during test in DDR Settings: 1 = ddr_clk_root is coming from an external source 0 = ddr_clk_root generated in CCM
29–28 init_ssi_apm_clk_sel	These bits connected to Clock Control Manager (CCM). They define initialization values to various clock MUX selects and clock dividers, thus defining initial system clocks frequencies. See CCM block guide for more details.
27 init_periph_clk_sel	This bit connected to Clock Control Manager (CCM). They define initialization values to various clock MUX selects and clock dividers, thus defining initial system clocks frequencies. See CCM block guide for more details.
26–25 ~init_periph_apm_sel[1:0]	These bits connected to Clock Control Manager (CCM). They define initialization values to various clock MUX selects and clock dividers, thus defining initial system clocks frequencies. See CCM block guide for more details.
24–22 init_nfc_podf[2:0]	These bits connected to Clock Control Manager (CCM). They define initialization values to various clock MUX selects and clock dividers, thus defining initial system clocks frequencies. See CCM block guide for more details.
21–19 init_emi_slow_podf_[2:0]	These bits connected to Clock Control Manager (CCM). They define initialization values to various clock MUX selects and clock dividers, thus defining initial system clocks frequencies. See CCM block guide for more details.

Table continues on the next page...

SJC_GPUCR2 field descriptions (continued)

Field	Description
18–16 init_axi_b_ podf[2:0]	These bits connected to Clock Control Manager (CCM). They define initialization values to various clock MUX selects and clock dividers, thus defining initial system clocks frequencies. See CCM block guide for more details.
15–13 init_axi_a_ podf[2:0]	These bits connected to Clock Control Manager (CCM). They define initialization values to various clock MUX selects and clock dividers, thus defining initial system clocks frequencies. See CCM block guide for more details.
12–10 init_ahb_ podf[2:0]	These bits connected to Clock Control Manager (CCM). They define initialization values to various clock MUX selects and clock dividers, thus defining initial system clocks frequencies. See CCM block guide for more details.
9–8 init_ipg_podf[1:0]	These bits connected to Clock Control Manager (CCM). They define initialization values to various clock MUX selects and clock dividers, thus defining initial system clocks frequencies. See CCM block guide for more details.
7–5 init_perclk_ podf[2:0]	These bits connected to Clock Control Manager (CCM). They define initialization values to various clock MUX selects and clock dividers, thus defining initial system clocks frequencies. See CCM block guide for more details.
4–2 init_perclk_ pred2[2:0]	These bits connected to Clock Control Manager (CCM). They define initialization values to various clock MUX selects and clock dividers, thus defining initial system clocks frequencies. See CCM block guide for more details.
1–0 init_perclk_ pred1[1:0]	These bits connected to Clock Control Manager (CCM). They define initialization values to various clock MUX selects and clock dividers, thus defining initial system clocks frequencies. See CCM block guide for more details.

6.12.12 General Purpose Unsecured Control Register 3 n (SJC_GPUCR3)

Address: SJC_GPUCR3 is 0h base + Bh offset = 0000_000Bh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPUCR3[31]	scb_or_ipt	sjc_ owr	sjc_ddr_input	ddr_sel[1:0]		sjc_odt_en	sjc_odt2	sjc_odt1	sjc_odt0	sjc_dse_en	sjc_dse2	sjc_dse1	sjc_dse0	sjc_ipt_io_tail_bypass	sjc_interrupttzic.intin_b[90]
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	arm_clk_off_on_lpm	gpio_18_alt7_hard_en	en_tester_control	smart_byoass_func	GPUCR[11]	sjc_ sre	sjc_oen_dis	sjc_ hys	sjc_ pke	sjc_ pue	sjc_ od	ahb_bypass	slow_bypass	axi_a_bypass	sjc_pus1	sjc_pus0
Reset	0	0	0	0	0	0	0	0	0	0	0	0				

SJC_GPUCR3 field descriptions

Field	Description
31 GPUCR3[31]	Reserved
30 scb_or_ipt	OR with scb._ipt_mode[3] and OR with scb._ipt_mode[1]
29 sjc_owr	SJC pad drive strength value
28 sjc_ddr_input	Select the ddr_input type; 1 - DDR2 0 - CMOS
27–26 ddr_sel[1:0]	ddr_sel[1:0] value - selects between different types 00 - DDR2_MODE1 01 - DDR2_MODE2 10 - LPDDR2 11 - RESERVED
25 sjc_odt_en	ODT enable Settings: 0 = On Die Termination value does not come from SJC 1= On Die Termination value comes from SJC
24 sjc_odt2	ODT value according to PAD description
23 sjc_odt1	ODT value according to PAD description
22 sjc_odt0	ODT value according to PAD description
21 sjc_dse_en	SJC pad strength enable Settings: 0 = Drive Strength value does not come from SJC 1= Drive Strength value comes from SJC
20 sjc_dse2	SJC pad drive strength value
19 sjc_dse1	SJC pad drive strength value
18 sjc_dse0	SJC pad drive strength value
17 sjc_ipt_io_tail_bypass	Bypasses the FF at the output to the pad Settings: 0 = Flip flop at pad output not bypassed 1 = Flip flop at pad output bypassed
16 sjc_interrupttzic.intin_b[90]	This interrupt allows the system to exit stop mode while the chip is in debug mode
15 arm_clk_off_on_lpm	Gates ARM clock Settings: 0 = ARM clock not gated 1 = ARM clock gated
14 gpio_18_alt7_hard_en	alt7_hard_en for IOMUXC cell GPIO_18. In this mode the system reset is reflected on GPIO_18 PAD. Settings: 0 = GPIO_18 pad is not hard_en in alt7 1 = GPIO_18 pad is hard_en in alt7
13 en_tester_control	Allows tester to receive determinism of reset sequence Settings: 0 = Tester cannot receive determinism of reset sequence 1 = Tester receives determinism of reset sequence

Table continues on the next page...

SJC_GPUCR3 field descriptions (continued)

Field	Description
12 smart_byoass_func	Memory repair bypass during Power On Reset (POR) - functional mode Settings: 0 = Memory Repair is not bypassed 1 = Memory Repair is bypassed
11 GPUCR[11]	Reserved
10 sjc_sre	SJC slew rate controller for IOMUXC Settings: 0 = Slow slew rate 1 = Fast slew rate
9 sjc_oen_dis	SJC pad output enable disable Settings: 1 = disable output enable 0 = do not disable output enable
8 sjc_hys	SJC pad hysteresis Settings: 1 = pad hysteresis 0 = no pas hysteresis
7 sjc_pke	SJC pad pull keeper enable Settings: 1 = pull up/keeper enable 0 = no pull up/keeper enable
6 sjc_pue	SJC pad pull up enable Settings: 1 = pull up 0 = keeper
5 sjc_od	SJC pad open drain Settings: 1 = open drain 0 = no open drain
4 ahb_bypass	Use bypass clock for AHB root in CCM Settings: 0 = regular 1 = bypass from PAD
3 slow_bypass	Use bypass clock for EMI_SLOW root in CCM Settings: 0 = regular 1 = bypass from PAD
2 axi_a_bypass	Use bypass clock for AXI_A root in CCM Settings: 0 = regular 1 = bypass from PAD
1 sjc_pus1	SJC pad pull up strength Settings: See IOMUXC block guide
0 sjc_pus0	SJC pad pull up strength Settings: See IOMUXC block guide

6.12.13 General Purpose Secured Control Register (SJC_GPSCR)

This register is used to configure WMSG IEEE1149.1 JTAG for special Test or debug modes. This register is secured (accessible in secure JTAG mode #3, #4 and #2 with response entered). Those bits are directly connected to SJC outputs.

Address: SJC_GPSCR is 0h base + Ch offset = 0000_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPSCR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_GPSCR field descriptions

Field	Description
31–0 GPSCR	General Purpose Control Register Reserved - Used for testing and debug.

6.12.14 Test Register (SJC_TESTREG)

This register is used to configure SoC Test modes for Manufacturing Test.

Address: SJC_TESTREG is 0h base + Dh offset = 0000_000Dh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	TESTREG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_TESTREG field descriptions

Field	Description
15–0 TESTREG	Reserved - Used for testing and debug. 0x0000 Normal operation 0x0001 Spare Test Mode #1 0x0002 Spare Test Mode #2 0xFFFF Spare Test Mode #15

6.12.15 Serial Access Select Register (SJC_SASR)

This register is used to select which of the SOC blocks are accessed serially (either PLL BIST or Latch divergence chain or ARM BIST, and so on) via JTAG access when using the "Serial access" IR instruction. Channels 7 and 8 can be accesses only in secure mode.

Address: SJC_SASR is 0h base + Eh offset = 0000_000Eh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-																										SASR[5:0]					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_SASR field descriptions

Field	Description
31–6 -	Reserved
5–0 SASR[5:0]	Select which of the modules is accessed serially Settings: h0 - bistcr1 selected module h1 - bistcr2 selected module h2 - bistcr3 selected module h4 - reserved h5 - ipt_so_lch h5-h31 - reserved

6.12.16 BIST Configuration Register 1 (SJC_BISTCR1)

These registers are used to configure GBIST engines - Asynchronous Sample Rate Converter (ASRC) ROM, Smart Direct Memory Access (SDMA) ROM and ARM platform ROM.

Address: SJC_BISTCR1 is 0h base + Fh offset = 0000_000Fh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BISTCR1[31:29]			sdma_rom_gbist_addr[9:0]										BISTCR1[18:17]		gbist_rowfast
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

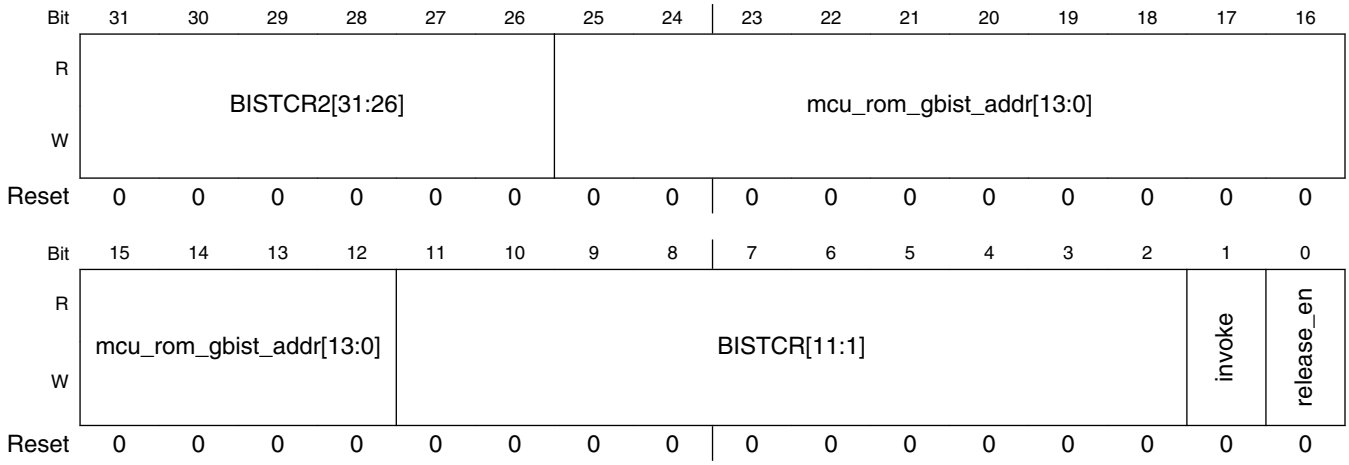
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	gbist_mode[2:0]			gbist_reset	BISTCR1[11:5]							asrc_gbist_select	mcu_gbist_select	sdma_gbist_select	invoke	release_en
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_BISTCR1 field descriptions

Field	Description
31–29 BISTCR1[31:29]	Reserved - Used for testing and debug.
28–19 sdma_rom_gbist_ addr[9:0]	SDMA ROM gbist end address
18–17 BISTCR1[18:17]	Reserved
16 gbist_rowfast	ASRC, SDMA and ARM platform gbist row fast mode Settings: See BIST guide for supported modes
15–13 gbist_mode[2:0]	ASRC, SDMA and ARM platform gbist configuration Settings: configuration See BIST guide for supported modes
12 gbist_reset	ASRC, SDMA and ARM platform gbist reset Settings: 0 = BIST clock disabled 1 = BIST clock enabled
11–5 BISTCR1[11:5]	Reserved
4 asrc_gbist_select	ASRC ROM gbist select Settings: 0 = BIST disabled 1 = BIST selected
3 mcu_gbist_select	ARM platform ROM gbist select Settings: 0 = BIST disabled 1 = BIST selected
2 sdma_gbist_ select	SDMA ROM gbist select Settings: 0 = BIST disabled 1 = BIST selected
1 invoke	Bist invoke - Invoke BIST1 engine
0 release_en	Release_en - Enable generating release signal for certain BIST modes when passing through RTI, IR = "serial access", and value of SASR corresponds to this register No use in i.MX53

6.12.17 BIST Configuration Register 2 (SJC_BISTCR2)

Address: SJC_BISTCR2 is 0h base + 10h offset = 0000_0010h



SJC_BISTCR2 field descriptions

Field	Description
31–26 BISTCR2[31:26]	Reserved - Used for testing and debug.
25–12 mcu_rom_gbist_ addr[13:0]	ARM platform ROM gbist end address
11–2 BISTCR[11:1]	Reserved
1 invoke	Bist2 invoke - Invoke BIST engine. No use in i.MX53
0 release_en	Release_en - Enable generating release signal for certain BIST modes when passing through RTI, IR = “serial access”, and value of SASR corresponds to this register. No Use in i.MX53

6.12.18 BIST Configuration Register 3 (SJC_BISTCR3)

Address: SJC_BISTCR3 is 0h base + 11h offset = 0000_0011h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BISTCR3[31:16]															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	asrc_rom_gbist_addr[10:0]											asrc_gbist_rom_array_sel		BISTCR3[2:1]		release_en
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_BISTCR3 field descriptions

Field	Description
31–16 BISTCR3[31:16]	Reserved - Used for testing and debug.
15–5 asrc_rom_gbist_addr[10:0]	ASRC ROM gbist end address
4–3 asrc_gbist_rom_array_sel	Selects which ROM is being tested 3: ASRC yrom (2048x24) 4: ASRC prom (512x48) Settings: 1: ROM selected 0: ROM not selected
2–1 BISTCR3[2:1]	Reserved
0 release_en	Release_en - Enable generating release signal for certain BIST modes when passing through RTI, IR = “serial access”, and value of SASR corresponds to this register. No Use in i.MX53

6.12.19 BIST Configuration Register 4 n (SJC_BISTCR4)

These registers are used to configure any of SoC BIST engines (VBIST, ARM BIST, ROM BIST, etc.). All the bits in the register except those specified are for general purpose use.

Address: SJC_BISTCR4 is 0h base + 12h offset = 0000_0012h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	bist_en_wrck	BISTCR4[30]	gpu2d_bist_clk_en	bist_ en_ sms_clk_tve	bist_en_ rom_36k_ gbist	~L2Data_bist_ clkdiv[2]	L2Data_bist_ clkdiv[1:0]		L2Cache_bist_ clkdiv[2]	~L2Cache_ bist_clkdiv[1:0]		ETB_bist_ clkdiv[2:1]		~ETB_bist_ clkdiv[0]	VPU_bist_ en_sms_clk	p_vl_bist_en
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	megamix_bist_ en_sms_clk	IPU_bist_ en_sms_clk	BISTCR4[13]	GPU_bist_ en_sms_clk	EMI_bist_ en_sms)clk	sel_sms_wdr	sel_sms_wir	sel_ipc_wdr	sel_ipc_wir	BISTCR4[6:1]						release_en
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_BISTCR4 field descriptions

Field	Description
31 bist_en_wrck	Enables STP functional clock wrck Settings: 0 = wrck clock disabled 1 = wrck clock enabled
30 BISTCR4[30]	Reserved
29 gpu2d_bist_clk_ en	Enables GPU2D Memory BIST clocks Settings: 0 = BIST clock disabled 1 = BIST clock enabled
28 bist_ en_sms_ clk_tve	Enables TVE Memory BIST clocks Settings: 0 = BIST clock disabled 1 = BIST clock enabled
27 bist_en_rom_ 36k_gbist	Enables ROM GBIST Settings: 0 = BIST disabled 1 = BIST enabled
26 ~L2Data_bist_ clkdiv[2]	Platform vbists clock dividers - L2_DATA, L2_CACHE and ETB memories
25–24 L2Data_bist_ clkdiv[1:0]	Platform vbists clock dividers - L2_DATA, L2_CACHE and ETB memories
23 L2Cache_bist_ clkdiv[2]	Platform vbists clock dividers - L2_DATA, L2_CACHE and ETB memories

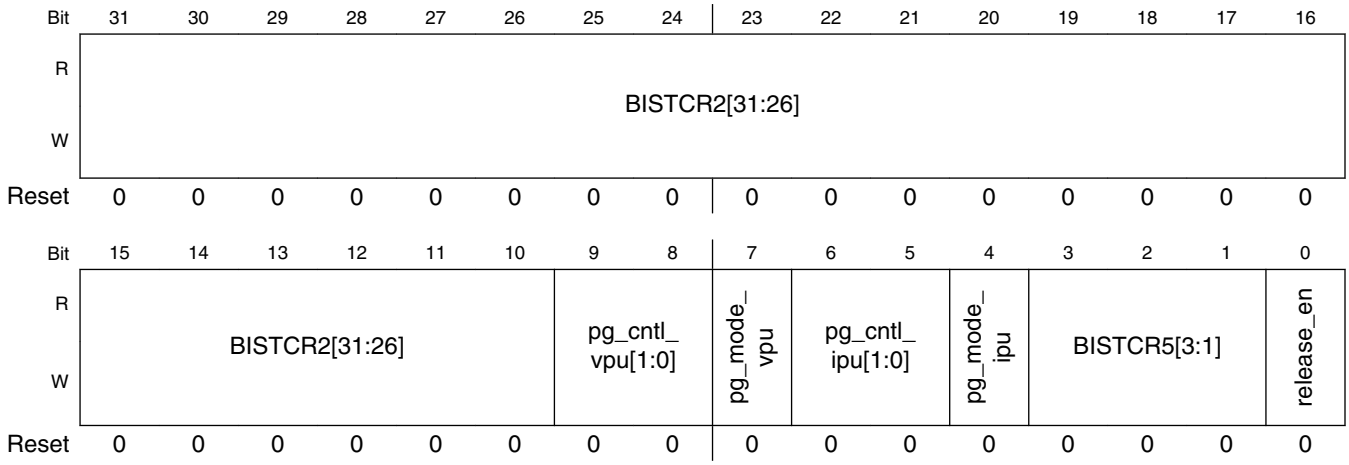
Table continues on the next page...

SJC_BISTCR4 field descriptions (continued)

Field	Description
22–21 ~L2Cache_bist_ clkdiv[1:0]	Platform vbists clock dividers - L2_DATA, L2_CACHE and ETB memories
20–19 ETB_bist_ clkdiv[2:1]	Platform vbists clock dividers - L2_DATA, L2_CACHE and ETB memories
18 ~ETB_bist_ clkdiv[0]	Platform vbists clock dividers - L2_DATA, L2_CACHE and ETB memories
17 VPU_bist_en_ sms_clk	Enables VPU Memory BIST clocks Settings: 0 = BIST clock disabled 1 = BIST clock enabled
16 p_vl_bist_en	Enables platform memories BIST clocks Settings: 0 = BIST clock disabled 1 = BIST clock enabled
15 megamix_bist_ en_sms_clk	Enables MEGAMIX memories BIST clocks Settings: 0 = BIST clock disabled 1 = BIST clock enabled
14 IPU_bist_en_ sms_clk	Enables IPU Memory BIST clocks Settings: 0 = BIST clock disabled 1 = BIST clock enabled
13 BISTCR4[13]	Reserved
12 GPU_bist_en_ sms_clk	Enables GPU Memory BIST clocks Settings: 0 = BIST clock disabled 1 = BIST clock enabled
11 EMI_bist_en_ sms)clk	Enables EXTMC Memory BIST clocks Settings: 0 = BIST clock disabled 1 = BIST clock enabled
10 sel_sms_wdr	Enables STP data register Settings: 0 = STP data register disabled 1 = STP data register enabled
9 sel_sms_wir	Enables STP instruction register Settings: 0 = STP instruction register disabled 1 = STP instruction register enabled
8 sel_ipc_wdr	Enables JPC data register Settings: 0 = JPC data register disabled 1 = JPC data register enabled
7 sel_ipc_wir	Enables JPC instruction register Settings: 0 = JPC instruction register disabled 1 = JPC instruction register enabled
6–1 BISTCR4[6:1]	Reserved
0 release_en	Release_en - Enable generating release signal for certain BIST modes when passing through RTI, IR = “serial access”, and value of SASR corresponds to this register. No Use in i.MX53

6.12.20 BIST Configuration Register 5 (SJC_BISTCR5)

Address: SJC_BISTCR5 is 0h base + 13h offset = 0000_0013h



SJC_BISTCR5 field descriptions

Field	Description
31–10 BISTCR2[31:26]	Reserved - Used for testing and debug.
9–8 pg_cntl_vpu[1:0]	Power gating test control for VPU used during unique scan mode to prevent scan from gating clocks Settings: pg_cntl_vpu[1] = iso cntl pg_cntl_vpu[0] = switch cntl
7 pg_mode_vpu	Power gating test mode for VPU Settings: 0 = power gating test mode off 1= power gating test mode on
6–5 pg_cntl_ipu[1:0]	Power gating test control for IPU (note: iso_cntl controls the power down in IPU memories and switch_cntl controls the voltage switch box going to IPU) Settings: pg_cntl_ipu[1] = iso cntl pg_cntl_ipu[0] = switch cntl
4 pg_mode_ipu	Power gating test mode for IPU used during unique scan mode to prevent scan from gating clocks Settings: 0 = power gating test mode off 1= power gating test mode on
3–1 BISTCR5[3:1]	Reserved
0 release_en	Release_en - Enable generating release signal for certain BIST modes when passing through RTI, IR = “serial access”, and value of SASR corresponds to this register. No Use in i.MX53

6.12.21 Bist Configuration Register 6 (SJC_BISTCR6)

These registers are used to configure any of SoC BIST engines.

Address: SJC_BISTCR6 is 0h base + 14h offset = 0000_0014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BISTCR6[31:1]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BISTCR6[31:1]															release_en
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_BISTCR6 field descriptions

Field	Description
31–1 BISTCR6[31:1]	Reserved
0 release_en	Release_en - Enable generating release signal for certain BIST modes when passing through RTI, IR = “serial access”, and value of SASR corresponds to this register. No Use in i.MX53

6.12.22 Bist Configuration Register 7 (SJC_BISTCR7)

Address: SJC_BISTCR7 is 0h base + 15h offset = 0000_0015h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BISTCR7[31:1]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BISTCR7[31:1]															release_en
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_BISTCR7 field descriptions

Field	Description
31–1 BISTCR7[31:1]	Reserved
0 release_en	Release_en - Enable generating release signal for certain BIST modes when passing through RTI, IR = “serial access”, and value of SASR corresponds to this register. No Use in i.MX53

6.12.23 Memory BIST Pass-Fail Register 1 (reserved for Test) (SJC_MBISTPASSR1)

These registers are made of all the pass-fail flags of all the Memory BIST engines used in the chip. One bit is used for each memory tested.

Address: SJC_MBISTPASSR1 is 0h base + 16h offset = 0000_0016h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MBISTPASSR1													ASRC_ GBIST_fail		GBIST_rom_ failed
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SDMA_rom_ bist_failed	TVE_asap_ fail_sms	GPU2D_ reg_fail_sms	GPU2D_ asap_fail_ sms	VPU_reg_ fail_sms	VPU_asap_ fail_sms	L2Data_fail_ sms	L2Cache_fail_ sms	ETB_fail_ sms	SIPMIX_ star_fail_sms	SIPMIX_ asap_fail_ sms	LPMIX_reg_ fail_sms	IPU_asap_ fail_sms	GPU_star_ fail_sms	GPU_reg_ fail_sms	EMI_fail_ sms
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_MBISTPASSR1 field descriptions

Field	Description
31–19 MBISTPASSR1	Reserved
18–17 ASRC_GBIST_ fail	Indicates the ASRC BIST test failed 18 - checks ASRC PROM rom failure 17 - checks ASRC YROM rom failure Settings: 0 = BIST test passed 1 = BIST test failed

Table continues on the next page...

SJC_MBISTPASSR1 field descriptions (continued)

Field	Description
16 GBIST_rom_failed	Indicates the ROM BIST test failed Settings: 0 = BIST test passed 1 = BIST test failed
15 SDMA_rom_bist_failed	Indicates the SDMA ROM BIST test failed Settings: 0 = BIST test passed 1 = BIST test failed
14 TVE_asap_fail_sms	Indicates the TVE ASAP Memory BIST test failed Settings: 0 = BIST test passed 1 = BIST test failed
13 GPU2D_reg_fail_sms	Indicates the GPU2D REG Memory BIST test failed Settings: 0 = BIST test passed 1 = BIST test failed
12 GPU2D_asap_fail_sms	Indicates the GPU2D ASAP Memory BIST test failed Settings: 0 = BIST test passed 1 = BIST test failed
11 VPU_reg_fail_sms	Indicates the VPU REG Memory BIST test failed Settings: 0 = BIST test passed 1 = BIST test failed
10 VPU_asap_fail_sms	Indicates the VPU ASAP Memory BIST test failed Settings: 0 = BIST test passed 1 = BIST test failed
9 L2Data_fail_sms	Indicates the L2DATA Memory BIST test failed Settings: 0 = BIST test passed 1 = BIST test failed
8 L2Cache_fail_sms	Indicates the L2CACHE Memory BIST test failed Settings: 0 = BIST test passed 1 = BIST test failed
7 ETB_fail_sms	Indicates the ETB Memory BIST test failed Settings: 0 = BIST test passed 1 = BIST test failed
6 SIPMIX_star_fail_sms	Indicates the SIPMIX STAR Memory BIST test failed Settings: 0 = BIST test passed 1 = BIST test failed
5 SIPMIX_asap_fail_sms	Indicates the SIPMIX ASAP Memory BIST test failed Settings: 0 = BIST test passed 1 = BIST test failed
4 LPMIX_reg_fail_sms	Indicates the LPMIX REG Memory BIST test failed Settings: 0 = BIST test passed 1 = BIST test failed
3 IPU_asap_fail_sms	Indicates the IPU ASAP Memory BIST test failed Settings: 0 = BIST test passed 1 = BIST test failed
2 GPU_star_fail_sms	Indicates the GPU STAR Memory BIST test failed Settings: 0 = BIST test passed 1 = BIST test failed

Table continues on the next page...

SJC_MBISTPASSR1 field descriptions (continued)

Field	Description
1 GPU_reg_fail_sms	Indicates the GPU REG Memory BIST test failed Settings: 0 = BIST test passed 1 = BIST test failed
0 EMI_fail_sms	Indicates the EXTMC Memory BIST test failed Settings: 0 = BIST test passed 1 = BIST test failed

6.12.24 Memory BIST Pass-Fail Register 2 (SJC_MBISTPASSR2)

Address: SJC_MBISTPASSR2 is 0h base + 17h offset = 0000_0017h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MBISTPASSR2[31:0]																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

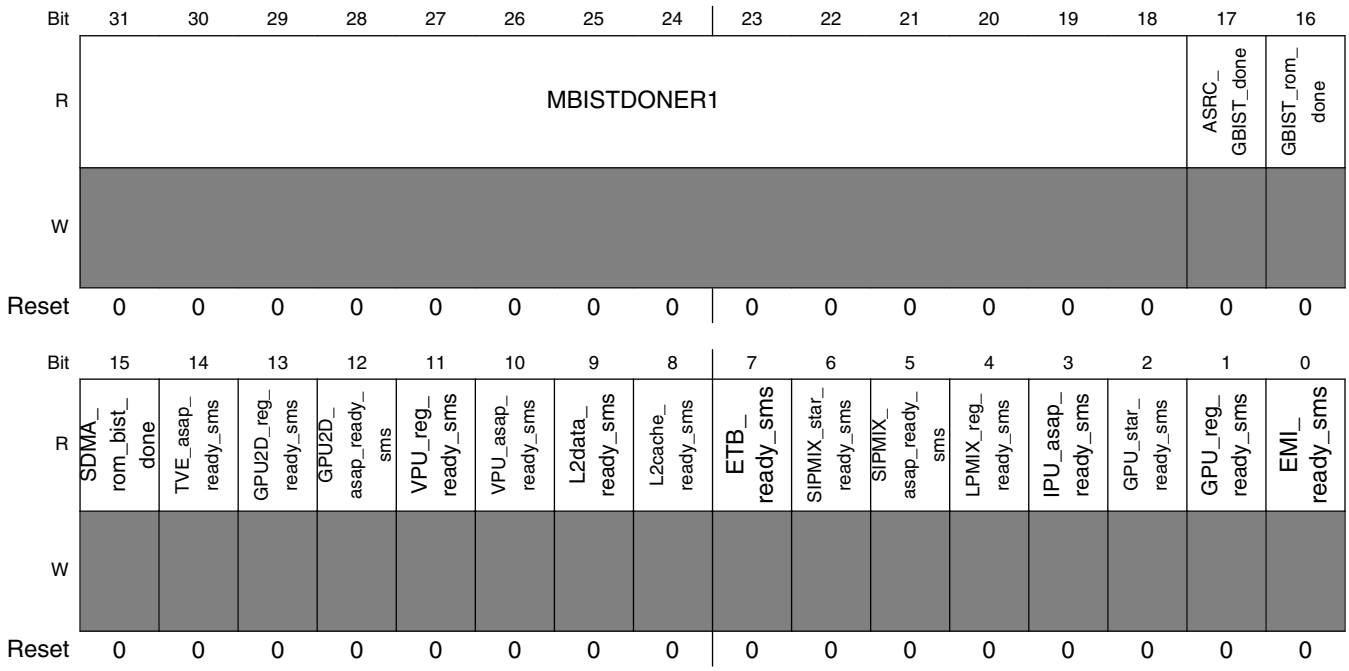
SJC_MBISTPASSR2 field descriptions

Field	Description
31–0 MBISTPASSR2[31:0]	Reserved

6.12.25 Memory BIST Done Register 1 (SJC_MBISTDONER1)

These registers are made of all the done flags of all the Memory BIST engines used in the chip. 1 bit is used for each Bist engine.

Address: SJC_MBISTDONER1 is 0h base + 18h offset = 0000_0018h



SJC_MBISTDONER1 field descriptions

Field	Description
31–18 MBISTDONER1	Reserved - Used for testing and debug.
17 ASRC_GBIST_done	Indicates the ASRC GBIST test done Settings: 0 = BIST test not done 1 = BIST test done
16 GBIST_rom_done	Indicates the ROM BIST test done Settings: 0 = BIST test not done 1 = BIST test done
15 SDMA_rom_bist_done	Indicates the SDMA ROM BIST test done Settings: 0 = BIST test not done 1 = BIST test done
14 TVE_asap_ready_sms	Indicates the TVE ASAP Memory BIST is ready Settings: 0 = BIST test not ready 1 = BIST test ready
13 GPU2D_reg_ready_sms	Indicates the GPU2D REG Memory BIST is ready Settings: 0 = BIST test not ready 1 = BIST test ready

Table continues on the next page...

SJC_MBISTDONER1 field descriptions (continued)

Field	Description
12 GPU2D_asap_ready_sms	Indicates the GPU2D ASAP Memory BIST is ready Settings: 0 = BIST test not ready 1 = BIST test ready
11 VPU_reg_ready_sms	Indicates the VPU REG Memory BIST is ready Settings: 0 = BIST test not ready 1 = BIST test ready
10 VPU_asap_ready_sms	Indicates the VPU ASAP Memory BIST is ready Settings: 0 = BIST test not ready 1 = BIST test ready
9 L2data_ready_sms	Indicates the L2DATA Memory BIST is ready Settings: 0 = BIST test not ready 1 = BIST test ready
8 L2cache_ready_sms	Indicates the L2CACHE Memory BIST is ready Settings: 0 = BIST test not ready 1 = BIST test ready
7 ETB_ready_sms	Indicates the ETB Memory BIST is ready Settings: 0 = BIST test not ready 1 = BIST test ready
6 SIPMIX_star_ready_sms	Indicates the SIPMIX STAR Memory BIST is ready Settings: 0 = BIST test not ready 1 = BIST test ready
5 SIPMIX_asap_ready_sms	Indicates the SIPMIX ASAP Memory BIST is ready Settings: 0 = BIST test not ready 1 = BIST test ready
4 LPMIX_reg_ready_sms	Indicates the LPMIX REG Memory BIST is ready Settings: 0 = BIST test not ready 1 = BIST test ready
3 IPU_asap_ready_sms	Indicates the IPU ASAP Memory BIST is ready Settings: 0 = BIST test not ready 1 = BIST test ready
2 GPU_star_ready_sms	Indicates the GPU STAR Memory BIST is ready Settings: 0 = BIST test not ready 1 = BIST test ready
1 GPU_reg_ready_sms	Indicates the GPU REG Memory BIST is ready Settings: 0 = BIST test not ready 1 = BIST test ready
0 EMI_ready_sms	Indicates the EXTMC Memory BIST is ready Settings: 0 = BIST test not ready 1 = BIST test ready

6.12.26 Memory BIST Done Register 2 (SJC_MBISTDONER2)

Address: SJC_MBISTDONER2 is 0h base + 19h offset = 0000_0019h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MBISTDONER2[31:0]																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_MBISTDONER2 field descriptions

Field	Description
31–0 MBISTDONER2[31:0]	Reserved

6.12.27 Memory BIST Mask Register 2 (SJC_MBISTMASKR2)

Bits in Memory BIST pass/fail, done, and mask registers correspond to each other, that is, 16th bit in Memory BIST pass/fail register 2 corresponds to 16th bit in Memory BIST Done register 2 and to 16th bit in Memory BIST Mask register 2.

Address: SJC_MBISTMASKR2 is 0h base + 1Bh offset = 0000_001Bh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	MBISTMASKR2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_MBISTMASKR2 field descriptions

Field	Description
15–0 MBISTMASKR2	Reserved

6.12.28 BIST Pass-Fail Register (SJC_BISTPASSR)

This register is made of all the pass-fail flags of all the non Memory BIST engines used in the chip - PLL BIST, and so on. These bits cannot be masked. One bit is used for each memory tested.

Address: SJC_BISTPASSR is 0h base + 1Ch offset = 0000_001Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BISTPASSR[31:0]																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_BISTPASSR field descriptions

Field	Description
31–0 BISTPASSR[31:0]	Reserved - Used for testing and debug.

6.12.29 BIST Done Register (SJC_BISTDONER)

This register is made of all the done flags of all the non Memory BIST engines used in the chip - PLL BIST, and so on. These bits cannot be masked. One bit is used for each memory tested.

Address: SJC_BISTDONER is 0h base + 1Dh offset = 0000_001Dh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BISTDONER[31:0]																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_BISTDONER field descriptions

Field	Description
31–0 BISTDONER[31:0]	Reserved - Used for testing and debug.

6.12.30 Monitor BIST Select Register (SJC_MONBISTSELR)

Address: SJC_MONBISTSELR is 0h base + 1Eh offset = 0000_001Eh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	-[bit 13]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-[12:0]													MONBISTSELR[2:0]		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_MONBISTSELR field descriptions

Field	Description
31–3 -	Reserved
2–0 MONBISTSELR[2:0]	Reserved

6.12.31 RVAL/WVAL Control Register (SJC_RWVALCR)

Address: SJC_RWVALCR is 0h base + 1Fh offset = 0000_001Fh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RWVALCR[31:0]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_RWVALCR field descriptions

Field	Description
31–0 RWVALCR[31:0]	Reserved

Chapter 7
System Boot

7.1 Introduction

The i.MX53 boot process begins at Power On Reset (POR) where the hardware reset logic forces the ARM core to begin execution starting from the on-chip boot ROM. Boot ROM code uses the statet of the internal register BOOT_MODE[1:0] as well as the state of various eFUSEs and/or GPIO settings to determine the boot flow behavior of the i.MX53.

The main features of the ROM include:

- Support for booting from various boot devices
- Serial Downloader support (UART and USB)
- Device Configuration Data (DCD)
- Digital signature based High Assurance Boot (HAB)

The i.MX53 boot ROM supports the following boot devices:

- NOR Flash
- NAND Flash
- OneNAND Flash
- SD/MMC
- Parallel ATA (PATA)/Serial ATA (SATA) HDD
- Serial ROM devices

In normal operation, the boot ROM uses the state of the BOOT_MODE register and eFUSEs to determine the boot device. For development purposes eFUSEs used to determine the boot device may be overridden using GPIO pin inputs.

Boot ROM code also allows the downloading of programs to be run on the i.MX53. An example is a provisioning program that can make further use of the serial connection to provision a boot device with a new image. Typically the provisioning program is

downloaded to internal RAM and allows the programming boot devices such as a NAND Flash. The ROM Serial Downloader uses either a high speed USB in non-stream mode or a UART connection.

The Device Configuration Data (DCD) feature allows boot ROM code to obtain IC configuration data from an external Program Image residing on the boot device. For example, DCD can be used to program the SDRAM controller for optimal settings improving the boot performance. DCD is restricted to memory areas and peripheral addresses that are considered essential for boot purposes.

A key feature of the i.MX53 BOOT ROM is the ability to perform a secure or High Assurance Boot (HAB). This is supported by the HAB security library which is a subcomponent of the MCIMX35 ROM code. HAB uses a combination of hardware and software together with a Public Key Infrastructure (PKI) protocol to protect the system from executing unauthorized programs. Before the HAB allows a user's image to execute, the image must be signed. The signing process is done during the image build process by the private key holder and the signatures are then included as part of the final Program Image. If configured to do so, the i.MX53 verifies the signatures using the public keys included in the Program Image. A secure boot with HAB can be performed on all boot devices supported on i.MX53 in addition to the Serial Downloader. The HAB library in the i.MX53 boot ROM also provides API functions, allowing additional boot chain components (bootloaders) to extend the secure boot chain. The out-of-fab setting for SEC_CONFIG is the Open configuration in which the ROM/HAB performs image authentication but all authentication errors are ignored and the image is still allowed to execute.

NOTE

Contact your local Freescale representative for further details on using the i.MX53 system boot feature.

The remainder of this chapter provides the details on how to configure and use the boot features of the i.MX53.

7.2 Boot Modes

7.2.1 Boot Mode Pin Settings

The i.MX53 has four boot modes (one is reserved for Freescale use). Boot mode is selected based on the binary value stored in the internal BOOT_MODE register. BOOT_MODE is initialized by sampling the BOOT_MODE0 and BOOT_MODE1 pins on the rising edge of RESET_B. After these pins are sampled, their subsequent state does

not affect the contents of the BOOT_MODE internal register. The state of the internal BOOT_MODE register may be read from the BMOD[1:0] field of the SRC Boot Mode Register (SRC_SBMR) register SRC Boot Mode Register (SRC_SBMR). The available boot modes are: Internal Boot, Boot From Fuses, and serial boot through USB/UART. Refer to Table 7-1 for settings.

Table 7-1. Boot MODE Pin Settings

BOOT_MODE[1:0][1]	Boot Type
00	Internal Boot (Development)
01	Reserved
10	Boot From Fuses
11	Serial Downloader

1. BOOT_MODE[1:0] <- {BOOT_MODE1,BOOT_MODE0} (at rising edge of RESET_B)

7.2.2 High Level Boot Sequence

Figure 7-1 shows the High Level boot ROM code flow followed in the i.MX53.

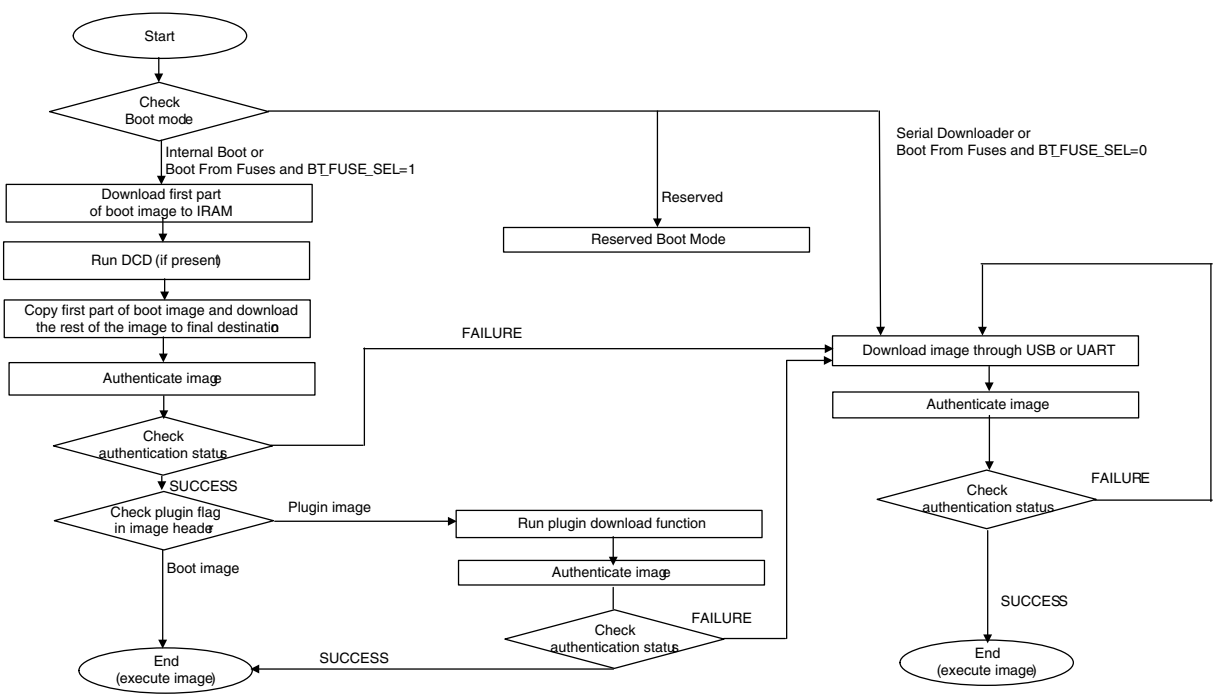


Figure 7-1. Boot Flow

NOTE

For External Interface Module (EIM) boot devices, downloading initial load region to IRAM is skipped. IVT is read from EIM address space (see IVT description Image Vector Table and Boot Data). Copying initial load region and the rest of the program image is done only if the absolute start address of the image is not equal to EIM CS0 start address.

7.2.3 Internal Boot (BOOT_MODE[1:0] = 0b00)

A value of 0b00 in the BOOT_MODE[1:0] register selects the internal Boot mode. In this mode, the processor continues to execute boot code from the internal boot ROM. The boot code performs hardware initialization, loads the Program Image from the chosen boot device, performs image validation using the HAB library (see Boot Security Settings), and then jumps to an address derived from the Program Image. If any error occurs during internal boot, the boot code jumps to the Serial Downloader (see Serial Downloader (BOOT_MODE[1:0] = 0b11)). A secure boot using the HAB on the i.MX53 is possible in all the three boot modes.

When set to Internal Boot, the boot flow may be controlled by a combination of eFUSE settings with an option of overriding the fuse settings using General Purpose I/O (GPIO) pins. Whether the ROM uses GPIO pins for a select number of configuration parameters or eFUSEs in this mode is determined by the GPIO Boot Select (BT_FUSE_SEL) eFUSE.

- If BT_FUSE_SEL = 1, all boot options are controlled by the eFUSEs described in Table 7-2.
- If BT_FUSE_SEL = 0, specific boot configuration parameters may be set using GPIO pins rather than eFUSES. The fuses that can be overridden when in this mode are indicated in the GPIO column of Table 7-2. Table 7-3 provides the details on the GPIO pins.

The use of GPIO overrides is intended for development as these pads are used for other purposes in deployed products. Freescale recommends controlling the boot configuration by eFUSEs in deployed products and reserve the use of the GPIO mode for development and testing purposes only.

7.2.4 Boot From Fuses (BOOT_MODE[1:0] = 0b10)

A value of 0b10 in the BOOT_MODE[1:0] register selects the Boot From Fuses mode. This mode is similar to the Internal Boot mode, with one difference. In this mode the GPIO boot override pins are ignored. The boot ROM code uses the boot eFUSE settings only. This mode also supports a secure boot using HAB.

If set to Boot From Fuses, the boot flow is controlled by the BT_FUSE_SEL eFUSE value. If BT_FUSE_SEL = 0, indicating that the boot device (for example, Flash, SD/MMC) has not yet been programmed, the boot flow jumps directly to the Serial Downloader. If BT_FUSE_SEL = 1, then the normal boot flow is followed, where the ROM attempts to boot from the selected boot device.

The first time a board is used, the default eFUSES may be configured incorrectly for the hardware on the platform. In such a case, the Boot ROM code may try to boot from a device that does not exist. This may cause an electrical/logic violation on some pads. Using Boot From Fuses Boot mode solves this problem. The first time the BT_FUSE_SEL eFUSE is encountered, it is not blown (thus setting BT_FUSE_SEL). This forces the ROM code to jump directly to the Serial Downloader. This allows a bootloader to be downloaded which can then provision the boot device with a Program Image and blow the BT_FUSE_SEL and the other boot configuration eFUSEs. After reset, the Boot ROM code determines that BT_FUSE_SEL is blown (BT_FUSE_SEL = 1) and the ROM code performs internal boot according to the new eFUSE settings. This allows a user to set BOOT_MODE[1:0]=0b10 on a production device and burn fuses on the same device (by forcing entry to the Serial Downloader), without changing the value of BOOT_MODE[1:0] or pull-ups/pull-downs on the BOOT_MODE pins.

7.2.5 Mode: Serial Downloader (BOOT_MODE[1:0] = 0b11)

The serial downloader is invoked if the external Flash device is not programmed, when a failure is encountered during the boot flow process or any of the following conditions are met:

- BOOT_MODE[1:0] = 0b11 (serial downloader mode)
- BOOT_MODE[1:0] = 0b10 (boot from fuses) and the eFUSE BT_FUSE_SEL = 0
- BOOT_MODE[1:0] = 0b00 or 0b10 (internal or boot from fuses) and there is not a valid image in the Flash device
- Internal failure
- Runtime exception occurs
- Error returned by the HAB functions while in Closed security configuration. Errors are ignored in Open security configuration.See Boot Security Settings.

To determine the active serial port, either UART or USB, the processor ROM program polls the UART and USB status register for approximately 32 seconds. If there is no activity on either port within the predefined polling time, then the ROM program will power down the SoC using Watchdog Timer (WDOG-1). When the serial bootloader is active WDOG-1 is reset periodically. If the communication between the serial Host and the i.MX53 is idle for more than 32 seconds or the processor goes into an endless loop, then WDOG-1 expires and powers down the device. For details on the Serial Downloader see Serial Downloader (BOOT_MODE[1:0] = 0b11).

7.2.6 Boot Security Settings

Internal boot modes use one of two security configurations:

- Closed: This level is intended for use with shipping products. All HAB functions are executed and security hardware is initialized (the Security Controller, or SCC, enters Secure state), DCD is processed if present, and the program image is authenticated by HAB prior to its execution. All detected errors will be logged, and the boot flow aborted with control passing to the serial downloader. At this level, execution does not leave the internal ROM unless the target executable image has been authenticated.
- Open: This level is intended for use in non-secure products or during the development phases of a product. All HAB functions are executed as for a production device. Security hardware is initialized (except the SCC is left in Non-Secure state), DCD is processed if present, and the program image is authenticated by HAB prior to its execution. All detected errors will be logged, but have no influence on the boot flow, which continues as if the errors did not occur. This configuration is useful for secure product development, as the Program Image will run even if the authentication data is missing or incorrect, and the error log can be examined to determine the cause of authentication failure.

NOTE

If DIR_BT_DIS eFuse is not blown, authentication may be bypassed.

7.3 Device Configuration

This section describes the external inputs defined that control the behavior of the i.MX53 Boot ROM code. This includes boot device selection (NAND, NOR, MMC), boot device configuration (NAND address cycles), and so on. In general, the source for this

configuration comes from eFUSES embedded inside the i.MX53. However, certain configuration parameters can be sourced from GPIO pins allowing further flexibility during the development process.

7.3.1 Boot eFUSE Descriptions

Table 7-2 below is a comprehensive list of the configuration parameters that the i.MX53 ROM uses.

Table 7-2. Boot eFUSE Descriptions

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings[2]
DIR_BT_DIS	OEM	Reserved Mode Disable	NA	0	0 Reserved Mode is allowed 1 Reserved Mode is not allowed
OSC_FREQ_SEL	OEM	OSC Frequency Select. Used by boot code for DPLLC programming.	Yes	0	0 19.2, 24, 26, 27 MHz - auto detection 1 OSC Frequency is 24MHz
BT_FUSE_SEL	OEM	In internal Boot mode, the BT_FUSE_SEL fuse determines weather the boot settings indicated by a Yes in the GPIO column are controlled by GPIO pins or eFUSE settings in the IC Identification Module (IIM). In Boot From Fuse mode, BT_FUSE_SEL fuse BOOT_MODE[1:0] = 10 indicates weather biit configuration eFuses have been programmed.	NA	0	If BOOT_MODE[1:0] = 0b00 0 Bits of SBMR are overridden by GPIO pins. 1 Specific bits of SBMR are controlled by eFUSE settings. If BOOT_MODE[1:0] = 0b10 0 BOOT configuration eFuses are not yet programmed. Boot flow jumps to serial downloader. 1 BOOT configuration eFuses have been programmed. Regular boot flow is performed.
SEC_CONFIG[1:0]	OEM	Security Configuration as defined in Boot Security Settings	NA	01	00 Reserved 01 Open (allows any program image, even if authentication fails) 1x Closed (Program image executes only if authenticated)
SRK_HASH[255:0]	OEM	256-bit hash value of super root key (SRK_HASH)	NA	0	Settings vary - used by HAB

Table continues on the next page...

Table 7-2. Boot eFUSE Descriptions (continued)

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings[2]
DIE-X-CORDINATE[7:0] DIE-Y-CORDINATE[7:0] WAFER_NO[4:0] LOT_NO_ENC[42:40] LOT_NO_ENC[39:32] LOT_NO_ENC[31:24] LOT_NO_ENC[23:16] LOT_NO_ENC[15:8] LOT_NO_ENC[7:0]	FSL	Device Unique ID, 64-bit UID.	NA	Unique ID	Settings vary - used by HAB
SRTC_SECMODE[1:0]	OEM	Security Mode for Secure RTC. Determines the level of security of the Secure Real Time Clock (SRTC) block.	No	00	00 Low Security 01 Medium Security 10 High Security 11 Reserved
BT_MMU_ENABLE	OEM	MMU/Cache enable bit used by boot ROM for fast HAB processing	Yes	0	0 MMU/Cache is disabled by ROM during the boot 1 MMU/Cache is enabled by ROM during the boot
BT_FREQ	OEM	ARM Frequency Selection	Yes	0	0 800MHz 1 400MHz
AXI/DDR Freq	OEM	AXI/DDR Frequency Selection	Yes	0	0 200MHz AXI/400MHz DDR 1 166MHz AXI/333MHz DDR
BOOT_CFG1[7:2]	OEM	Boot Configuration1	Yes	0	Specific to selected boot mode
BOOT_CFG2[7:5,2]	OEM	Boot Configuration2	Yes	0	Specific to selected boot mode
BOOT_CFG3[7:1]	OEM	Boot Configuration3	Yes	0	Specific to selected boot mode
WDOG_ENABLE	OEM	Watchdog reset counter enable	No	0	0 - watchdog reset counter is disabled during serial downloader 1 - watchdog reset counter is enabled during serial downloader
NFC_DLL_DLY[2:0]	OEM	NFC Delay Line settings	No	0	NFC Delay Line settings
MMC_DLL_DLY[2:0]	OEM	eMMC4.4 Delay Line settings	No	0	eMMC4.4 Delay Line settings
SRK_REVOKE[2:0]	OEM	SRK revocation mask	No	0	SRK revocation mask

1. Setting can be overridden by GPIO settings when GPIO_FUSE_SEL fuse is intact. See Table 7-3 for corresponding GPIO pin.
2. 0 = intact fuse and 1= blown fuse

7.3.2 GPIO Boot Overrides

Table 7-3 provides a list of GPIO boot overrides. These input pins are sampled at boot, and can be used to override corresponding eFUSE values, depending on the setting of the BT_FUSE_SEL fuse. The GPIO override feature is only available when BT_FUSE_SEL is 0 (fuse is unblown) and BOOT_MODE[1:0] = 00.

Table 7-3. GPIO Override Contact Assignments

Package Pin	Direction on reset	eFuse
BOOT_MODE1	Input	Boot Mode Selection
BOOT_MODE0	Input	
EIM_A22	Input	BOOT_CFG1[7]
EIM_A21	Input	BOOT_CFG1[6]
EIM_A20	Input	BOOT_CFG1[5]
EIM_A19	Input	BOOT_CFG1[4]
EIM_A18	Input	BOOT_CFG1[3]
EIM_A17	Input	BOOT_CFG1[2]
EIM_A16	Input	BT_FREQ
EIM_LBA	Input	BT_MMU_ENABLE
EIM_EB0	Input	BOOT_CFG2[7]
EIM_EB1	Input	BOOT_CFG2[6]
EIM_DA0	Input	BOOT_CFG2[5]
EIM_DA1	Input	AXI / DDR Freq
EIM_DA2	Input	OSC_FREQ_SEL
EIM_DA3	Input	BOOT_CFG2[2]
EIM_DA4	Input	BOOT_CFG3[7]
EIM_DA5	Input	BOOT_CFG3[6]
EIM_DA6	Input	BOOT_CFG3[5]
EIM_DA7	Input	BOOT_CFG3[4]
EIM_DA8	Input	BOOT_CFG3[3]
EIM_DA9	Input	BOOT_CFG3[2]
EIM_DA10	Input	BOOT_CFG3[1]

7.3.3 Device Configuration Data

See Device Configuration Data (DCD), for more details on Device Configuration Data.

7.4 Device Initialization

This section describes the details on the i.MX53 ROM and the provides initialization details. This includes details on:

- The iROM Memory Map
- The iRAM Memory Map
- On-chip blocks that the ROM should make use of or change POR register default values
- Clock initialization
- Enabling the MMU/L2 cache when performing a secure boot (SEC_CONFIG = Closed)
- Exception handling, and interrupt handling

7.4.1 Internal ROM /RAM Memory Map

Figure 7-2 below shows the iROM memory map for the i.MX53.

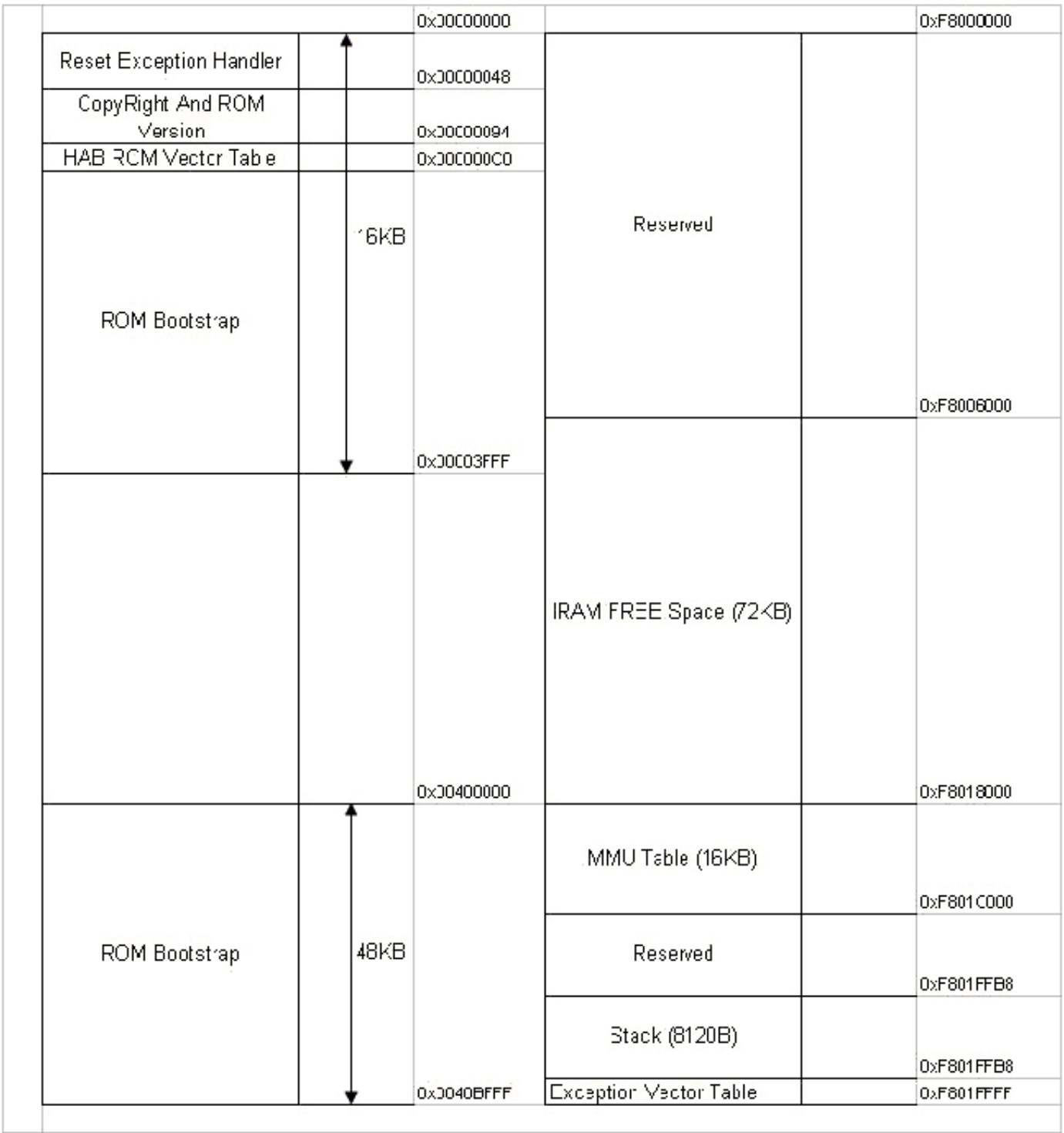


Figure 7-2. Internal ROM and RAM Memory Map

7.4.2 Boot Block Activation

The i.MX53 boot ROM affects a number of different hardware blocks which are activated and play a vital role in the boot flow. The i.MX53 ROM configures and uses the following blocks (listed in alphabetical order) during the boot process. Note that the blocks actually used depend on the boot mode and boot device selection:

- CCM - Clock Control Module
- CSU - Central Security Unit.
- ECSPI - Enhanced Configurable Serial Peripheral Interface
- EXTMC (EIM/NFC/ESDCTL) - External Memory Controller
- ESDHCV2/ESDHCV3- Enhanced Secure Digital Host Controller
- I2C - I2C Controller
- IIM - IC Identification Module. The IIM contains the eFUSEs.
- IOMUXC - I/O Multiplexer Control allows GPIO use to override eFUSE boot settings
- DPLLC - Digital Phase-Locked Loop Controller
- RTIC - Real Time Integrity Checker
- SAHARA- SAHARA Security Accelerator
- SCC-Security Controller
- SRC - System Reset Controller
- SRTC - Secure Real Time Clock
- UART and USB - Used for serial download of a boot device provisioning program
- WDOG-1 - Watchdog Timer

7.4.3 Clocks at Boot Time

The following frequencies are available as input clock to DPLLC1-4 blocks. Selection of these frequencies is accomplished as follows:

- If OSC_FREQ is 0, then ROM automatically detects one of the following frequencies - 26MHz, 19.2MHz, 27MHz or 24MHz.
- If OSC_FREQ is 1, then 24MHz oscillator.

NOTE

If OSC_FREQ is 0, then automatic detection will take approximately 10ms. For 24MHz input clock it is recommended to set OSC_FREQ=1 to save boot time.

Table 7-4 shows DPLLC1-DPLLC4 with their associated frequencies. Table 7-5 shows the various clocks and their sources used by ROM. The BOOT ROM has the capability to change the out of reset frequencies with the values shown in Table 7-5. (See "Reset values for DPLLC")

Table 7-4. DPLLC Frequencies

Clock	Frequency (MHz)
DPLLC-1	800/400
DPLLC-2	400/333
DPLLC-3	216
DPLLC-4	595

Table 7-5. Normal Frequency Clocks Configuration

Clock	CCM signal	Source	Frequency(MHz) AXI/DDR Freq=0, BT_FREQ=0	Frequency(MHz) AXI/DDR Freq=0, BT_FREQ=1	Frequency(MHz) AXI/DDR Freq=1, BT_FREQ=0	Frequency(MHz) AXI/DDR Freq=1, BT_FREQ=1
ARM core clock	arm_clk_root	DPLLC1	800	400	800	400
EXTMC	emi_slow_clk_root	DPLLC2	100		83.25	
NFCv2	enfc_clk_root	DPLLC2	33.3(NFC_FREQ_SEL=0) 14.29(NFC_FREQ_SEL=1)		27.75(NFC_FREQ_SEL=0) 11.9(NFC_FREQ_SEL=1)	
AHB	ahb_clk_root	DPLLC2	133		110	
IPG	ipg_clk_root	DPLLC2	66.5		55	
AXI_A	axi_a	DPLLC2	400		333	
AXI_B	axi_b	DPLLC2	200		166.5	
USB	usboh3_clk_root	DPLLC2	66.5		55	
UART	uart_clk_root	DPLLC3	21.6		21.6	
ESDHC	esdhc1_clk_root esdhc2_clk_root	DPLLC2	80		66.6	
CSPI	ecspi_clk_root	DPLLC3	54		54	
I2C	per_clk_root	DPLLC2	20		16.6	

On reset the processor has access to all peripherals. ROM code will disable the clocks listed in Table 7-6, except for the boot devices listed in the second column.

Table 7-6. List Of Disabled Clocks

Clock Name	Enabled For Boot Device
TMAX2_CLK	ESDHCV2-1, ESDHCV2-2

Table continues on the next page...

Table 7-6. List Of Disabled Clocks (continued)

TMAX3_CLK	PATA
UART1_IPG_CLK	UART-1
UART1_PER_CLK	UART-1
UART2_IPG_CLK	UART-2
UART2_PER_CLK	UART-2
UART3_IPG_CLK	UART-3
UART3_PER_CLK	UART-3
I2C1_CLK	I2C-1
I2C2_CLK	I2C-2
I2C3_CLK	I2C-3
FIRI_IPG_CLK	-
FIRI_SER_CLK	-
EPIT1_IPG_CLK	-
EPIT1_HIGH_FREQ_CLK	-
EPIT2_IPG_CLK	-
EPIT2_HIGH_FREQ_CLK	-
PWM1_IPG_CLK	-
PWM1_HIGH_FREQ_CLK	-
PWM2_IPG_CLK	-
PWM2_HIGH_FREQ_CLK	-
OWIRE_CLK	-
FEC_CLK	-
USB_OH3_CLK	USB
USB_OH3_SERIAL_CLK	USB
TVE_CLK	-
ESDHC1_IPG_CLK	ESDHCV2-1
ESDHC1_PER_CLK	ESDHCV2-1
ESDHC2_IPG_CLK	ESDHCV2-2
ESDHC2_PER_CLK	ESDHCV2-2
ESDHC3_IPG_CLK	ESDHCV3-3
ESDHC3_PER_CLK	ESDHCV3-3
ESDHC4_IPG_CLK	ESDHCV2-4
ESDHC4_PER_CLK	ESDHCV2-4
SSI1_IPG_CLK	-
SSI1_SSI_CLK	-
SSI2_IPG_CLK	-

Table continues on the next page...

Table 7-6. List Of Disabled Clocks (continued)

SSI2_SSI_CLK	-
SSI3_IPG_CLK	-
SSI3_SSI_CLK	-
SSI_EXT1_CLK	-
SSI_EXT2_CLK	-
PATA_CLK	PATA
SATA_CLK	SATA
CAN2_IPG_CLK	-
CAN2_SERIAL_CLK	-
USB_PHY1_CLK	USB
USB_PHY2_CLK	-
ECSPI1_IPG_CLK	ECSPI-1
ECSPI1_PER_CLK	ECSPI-1
ECSPI2_IPG_CLK	ECSPI-2
ECSPI2_PER_CLK	ECSPI-2
CSPI_IPG_CLK	CSPI
SDMA_CLK	-
GPU_CLK	-
VPU_CLK	-
VPU_REF_CLK	-
IPU_CLK	-
EMI_SLOW_CLK	NAND, EIM
EMI_ENFC_CLK	NAND
GPC_IPG_CLK	-
SPDIF0_CLK	-
SPDIF_IPG_CLK	-
CSI_MCLK1_CLK	-
CSI_MCLK2_CLK	-
IPU_DI0_CLK	-
IPU_DI1_CLK	-
GPU2D_CLK	-
ESAI_IPG_CLK	-
ESAI_SERIAL_CLK	-
CAN1_IPG_CLK	-
CAN1_SERIAL_CLK	-
PL301_4x1_CLK	SATA

Table continues on the next page...

Table 7-6. List Of Disabled Clocks (continued)

LDB_DI0_CLK	-
LDB_DI1_CLK	-
ASRC_IPG_CLK	-
ASRC_ASRCK_CLK	-
MLB_CLK	-
IEEE1588_CLK	-
UART4_IPG_CLK	UART-4
UART4_PER_CLK	UART-4
UART5_IPG_CLK	UART-5
UART5_PER_CLK	UART-5
ESAI_HCKR_CLK	-
ESAI_HCKT_CLK	-

7.4.4 Enabling MMU and Caches

Boot ROM includes a feature of enabling the Memory Management Unit (MMU) and caches to improve boot speed when performing a secure boot with SEC_CONFIG=Closed (High Assurance Boot (HAB)). L1 instruction cache is enabled at the start of image download. L1 data cache, L2 cache and MMU are enabled during image authentication.

Enabling the MMU when booting non-securely by setting, SEC_CONFIG=Open and setting the CSF pointer in the Image Vector Table set to NULL, has no impact on the boot performance. With this configuration it is recommended not to blow the MMU_EN fuse.

7.4.5 Exception Handling

The exception vectors located at the start of iROM are used to map all the ARM exceptions (except the reset exception) to a duplicate exception vector table in internal RAM.

During the boot phase, the iRAM vectors point to the serial downloader in iROM. After boot the OS loader can overwrite the vectors as required. The code shown in is used to map the ROM exception vector table to the duplicate one in iRAM.

Mapping ROM Exception Vector Table

```
;; Define linker area for iROM exception vector table
AREA IROM_VECTORS, CODE, READONLY
LDR     PC, Reset_Addr
LDR     PC, Undefined_Addr
LDR     PC, SWI_Addr
LDR     PC, Prefetch_Addr
LDR     PC, Abort_Addr
NOP                         ; Reserved vector
LDR     PC, IRQ_Addr
LDR     PC, FIQ_Addr
LDR       PC, SW_monitor_addr
;; Define exception vector table
Reset_Addr      DCD     start_address
Undefined_Addr  DCD     iRAM_Undefined_Handler
SWI_Addr        DCD     iRAM_SWI_Handler
Prefetch_Addr   DCD     iRAM_Prefetch_Handler
Abort_Addr      DCD     iRAM_Abort_Handler
                DCD     0               ; Reserved vector
IRQ_Addr        DCD     iRAM_IRQ_Handler
FIQ_Addr        DCD     iRAM_FIQ_Handler
SW_monitor_add  DCD                 SW monitor exception
start_address DCD start ;reset handler vector
```

7.4.6 Interrupt Handling During Boot

No special interrupt handling routines are required during the boot process. Interrupts are disabled during boot ROM execution and may be enabled in a later boot stage.

7.4.7 Persistent Bits

Some modes of boot ROM require registers that keep their values even after a reset. SRTC LP General Purpose register is used for this purpose. See Table 7-7 for persistent bits list and description.

Table 7-7. Persistent Bits

Bit Name	Bit Location	Description
PERSIST_SECONDARY_BOOT	SRTC_LPGR[30]	This bit is identifies which image must be used - primary and secondary. Used only for boot modes that support redundant boot.
PERSIST_BLOCK_REWRITE	SRTC_LPGR[29]	This bit is used as warning. It identifies that there are errors in NAND blocks that hold the application image. See NAND Flash for more details.
PERSIST_WDOG_BOOT	SRTC_LPGR[28]	This bit is set for enabling SBMR shadow register during Watchdog Reset Boot Mode. See Watchdog Reset Boot Mode for more details.
PERSIST_SBMR_SHADOW	SRTC_LPGR[25:0]	These bit s are used as shadow SBMR registers during Watchdog Reset Boot Mode. See Watchdog Reset Boot Mode for more details.

7.5 Boot Devices (Internal Boot)

The i.MX53 supports the following boot Flash devices:

- NOR Flash with External Interface Module (EIM), located on CS0, 16-bits bus width.
- OneNAND Flash with EIM interface, located on CS0, 16-bits bus width.
- MLC NAND, SLC NAND, and LBA NAND Flash via NAND Flash Controller (NFC) interface. Page sizes of 512 bytes, 2 Kbytes or 4 Kbytes. Bus widths of 8-bit or 16-bit, with 4/8/14/16-bit Error Checking (ECC) are supported.
- SD/MMC/eMMC and eSD through ESDHC interface, supporting high capacity cards.eMMC4.3/eMMC4.4 "boot mode (FAST BOOT)" through ESDHCV3-3 interface.
- eSD FAST BOOT mode through all the ESDHC ports.
- EEPROM boot through SPI (serial flash) and I2C/HS- I2C (via CSPI, HS- I2C, and I2C blocks respectively)
- PATA boot through PATA interface
- Serial ATA (SATA) boot through SATA interface

The selection of external boot device type is controlled by BOOT_CFG1[7:4] eFUSEs. See Table 7-8 for more details.

Table 7-8. Boot Device Selection

BOOT_CFG1[7:4]	Boot Device
0000	NOR/OneNand (EIM)
0001	Reserved
0010	Hard Disk (PATA/SATA)
0011	Serial ROM (I2C/SPI)
010x	SD/eSD
011x	MMC/eMMC
1xxx	NAND

7.5.1 NOR Flash/OneNand using EIM Interface

The External Interface Module (EIM) works in the asynchronous mode, and supports either muxed, Address/Data, or non-muxed schemes based on fuse settings:

Table 7-9. EIM Boot eFUSE Descriptions

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings
BOOT_CFG1[7:4]	OEM	Boot Device Selection	Yes	0000	0000 - Boot from EIM Interface
BOOT_CFG1[3]	OEM	NOR/OneNand Selection	Yes	0	0 - NOR 1 - OneNand
BOOT_CFG2[7:6]	OEM	Muxing Scheme	Yes	00	00 - Muxed, 16-bit data (low half) interface 01 - Not muxed, 16-bit data (high half) interface 10 - Reserved 11 - Reserved
BOOT_CFG3[7:6]	OEM	OneNAND Page Size	Yes	00	00 - 1K 01 - 2K 10 - 4K 11 - Reserved

1. Setting can be overridden by GPIO settings when BT_FUSE_SEL fuse is intact. See Table 7-3 for corresponding GPIO pin.

7.5.1.1 NOR Flash Boot Operation

Booting from the NOR Flash is supported via EIM interface. The ROM reads Image Vector Table and Boot Data structures to determine if the image can be executed directly from EIM address space or should be copied to other memory. The start field of Boot Data Structure specifies the final location of the image (see Image Vector Table and Boot Data).

7.5.1.2 OneNAND Flash Boot Operation

At system power-up, the OneNAND device automatically copies an Initial Load Region of 1 Kbyte from the start of the flash array (sector 0 and sector 1, page 0, block 0) to its Boot RAM (oneNAND's internal RAM).

NOTE

The OneNAND boot RAM memory containing the Initial 1K Load Region must contain the IVT, DCD and the Boot Data structures.

Next, the i.MX53 ROM processes the DCD and then proceeds to copy the Program Image contents to the application destination pointer (located in the start entry of Boot Data (see Image Vector Table and Boot Data). The ROM determines the size of the

Program Image by the length specified by size entry in Boot Data structure (see Image Vector Table and Boot Data). A failure loading data from the OneNAND device for any reason forces the i.MX53 Boot ROM to enter the Serial Downloader, otherwise the booting from the OneNAND device continues.

Figure 7-3 illustrates the layout of the Program Image on a OneNand boot device.

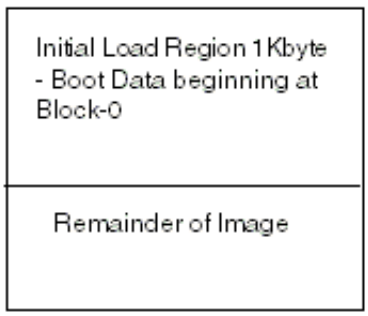


Figure 7-3. Program Image Layout on a OneNand Flash Device

Prior to accessing the OneNAND device the i.MX53 ROM waits approximately 500 µs after Power On Reset. This delay is required for the OneNAND device to become ready. After this initial 500 µs delay it can take an addition 70 µs for the OneNAND device to load the Initial Load Region of 1 Kbyte into its boot RAM. The i.MX53 ROM polls the OneNAND device Interrupt Status Register to confirm that the first 1 Kbytes has been loaded to the OneNAND boot RAM before continuing with the boot flow.

7.5.1.3 IOMUX Configuration for EIM Devices

The EIM interface signals are not configured in the IOMUX for 16-bit muxed mode. The EIM interface uses dedicated contacts on the IC. The contacts assigned to the data signals used by EIM in non-muxed mode is shown in Table 7-10.

Table 7-10. EIM Non-Muxed Mode IOMUX Pin Configuration

Signal	EIM
DATA0	EIM_DA0.alt0
DATA1	EIM_DA1.alt0
DATA2	EIM_DA2.alt0
DATA3	EIM_DA3.alt0
DATA4	EIM_DA4.alt0
DATA5	EIM_DA5.alt0

Table continues on the next page...

Table 7-10. EIM Non-Muxed Mode IOMUX Pin Configuration (continued)

DATA6	EIM_DA6.alt0
DATA7	EIM_DA7.alt0
DATA8	EIM_DA8.alt0
DATA9	EIM_DA9.alt0
DATA10	EIM_DA10.alt0
DATA11	EIM_DA11.alt0
DATA12	EIM_DA12.alt0
DATA13	EIM_DA13.alt0
DATA14	EIM_DA14.alt0
DATA15	EIM_DA15.alt0

7.5.2 NAND Flash

A number of MLC/SLC NAND Flash devices from different vendors and LBA NAND Flash devices are supported by the boot ROM. The Error Correction and Control (ECC) sub-block is used to detect the errors.

7.5.2.1 NAND eFUSE Configuration

The boot ROM determines the configuration of external the NAND flash by parameters, either provided by eFUSE, or sampled on GPIO pins, during boot. See the Table 7-11 for parameters details.

Table 7-11. NAND Boot eFUSE Descriptions

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings
BOOT_CFG1[7]	OEM	Boot Device Selection	Yes	0	1 - Boot from Nand Interface
BOOT_CFG1[6]	OEM	Muxed On	Yes	0	0 - PATA Pads 1 - EIM Pads
BOOT_CFG1[5:4]	OEM	Interleaving Scheme	Yes	00	00 - No Interleaving 01- 2 device 10- 4 device 11- Reserved

Table continues on the next page...

Table 7-11. NAND Boot eFUSE Descriptions (continued)

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings
BOOT_CFG1[3:2]	OEM	Address Cycles	Yes	00	00 - 3 01 - 4 10 - 5 11 - 6
BOOT_CFG2[7:6]	OEM	Page Size	Yes	00	00 - 512 + 16 Bytes (4-bit ECC) 01 - 2KB + 64 Bytes (4-bit ECC) 10 - 4KB + 128 Bytes (4-bit ECC) 11 - 4KB + 218 Bytes (8/14/16-bit ECC)
BOOT_CFG2[5]	OEM	NAND Interface Bus Width	Yes	0	0 - 8 bit 1- 16 bit
BOOT_CFG2[2]	OEM	NFC_FREQ_SEL	Yes	0	0 - AXI DDR divide by 12 1 - AXI DDR divide by 28
BOOT_CFG3[7]	OEM	Stride Size (Bad Block skip step)	Yes	0	0 - 1 Block 1 - 8 Blocks
BOOT_CFG3[6]	OEM	LBA-Nand Select	Yes	0	0 - Non LBA (11ms delay) 1 - LBA (22ms delay)
BOOT_CFG3[5]	OEM	NAND Use R/B Signals:	Yes	0	0 - No 1 - Yes
BOOT_CFG3[4:3]	OEM	ECC / Spare select:	Yes	00	Page Size=00/01/10 00 - 4-bit ECC 01 - 4-bit ECC 10 - 4-bit ECC 11 - ECC OFF Page Size=11 00 - 8-bit ECC 01 - 14-bit ECC 10 - 16-bit ECC 11 - ECC OFF
BOOT_CFG3[2:1]	OEM	Pages In Block	Yes	00	00 - 32 01 - 64 10 - 128 11 - 256

1. Setting can be overridden by GPIO settings when BT_FUSE_SEL fuse is intact. See Table 7-3 for corresponding GPIO pin.See Table 7-47 for NAND eFuse configuration in DCD Example.

7.5.2.2 NAND Flash Boot Flow

On device power on the boot ROM will start searching for the Firmware Configuration Block (FCB). FCB is used by the ROM to find the program images and Discovered Bad Blocks Table (DBBT) for redundant boot and bad block management purposes. It has pointers to Primary and Secondary images, and to DDBT.

See the flow of FCB search in Figure 7-4

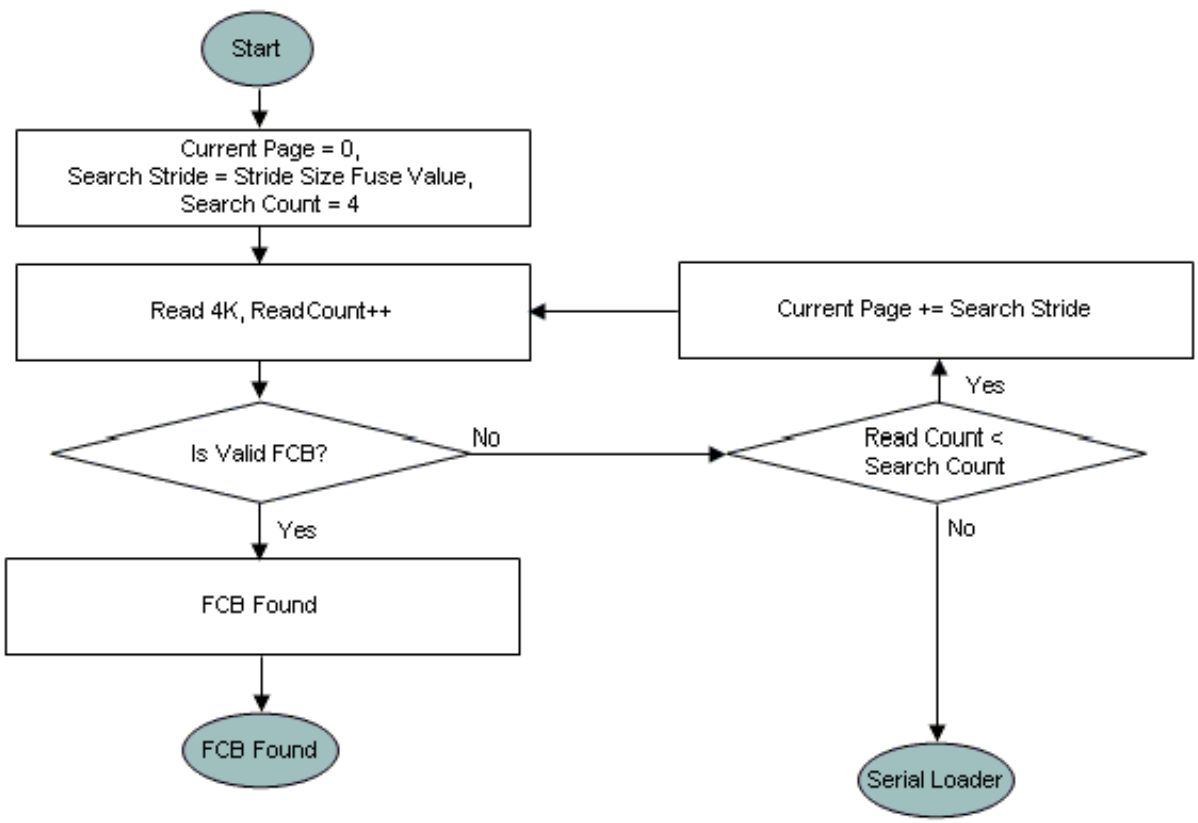


Figure 7-4. FCB Search Flow

See Table 7-12 for FCB format.

Table 7-12. NAND FCB Format

Byte Index	Byte0	Byte1	Byte2	Byte3	Comments
0x0000	Reserved				

Table continues on the next page...

Table 7-12. NAND FCB Format (continued)

Byte Index	Byte0	Byte1	Byte2	Byte3	Comments
0x0004	0x46	0x43	0x42	0x20	Fingerprint #2 (ASCII Code, FCB)
0x0008	0x01	0x00	0x00	0x00	Fingerprint #3 (FCB Version)
0x000C-0x0067	Reserved				
0x0068	Primary Image Starting Page Number				Start page number of primary firmware on the media
0x006C	Secondary Image Starting Page Number				Start page number of secondary firmware on the media
0x0070-0x0077	Reserved				Reserved
0x0078	DBBT Search Area Starting Page Number				Start page address of DBBT Search Area
0x007C	Bad Block Marker Offset				Band Block Marker Offset in page data buffer
0x0080-0x00AB	Reserved				
0x00AC	Bad Block Marker Swapping Enable				Bad Block Marker Swapping Enable
0x00BO	Bad Block Marker Offset to Spare Byte				Bad Block Marker Offset to Spare Byte
0x00B4-0x01FF	Reserved				

Once FCB found the boot ROM starts searching for Discovered Bad Blocks Table (DBBT). If DBBT Search Area is 0 in FCB, then ROM assumes that there are no bad blocks on NAND device. See Figure 7-5 for DBBT search flow.

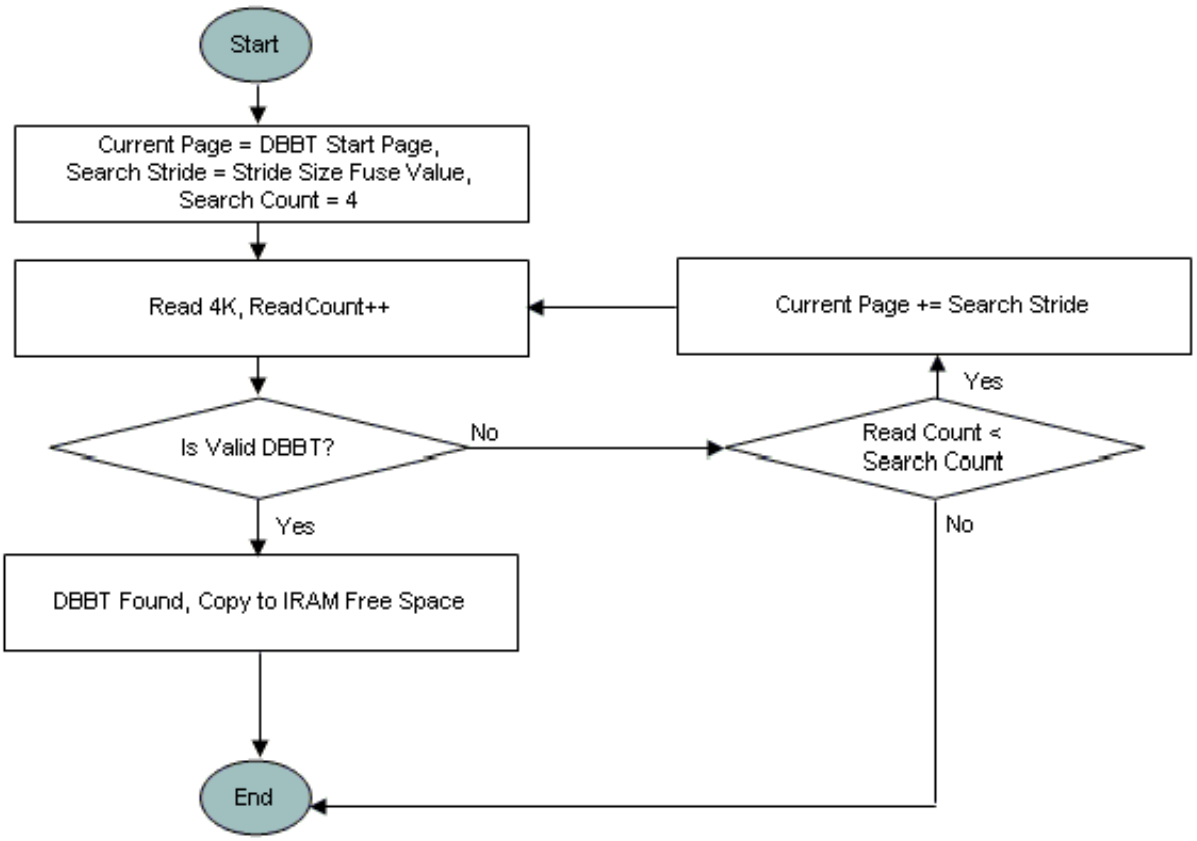


Figure 7-5. DBBT Search Flow

See Table 7-13 for DBBT format.

Table 7-13. NAND DBBT Format

Byte Index	Byte0	Byte1	Byte2	Byte3	Comments
0x0000	Reserved				
0x0004	0x54	0x42	0x42	0x44	Fingerprint #2 (ASCII Code for TBBD)[1]
0x0008	0x01	0x00	0x00	0x00	Fingerprint #3 (DBBT Version)
0x000C	Reserved				
0x0010	Reserved				
0x0014-4*PageSize	Reserved				
4*PageSize + 0x0000	Reserved				
4*PageSize + 0x0004	Number of Entries (N<425)				
4*PageSize + 0x0008	Bad Block Number				

Table continues on the next page...

Table 7-13. NAND DBBT Format (continued)

Byte Index	Byte0	Byte1	Byte2	Byte3	Comments
4*PageSize + 0x000C	Bad Block Number				
.....					
4*PageSize + 0x06A4	Bad Block Number				
4*PageSize + 4*425	Bad Block Number				

1. It should have been DBBT, but it is actually coded as TBBD due to a typo.

From that point ROM will start reading the image and copy it to final destination. If PERSIST_SECONDARY_BOOT (see Table 7-7) bit is 0, then ROM will use primary image start page from FCB for reading the image. Otherwise it will use the secondary image.

NOTE

The Initial 4K of Program Image must contain the IVT, DCD and the Boot Data structures.

ROM will skip blocks marked as bad in DBBT.

If during read there was a page with number of errors that ECC can correct, ROM will turn on PERSIST_BLOCK_REWRITE bit (see Table 7-7). This is done as warning for maintenance task of higher layer software.

If during primary image read there are uncorrected ECC errors or authentication failures (for Closed mode only), the boot ROM will turn on PERSIST_SECONDARY_BOOT bit (see Table 7-7) and perform software reset. (After software reset secondary image will be used).

If during secondary image read there was a page with number of errors higher than ECC can correct, the boot ROM will go to serial loader.

7.5.2.3 Bad Block Marker Swapping

NAND vendors mark bad blocks during production. All device locations are erased (FFh) except locations where the initial invalid block(s) information is written prior to shipping. The initial invalid block(s) status is defined by the one byte in the spare area. As vendors' layout of data and spare area on NAND flash page is different from MCIMX53 NAND Flash Controller layout, swapping one byte between data and spare area is required, in order to preserve Bad Block marker at the original location on NAND flash page.

Bad Block swapping is enabled depending on Bad Block Swapping Enable FCB parameter (see Table 7-12). The swapping is done between one byte in data area with offset specified in FCB, and one byte in spare area segment of the page. The offset to spare area segment byte is specified in FCB.

7.5.2.4 IOMUX Configuration for NAND

The NAND Flash Controller (NFC) signals can use 2 sets of pins, selected by eFuse BOOT_CFG1[6] (MUXED_ON), see Table 7-11. The contacts assigned to the signals used by NFC is shown in Table 7-14

Table 7-14. NAND IOMUX Pin Configuration

Signal	MUXED_ON=0 (PATA)	MUXED_ON=1 (EIM)
CLE	NANDF_CLE.alt0	NANDF_CLE.alt0
ALE	NANDF_ALE.alt0	NANDF_ALE.alt0
WP_B	NANDF_WP_B.alt0	NANDF_WP_B.alt0
WE_B	NANDF_WE_B.alt0	NANDF_WE_B.alt0
RE_B	NANDF_RE_B.alt0	NANDF_RE_B.alt0
RB0	NANDF_RB0.alt0	NANDF_RB0.alt0
CS0	NANDF_CS0.alt0	NANDF_CS0.alt0
CS1	NANDF_CS1.alt0	NANDF_CS1.alt0
CS2	NANDF_CS2.alt0	NANDF_CS2.alt0
D0	PATA_DATA0.alt3	EIM_DA0.alt0
D1	PATA_DATA1.alt3	EIM_DA1.alt0
D2	PATA_DATA2.alt3	EIM_DA2.alt0
D3	PATA_DATA3.alt3	EIM_DA3.alt0
D4	PATA_DATA4.alt3	EIM_DA4.alt0
D5	PATA_DATA5.alt3	EIM_DA5.alt0
D6	PATA_DATA6.alt3	EIM_DA6.alt0
D7	PATA_DATA7.alt3	EIM_DA7.alt0
D8	PATA_DATA8.alt3	EIM_DA8.alt0
D9	PATA_DATA9.alt3	EIM_DA9.alt0
D10	PATA_DATA10.alt3	EIM_DA10.alt0
D11	PATA_DATA11.alt3	EIM_DA11.alt0
D12	PATA_DATA12.alt3	EIM_DA12.alt0
D13	PATA_DATA13.alt3	EIM_DA13.alt0
D14	PATA_DATA14.alt3	EIM_DA14.alt0
D15	PATA_DATA15.alt3	EIM_DA15.alt0

7.5.3 Expansion Device

The i.MX53 ROM supports booting from MMC/eMMC and SD/eSD compliant devices.

7.5.3.1 Expansion Device eFUSE Configuration

SD/MMC/eSD/eMMC boot can be performed using either ESDHCV2-1, ESDHCV2-2, ESDHCV3-3, or ESDHCV2-4 ports, based on setting of the BOOT_CFG3[5:4] (Port Select) fuse or its associated GPIO input value at boot. For eMMC4.3 and eMMC4.4 with BOOT ACK support boot mode, only ESDHCV3-3 can be used. Refer to Table 7-15 for details. The boot flow for expansion devices is shown in Figure 7-6 through Figure 7-10, beginning Figure 7-6.

Table 7-15. ESDHC Boot eFUSE Descriptions

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings
BOOT_CFG1[7:6]	OEM	Boot Device Selection	Yes	00	01 - Boot from ESDHC Interfaces
BOOT_CFG1[5]	OEM	SD/MMC Selection	Yes	0	0 - SD/eSD 1 - MMC/eMMC
BOOT_CFG1[4]	OEM	Fast Boot Support	Yes	0	0 - Normal Boot 1 - Fast Boot
BOOT_CFG1[3]	OEM	SD/MMC Speed Mode	Yes	0	0 - High Speed Mode 1 - Normal Speed Mode
BOOT_CFG2[7:5]	OEM	Bus Width	Yes	000	SD/eSD (BOOT_CFG1[5]=0) xx0 - 1-bit xx1 - 4-bit MMC/eMMC (BOOT_CFG1[5]=1) 000 - 1-bit 001 - 4-bit 010 - 8-bit 101 - 4-bit DDR (MMC 4.4) 110 - 8-bit DDR (MMC 4.4) Else - reserved.
BOOT_CFG3[5:4]	OEM	Port Select	Yes	00	00 - ESDHCV2-1 01 - ESDHCV2-2 10 - ESDHCV3-3 11 - ESDHCV2-4

Table continues on the next page...

Table 7-15. ESDHC Boot eFUSE Descriptions (continued)

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings
BOOT_CFG3[3]	OEM	DLL Override	Yes	0	0 - Boot ROM default. 1 - Apply value per fuse field MMC_DLL_DLY[3:0]
BOOT_CFG3[2]	OEM	Boot Acknowledge Disable	Yes	0	0 - Boot Acknowledge Enabled. 1 - Boot Acknowledge Disabled.
MMC_DLL_DLY[3:0]	OEM	MMC DLL Value	No	0000	MMC DLL Value

1. Setting can be overridden by GPIO settings when BT_FUSE_SEL fuse is intact. See Table 7-3 for corresponding GPIO pin.

Boot code supports following standards.

- MMCv4.4 or less
- eMMCv4.4 or less
- SDv2.0 or less
- eSDv2.10 rev-0.9, with or without FAST_BOOT.

MMC/SD/eSD/eMMC can be connected to any of ESDHCV2-1,2,4/V3-3 blocks and can be booted by copying 2 Kbyte of data from MMC/SD/eSD/eMMC device to internal RAM. After checking the Image Vector Table header value (0xD1) from Program Image, the ROM code performs a DCD check. After successful DCD extraction, the Rom code extracts from Boot Data Structure the destination pointer and length of image to be copied to RAM device from where code execution occurs.

NOTE

The Initial 2Kbyte of Program Image must contain the IVT, DCD and the Boot Data structures.

Table 7-16. SD/MMC Frequencies

Frequency	AXI/DDR Freq=0	AXI/DDR Freq=0
Identification (KHz)	357.143	297.321
Normal Speed Mode (MHz)	20	16.6
High Speed Mode (MHz)	40	33.3

NOTE

The application image length and destination pointer should be specified in image as the Boot ROM code reads application image length as well as application destination pointer from image.

7.5.3.2 MMC and eMMC Boot

During initialization (normal boot mode) the MMC frequency is set to 357.143 KHz. When the MMC card enters the identification portion of the initialization, voltage validation is performed and the ROM boot code checks high voltage settings and card capacity. The ROM boot code supports both high capacity and low capacity MMC/eMMC cards. After initialization phase is complete, the ROM boot code switches to a higher frequency (20 MHz in Normal boot mode or 40MHz in High Speed mode). eMMC is also interfaced through ESDHC and follows same flow as does by MMC.

The boot partition can be selected for an MMC4.x card after the card initialization is complete. The ROM code reads the BOOT_PARTITION_ENABLE field in the Ext_CSD[179] to get the boot partition to be set. If there is no boot partition mentioned in BOOT_PARTITION_ENABLE field or the user partition has been mentioned, ROM boots from the user partition.

If using an eMMC4.3 or eMMC4.4 device supporting special "Boot mode", it can be initiated by pulling the CMD line low. If BOOT ACK is enabled, the eMMC4.3/eMMC4.4 device sends the BOOT ACK via DATA lines and ROM can read the BOOT ACK [S010E] to identify the eMMC4.3/eMMC4.4 device. eMMC4.3/eMMC4.4 device with "Boot mode" feature can only be supported via ESDHCV3-3 and with or without BOOT ACK. If BOOT ACK is enabled ROM waits 50 ms to get the BOOT ACK and if BOOT ACK is received by ROM. If BOOT ACK is disabled ROM waits 1 second for data. If BOOT ACK or data was received then eMMC4.3/eMMC4.4 is booted in "Boot mode", otherwise eMMC4.3/eMMC4.4 boots as a normal MMC card from the selected boot partition. This boot mode can be selected by BOOT_CFG1[4] (Fast Boot) fuse. BOOT ACK is selected by BOOT_CFG2[2].

If using eMMC4.4 device, Double Data Rate (DDR) mode can be used. This mode can be selected by BOOT_CFG2[7:5] (Bus Width) fuse.

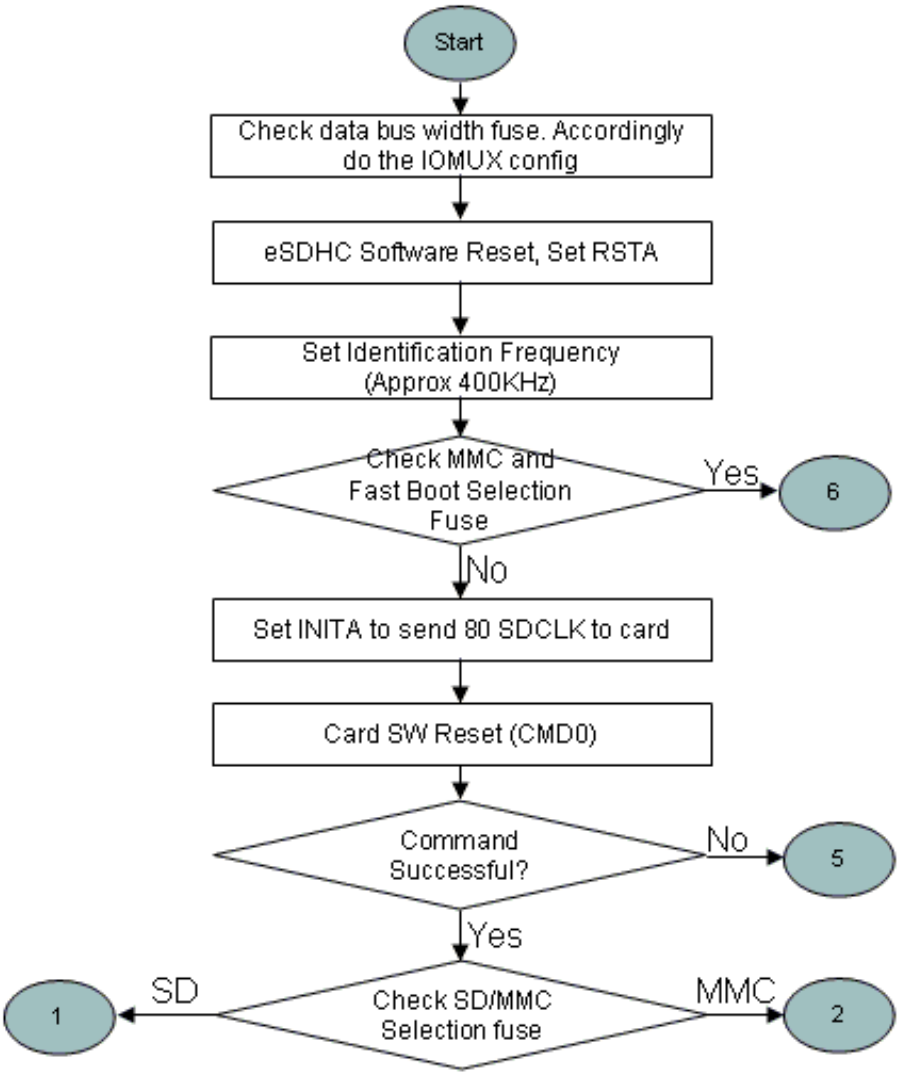


Figure 7-6. Expansion Device Boot Flow (1 of 5)

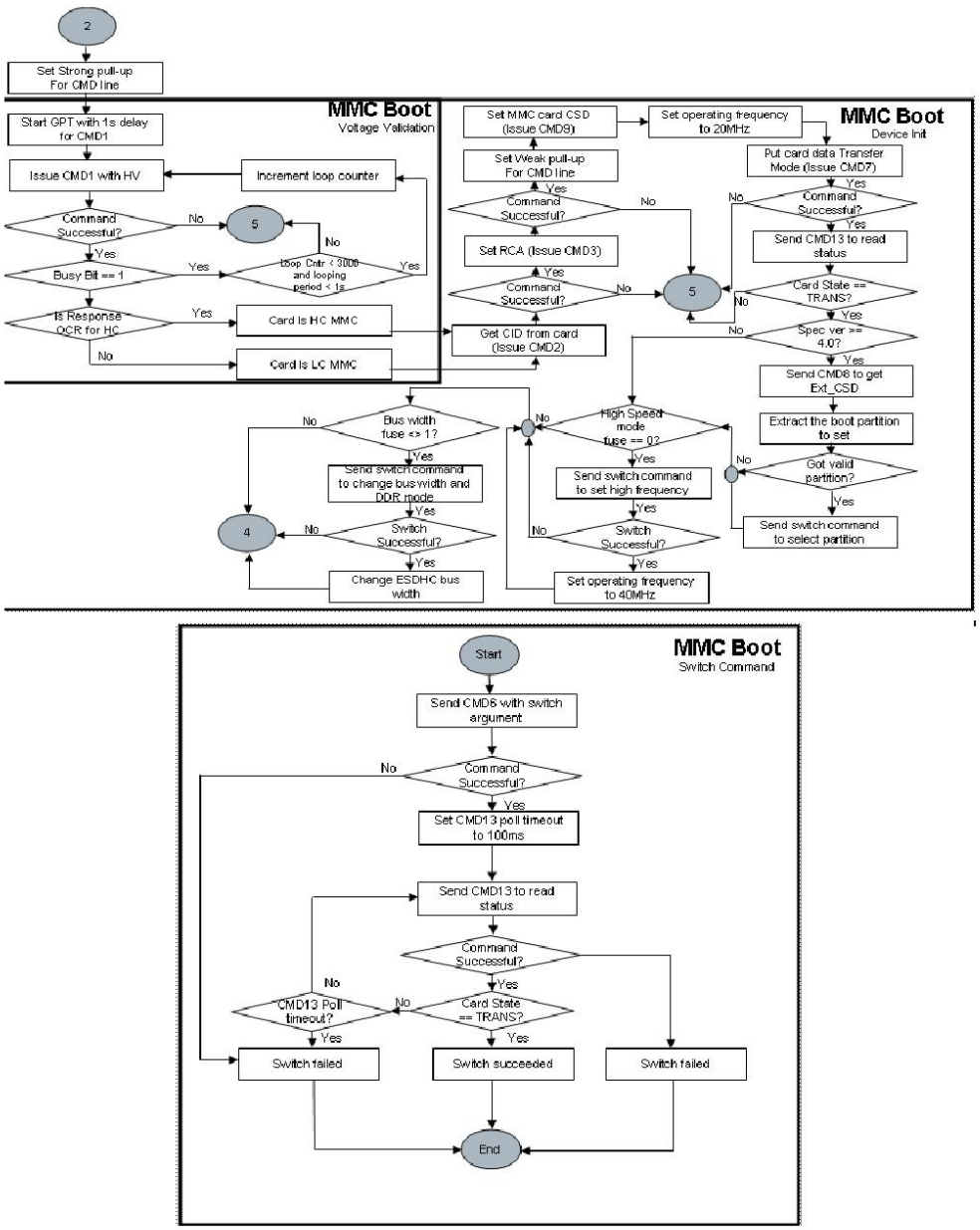


Figure 7-7. Expansion Device (MMC) Boot Flow (2 of 5)

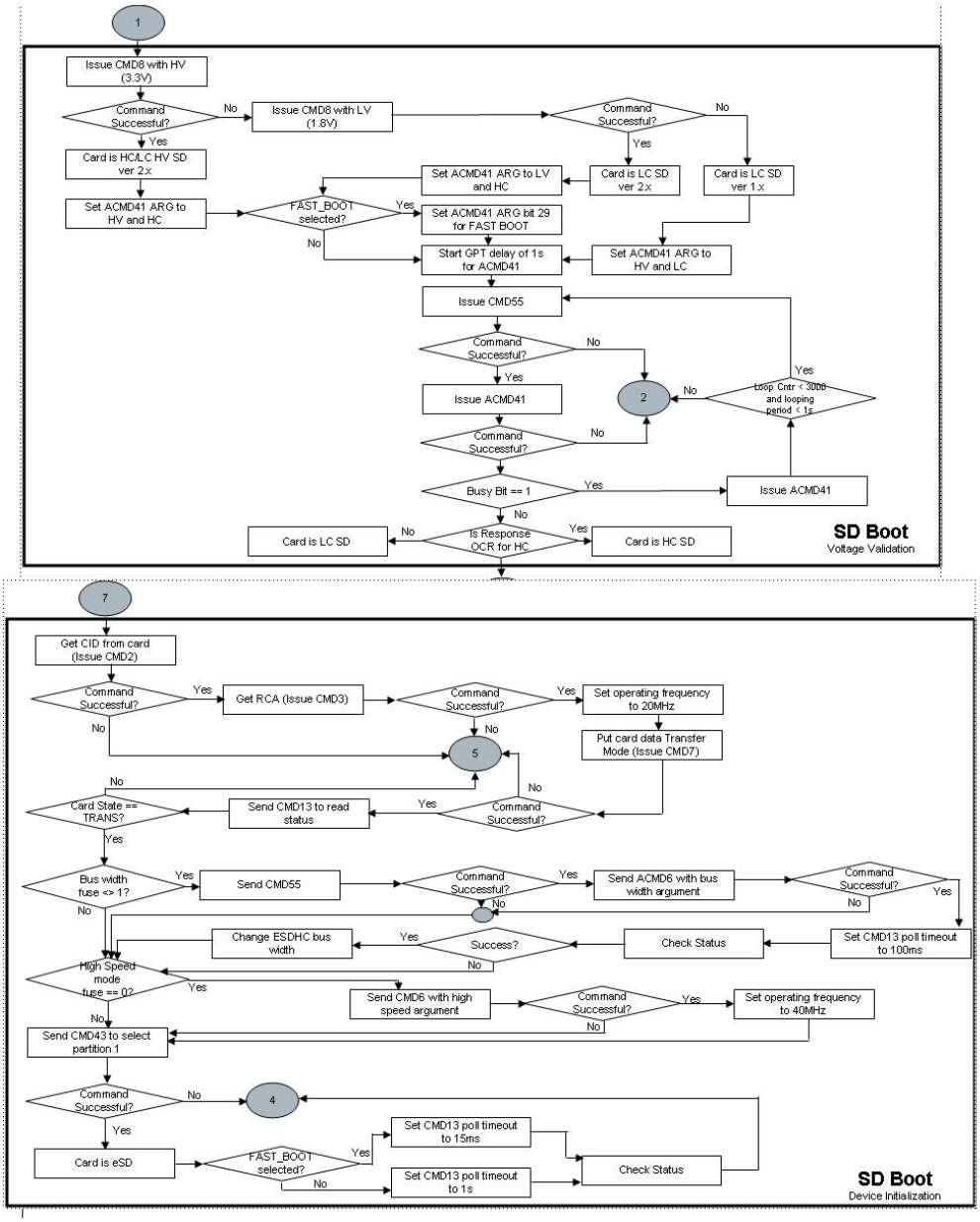


Figure 7-8. Expansion Device (SD/eSD) Boot Flow (3 of 5)

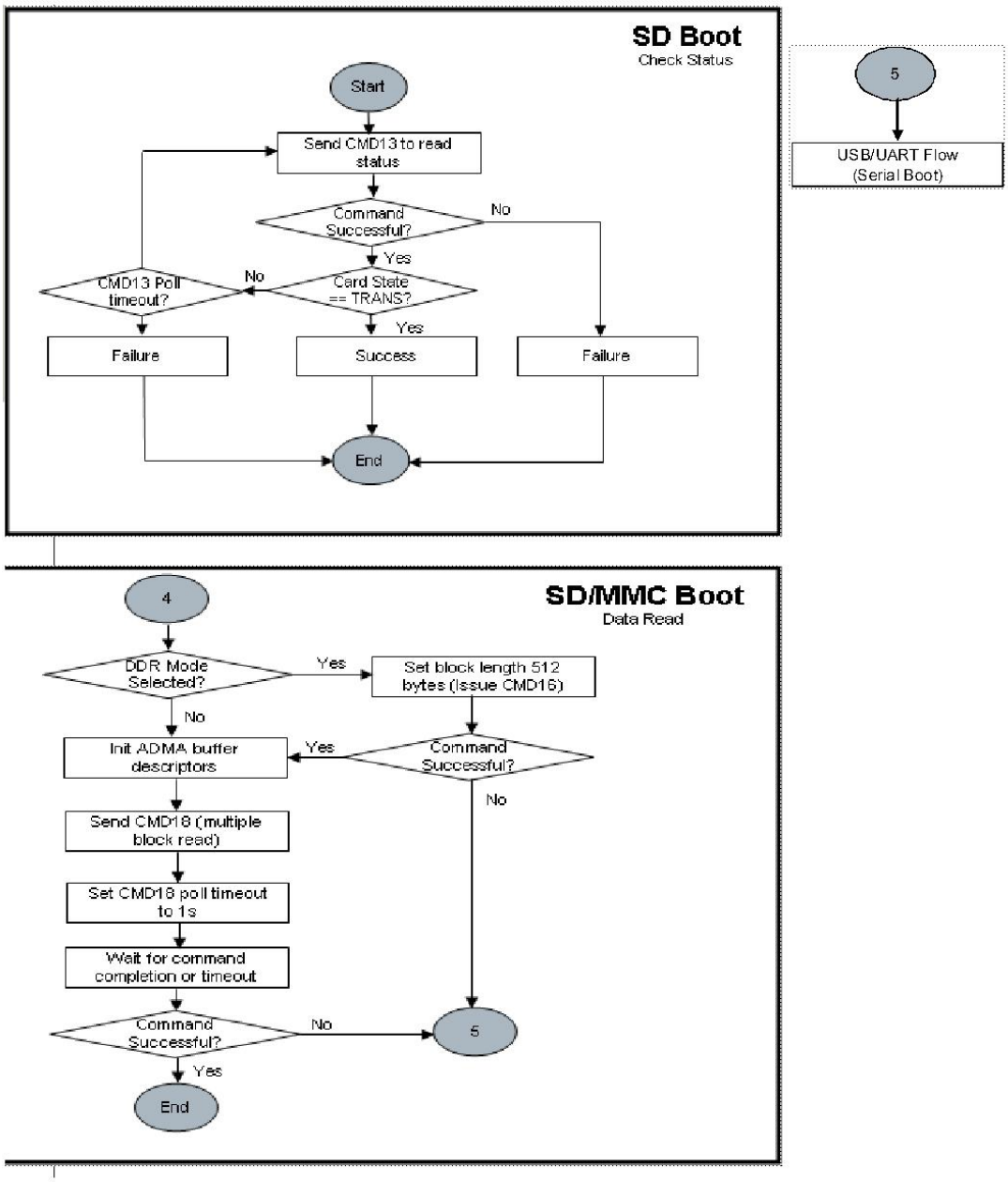


Figure 7-9. Expansion Device Boot Flow (4 of 5)

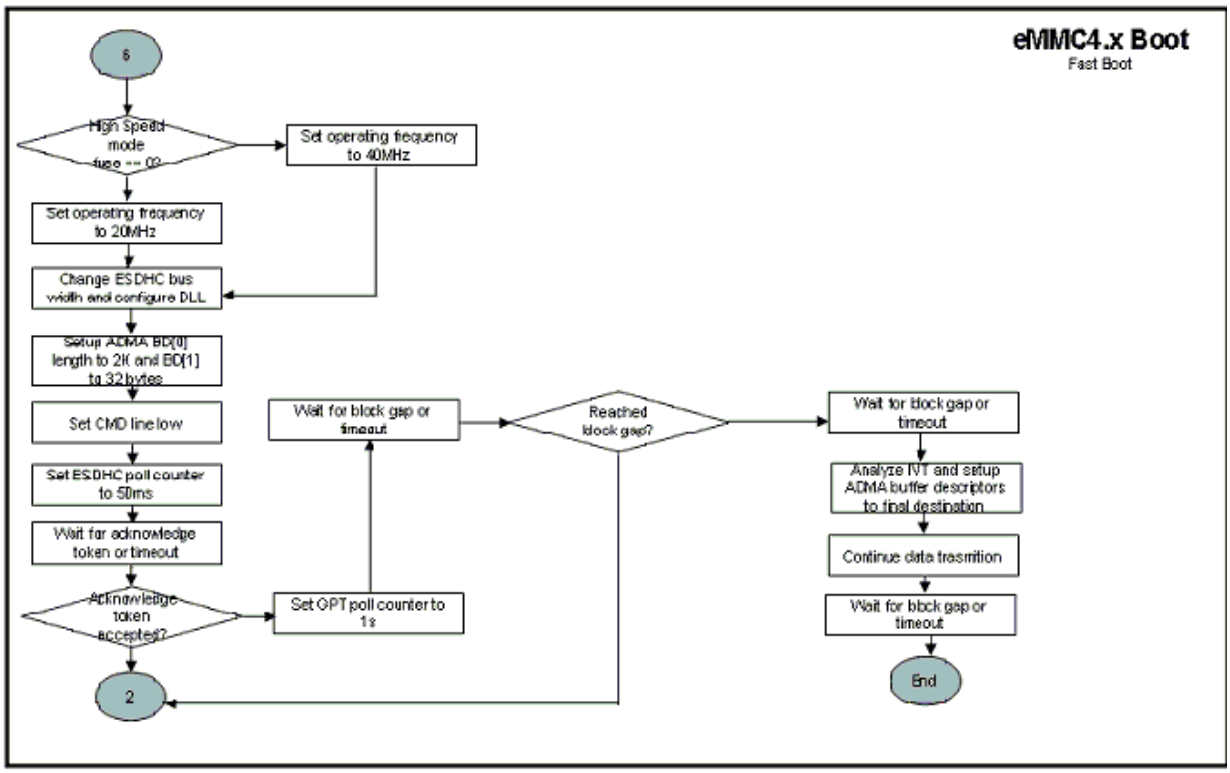


Figure 7-10. Expansion Device Boot Flow (5 of 5)

7.5.3.3 SD and eSD

After the normal boot mode initialization begins, the SD and eSD frequency is set to 357.143 KHz. During the identification phase SD and eSD card voltage validation is performed. During voltage validation, boot code first checks with high voltage settings and if it fails it checks with low voltage settings. Capacity of card is also checked. Boot code supports high capacity and low capacity SD and eSD cards. After voltage validation card initialization is done.

During card initialization the ROM boot code attempts to set the boot partition (both SD and eSD devices). If this fails, the boot code assumes card is a normal SD card otherwise as eSD card. After initialisation phase is over, boot code switches to a higher frequency (20 MHz in Normal Speed mode or 40 MHz in High Speed Mode). ROM also supports FAST_BOOT mode booting from eSD card. This mode can be selected by BOOT_CFG1[4] (Fast Boot) fuse described in Table 7-15.

7.5.3.4 IOMUX Configuration for SD/MMC

Table 7-17. SD/MMC IOMUX Pin Configuration

Signal	ESDHCV2-1	ESDHCV2-2	ESDHCV3-3	ESDHCV2-4
CLK	SD1_CLK.alt0	SD2_CLK.alt0	PATA_RESET_B.alt2	PATA_DA_1.alt2
CMD	SD1_CMD.alt0	SD2_CMD.alt0	PATA_IORDY.alt2	PATA_DA_2.alt2
DAT0	SD1_DATA0.alt0	SD2_DATA0.alt0	PATA_DATA8.alt4	PATA_DATA12.alt4
DAT1	SD1_DATA1.alt0	SD2_DATA1.alt0	PATA_DATA9.alt4	PATA_DATA13.alt4
DAT2	SD1_DATA2.alt0	SD2_DATA2.alt0	PATA_DATA10.alt4	PATA_DATA14.alt4
CD/DAT3	SD1_DATA3.alt0	SD2_DATA3.alt0	PATA_DATA11.alt4	PATA_DATA15.alt4
DAT4	PATA_DATA8.alt2	PATA_DATA12.alt2	PATA_DATA0.alt4	PATA_DATA4.alt4
DAT5	PATA_DATA9.alt2	PATA_DATA13.alt2	PATA_DATA1.alt4	PATA_DATA5.alt4
DAT6	PATA_DATA10.alt2	PATA_DATA14.alt2	PATA_DATA2.alt4	PATA_DATA6.alt4
DAT7	PATA_DATA11.alt2	PATA_DATA15.alt2	PATA_DATA3.alt4	PATA_DATA7.alt4

7.5.3.5 Redundant Boot Support for Expansion Device

i.MX53 ROM supports redundant boot for expansion device. Primary or Secondary image is selected depending on PERSIST_SECONDARY_BOOT setting (see Table 7-7).

If PERSIST_SECONDARY_BOOT is 0, the boot ROM uses address 0x0 for primary image.

If PERSIST_SECONDARY_BOOT is 1, the boot ROM will read secondary image table from address 0x200 on boot media and will use address specified in the table

.

Table 7-18. Secondary Image Table Format

Reserved (chipNum)
Reserved (driveType)
tag
firstSectorNumber
Reserved (sectorCount)

Where:

- tag: used as indication of valid secondary image table. Must be 0x00112233.
- firstSectorNumber is the first 512B sector number of the secondary image.

For secondary image support, the primary image must reserve space for secondary image table. See Figure 7-11 for typical structures layout on expansion devic

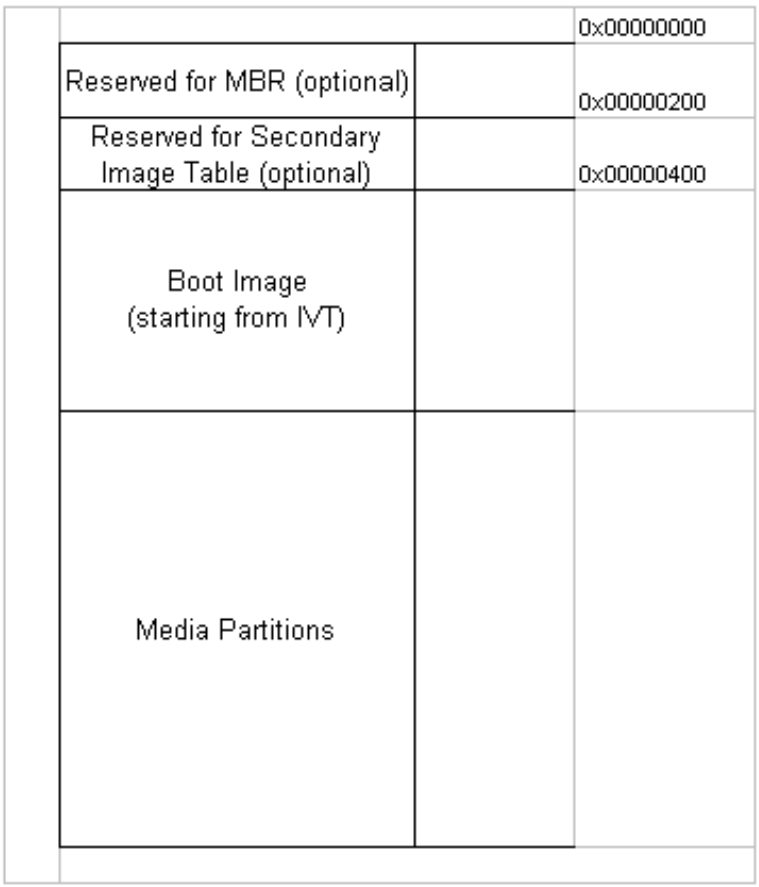


Figure 7-11. Expansion Device Structures Layout

For Closed mode, if there are failures during primary image authentication, the boot ROM will turn on PERSIST_SECONDARY_BOOT bit (see Table 7-7) and perform software reset. (After software reset secondary image will be used).

7.5.4 Hard Disk and SSD

The i.MX53 supports boot from Hard Disk and SSD devices using the PATA and SATA interfaces.

7.5.4.1 Hard Disk and SSD eFUSE Configuration

The boot ROM code determines the type of device using the following parameters, either provided by eFUSE settings or sampled on the I/O pins, during boot (See Table 7-19 for details):

Table 7-19. HDD eFUSE Descriptions

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings
BOOT_CFG1[7:4]	OEM	Boot Device Selection	Yes	0000	0010 - Boot from Hard Disk
BOOT_CFG1[3]	OEM	HD Type	Yes	0	0 - PATA 1- SATA

1. Setting can be overridden by GPIO settings when BT_FUSE_SEL fuse is intact. See Table 7-3 for corresponding GPIO pin.

The boot ROM will send IDENTIFY command to hard disk during initialization. When identification block received, boot ROM assumes that the device is ready. The boot ROM sends separate command for each sector of 512 bytes in PIO mode.

The boot ROM will copy 2 Kbyte of data from Hard Disk or SSD device to internal RAM. After checking the Image Vector Table header value (0xD1) from Program Image, the ROM code performs a DCD check. After successful DCD extraction, the Rom code extracts from Boot Data Structure the destination pointer and length of image to be copied to RAM device from where code execution occurs.

NOTE

The Initial 2 Kbyte of Program Image must contain the IVT, DCD and the Boot Data structures.

7.5.4.2 IOMUX Configuration for PATA

Table 7-20 below provides the IOMUX configuration details for Parallel ATA (PATA).

Table 7-20. PATA IOMUX Pin Configuration

Signal	PATA
DIOW	PATA_DIOW.alt0
DMACK	PATA_DMACK.alt0
DMARQ	PATA_DMARQ.alt0
BUFFER_EN	PATA_BUFFER_EN.alt0
INTRQ	PATA_INTRQ.alt0
DIOR	PATA_DIOR.alt0
RESET_B	PATA_RESET_B.alt0
IORDY	PATA_IORDY.alt0
DA_0	PATA_DA_0.alt0
DA_1	PATA_DA_1.alt0
DA_2	PATA_DA_2.alt0

Table continues on the next page...

Table 7-20. PATA IOMUX Pin Configuration (continued)

CS_0	PATA_CS_0.alt0
CS_1	PATA_CS_1.alt0
D0	PATA_DATA0.alt0
D1	PATA_DATA1.alt0
D2	PATA_DATA2.alt0
D3	PATA_DATA3.alt0
D4	PATA_DATA4.alt0
D5	PATA_DATA5.alt0
D6	PATA_DATA6.alt0
D7	PATA_DATA7.alt0
D8	PATA_DATA8.alt0
D9	PATA_DATA9.alt0
D10	PATA_DATA10.alt0
D11	PATA_DATA11.alt0
D12	PATA_DATA12.alt0
D13	PATA_DATA13.alt0
D14	PATA_DATA14.alt0
D15	PATA_DATA15.alt0

7.5.4.3 IOMUX Configuration for SATA

The interface signals of the SATA PHY are not configured in the IOMUX. The SATA PHY interface uses dedicated contacts on the IC. See the i.MX53 data sheet for details.

7.5.4.4 Redundant Boot Support for Hard Disk and SSD

i.MX53 ROM supports redundant boot for hard disk and SSD. Primary or Secondary image is selected depending on PERSIST_SECONDARY_BOOT setting (see Table 7-7).

If PERSIST_SECONDARY_BOOT is 0, the boot ROM uses address 0x0 for primary image.

If PERSIST_SECONDARY_BOOT is 1, the boot ROM will read secondary image table from address 0x200 on boot media and will use address specified in the table.

Table 7-21. Secondary Image Table Format

Reserved (chipNum)

Table continues on the next page...

Table 7-21. Secondary Image Table Format (continued)

Reserved (driveType)
tag
firstSectorNumber
Reserved (sectorCount)

Where:

- tag: used as indication of valid secondary image table. Must be 0x00112233.
- firstSectorNumber is the first 512B sector number of the secondary image.

For secondary image support, the primary image must reserve space for secondary image table. See Figure 7-12 for typical structures layout on expansion device.

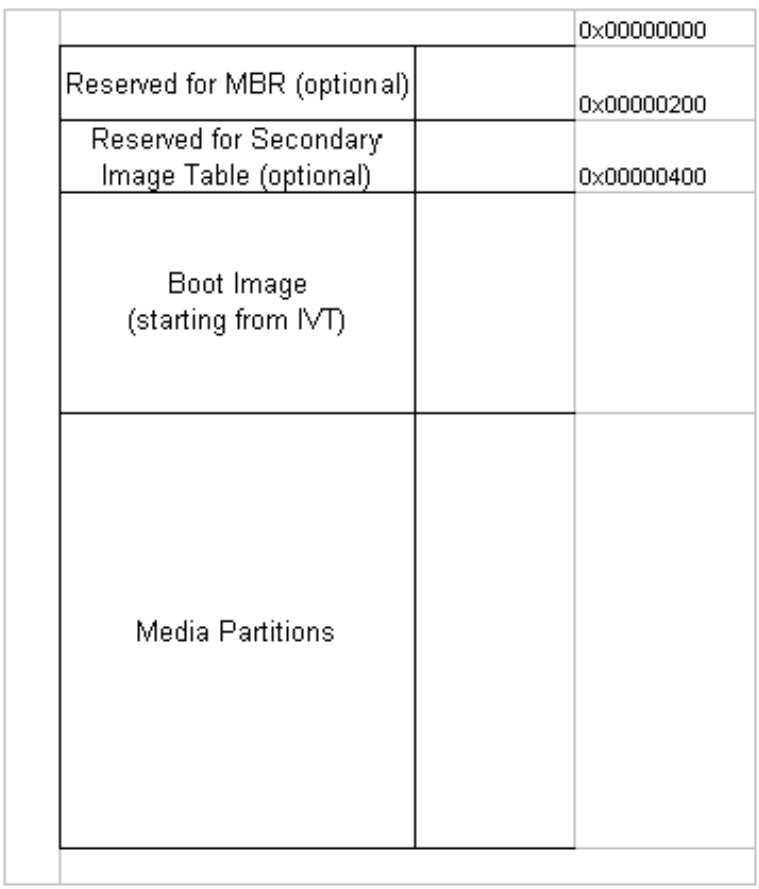


Figure 7-12. Hard Disk Structures Layout

For Closed mode, if there are failures during primary image authentication, the boot ROM will turn on PERSIST_SECONDARY_BOOT bit (see Table 7-7) and perform software reset. (After software reset secondary image will be used).

7.5.5 Serial ROM through SPI and I2C

The i.MX53 supports boot from serial memory devices, such as EEPROM, and Serial Flash using the SPI

(CSPI, ECSPI-1, ECSPI-2), and I2C Controller (I2C-1, I2C-2 and I2C-3) interfaces.

7.5.5.1 Serial ROM eFUSE Configuration

The boot ROM code determines the type of device using the following parameters, either provided by eFUSE settings or sampled on the I/O pins, during boot (See Table 7-22 for details):

Table 7-22. Serial ROM Boot eFUSE Descriptions

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings
BOOT_CFG1[7:4]	OEM	Boot Device Selection	Yes	0000	0011 - Boot from Serial ROM
BOOT_CFG1[3]	OEM	Serial ROM select	Yes	0	0 - I2C 1- SPI
BOOT_CFG2[5]	OEM	SPI Addressing	Yes	0	0 - 2-bytes (16-bit) 1 - 3-bytes (24-bit)
BOOT_CFG3[5:4]	OEM	Port Select	Yes	00	00 - I2C-1 / ECSPI-1 01 - I2C-2 / ECSPI-2 10 - I2C-3 / CSPI 11 - Reserved
BOOT_CFG3[3:2]	OEM	CS select (SPI only)	Yes	00	00 - CS#0 01 - CS#1 10 - CS#2 11 - CS#3

1. Setting can be overridden by GPIO settings when BT_FUSE_SEL fuse is intact. See Table 7-3 for corresponding GPIO pin.

The I2C-1/I2C-2/I2C-3 block can be used as boot device using I2C interface, for serial ROM boot. The I2C interface is configured to operate at 384 Kbps.

The boot ROM will copy 2 Kbyte of data from Serial ROM device to internal RAM. After checking the Image Vector Table header value (0xD1) from Program Image, the ROM code performs a DCD check. After successful DCD extraction, the Rom code extracts from Boot Data Structure the destination pointer and length of image to be copied to RAM device from where code execution occurs.

NOTE

The Initial 2K of Program Image must contain the IVT, DCD and the Boot Data structures.

7.5.5.2 I2C Boot

The boot flow when booting from an I2C device is shown in Figure 7-13. The boot ROM code reads the fuses BOOT_CFG1[7:4] (Boot Device Selection) and BOOT_CFG1[3] (Serial ROM select) to detect EEPROM device type. The ROM program copies 2K data from the EEPROM device to internal RAM. The boot ROM code next copies the initial 2 Kbyte of data as well as rest of image directly to application destination extracted from application image.

The i.MX53 uses the Device Select Code/Device Address in Table 7-23 to boot from an EEPROM.

Table 7-23. EEPROM via I2C Device Select Code

Bits	Device Type Identifier				Chip Enable Address[1]			R/W
	7	6	5	4	3	2	1	0
Device Select Code	1	0	1	0	0	0	0	R/W

1. These address bits, should be configured at the memory device, to match this '000' value.

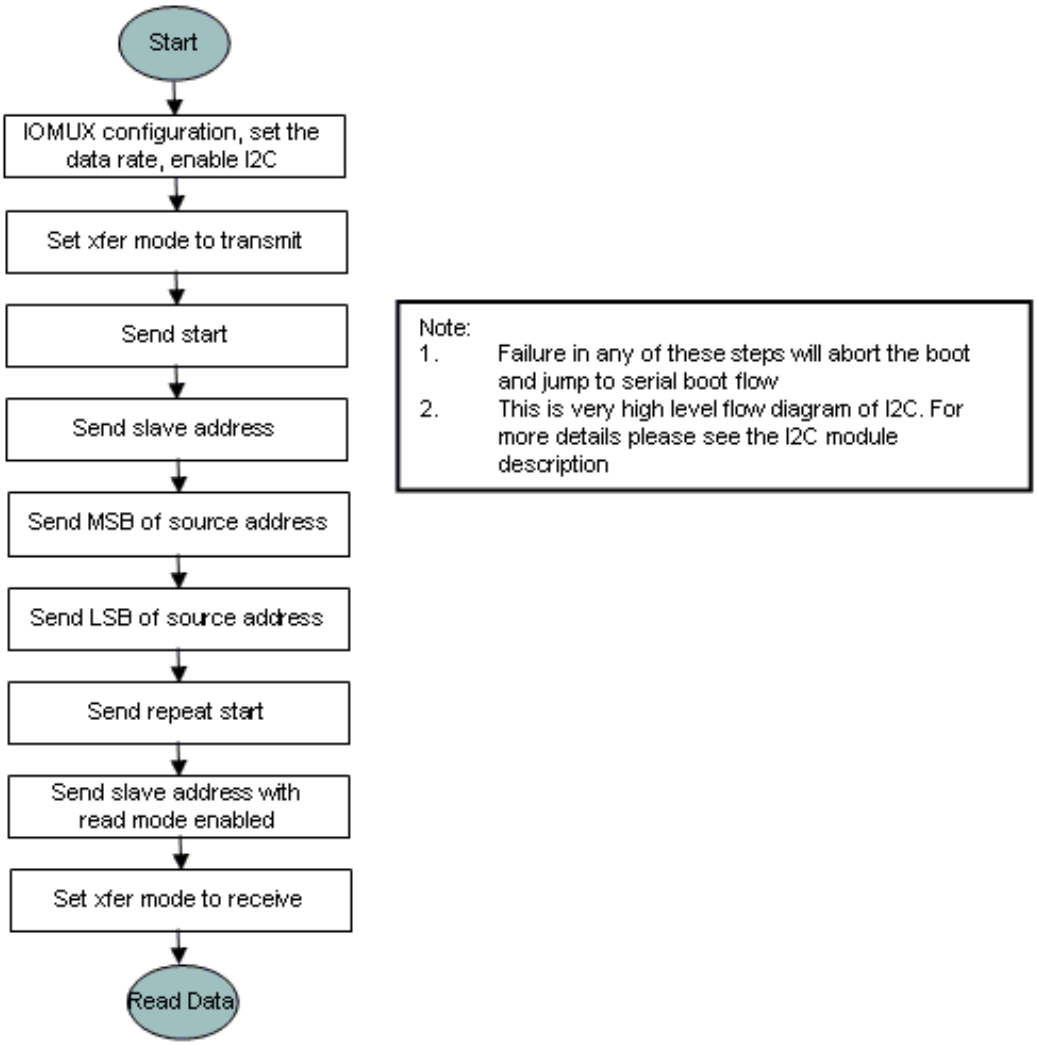


Figure 7-13. I2C Flow Chart

7.5.5.2.1 I2C IOMUX Pin Configuration

The contacts assigned to the signals used by the three I2C blocks is shown in Table 7-24.

Table 7-24. I2C IOMUX Pin Configuration

Signal	I2C-1	I2C-2	I2c-3
SDA	EIM_D28.alt5	EIM_D16.alt5	EIM_D18.alt5
SCL	EIM_D21.alt5	EIM_EB2.alt5	EIM_D17.alt5

7.5.5.3 CSPI Boot

The Configurable SPI (CSPI) interface is configured in Master mode and the EEPROM device is connected to CSPI interface as slave. The boot ROM code copies 2 Kbyte data from EEPROM device to the internal RAM. If DCD verification is successful, the ROM code copies the initial 2 Kbyte data as well as the rest of image directly to application destination extracted from application image. The CSPI can read data from EEPROM using 2 or 3 byte addressing and its burst length is 32 bytes.

NOTE

The Serial ROM Chip Select Number is determined by BOOT_CFG3[3:2] (Chip Select) fuse

When using the SPI as boot device, the i.MX53 supports booting from both Serial EEPROMS and Serial Flash devices. The boot code determines which device is being used by reading the appropriate eFUSE/I/O values at boot (See Table 7-22 for details).

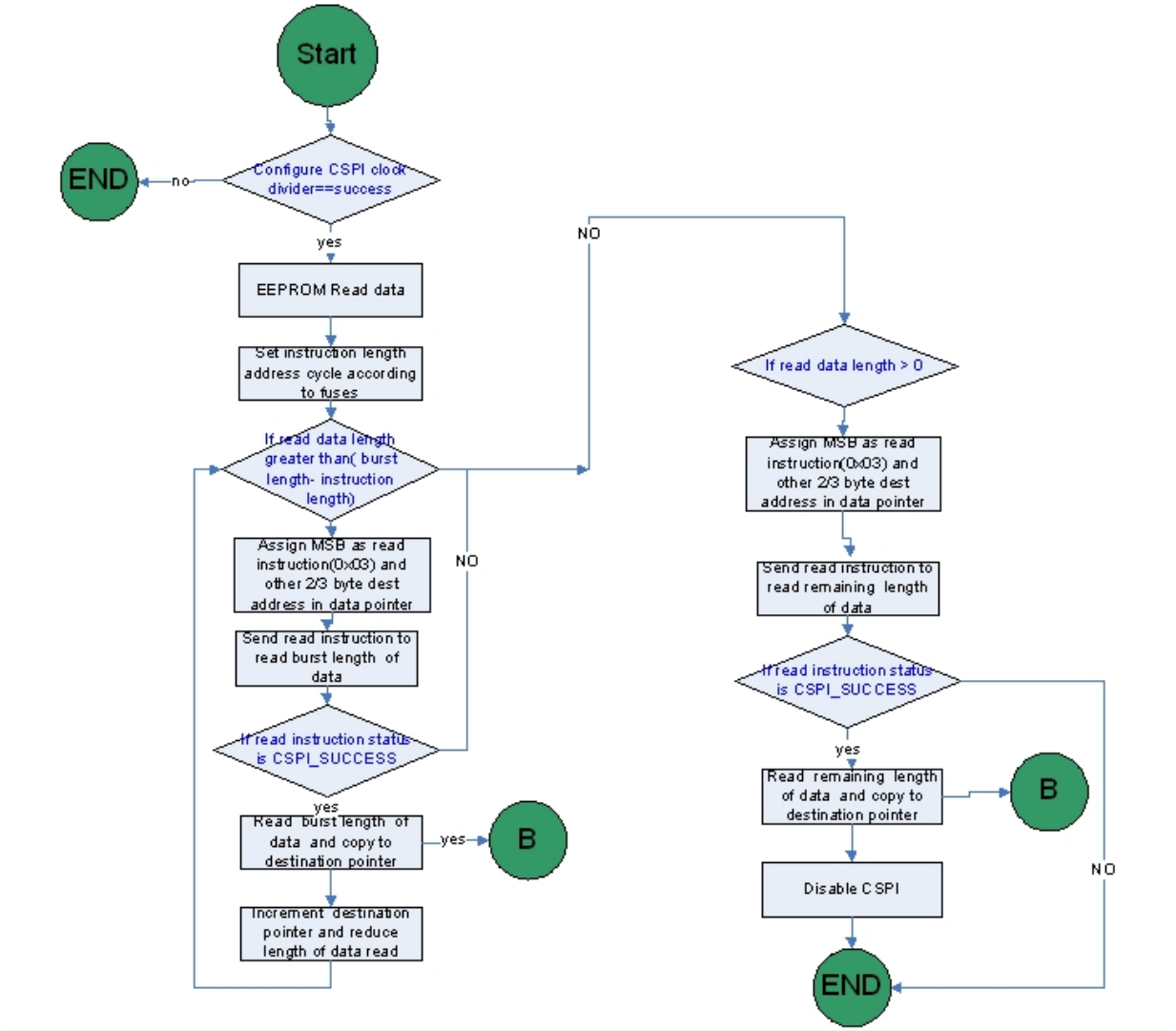


Figure 7-14. CSPI Flow chart

7.5.5.3.1 ECSPI/CSPI IOMUX Pin Configuration

The contacts assigned to the signals used by the three CSPI blocks is shown in Table 7-25.

Table 7-25. SPI IOMUX Pin Configuration

Signal	ECSPI-1	ECSPI-2	CSPI
MISO	EIM_D17.alt4	CSI0_DAT9.alt3	EIM_D22.alt4
MOSI	EIM_D18.alt4	CSI0_DAT10.alt3	EIM_D28.alt4
RDY	N/A[1]	N/A	N/A
SCLK	EIM_D16.alt4	CSI0_DAT8.alt3	EIM_D21.alt4
SS0	EIM_EB2.alt4	CSI0_DAT11.alt3	EIM_D20.alt4

Table continues on the next page...

Table 7-25. SPI IOMUX Pin Configuration (continued)

SS1	EIM_D19.alt4	EIM_LBA.alt2	EIM_A25.alt4
SS2	EIM_D25.alt3	EIM_D25.alt6	EIM_D24.alt4
SS3	EIM_D26.alt3	EIM_D26.alt6	EIM_D24.alt4

1. N/A in the ROM code indicates the pins are not available.

7.6 Program Image

This section describes the data structures that are required to be included in a user's Program Image. A Program Image consists of:

- Image Vector Table- A list of pointers located at a fixed address that the ROM examines to determine where other components of the Program Image are located.
- Boot Data- A table indicating the Program Image Location, Program Image size in bytes and plugin flag.
- Device Configuration Data- IC configuration data.
- User code and data

7.6.1 Image Vector Table and Boot Data

The Image Vector Table (IVT) is the data structure that the ROM reads from the boot device supplying the program image containing the required data components to perform a successful boot. The IVT includes the program image entry point, a pointer to Device Configuration Data (DCD) and other pointers used by the ROM during the boot process. The ROM locates the IVT at a fixed address that is determined by the boot device connected to i.MX53. The IVT offset from the base address and initial load region size for each boot device type is defined in Table 7-26. The location of the IVT is the only fixed requirement by the ROM. The remainder or the image memory map is flexible and is determined by the contents of the IVT.

Table 7-26. Image Vector Table Offset and Initial Load Region Size

Boot Device Type	Image Vector Table Offset	Initial Load Region Size
NOR	4 Kbyte = 0x1000 bytes	Entire Image Size
NAND	1 Kbyte = 0x400 bytes	4 Kbyte
OneNAND	256 bytes = 0x100 bytes	1 Kbyte
SD/MMC/eSD/eMMC	1 Kbyte = 0x400 bytes	2 Kbyte
I2C/HS-I2C/SPI EEPROM	1 Kbyte = 0x400 bytes	2 Kbyte

Table continues on the next page...

Table 7-26. Image Vector Table Offset and Initial Load Region Size (continued)

PATA/SATA	1 Kbyte = 0x400 bytes	2 Kbyte

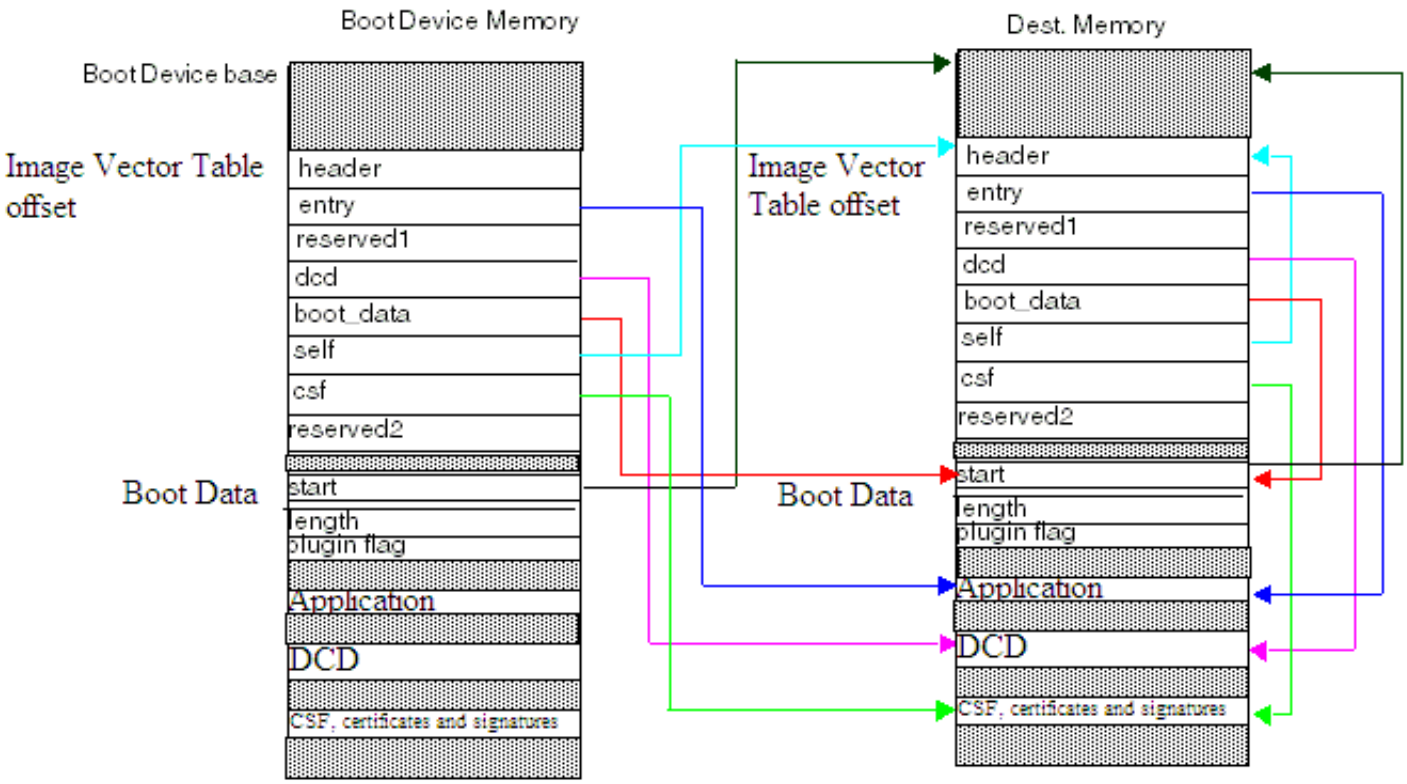


Figure 7-15. Image Vector Table

7.6.1.1 Image Vector Table Structure

The IVT has the following format (each entry is a 32 bit word):

Table 7-27. i.MX53 IVT Format

header
entry
reserved1
dcd
boot data
self
csf
reserved2

- header: The IVT header has the following format:

Table 7-28. IVT Header Format

Tag	Length	version

where:

```
Tag: A single byte field set to 0xD1
Length: a two byte field in big endian format containing the overall length of the IVT,
in bytes, including the header. (the length is fixed and must have a value of
32 bytes)
Version: A single byte field set to 0x40
```

- entry: Absolute address of the first instruction to execute from the image.
- reserved1: Reserved and should be zero.
- dcd: Absolute address of the image DCD. The DCD is optional so this field may be set to NULL if no DCD is required. See Device Configuration Data (DCD) for further details on DCD.
- boot data: Absolute address of the Boot Data
- self: Absolute address of the IVT. Used internally by the ROM.
- csf: Absolute address of Command Sequence File (CSF) used by the HAB library. See High Assurance Boot (HAB) for details on secure boot using HAB. This field must be set to NULL when not performing a secure boot.
- reserved2: Reserved and should be zero.

7.6.1.2 Boot Data Structure

The Boot Data must follow the format defined in Table 7-29, each entry is a 32-bit word.

Table 7-29. i.MX53 Boot Data Format

start
length
plugin

Where:

- start: Absolute address of the image
- length: Size of the program image
- plugin: Plugin flag

7.6.2 Device Configuration Data (DCD)

Upon reset the i.MX53 uses the default register values for all peripherals in the system. However, these settings typically are not ideal for achieving optimal system performance and there are even some peripherals that must configured before they can be used. The DCD is configuration information contained in a Program Image, external to the ROM, that the ROM interprets to configure various peripherals on i.MX53. For example, the EIM default settings allow core to interface to a NOR flash device immediately out of reset. This allows i.MX53 to interface with any NOR flash device, but has the cost of slow performance. Additional, some components such as SDRAM require some sequence of register programming as part of configuration before it is ready to be used. The DCD feature can be used to program the EIM registers and Enhanced SDRAM Controller (ESDCTL) registers to the optimal settings.

The ROM determines the location of the DCD table based on information located in the Image Vector Table (IVT). See Image Vector Table and Boot Data for more details. The DCD table shown in Table 7-30 is a big endian byte array of the allowable DCD commands. The maximum size of the DCD limited to 1768 bytes.

Table 7-30. DCD Data format

Header
[CMD]
[CMD]
...

The DCD header is 4 bytes with the following format:

Table 7-31. DCD Header

Tag	Length	Version

where:

```
Tag: A single byte field set to 0xD2
Length: a two byte field in big endian format containing the overall length of the DCD,
in bytes, including the header
Version: A single byte field set to 0x40
```

7.6.2.1 Write Data Command

The Write Data Command is used to write a list of given 1, 2 or 4-byte values or bitmasks to a corresponding list of target addresses. The format of Write Data Command, again a big endian byte array, is shown in Table 7-32.

Table 7-32. Write Data Command Format

Tag	Length	Parameter
Address		
Value/Mask		
[Address]		
[Value/Mask]		
...		
[Address]		
[Value/Mask]		

where:

```
Tag: A single byte field se to 0xCC
Length: A two byte field in big endian format containing the length of the Write Data
Command, in bytes, including the header
Address: target address to which data should be written
Value/Mask: data value or bitmask to be written to preceding address
```

The Parameter field is a single byte divided into bitfields as follows:

Table 7-33. Write Data Command Parameter field

7	6	5	4	3	2	1	0
flags					bytes		

where

```
bytes: width of target locations in bytes. Either 1, 2 or 4
flags: control flags for command behavior.
Data Mask = bit 3: if set, only specific bits may be overwritten at target address
(otherwise all bits may be overwritten)
Data Set = bit 4: if set, bits at the target address overwritten with this flag
(otherwise it is ignored)
```

One or more target address and value/bitmask pairs can be specified. The same bytes and flags parameters apply to all locations in the command.

When successful, this command writes to each target address in accordance with the flags as follows:

Table 7-34. Interpretation of Write Data Command Flags

"Mask"	"Set"	Action	Interpretation
0	0	*address = val_msk	Write value
0	1	*address = val_msk	Write value
1	0	*address &= ~val_msk	Clear bitmask
1	1	*address |= val_msk	Set bitmask

NOTE

If any of the target addresses does not have the same alignment as the data width indicated in the parameter field, none of the values are written.

If any of the values is larger or any of the bitmasks is wider than permitted by the data width indicated in the parameter field, none of the values are written.

If any of the target addresses do not lie within an allowed region, none of the values are written. The list of allowable blocks and target addresses for i.MX53 are given below.

Table 7-35. Valid DCD Address Ranges

Address range	Start address	Last Address
CCM register set	0x53FD4000	0x53FD7FFF
EIM register set	0x63FDA000	0x63FDAFFF
NANDFC register set	0xF7FF0000	0xF7FFFFFF
IOMUX Control (IOMUXC) registers	0x53FA8000	0x53FABFFF
DPLLC1 register	0x63F80000	0x63F83FFF
DPLLC2 register	0x63F84000	0x63F87FFF
DPLLC3 register	0x63F88000	0x63F8BFFF
DPLLC4 register	0x63F8C000	0x63F8FFFF
ESD RAM controller register	0x63FD9000	0x63FD9FFF
M4IF register	0x63FD8000	0x63FD8FFF
DDR	0x70000000	0xEFFFFFFF
EIM	0xF0000000	0xF7FEFFFF
NANDFC Buffers	0xF7FF0000	0xF7FFFFFF
IRAM Free Space	0xF8006000	0xF8017FF0
GPU Memory	0xF8020000	0xF805FFFF

7.6.2.2 Check Data Command

The Check Data Command is used to test for a given 1, 2 or 4-byte bitmasks from a source address. The Check Data Command is a big endian byte array with format shown in Table 7-36.

Table 7-36. Check Data Command Format

Tag	Length	Parameter

Table continues on the next page...

Table 7-36. Check Data Command Format (continued)

Address
Mask
[Count]

where:

```
Tag: A single byte field se to 0xCF
Length: A two byte field in big endian format containing the length of the Check Data
Command, in bytes, including the header
Address: source address to test
Mask: bit mask to test
Count: optional poll count. If count is not specified this command will poll indefinitely
until the exit condition is met. If count = 0, this command behaves as for NOP.
```

The Parameter field is a single byte divided into bitfields as follows:

Table 7-37. Check Data Command Parameter field

7	6	5	4	3	2	1	0
flags					bytes		

where

```
bytes: width of target locations in bytes. Either 1, 2 or 4
flags: control flags for command behavior.
Data Mask = bit 3: if set, only specific bits may be overwritten at target address
(otherwise all bits may be overwritten)
Data Set = bit 4: if set, bits at the target address overwritten with this flag
(otherwise it is ignored)
```

This command polls the source address until either the exit condition is satisfied, or the poll count is reached. The exit condition is determined by the flags as follows

Table 7-38. Interpretation of Check Data Command Flags

"Mask"	"Set"	Action	Interpretation
0	0	(*address & mask) == 0	All bits clear
0	1	(*address & mask) == mask	All bits set
1	0	(*address & mask)!= mask	Any bit clear
1	1	(*address & mask)!= 0	Any bit set

NOTE

If the source address does not have the same alignment as the data width indicated in the parameter field, the value is not read.

If the bitmask is wider than permitted by the data width indicated in the parameter field, the value is not read.

See DCD Table Structure of Device Configuration Data in DCD Example

7.6.2.3 NOP Command

This command has no effect. The format of Check Data Command is a little endian four byte array as shown in Table 7-36

Table 7-39. NOP Command Format

Tag	Length	Undefined

where:

```
Tag: A single byte field se to 0xC0
Length: A two byte field containing the length of the NOP Command in bytes. Fixed to a
value of 4.
Undefined: This byte is ignored and can be set to any value.
```

7.7 Plugin Image

i.MX53 ROM supports a limited number of boot devices. For using other devices as boot source (for example, ethernet, CDROM, or USB), the supported boot device must be used (typically serial ROM) for firmware with the missing boot drivers.

Boot ROM will detect the image type by plugin flag of boot data structure (see Boot Data Structure). If the plugin flag is 1, then ROM will use the image as plugin function. The function must initialize boot device and copy program image to final location. At the end the plugin function must return with the parameters of program image.

The boot ROM will authenticate the plugin image prior running plugin function and then will authenticate the program image.

The plugin function must follow the API described below:

typedef unsigned char (*) plugin_download_f(void **start, size_t *bytes, UINT32 *ivt_offset)

ARGUMENTS PASSED:

- start - Image load address on exit.
- bytes - Image size on exit.
- ivt_offset - Offset in bytes of the IVT from the image start address on exit.

RETURN VALUE:

- 1 - on success
- 0 - on failure

7.8 Serial Downloader (BOOT_MODE[1:0] = 0b11)

The Serial Downloader provides a means to download a Program Image to i.MX53 over a UART or USB serial connection. In this mode the ROM programs WDOG-1 for a 32-second time-out if WDOG_ENABLE eFuse is 1, and then continuously polls for USB and UART actively in an alternating fashion. If no activity is found on either USB or UART and the watchdog timer expires the ARM core is reset.

NOTE

The downloaded image must continue to service the watchdog timer to avoid an undesired reset from occurring.

The USB/UART boot flow is shown in Figure 7-16.

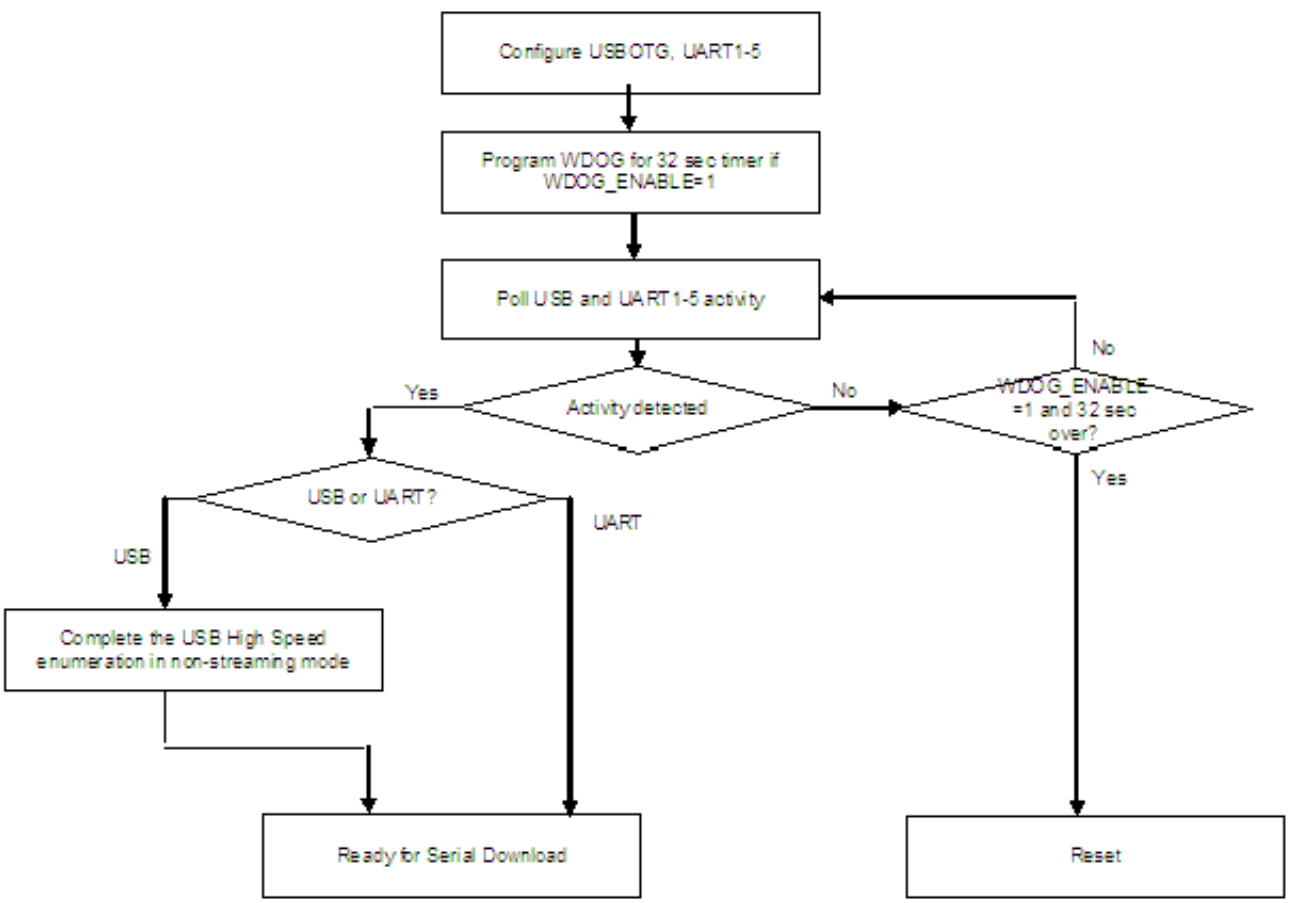


Figure 7-16. USB/UART Boot Flow

7.8.1 USB

The USB support is composed of the USBOH3 (USB OTG core controller, compliant with the USB 2.0 specification and three additional USB hosts) and the USBPHY (HS USB transceiver).

The USB OTG port is the bootable USB device. Additional USB hosts are intended for the intraplatform communication and not bootable.

To enumerate the ROM boot USB function the host PC needs a compatible windows driver and a specific application "Manufacturing Tool". Once enumerated, the USB boot device uses a specific protocol "Download protocol" to download and launch the application.

For USB boot, the USB controller is used, with the integrated PHY. The Boot ROM USB driver configures the USB controller to function in High Speed non-stream bulk mode with 512B maximal packet size. The control endpoints EP0IN and EP0OUT are configured for control transfer and for data transfer in bulk mode EP1OUT and EP2IN are configured as IN and OUT transactions respectively. The supported transceivers are in Table 7-40.

Table 7-40. Supported USB Transceivers

Transceiver	Speed	Interface	USB Controller Mode
Internal PHY	High Speed (HS)	UTMI	High Speed (HS) non-stream bulk mode with a maximal packet size 512B

7.8.1.1 USB Configuration Details

The USB OTG function device driver supports a high speed (HS for UTMI) non-stream mode with a maximal packet size of 512 B and a low-level USB OTG function.

The VID/PID and strings for USB device driver are listed in Table 7-41.

Table 7-41. VID/PID and Strings for USB Device Driver

Descriptor	Value
VID	0x15A2 (Freescale vendor ID)
PID[1]	0x004E
Screen Descriptor1 (manufacturer)	Freescale Semiconductor, Inc.

Table continues on the next page...

Table 7-41. VID/PID and Strings for USB Device Driver (continued)

Descriptor	Value
Screen Descriptor2 (product)	S Blank RITA SE Blank RITA NS Blank RITA
Screen Descriptor4	Freescale Flash
Screen Descriptor5	Freescale Flash

1. Allocation based on BPN (Before Part Number)

7.8.1.2 IOMUX Configuration for USB

The interface signals of the UTMI PHY are not configured in the IOMUX. The UTMI PHY interface uses dedicated contacts on the IC. See the i.MX53 data sheet for details.

7.8.2 UART

All the UART (UART-1, UART-2, UART-3, UART-4 and UART-5) ports can be used as a boot device. The ROM boot functions polls all the ports for data activity indication. Once indicated, the UART boot device uses a specific protocol, called the Serial Downloader Protocol, to download and launch a program image. The UART driver uses the communication parameters listed in Table 7-42.

Table 7-42. UART Communication Parameters

Parameter	Value
Baud rate	115.2 kbaud
Parity check	Disabled
Word size	8 bits
Stop bits	1 bit
RTS	Ignored
CTS	Controlled by host
Receive pin	RXD1
Transmit pin	TXD1

7.8.2.1 IOMUX Configuration for UART

The contacts assigned to the signals used by the three UART blocks is shown in Table 7-43.

Table 7-43. UART IOMUX Pin Configuration

Signal	UART-1	UART-2	UART-3	UART-4	UART-5
TXD	CSI0_DAT10.alt2	PATA_DMARQ.alt 3	EIM_D25.alt2	CSI0_DAT12.alt2	CSI0_DAT14.alt2
RXD	CSI0_DAT11.alt2	PATA_BUFFER_EN.alt3	EIM_D24.alt2	CSI0_DAT13.alt2	CSI0_DAT15.alt2

NOTE

ROM polls activity on all UART ports.When activity is detected on one of these ports, ROM starts waiting for data on this particular port and never goes to another port. UART pads can be muxed with different devices, therefore board design should take this into account and don't connect to UART RX pads devices that can drive these pads during boot.

7.8.3 Serial Download Protocol

This section describes the serial download protocol used for all boot devices in the serial bootloader on the i.MX53.

Each stage in the protocol begins with a command issued by the host to the device, followed by a response from the device to the host. For most commands, that completes the protocol stage. The exception is the Write File command, which has an additional data stream sent from the host to the device after the response.

The protocol is terminated by the host issuing a Write File command with application file type. After processing the Write File commands, the device interprets the next command as a Completed command, and resumes execution of the boot flow in order to authenticate the downloaded application (if required) and then execute it.

7.8.3.1 Get Status

The Get Status command retrieves the error log stored during a failed boot (or the success code when the Serial bootloader is selected deliberately).

Table 7-44. Get Status Command

Command	0x05	0x05	-	-	-	-	-	-	-	-	-
Response	SC	SC	SC	SC	SC						

The fields have the following interpretation:

- SC Status Code. Four copies of the status code are sent.
- - Don't Care (but must be present)

7.8.3.2 Read Memory

The Read Memory command reads a stream of bytes, half-words or words of data starting from a given address in memory. For Closed security configuration, this command is ignored.

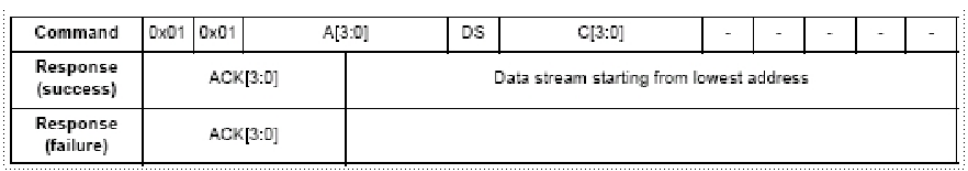


Figure 7-17. Read Memory Command

The fields have the following interpretation:

- A[3:0] = Target address (most-significant byte first).
- DS = Data Size (0x08 = byte, 0x10 = half-word, 0x20 = word). Unsupported DS values result in a failure response.
- C[3:0] = Number of data elements (bytes, half-words or words as appropriate) in data stream (most-significant byte first).
- ACK[3:0] = 0x56, 0x78, 0x78, 0x56.
- - = Don't Care (but must be present)

7.8.3.3 Write Memory

The Write Memory command writes a byte, half-word or word of data to a given address in memory.

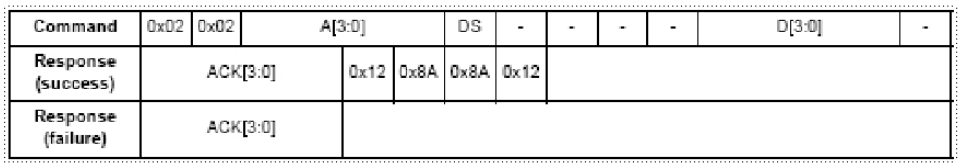


Figure 7-18. Write Memory Command

The fields have the following interpretation:

- A[3:0] = Target address (most-significant byte first).
- DS = Data Size (0x08 = byte, 0x10 = half-word, 0x20 = word). Unsupported DS values result in a failure response.
- D[3:0] = Data to write(most-significant byte first). For DS = 0x08, only D[0] is used. For DS = 0x10, only D[1:0] is used.
- ACK[3:0] = 0x12, 0x34, 0x34, 0x12 for Closed security configuration, and 0x56, 0x78, 0x78, 0x56 otherwise.
- - = Don't Care (but must be present)

For Closed security configuration, the address ranges which may be written are restricted to those listed for DCD, DDR memory space and internal memory regions (IRAM, GMEM and NFC buffer). Attempts to write outside of the valid ranges are ignored and result in the failure response. For other security levels, no restrictions apply to the target address.

7.8.3.4 Re-enumerate

The Re-enumerate command resets the USB connection with an updated descriptor.

Table 7-45. Re-enumerate Command

Command	0x09	0x09	-	-	-	-	-	-	-	SN[3:0]	-
Response	0x89	0x23	0x23	0x89							

The fields have the following interpretation:

- SN[3:0] = Serial number for enumeration descriptor.
- - = Don't Care (but must be present)

7.8.3.5 Write File

The Write File command writes a stream of bytes to a given address in memory. The byte stream may be assigned a file type in order to distinguish CSF, DCD and application files. An application file type leads to the termination the serial download protocol, with the next command (whatever the command byte values) being treated as the Completed command.

Command	0x04	0x04	A[3:0]	-	C[3:0]	-	-	-	-	FT
Response	ACK[3:0]									
Data	Byte stream with data for lowest address first									

Figure 7-19. Write File Command

The fields have the following interpretation:

- A[3:0] = Target address (most-significant byte first).
- C[3:0] = Number of bytes in data stream (most-significant byte first).
- FT= File Type (0xAA = application (terminates protocol), 0xCC = CSF, 0xEE = DCD). With unrecognized FT values, the file is still downloaded, but the pointers to the three essential files are not modified.
- ACK[3:0] = 0x12, 0x34, 0x34, 0x12 for Closed security configuration, and 0x56, 0x78, 0x78, 0x56 otherwise.
- - = Don't Care (but must be present)

For Closed security configuration, the address ranges which may be written are restricted to those listed for DCD DDR memory space and internal memory regions (IRAM, GMEM and NFC buffer). Attempts to write outside of the valid ranges are ignored and result in the failure response. For other security levels, no restrictions apply to the target address.

7.8.3.6 Completed

The Completed command is required after the Write File command with application file type in order to terminate the protocol. The content of the command is irrelevant, but a command must be sent. This command triggers authentication and DCD processing (if required) followed by execution of the application using entry pointer from IVT (see Image Vector Table Structure). For Closed security configuration, if application authentication fails, the serial download protocol resumes, with the status code for authentication failure available through the Get Status command.

Table 7-46. Completed Command

Command	-	-	-	-	-	-	-	-	-	-	-
Response	0x88	0x88	0x88	0x88							

The fields have the following interpretation:

- - = Don't Care (but must be present)

7.9 Watchdog Reset Boot Mode

For testing purposes, shadow SBMR register and SW watchdog reset can be used. SW can write to PERSIST_SBMR_SHADOW register value to select override boot mode settings. Then PERSIST_WDOG_RESET must be set and SW reset must be performed (see Table 7-7).

In case of SW reset the ROM code checks if PERSIST_WDOG_RESET is set. If it is set it uses PERSIST_SBMR_SHADOW register. Otherwise it uses SRC shadow register.

This mode is disabled if DIR_BT_DIS eFuse is blown.

7.10 High Assurance Boot (HAB)

The High Assurance Boot (HAB) component of the ROM protects against the potential threat of attackers modifying areas of code or data in programmable memory to make it behave in an incorrect manner. The HAB also prevents attempts to gain access to features which should not be available. The integration of the HAB feature with the ROM code ensures that i.MX53 does not enter an operational state if the existing hardware security blocks have detected a condition that may be a security compromise or areas of memory deemed to be important have been modified. The HAB uses RSA digital signatures to enforce these policies.

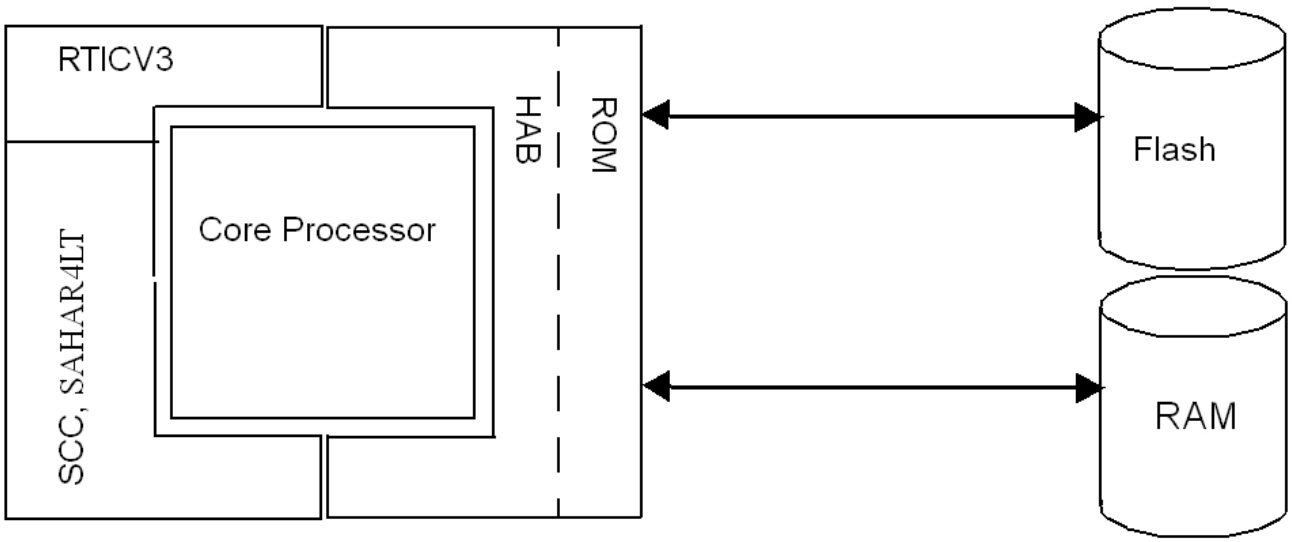


Figure 7-20. Secure Boot Components

Figure 7-20 illustrates the components used during a secure boot using HAB. The HAB interfaces with the SCC to ensure the system security state is as expected. The HAB also makes use of SAHARA and/or RTIC hardware blocks to accelerate SHA-256 message digest operations performed during signature verifications. The HAB also includes a software implementation of SHA-256 for cases where a hardware accelerator cannot be used. The core RSA signature verification operations are performed by a software implementation contained in the HAB library. The main features supported by HAB are:

- X.509 Public key certificate support
- CMS signature format support

NOTE

For further details on making use of the i.MX53 secure boot feature using HAB contact your local Freescale representative.

7.10.1 ROM Vector Table Addresses

For devices that perform a secure boot, the HAB library may be called by boot stages that execute after ROM code. The RVT table contains the pointers to the HAB API functions and is located at 0x00000094

NOTE

For additional information on secure boot on i.MX53 including the HAB API contact your local Freescale representative

7.10.2 SRTC Initialization

NOTE

Refer to the i.MX53 Security Reference Manual for information on Secure Real Time Clock (SRTC) initialization. For details on SRTC initialization when performing a secure boot with HAB contact your local Freescale representative.

7.11 Examples

7.11.1 NAND eFuse Configuration Example

Table 7-47. NAND eFuse Configuration Example

Fuse	Definition	Setting	Description
BOOT_CFG1[7]	Boot Device Selection	1	NAND device is used
BOOT_CFG1[6]	Muxed On	1	NAND data bus uses EIM pads EIM_DA[16:0]
BOOT_CFG1[5:4]	Interleaving Scheme	00	No Interleaving. ROM code will use CS0 only for reading data from NAND device.
BOOT_CFG1[3:2]	Address Cycles	10	5 address cycles
BOOT_CFG2[7:6]	Page Size	01	Page size is 2KB + 64 Bytes spare area, 4-bit ECC is used
BOOT_CFG2[5]	NAND Interface Bus Width	0	8 bit device is used
BOOT_CFG2[2]	NFC_FREQ_SEL	1	NAND frequency will be (AXI DDR divide by 28): 14.29MHz if AXI/DDR Freq eFuse is 0 11.9MHz if AXI/DDR Freq eFuse is 1).
BOOT_CFG3[7]	Stride Size (Bad Block skip step)	1	FCB search stride is 8 blocks
BOOT_CFG3[6]	LBA-Nand Select	0	Non LBA device. 11ms delay will be used after device reset.
BOOT_CFG3[5]	NAND Use R/B Signals:	1	Use R/B signals.
BOOT_CFG3[4:3]	ECC / Spare select:	00	4-bit ECC
BOOT_CFG3[2:1]	Pages In Block	01	NAND device has 64 pages in block

7.11.2 DCD Example

DCD Table Structure

```
#define HWC_LINES 63
#define HWC_WORDS (HWC_LINES*2+2)
#define HWC_VALUE (HWC_LINES*3)
#define HWC_BYTES (HWC_VALUE*4)
#define LE_2_BE_32(word) \
    (((word&0x000000FF) << 24)|         \
      ((word&0x0000FF00) << 8) |        \
      ((word&0x00FF0000) >> 8) |        \
      ((word&0xFF000000) >> 24)  )
#define LE_2_BE_16(word) \
    ( ((word&0x00FF) << 8) |            \
      ((word&0xFF00) >> 8) )
typedef unsigned int UINT32;
const unsigned int hwc[HWC_WORDS] = {
    (UINT32)0xD2<<0 |
    LE_2_BE_16(HWC_WORDS*4)<<8 |
    (UINT32)4<<28 |
    (UINT32)0<<24,
    (UINT32)0xCC<<0 | LE_2_BE_16(HWC_WORDS*4-4)<<8 | 4<<24,
    /* DDR2 IOMUX configuration */
    LE_2_BE_32(0x53fa8554), LE_2_BE_32(0x00380000),  /* IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM3 */
    LE_2_BE_32(0x53fa8558), LE_2_BE_32(0x00380040),  /* IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS3 */
    LE_2_BE_32(0x53fa8560), LE_2_BE_32(0x00380000),  /* IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM2 */
    LE_2_BE_32(0x53fa8564), LE_2_BE_32(0x00380040),  /* IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT1 */
    LE_2_BE_32(0x53fa8568), LE_2_BE_32(0x00380040),  /* IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS2 */
    LE_2_BE_32(0x53fa8570), LE_2_BE_32(0x00380000),  /* IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_1 */
    LE_2_BE_32(0x53fa8574), LE_2_BE_32(0x00380000),  /* IOMUXC_SW_PAD_CTL_PAD_DRAM_CAS */
    LE_2_BE_32(0x53fa8578), LE_2_BE_32(0x00380000),  /* IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_0 */
    LE_2_BE_32(0x53fa857c), LE_2_BE_32(0x00380040),  /* IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS0 */
    LE_2_BE_32(0x53fa8580), LE_2_BE_32(0x00380040),  /* IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT0 */
    LE_2_BE_32(0x53fa8584), LE_2_BE_32(0x00380000),  /* IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM0 */
    LE_2_BE_32(0x53fa8588), LE_2_BE_32(0x00380000),  /* IOMUXC_SW_PAD_CTL_PAD_DRAM_RAS */
    LE_2_BE_32(0x53fa8590), LE_2_BE_32(0x00380040),  /* IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS1 */
    LE_2_BE_32(0x53fa8594), LE_2_BE_32(0x00380000),  /* IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM1 */
    LE_2_BE_32(0x53fa86f0), LE_2_BE_32(0x00380000),  /* IOMUXC_SW_PAD_CTL_GRP_ADDDS */
    LE_2_BE_32(0x53fa86f4), LE_2_BE_32(0x00000200),  /* IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL */
    LE_2_BE_32(0x53fa86fc), LE_2_BE_32(0x00000000),  /* IOMUXC_SW_PAD_CTL_GRP_DDRPKE */
    LE_2_BE_32(0x53fa8714), LE_2_BE_32(0x00000000),  /* IOMUXC_SW_PAD_CTL_GRP_DDRMODE - CMOS
mode */
    LE_2_BE_32(0x53fa8718), LE_2_BE_32(0x00380000),  /* IOMUXC_SW_PAD_CTL_GRP_B0DS */
    LE_2_BE_32(0x53fa871c), LE_2_BE_32(0x00380000),  /* IOMUXC_SW_PAD_CTL_GRP_B1DS */
    LE_2_BE_32(0x53fa8720), LE_2_BE_32(0x00380000),  /* IOMUXC_SW_PAD_CTL_GRP_CTLDS */
    LE_2_BE_32(0x53fa8724), LE_2_BE_32(0x02000000),  /* IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE -
DDR_SEL0=  */
    LE_2_BE_32(0x53fa8728), LE_2_BE_32(0x00380000),  /* IOMUXC_SW_PAD_CTL_GRP_B2DS */
    LE_2_BE_32(0x53fa872c), LE_2_BE_32(0x00380000),  /* IOMUXC_SW_PAD_CTL_GRP_B3DS */

    /* Initialize DDR2 memory */
    LE_2_BE_32(0x63fd9088), LE_2_BE_32(0x2d313331), /* ESDCTL */
    LE_2_BE_32(0x63fd9090), LE_2_BE_32(0x393b3836), /* ESDCTL */
    LE_2_BE_32(0x63fd90F8), LE_2_BE_32(0x00000800), /* ESDCTL */
    LE_2_BE_32(0x63fd907c), LE_2_BE_32(0x020c0211), /* ESDCTL */
    LE_2_BE_32(0x63fd9080), LE_2_BE_32(0x014c0155), /* ESDCTL */
    LE_2_BE_32(0x63fd9018), LE_2_BE_32(0x000016d0), /* ESDCTL */
    LE_2_BE_32(0x63fd9000), LE_2_BE_32(0xc4110000), /* ESDCTL */
    LE_2_BE_32(0x63fd900C), LE_2_BE_32(0x4d5122d2), /* ESDCTL */
    LE_2_BE_32(0x63fd9010), LE_2_BE_32(0x92d18a22), /* ESDCTL */
    LE_2_BE_32(0x63fd9014), LE_2_BE_32(0x00c70092), /* ESDCTL */
    LE_2_BE_32(0x63fd902c), LE_2_BE_32(0x000026d2), /* ESDCTL */
    LE_2_BE_32(0x63fd9030), LE_2_BE_32(0x009f000e), /* ESDCTL */
    LE_2_BE_32(0x63fd9008), LE_2_BE_32(0x12272000), /* ESDCTL */
    LE_2_BE_32(0x63fd9004), LE_2_BE_32(0x00030012), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x04008010), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x00008032), /* ESDCTL */
```

```
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x00008033), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x00008031), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x0b5280b0), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x04008010), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x00008020), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x00008020), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x0a528030), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x03c68031), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x00468031), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x04008018), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x0000803a), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x0000803b), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x00008039), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x0b528138), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x04008018), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x00008028), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x00008028), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x0a528038), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x03c68039), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x00468039), /* ESDCTL */
    LE_2_BE_32(0x63fd9020), LE_2_BE_32(0x00005800), /* ESDCTL */
    LE_2_BE_32(0x63fd9058), LE_2_BE_32(0x00033337), /* ESDCTL */
    LE_2_BE_32(0x63fd901c), LE_2_BE_32(0x00000000)  /* ESDCTL */
};
```

Chapter 8
Multimedia

8.1 The Video/Graphics Sub-System

NOTE

The information in this chapter takes precedence if any discrepancy between the information here, and the information in the mentioned block guides (like IPU, GPU2D, GPU3D, VPU and so on) arises.

The video-graphics sub-system includes the following dedicated blocks (see Figure 8-1):

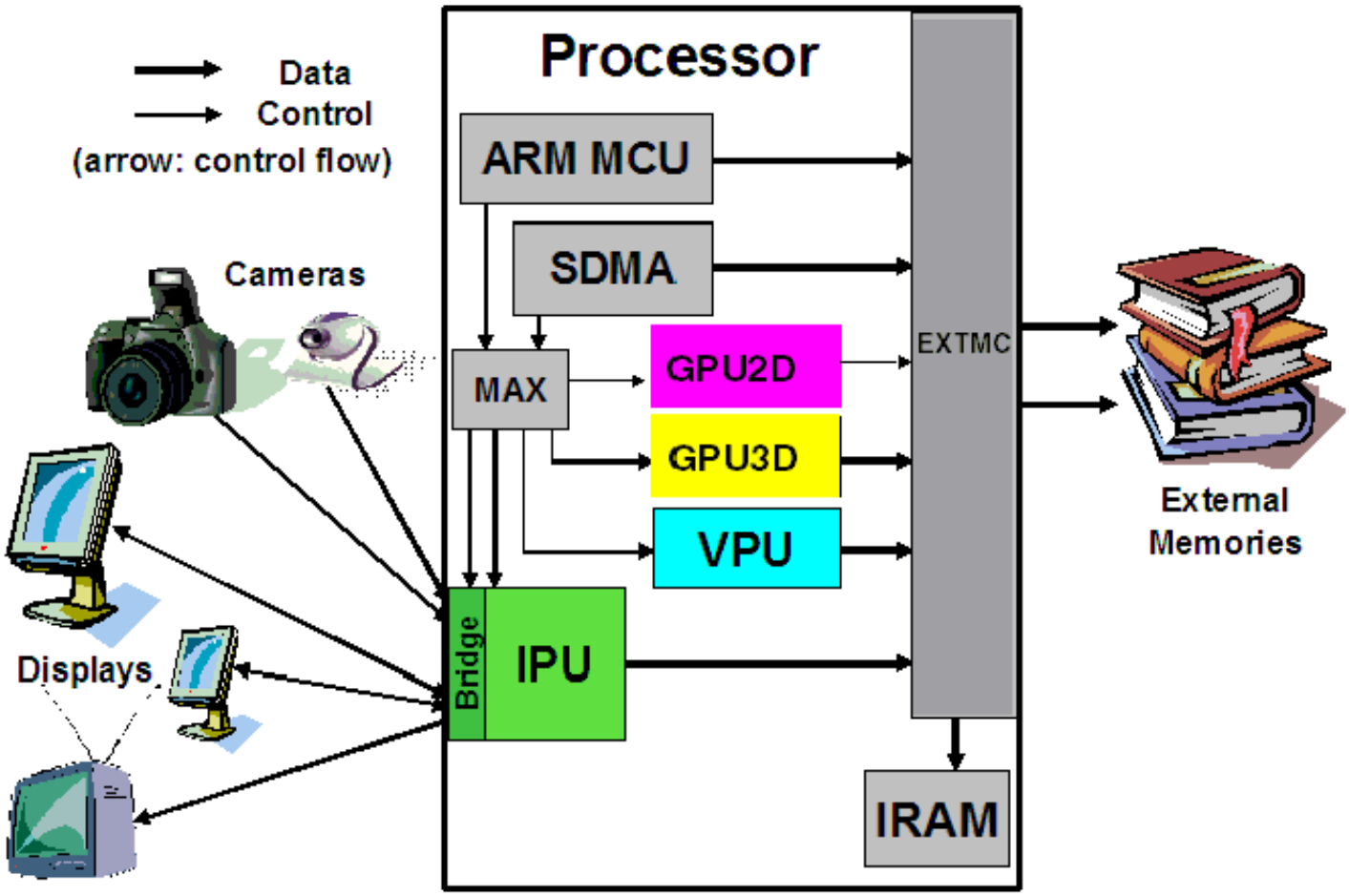


Figure 8-1. The Video-Graphics Subsystem

- Video Processing Unit (VPU): a multi-standard video/image codec
- Two Graphics Processing Units (GPUs): one for accelerating 3D graphics (OpenGL/ES), and one for accelerating vector graphics (OpenVG and 2D graphics BitBLT)
- Image Processing Unit (IPU): providing connectivity to cameras and displays, related processing, synchronization and control
- Display interface bridges: providing optional translation from the digital display interface supported by the IPU to other interfaces:
  - TV Encoder (TVE) bridge: composite, S-video, component and VGA interfaces
  - LVDS bridge: up to two LVDS interfaces

These blocks are connected to the other parts of the system as follows:

- The display ports are described in Figure 8-1.

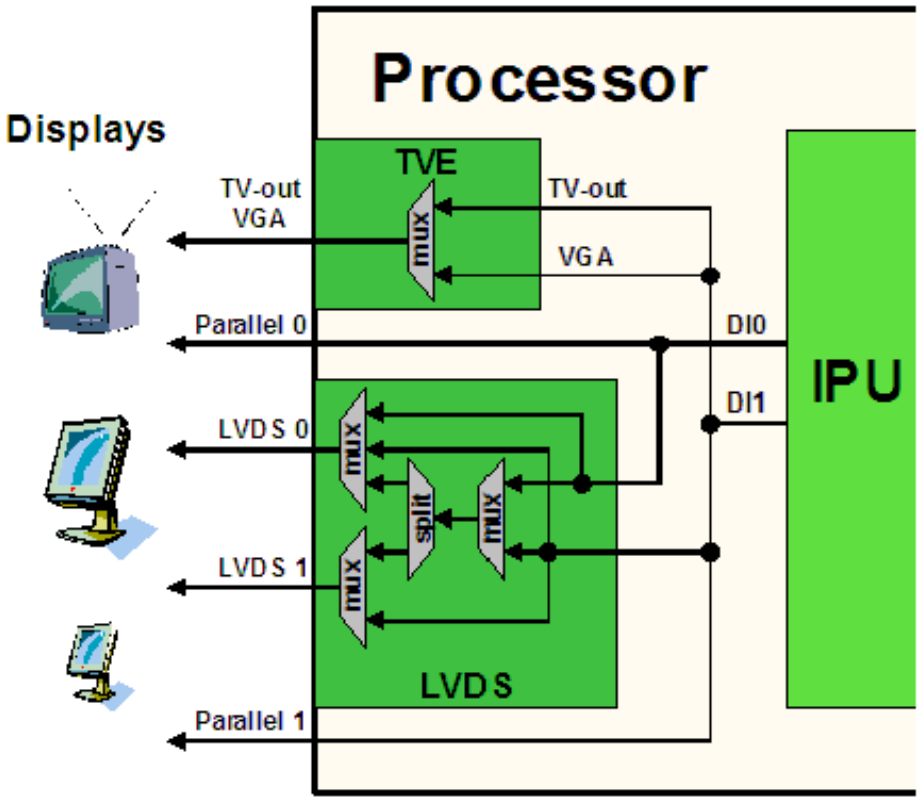


Figure 8-2. The Display Ports

There are five ports:

- Two parallel - driven directly by the IPU
- One analog (TV-out or VGA) - driven by the TVE
- Two LVDS channels, driven by the LVDS bridge

IPU has 2 display ports, so only up to two external ports can be active at any given time.

(Additional asynchronous data flows can be sent though the parallel ports.)

- IPU has 2 camera ports, used to connect to relevant external devices.
- The IPU, VPU and the GPUs have a master AXI port, providing access to system memory.
- The blocks are controlled by the ARM Cortex A8TM Platform (ARM) or the Smart Direct Memory Access (SDMA) as follows
  - The LVDS bridge is controlled by signals connected to top-level registers
  - All other blocks have host interfaces. The VPU and TVE blocks use slave IP ports and the GPU and IPU blocks use slave AHB ports.
- The slave AHB port of the GPUs also allows a direct access of the host to the GPUs internal memory, for the transfer of graphics commands, for example.

- The slave AHB port of the IPU also provides a direct access of the host to an external display controller, connected to the parallel display port.
- There is an interface between the IPU and the GPUs, allowing direct synchronization between them, to transfer graphics data from the GPUs to the IPU (through system memory) while avoiding tearing.

Further aspects of these blocks are outlined in the following sub-sections.

8.2 Image Processing Unit (IPU)

Table 8-1. IPU Block Parametric Table

Name	IPU
Function	Connectivity to cameras and displays; related processing; synchronization and control
External I/O Pins Notes: This is the pinout of the IPU block At chip level, some of the pins are muxed and some are omitted. Additional General Purpose I/O Module (GPIO) pins are required to construct the connection. This is not included in this list.	Parallel Display port: 24-bit data, ~18 clocks and controls. Regular CMOS IO type, 180 MHz max. May be also connected to the TVE or LVDS internal connectivity bridges. Slow Serial Display port: 2 bit data, 3 clocks and controls. Regular CMOS IO type, 120 MHz max. Parallel Sensor port: 20 bit data inputs, 5-6 clocks and controls. Regular CMOS IO type, 180 MHz max. Camera strobe port: 6 camera control outputs. Regular CMOS IO type, 180 MHz max.
SoC Buses	AXI master - for accessing the memory AHB slave - for programming, control and direct access of the ARM Core to the display
Interrupts	Two types of interrupts: one triggered by selected completion events and a second triggered by selected internal error events
DMA Requests	An integral DMA controller, with an AXI master port One DMA request to the SDMA
Number of instantiations	1
Clock sources and range	HSP_CLK - Internal high-speed processing clock: up to 200 MHz DI_CLK0, DI_CLK1 - Display interface clocks: up to 180 MHz In addition, a sensor may require a "master" clock source: 2-180 MHz

The goal of the IPU is to provide comprehensive support for the flow of data from an image sensor and/or a display device. This support covers all aspects of these activities:

- Connectivity to relevant devices - cameras, displays, graphics coprocessors, TV encoders and decoders.
- Related image processing and manipulation: image enhancements and conversions, etc.
- Synchronization and control capabilities (for example, to avoid tearing artifacts)

This integrative approach leads to several significant advantages:

- Automation:

  The involvement of the ARM Core in image management is minimized. In particular, display refresh/update and a camera preview (displaying the input from an image sensor) can be performed completely autonomously. The resulting benefits are reducing the overhead due to SW-HW synchronization, freeing the ARM Core to perform other tasks and reducing the power consumption (when the ARM Core is idle and can be powered down).

- Optimal data path:

  Access to system memory is minimized. In particular, significant processing can be performed on-the-fly while receiving data from an image sensor and/or sending data to a display. System memory is used essentially only when a change in pixel order or frame rate is needed. The resulting benefits are reduced load on the system bus and further reduction of power consumption.

- Resource sharing:

  Maximal HW reuse for different applications, resulting with the support of a wide range of requirements with minimal HW

The HW reuse mentioned above is enabled by a sophisticated configurability of each HW block. This configurability also allows the support of a wide range of external devices, data formats and operation modes. The resulting flexibility is important because the support requirements are evolving significantly, so expected future changes need to be anticipated and accounted for.

A simplified block diagram of the IPU is given in Figure 8-3. The role of each block is described in Table 8-2.

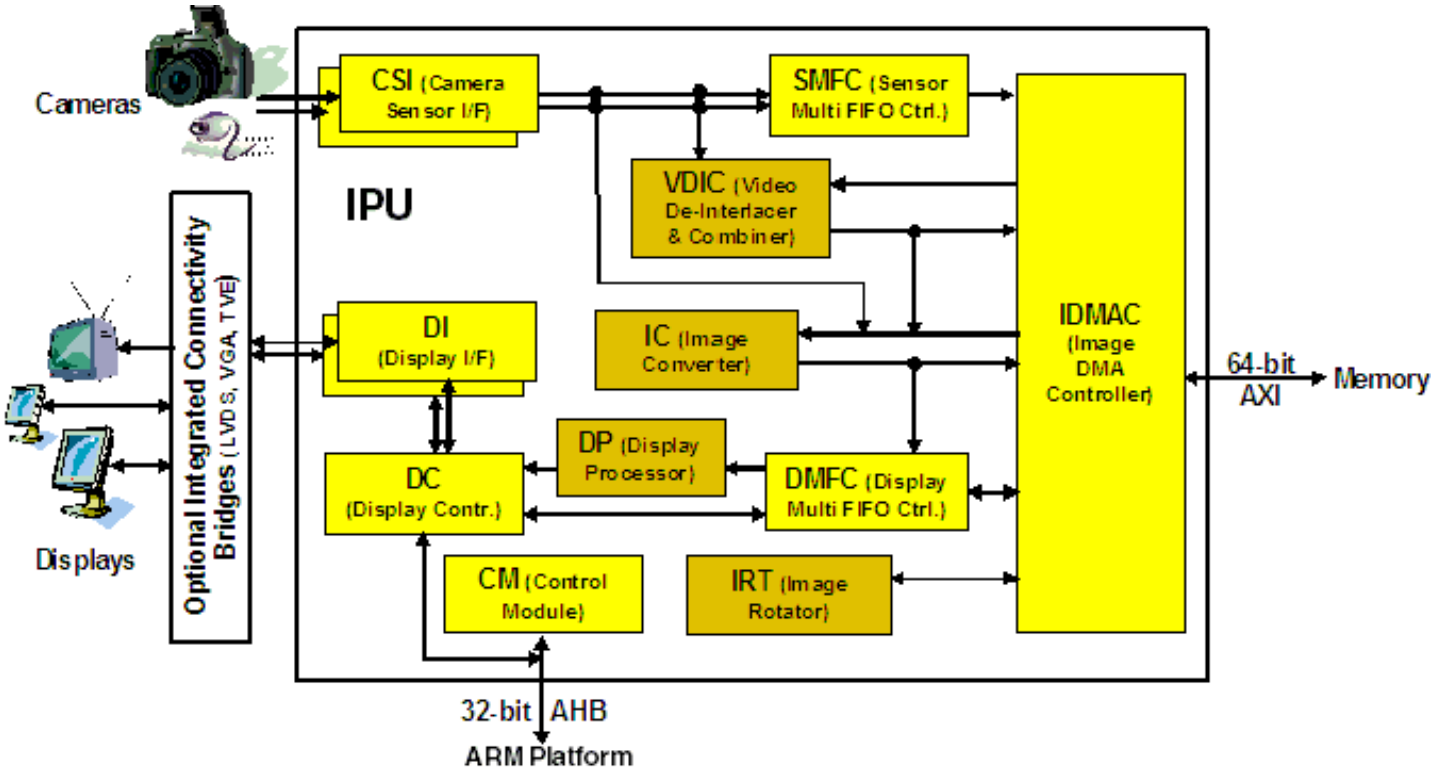


Figure 8-3. IPU - Block Diagram

Table 8-2. IPU - Block Description

Block	Description
CSI - Camera Sensor Interface	Controls a camera port; provides interface to an image sensor or a related device. IPU includes 2 such blocks
DI - Display Interface	Provides interface to displays, display controllers and related devices. IPU includes 2 such blocks
DC - Display Controller	Controls the display ports.
VDIC - Video De-Interlacer and Combiner	Performs deinterlacing - converting interlaced video to progressive - or combining
IC - Image Converter	Performs resizing, color conversion/correction, combining with graphics, and horizontal inversion
DP - Display Processor	Performs the processing required for data sent to display
IRT - Image Rotator	Performs rotation (90 or 180 degrees) and inversion (vertical/horizontal)
IDMAC - Image DMA Controller	Controls the memory port; transfers data to/from system memory
SMFC - Sensor Multi FIFO Controller	Controls FIFO's for output from the CSI's to system memory
DMFC - Display Multi FIFO Controller	Controls FIFO's for IDMAC channels related to the display system
CM - Control Module	Provides control and synchronization.

In the following, the main features of the IPU are listed.

8.2.1 External Ports-IPU

The IPU has the following ports:

- Two camera port - controlled by a CSI sub-block - providing a connection to image sensors and related devices.
- Two display ports - each controlled by a DI sub-block - providing a connection to displays and related devices.
- Memory port - AXI (AHB V3.0) master, controlled by the IDMAC - providing connection to the system memory.
- AHB-Lite slave port, providing connection to the ARM (and to any other master connected to the ARM's cross-bar switch)

8.2.1.1 Camera Port

The role of these ports is to receive input from image sensors (or TV decoders) and to provide support for time-sensitive control signals to the camera. Non-time-sensitive controls - such as configuration, reset - are performed by the ARM Core through an I2C interface or General Purpose I/O Module (GPIO) signal.

Each of the camera ports includes the following features:

- Direct connectivity to most relevant image sensors and to TV decoders.
- Parallel interface - up to 20-bit data bus
- Frame size: up to 8192 x 4096 pixels (including blanking intervals)
- Data formats supported include Raw (Bayer), RGB, YUV 4:4:4, YUV 4:2:2 and grayscale, up to 16 bits per value (component).
- Synchronization - video mode
  - The sensor is the master of the pixel clock (PIXCLK) and synchronization signals
  - Synchronization signals are received using either of the following methods:
    - Dedicated control signals -VSYNC, HSYNC - with programmable pulse width and polarity
    - Controls embedded in the data stream, following loosely the BT.656 protocol, with flexibility in code values and location.
- Synchronization - still image capture

- The image capture is triggered by the ARM Core or by an external signal (for example, a mechanical shutter)
- Synchronized strobes are generated for up to 6 outputs - the sensor and camera peripherals (flash, mechanical shutter...)

- Additional features
  - Frame rate reduction, by periodic skipping of frames
  - Window-of-interest selection
  - Pre-flash - for red-eye reduction and for measurements (such as focus) in low-light conditions

Several sensors can be connected to each of the CSIs. Simultaneous functionality (sending data) is supported as follows:

- Two sensors can send data independently, each through a different port.
- Only one of the streams can be transferred to the VDIC or IC for on-the-fly processing, while the other one is sent directly to system memory.

The input rate supported by the camera port is as follows:

- Peak: up to 180 MHz (values/sec)
- Average (assuming 35% blanking overhead), for YUV 4:2:2
  - Pixel in one cycle (BT.1120): up to 135 MP/sec, e.g. 9M pixels @ 15 fps
  - Pixel on two cycles (BT.656): up to 67 MP/sec, e.g. 4.5M pixels @ 15 fps.
- On-the-fly processing may be restricted to a lower input rate.

8.2.1.2 Display Ports

The role of these ports is to communicate with display devices, either directly or through a controller (such as a graphics accelerator) or a bridge (such as a TV encoder or an LVDS interface bridge).

8.2.1.2.1 Access Modes

Two access modes are supported.

8.2.1.2.1.1 Synchronous Access

In this mode, the IPU transfers a two-dimensional block of pixels to the display device, in synchronization with the screen refresh cycle.

This mode has a dual role:

- For a RAM-less display or a TV screen, this mode is used to perform the screen refresh process from a display buffer in system memory.
- For a "smart" display, this mode is used to transfer a rectangular block of pixels to the display's screen and, in some cases, also to the display buffer

In both cases, the IPU sends all the synchronization signals controlling the screen refresh to the display, and the block transfer is synchronized with these signals. This synchronization means that tearing effects are avoided when using this mode.

8.2.1.2.1.2 Asynchronous Access

This is the main mode used for communicating with an external display controller (possibly in a smart display or a graphics accelerator). In this mode, the IPU performs random access - read/write - to the memory and registers of the controller.

The following access types are provided:

- Data transfer to the external device, after on-the-fly processing in the IPU.
- Data transfer (DMA) - read/write - between the host's system memory and the external device, through the IPU's memory port (controlled by the IDMAC); the transfer of a rectangular block of pixels (possibly full screen).
- Host access - read/write - to an external device, through the slave AHB port
  - Access types
    - Direct access - emulating a directly-addressed access (see below)

      This includes burst access (incremental; up to 8 words/burst)

    - Low-level access - leaving to the host the explicit generation of the access protocol
  - The possible accessing blocks include the ARM Core and the system DMA controller (as well as any other AHB master connected to the ARM Core's cross-bar switch).

The asynchronous access requires the specification of an address. The display interface uses "indirect addressing", namely, there is no address bus, and the address, as well as control and configuration commands, are embedded in the data stream. The access procedure - including writing addresses and commands - can be managed autonomously by the interface, using an access template generated by the ARM Core.

8.2.1.2.2 The Interface

The display interface is very flexible and supports a wide variety of devices from major manufacturers. The following interface types are provided (in each of the two display ports)

- A parallel video interface (for synchronous access) - up to 24-bit data bus.
  - Supports BT.656 (8-bit) and BT.1120 (16-bit) protocols
  - Supports HDTV standards SMPTE274 (1080i/p) and SMPTE296 (720p)
- A parallel bidirectional bus interface (for asynchronous access) - up to 32-bit data bus.
- A serial interface - 3-wire, 4-wire and 5-wire (two flavors) (for asynchronous access)

The supported formats for pixel data are: RGB and YUV 4:2:2 (for TV encoder).

For the interface clock, there are the following options (independently for each port)

- Derived from the IPU internal clock (master mode)
- Provided by an external source (slave mode)

Transfer rates supported:

- Pixel clock frequency up to 200MHz when a single port is active.
- With both ports active
  - Each pixel clock may be up to170 MHz
  - The sum of pixel clock rates may not exceed 180 MHz

8.2.1.2.2.1 Connecting To Display Devices

IPU allows the connectivity to multiple display devices. In particular, it supports the following setup:

- Primary LCD display: can be smart, dumb (RAM-less) or dual-port; may use the parallel interface, or (through an integrated bridge) LVDS interface.
- Second LCD display: can be smart or dumb (RAM-less); may use parallel or legacy serial interface, or (through an integrated bridge) LVDS interface.
- TV Output: either digital parallel output, or (through the integrated TV encoder bridge) analog output.

Simultaneous functionality of the above devices is possible in each of the following ways:

- Two devices can be accessed (synchronously or asynchronously) independently, each through a different port: each using any of the available interfaces.
- Two devices can time-share asynchronous accesses through the legacy serial and parallel interfaces, using the CS signals.
- An asynchronous access can be performed during vertical blanking intervals of a synchronous access (screen refresh; to the same or other device).

8.2.2 Processing-IPU

The IPU processes rectangular blocks of pixels. The processing is performed in four sub-blocks - VDI, IC, DP and IRT (see Figure 8-3 and Table 8-2). Several time-shared data flows are supported, as described in Table 8-3.

Table 8-3. Time-Shared Data Flows through the IPU

Name	Number	Type	Flow	Target	Restrictions
Display Refresh/ Update	5 flows (at most two of them of type DS1)	DS1	Fmem -> DP -> Display	Synchronous Access (such as display refresh; controlled by the DI)	-
		DS2	Fmem -> DP -> Display	Asynchronous Access (such as display update)	-
		DS3	Fmem <-> Display	Generic Data Transfer	-
	1 flow	DS4	ARM Core<-> Display	Direct Access	-
Video Playback	1 flow	PL1	Bmem -> VDI -> IC -> Bmem -> IRT -> Fmem + DSx	Main option	-
		PL2	Fmem -> IRT -> Bmem -> IC -> DP	Low power (branching to DSx, as a video plane)	Progressive source No other video flows
		PL3	Fmem -> VDI -> IC -> DP	Low power (branching to DSx, as a video plane)	Interlaced source No other video flows
Camera Pre-view	2 flows (VF2 may be used also as a playback flow)	VF1	Sensor -> IC -> Bmem -> IRT -> Fmem + DSx	main option	Single progressive input
		VF2	Sensor -> Fmem -> VDI -> IC -> Bmem -> IRT -> Fmem + DSx	two inputs and/or interlaced input	When the VDI is used, one of the three input fields can go directly from the sensor to the VDI
		VF3	Sensor -> IC -> DP	Low power Smart Display Tearing-less (branching to DS2, as a video plane)	Single progressive input Refresh rate = 2x sensor frame rate No other video flows
		VF4	Sensor -> IC -> Fmem + DS1	Low power RAM-less Display Single Display Buffer (in internal memory) Tearing-less	Single progressive input Refresh rate = 2x sensor frame rate No other video flows
Video Record	2 flows	RCx	IC -> Bmem -> IRT -> Fmem	(branching from VFx)	-

Table continues on the next page...

Table 8-3. Time-Shared Data Flows through the IPU (continued)

Name	Number	Type	Flow	Target	Restrictions
Graphic Overlays	2 flows	GF1	Fmem -> IC	(combining with the main flow	-
	2 flows	GF2	Fmem -> DP		-

Comments

- System memory usage - legend
  - Fmem: frame double-buffer (page-flip) in system memory (typically external)
  - Bmem: two possibilities
    - A frame double buffer, as above
    - A band (4-256 rows) double-buffer (page-flip) in system memory (could be internal)
  - Direct arrow between two processing stages represents an internal pipelining
- Time-sharing
  - IC can time-share tightly three flows: one VFx, one RCx and one PLx (with independent processing parameters)
  - DP can time-share one DS1 flow and a one DS2 flow (each with different destinations and independent processing parameters)
  - Direct access to display (DS4) time-shares tightly the display port with other active DSx flows.
  - Other time-sharing (between PLx; between DS2&DS3; in IRT) is frame-by-frame
- Any of the processing stages in the above flows can be skipped.

8.2.2.1 Display Processor (DP)

The Display Processor performs all the processing required for data sent to a display:

- Combining 2 video/graphics planes
- Overlaying a simple HW cursor
- 32 x 32 pixels, uniform color; may be combined logically with the background.
- Color conversion/correction - linear (multiplicative and additive)
- Programmable; including:
  - YUV <-> RGB, YUV<->YUV conversions
  - where YUV stands for any one of the color formats defined in the MPEG-4 standard
  - Adjustments: brightness, contrast, color saturation...
  - Special effects: gray-scale, color inversion, sephia, blue-tone...

    - Hue-preserving gamut mapping - for minimal color distortion
    - Applied to the output of combining or to one of the inputs
- Gamma correction and contrast stretching - programmable piecewise-linear map

The DP processes a single data flow at any given time but it supports up to three data flows by time sharing. One of them may be synchronous.

The data throughput is up to 200M pixels/sec (peak; including blanking intervals)

8.2.2.2 Video Deinterlacer (VDIC)

The Video Deinterlacer and Combiner has two operation modes

- Deinterlacing: converts an interlaced video stream to progressive order.
- Combining: combines two video/graphics planes and a background color

The input and output to/from the VDIC is as follows:

- Input for deinterlacing: three consecutive fields
    - Source
        - The most recent field may come from the CSI or from system memory
        - The other two fields are read from memory
    - Field size: up to 968x600 pixels (may be a vertical stripe of a wider field; for example, 1920 pixels)
    - Pixel format: YUV 4:2:2/4:2:0, 8 bits/value
    - Typical video sources - SDTV: 480i30 (720x480 @ 30 fps) or 576i25 (720x576 @ 25 fps) and HDTV: 1080i30 (1920x1080 @ 30 fps)
- Input for combining: two progressive video/graphics planes
    - Source: system memory
    - Plane size: up to 1920x1200 pixels.
    - Pixel format: RGB/YUV 4:2:2, 8 bits/value
- Output: progressive frame
    - Destination: to system memory or to the Image Converter.
    - Frame size: up to 1920x1200 pixels.
    - Rate: up to 180 MP/sec.
    - Format: same as input format.

The deinterlacing is performed using a high-quality 3-field filter which is motion adaptive:

- For slow motion - retains the full resolution (of both top and bottom fields)
- For fast motion - prevents motion artifacts

The VDIC supports a single video stream at any given time.

8.2.2.3 Image Converter (IC)

The Image Converter performs various operations on a video stream. The operations performed are:

- Resizing
  - Fully flexible resizing ratio
  - Maximal downsizing ratio: 16:1
  - Subject to this limitation, any N->M resizing can be performed
  - Independent horizontal and vertical resizing ratios.
- Color conversion/correction - linear (multiplicative and additive)

  Programmable; including:

  - YUV <-> RGB, YUV<->YUV conversions where YUV stands for any one of the color formats defined in the MPEG-4 standard
  - Adjustments: brightness, contrast, color saturation...
  - Special effects: gray-scale, color inversion, sephia, blue-tone
- Combining with a graphics plane (application-specific overlay)
- Horizontal inversion

The IC supports three time-shared data flows: record, camera preview and playback (the first two share a common input).

The frame resolution supported is up to 4096x4096 for input and up to 1024x1024 for output. Wider frames can be processed by the IC by splitting them to vertical stripes.

The data throughput

- Up to 180M pixels/sec for input and up to 90M pixels/sec for output - for a single active task case.
- Up to 150M pixels/sec for input and up to 75M pixels/sec for output - for several active tasks case.

8.2.2.4 Image Rotator (IRT)

The Image Rotator performs any combination of the following:

- 90-degree rotation
- Horizontal inversion
- Vertical inversion

The data throughput

- When a single task is active: up to 90M pixels/sec.
- When more tasks are active: up to 75M pixels/sec.

8.2.3 Automatic Procedures

The IPU is equipped with powerful control and synchronization capabilities to perform its tasks with minimal involvement of ARM and minimal use of memory. In particular, it includes:

- An integrated DMA controller with an AXI master port, allowing autonomous access to the system memory.
- An integrated display controller, performing screen refresh of a RAM-less display.
- A page-flip double buffering mechanism, synchronizing read and write access to system memory, to prevent tearing effects.
- A double/triple buffer synchronization mechanism with a video/graphics source.
- Internal synchronization, such as that between input from sensor and output to display.

In most cases, the ARM Core is involved only when it also performs part of the processing (such as video coding). In particular, the following procedures are performed by the IPU completely autonomously:

- Screen refresh for RAM-less displays
- Camera preview (displaying a view finder - a video-stream from the image sensor to the display).

Typically, there are extended periods of time in which there is no other activity in the system. The ARM Core can be put into a low-power mode (idle), reducing the power consumption and significantly extending the battery life.

The IPU supports several techniques to reduce further the power consumption of the display system:

- Optimized update of the display buffer, using a "snooping" signal from the memory controller, indicating a modification of the source buffer.
- Dynamic backlight control, with low-light compensation by image enhancement

Further features and capabilities of the automatic procedures include:

- Automatic display of a changing image (animation) or moving image (scrolling).
- The timing of the display update can be adjusted to avoid tearing.
- The video stream from an image sensor can be sent directly to the display buffer used for screen refresh, while avoiding tearing.

The IPU supports direct synchronization with a video/graphics source. Up to three frame buffers are supported for this synchronization. It is performed using the following signals

- The source instructs the IPU which frame should be transferred to the display
- The IPU notifies the source which frame is currently transferred to the display
- The IPU provides to the source an end-of-frame trigger (after which it switches to the frame indicated by the source).

This mechanism is provided for up to two sources, each can be connected to any of the DP/DC channels supporting a synchronous flow.

The clock sources received by the IPU are listed in Table 8-4.

Table 8-4. IPU Clock Sources

Name	Symbol	Source	Rate	Comments
High-Speed Processing Clock	HSP_CLK	Clock Control Module	Up to 200 MHz	-
Display Interface Clocks	DI_CLK0 DI_CLK1	Clock Control Module or an external DPLL	up to 150 MHz	Optional; needed for synchronization with interface bridges

8.3 LVDS Display Bridge (LDB)

LVDS Display Bridge (LDB) will be used to connect the Image Processing Unit (IPU) to External LVDS Display Interface.

Relevant Standards:

- ANSI EIA-644-A. Electrical Characteristics of Low Voltage Differential Signaling (LVDS) Interface Circuits.
- SPWG Notebook Panel Specification (V3.8 from 03/2007)

  http://www.spwg.org/specifications.htm .
- PSWG standards (Panel Standardization Working Group) - set of standards for panels using LVDS.

  All available from http://www.vesa.org/.
- DISM Standard JEIDA-59-1999

Table 8-5. IP Parametric Table

Name	IPU
Function	Connectivity to displays with LVDS interface

Table continues on the next page...

Table 8-5. IP Parametric Table (continued)

Name	IPU
External I/O Pins Notes: Those are LVDS IO pads	LVDS Display port: 2 channels. Each channel consists of: • 1 clock pair • 4 data pairs Each pair contains - LVDS special differential pad (PadP, PadM). total of 20 IO pads.
SoC Buses	None. Only configuration signals.
Interrupts	None
DMA Requests	None
Number of instantiations	1
Clock sources and range	DI0_CLK, DI1_CLK- Display interface clock: 20-170 MHz DI0_SER_CLK, DI1_SER_CLK - Serializer clock: 140-595 MHz

The purpose of the LDB is to support flow of synchronous RGB data from the IPU to external display devices through the LVDS interface. This support covers all aspects of these activities:

- Connectivity to relevant devices - Displays with LVDS receivers.
- Arranging the data as required by the external display receiver and by LVDS display standards.
- Synchronization and control capabilities.

A block diagram of the LDB is given in the following figure.

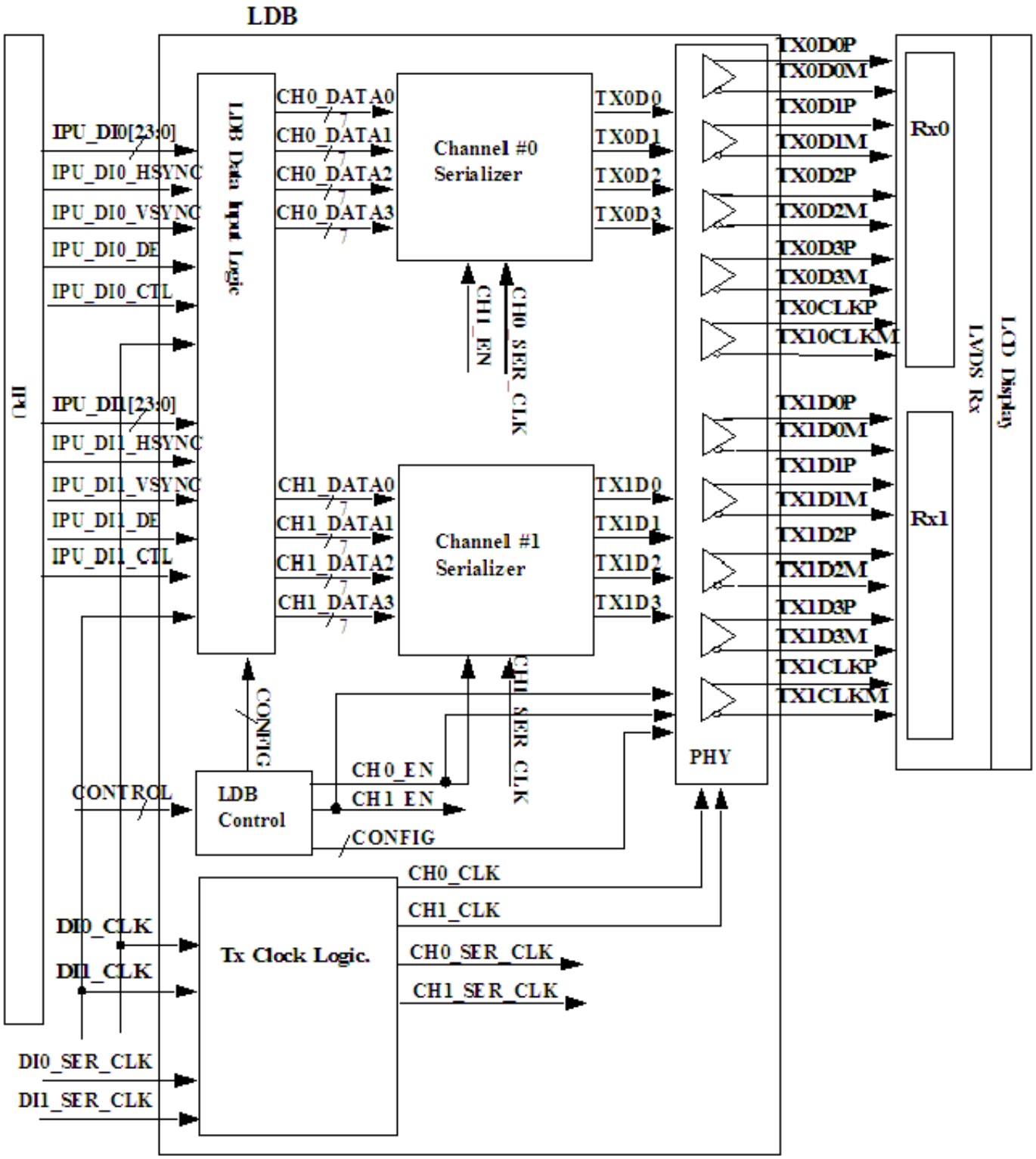


Figure 8-4. LDB - Block Diagram

Table 8-6. LDB - Block Description

Block	Description
LDB Data Input Logic	Rearranges input DI data from IPU according to configuration, and sends output to serializers. Is also responsible on Input bus split to Odd/Even busses.
Channel Serializers	LDB has 2 channel serializers. Each serializer does parallel to serial conversion 7:1 to 3 or 4 data lines
LDB Control	Gets control signals from SoC and determines parameters of LDB
Tx Clock Logic	Mux input clocks to Serializers and PHY.
PHY	Converts digital signals to analog LVDS signals.

8.3.1 External Ports-LDB

The LDB has the following ports:

- Two input parallel display ports.
- Two Output LVDS channels - Each channel consisting of 4 data pair, and 1 clock pair (pair =LVDS pad contains PadP, PadM).
- Control signals - to configure LDB parameters and operations.
- Clocks from SoC DPLLs.

8.3.1.1 Input Parallel Display Ports

One or Two (DI0, DI1) parallel RGB input ports are supported (configurable). Only synchronous access mode is supported.

Each RGB data interface contains the following:

- RGB Data of 18 or 24 bits
- Pixel clock
- Control signals: HSYNC, VSYNC, DE, and 1 additional optional general purpose control.

Total of up to 28 bits per data interface are transferred per pixel clock cycle.

Rates supported:

- For dual-channel output: Up to 170 MHz pixel clock (for example, UXGA - 1600x1200 @ 60 Hz + 35% blanking)
- For single-channel output: Up to 85 MHz per interface (for example, WXGA - 1366x768 @ 60 Hz + 35% blanking).

8.3.1.2 Output LVDS Ports

There are 2 LVDS channels. These inputs are used to communicate RGB data and controls to external LCD displays with LVDS interface, or through LVDS receivers.

The LVDS ports may be used as follows:

- One single-channel output
- One dual channel output: single input, split to two output channels
- Two identical outputs: single input sent to both output channels
- Two independent outputs: two inputs sent, each, to a different output channel

The output LVDS port must comply to #d346e12a1310.

8.3.1.3 Control Signals

Control signals are used to configure LDB operation. Needed configuration parameters are:

Table 8-7. LDB - Configuration Description

Parameter	Description
CHANNEL_MAPPING	Mapping of Parallel input interfaces (DI0, DI1) to output LVDS channels (Channel 1, Channel 2).
D0_DATA_WIDTH	18/24 bit (use 3/4 data pairs)
D1_DATA_WIDTH	18/24 bit (use 3/4 data pairs)
DI0_BIT_MAPPING	Bit mapping/ordering for DI0. Used only on modes where DI0 is enable. See muxing modes on "Processing"
DI1_BIT_MAPPING	Bit mapping/ordering for DI1. Used only on mods where DI1 is enabled. See muxing modes on "Processing"

8.3.1.4 Clock Sources

The clock sources received by the LDB are listed in Table 8-8.

Table 8-8. LDB Clock Sources

Name	Symbol	Source	Rate	Comments
IPU DI0 interface pixel clock	IPU_DI0_CLK	Clock Control Module	Up to 170 MHz	See note below [1, 1]
IPU DI1 interface pixel clock	IPU_DI1_CLK	Clock Control Module	Up to 170 MHz	This input also goes to IPU DI1 as input. See note below [1]

Table continues on the next page...

Table 8-8. LDB Clock Sources (continued)

Name	Symbol	Source	Rate	Comments
DI0 interface serializer clock	DI0_SER_CLK	Clock Control Module	Up to 595 MHz	This is x7 or x3.5 the rate of the DI0 interface pixel clock. See note below.
DI1 interface serializer clock	DI1_SER_CLK	Clock Control Module	Up to 595 MHz	This is x7 or x3.5 the rate of the DI1 interface pixel clock. See note below

1. LVDS DI_CLK should always be identical to the relevant IPU DI_CLK. In case of single-channel of separate-channels use-case - the serializer clock will be x7 faster than the DI_CLK. In case of dual-channel use-case - the serializer clock will be x3.5 faster than the DI_CLK

8.3.2 Processing-LDB

LDB data processing stages are as follows:

- Get input data from 1 or 2 (configurable) parallel input interfaces. 18/24 RGB data + up to 4 controls and map them to LVDS channels.
  - If needed (dual-channel) split the input bus to two half-rate busses.
- Re-arrange the input data according to channel configuration, and muxing scheme.
- Serialize the 22/28 bit input bus (per channel) on 3-4 output serial data lines (7:1)
- PHY converts the serial digital data to analog LVDS signals

8.3.2.1 LDB Data Input Logic

The Data Input Logic does the following processing on the 1-2 input busses:

8.3.2.1.1 Mapping of Input Data Busses

8.3.2.1.1.1 Channel Mapping

Mapping of Parallel input interfaces (DI0, DI1) to output LVDS channels (Channel 0, Channel 1). See Table 8-5.

Channel Mapping

Use Case	LVDS Channel 0	LVDS Channel 1
Single channel from DI0	DI0	Disabled
Single channel from DI1 to channel 1	Disabled	DI1
Single channel from DI1 to channel 0	DI1	Disabled
Dual Channel from DI0	DI0 (first pixel)	DI0 (second pixel)
Dual Channel from DI1	DI1 (first pixel)	DI1 (second pixel)

Table continues on the next page...

Use Case	LVDS Channel 0	LVDS Channel 1
Two single channels with identical content - from DI0	DI0	DI0
Two single channels with identical content - from DI1	DI1	DI1
Two independent single channels	DI0	DI1

8.3.2.1.1.2 Input Bus Split

For the dual-channel case, it is required to split the input bus into two half-rate busses (Odd/Even pixels).

- First pixel is mapped to Channel 0
- Second pixel is mapped to Channel 1

8.3.2.1.2 Bit Mapping

Several bit mapping options should be supported. The 22/28 bit bus is mapped into on 3/4 groups of 7 bits each.

SPWG/PSWG/VESA 18/24 bpp Data Mapping

Serializer input	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
CHx_DATA0	G0	R5	R4	R3	R2	R1	R0
CHx_DATA1	B1	B0	G5	G4	G3	G2	G1
CHx_DATA2	DE	VS	HS	B5	B4	B3	B2
CHx_DATA3 (for 24 bpp only)	CONT	B7	B6	G7	G6	R7	R6

JEIDA 24bpp Data Mapping

Serializer input	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
CHx_DATA0	G2	R7	R6	R5	R4	R3	R2
CHx_DATA1	B3	B2	G7	G6	G5	G4	G3
CHx_DATA2	DE	VS	HS	B7	B6	B5	B4
CHx_DATA3	CONT	B1	B0	G1	G0	R1	R0

NOTE

Several options of control usage can be available. Some display devices use only DE, some others use all three controls, and some use only HS and VS. "CONT" is an optional general purpose control which is usually unused by the display.

8.3.2.2 LDB Control

Receives control signals from SoC and configures LDB unit accordingly.

See also Control Signals.

8.3.2.3 LDB Tx Clock

The input pixel clock is used to mark pixel boundaries.

The serializer clock is needed to enable the 7:1 muxing on the channel serializers.

Following operations should be supported:

- Map input clocks to channel clocks
- Map serializer clocks to channel serializers
- Generate hold-rate channel clock and align it to the required timing by LVDS standards

See also Clock Sources.

LDB uses 2 Channel serializers. One per LVDS channel.

- Input: 3-4 busses of 7 bit each.
- Clock: serializer clock
- Processing: 7:1 muxing.
- Output 3-4 serial lines

8.3.2.4 PHY

PHY translates the digital serializer outputs to LVDS signals.

PHY consists of 2 channels (1 or 2 may be used according to use case).

Each channel consists of:

- 4 data pairs (LVDS pads) - 3 or 4 may be used according to use case
- 1 clock pair

The PHY output must comply with ANSI EIA-644-A.

Clock/data timing must comply with display standards such as mentioned in the LVDS Display Bridge (LDB).

Spread spectrum support - only as enabled by SoC DPLL's features.

8.4 Video Processing Unit (VPU)

The Video Processing Unit (VPU) is a multi-standard video codec (encoder/decoder) capable of handling multiple streams simultaneously through time multiplexing. The VPU is a very flexible block consisting of a programmable core surrounded by hardware accelerators. The VPU presents to the system a register mapped interface that is controlled by the embedded processor. Because this interface can be updated by the firmware it will not appear in this document; instead the designer should consult the documentation released with the firmware used. End users should only interface with the VPU using the API that is made available with each firmware release. This API isolates the user from possible changes in the register level interface.

Table 8-9 is a summary of the VPU specs. The VPU has its own DMA driven AXI masters that allow it to retrieve the required data directly from system memory (DDR and iRAM). The load in the host ARM platform is negligible because it only needs to interact with the VPU at the frame level.

Table 8-9. VPU Spec Summary

Name	VPU
Function	Decode video streams including optional video processing such as rotation, deringing and mirroring.
Supported encoders	MPEG-4 SP H.263 V2 + Annex J, K (RS=0 and ASO=0), and T H.264 BP MJPEG Baseline
Supported decoders	MPEG-2 MP VC-1 SP, MP, AP MPEG-4 SP, ASP H.263 V2 + Annex J, K (RS=0 and ASO=0), and T Sorenson H263 H.264 BP, MP, HP DivX v3,4,5 Real Video 8, 9, 10 MJPEG Baseline
External I/O Pins (List, Type, Schmidt Trigger, Speed)	No external I/O pins are needed
SoC Buses (List, Type, Bandwidth)	64 bit AXI master for accessing the system memory and search RAM IP Bus slave for host control
Interrupts	one interrupt

Table continues on the next page...

Table 8-9. VPU Spec Summary (continued)

Name	VPU
DMA Requests	Integrated DMA controller on the AXI master port
Number of instantiations	1
Clock sources and range	Core clock: up to 200 MHz AXI bus clock: up to 200 MHz IP Bus clock: up to 66.5 MHz

The i.MX53 processor has a high-performance Video Processing Unit (VPU) which covers many standard and high definition video decoders and encoders as a multi-standard video codec engine, as well as several important video processes such as rotation, mirroring and de-ringing. VPU is a licensed block provided by Chips&Media. The overview of VPU is summarized above.

8.4.1 Basic Structure

The following figure shows the basic structure of the VPU. The VPU is self-contained except for memory accesses. It is connected to the memory subsystem through two AXI master ports. The ARM configures the VPU operation through the IP Bus that is converted to APB in a gasket.

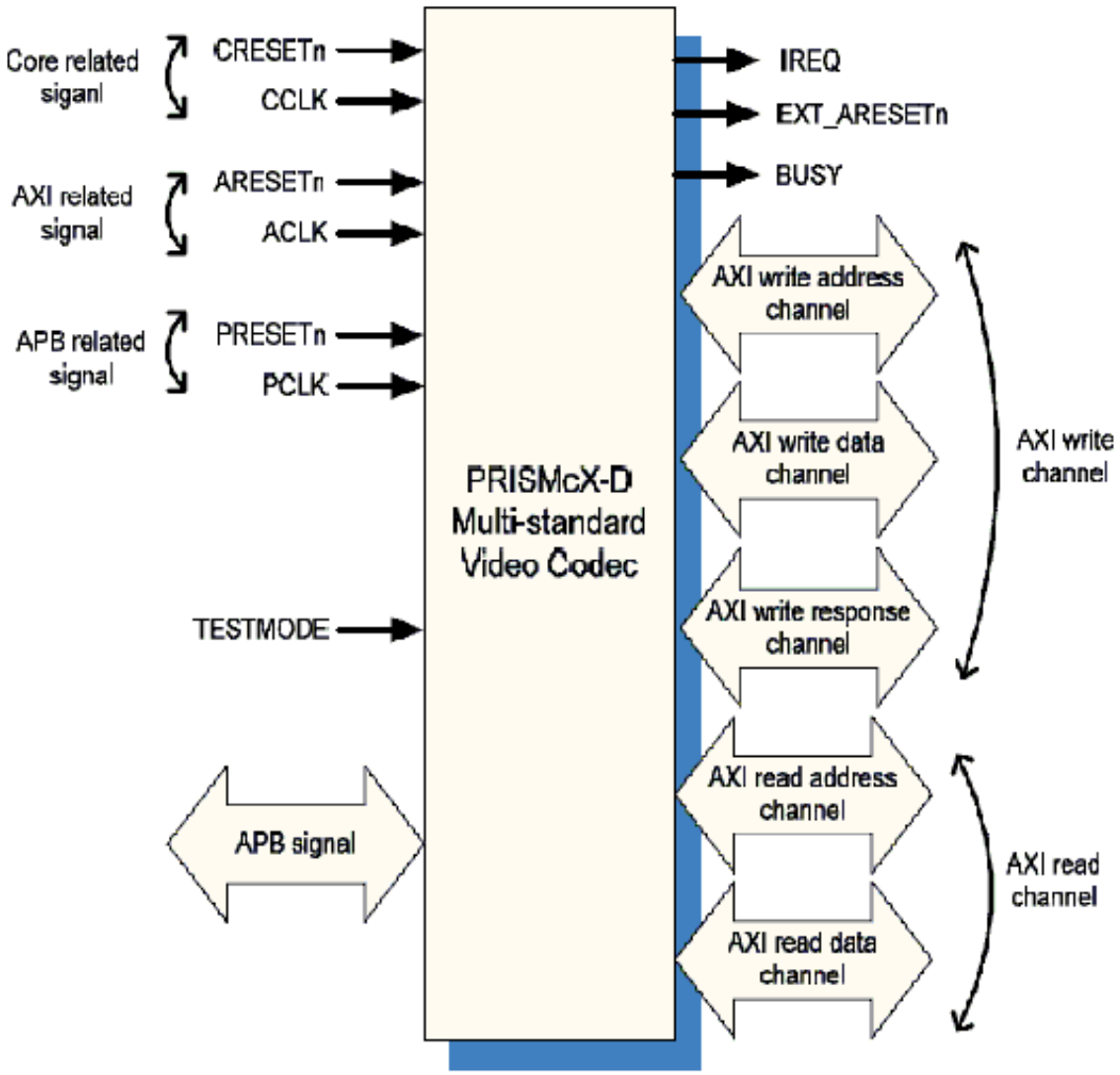


Figure 8-5. The Block Diagram of the VPU

8.4.2 Feature Summary

Table 8-10 below lists the VPU's encoding / decoding capabilities.

Table 8-10. VPU Decoding / Encoding Capabilities

Dec/Enc	Standard	Profile	Resolution	Bitrate	Comments
HW Decoder	MPEG-2	Main-High	1080 i/p	40Mbps	
	MPEG4/XviD	SP/ASP	1080 i/p	40Mbps	-
	H.263	P0/P3	16CIF	20Mbps	i.MX53 Platform may support up to D1@30fps
	Sorenson H.263	N/A	16CIF	20Mbps	i.MX53 Platform may support up to D1@30fps
	H.264	BP/MP/HP	1080 i/p	40Mbps	-
	VC1	SP/MP/AP	1080 i/p	40Mbps	-
	RealVideo	8/9/10	1080p	40Mbps	-
	DivX	3/4/5/6	1080 i/p	40Mbps	-
	MJPEG	Baseline	8192x8192	40Mpixel/sec (Up to YUV444)	40MP/sec is at YUV444 format and it will be higher for 422 and 420 formats
HW Encoder	MPEG2[1]	Main-Main	D1	15Mbps	-
	MPEG4	Simple	720p	20Mbps	VPU can generate higher bitrate than the maximum specified by the corresponding standard.
	H.263	P0/P3	4CIF	20Mbps	
	H.264	Baseline	720p	20Mbps	
	MJPEG	Baseline	8192x8192	80Mpixel/sec (Up to YUV422)	80MP/sec is for 422 format, and it will be higher for 420 format

1. Video partially performed in HW (70%)

8.4.3 Other Features of VPU

- Rotation and Mirroring: on-the-fly (90 x n) degree simultaneous rotation and mirroring (n = 0,1,2,3)
- Post-processing: de-blocking filtering for MPEG-4 and de-ringing filtering for MPEG-4 and H.264 decoder
- De-ringing support: de-ringing filtering is supported in VPU

8.4.4 Architectural Overview

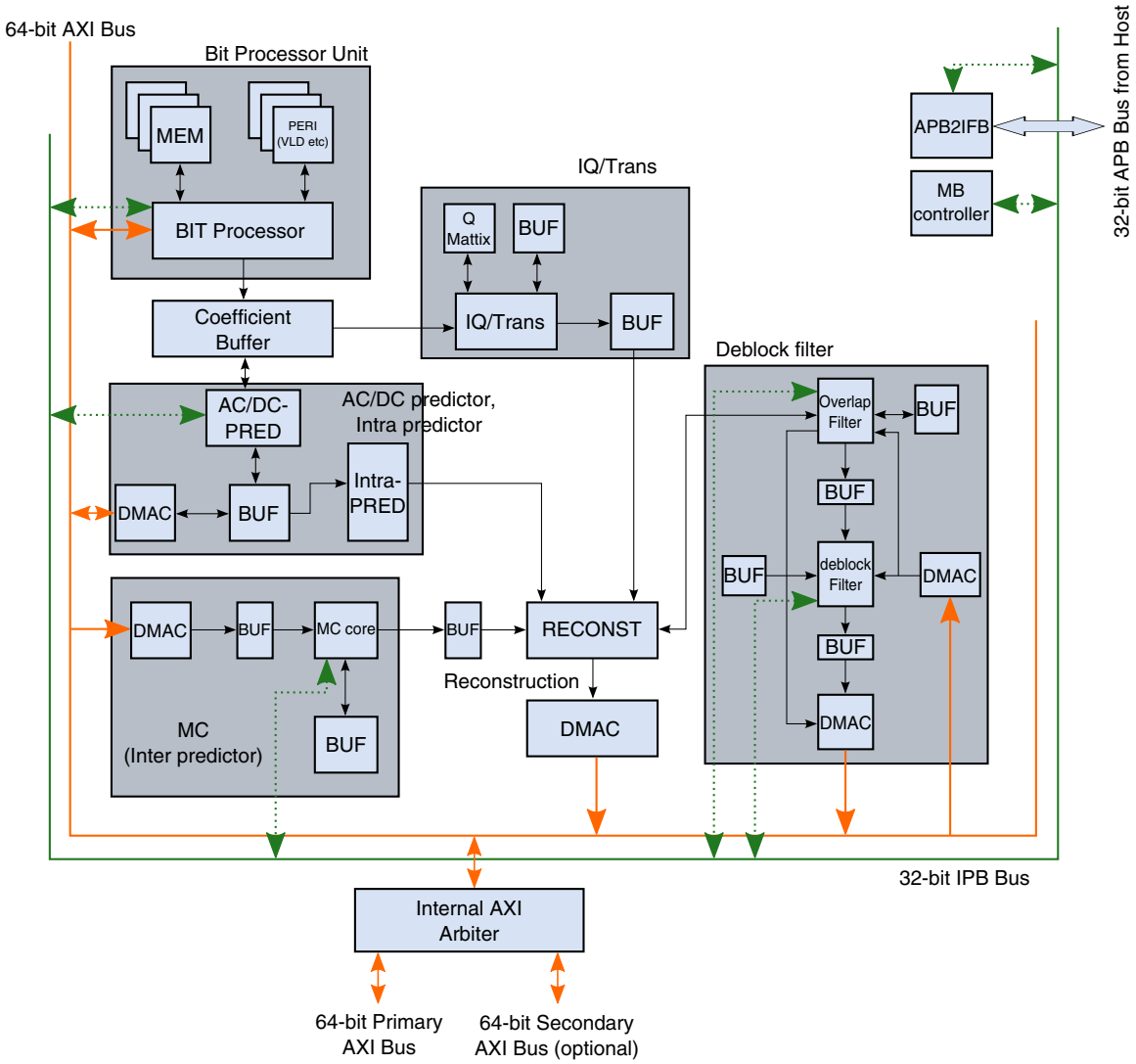


Figure 8-6. VPU Architectural Block Diagram

8.4.5 Interfaces

- Bus Interface
  - Primary AXI master: to System memory, 64-bit
  - Secondary AXI master: to iRAM, 64-bit
  - Host control: IP Bus slave, 32-bit
- Input/output Interface: partially interleaved 4:2:0 data format (YYY... CbCrCbCrCbCr...)
- Others

- One interrupt
- Integral DMA controller, with an AXI master port

(A)WID	Block	Description	External bus
0	BIT Processor	FW code, FW data, and bitstream	Primary for data related to FW.
1	Overlap Filter	Temporal buffer for overlap smoothing filtering processor (VC-1 only)	Either primary or secondary
2	Deblocking Filter	Temporal buffer for deblocking filtering process	Either primary or secondary
3	AC/DC prediction	Temporal buffer for AC/DC prediction(VC-1 only)	Either primary or secondary
4	Reconstruction	Reconstructed pixel row(s) for intra-prediction	Either primary or secondary
5	Deblocking Filter	Decoded frame	Primary
6	Range remapper	VC-1 range remapping/reduction	Primary
7	BIT Processor	MB information	Either primary or secondary

Figure 8-7. AXI Write Channel Traffic Sources

(A)RID	Block	Description	External bus
0	BIT Processor	FW code & data, bitstream	Primary for data related to FW
1	Overlap Filter	Temporal buffer for overlap smoothing filtering processor (VC-1 only)	Either primary or secondary
2	Deblocking Filter	Temporal buffer for deblocking filtering process	Either primary or secondary
3	ACDC –prediction	Temporal buffer for AC/DC prediction(VC-1 only)	Either primary or secondary
4	BIT Processor	MB information	Either primary or secondary
5	NA		
6	Range remapper	VC-1 range remapping/reduction	

Figure 8-8. AXI Read Channel Traffic Sources

8.4.6 Operating Frequencies

VPU operating frequency depends on the block task - decoder / encoder, standard, profile, resolution, bit rate, etc. Below are few estimations of operating frequencies:

- H.264 HP, 720p, 30fps, 14Mbps - 100MHz
- H.264 HP, 1080p, 30fps, 40Mbps - 200MHz

8.4.7 Architectural Features

- Ability to decode entire frames on a per frame basis.
- Upgradeable firmware.
- Multiple decoders and/or encoders can operate simultaneously in a time multiplex mode on frame-basis.
- Partial acceleration for any non-standard decoders is required as seen below

8.4.8 Memory Requirements

8.4.9 Internal Memory (iRAM)

The internal memory or iRAM is used for providing a temporal buffer video encoding and decoding process for the purpose of external memory bandwidth reduction. The iRAM can be accessed through the secondary AXI bus. The size of the iRAM should be determined by the video coding standard which uses the largest iRAM. This is the VC-1 as shown in the table below.

temporal buffer type		QCIF	CIF	VGA	525SD	625SD	720p	1080p
MB information		1,408	2,816	5,120	5,760	5,760	10,240	15,360
AC/DC prediction		1,408	2,816	5,120	5,760	5,760	10,240	15,360
Intra-prediction		704	1,408	2,560	2,880	2,880	5,120	7,680
overlap filter	VC-1	880	1,760	3,200	3,600	3,600	6,400	9,600
deblock filter	VC-1 MP	2,816	5,632	10,240	11,520	11,520	20,480	30,720
	VC-1 AP	5,632	11,264	20,480	23,040	23,040	40,960	61,440
	H.264 BP, H.263P3	1,408	2,816	5,120	5,760	5,760	10,240	15,360
	H.264 MP/HP	2,816	5,632	10,240	11,520	11,520	20,480	30,720
	RV10	2,816	5,632	10,240	11,520	11,520	20,480	30,720
Sub-total		4,400	8,800	16,000	18,000	18,000	32,000	48,000
Worse case deblocking		5,632	11,264	20,480	23,040	23,040	40,960	61,440
Worse case total (VC-1AP)		9328	18656	33920	38160	38160	67840	101760

Figure 8-9. Decoding Use Case (The Unit is byte for all the Numbers)

temporal buffer type		QCIF	CIF	VGA	525SD	625SD	720p
MB information		1408	2816	5120	5760	5760	10240
AC/DC prediction		1408	2816	5120	5760	5760	10240
Intra-prediction		704	1408	2560	2880	2880	5120
deblock filter	H.264 BP, H.263P3	1408	2816	5120	5760	5760	10240
ME search RAM		8384	14720	25088	27968	27968	48128
Worse case total (H263-P3)		12608	23168	40448	45248	45248	78848

Figure 8-10. Encoding Use Case (The Unit is byte for all the Numbers)

8.4.10 External Memory (SDRAM)

The following two tables present the typical amounts of external DRAM required by each specific encoder or decoder. For encoders, it can be seen that the total amount of SDRAM used for H264 and MPEG4 are the same. This is because only one reference frame is used for both encoders. For decoders, the worse case is for H264 and VC-1, which uses more amount of memory than other video coding standards.

		H.264		MPEG4	
Image Format		NTSC	720p	NTSC	720p
frame buf	current	1215	2700	1215	2700
	reconstructed	607.5	1350	607.5	1350
	reference	607.5	1350	607.5	1350
BIT processor		386	386	386	386
total		2816	5786	2816	5786

Figure 8-11. Typical Encoding External Memory Requirements (Unit are 1024 bytes. Note that current frame buffer = the frame being encoded + incoming frame)

		H.264		MPEG4		VC-1		RV10	
		720p	1080 i/p	720p	1080 i/p	720p	1080 i/p	720p	1080 i/p
frame buffer	reconstructed	3150	7140	3150	7140	3150	7140	3150	7140
	reference	7875	14280	3150	7140	3150	7140	3150	7140
	range red					3150	7140		
	post-process	2700	6120	2700	6120	2700	6120	2700	6120
BIT processor		514	514	514	514	514	514	514	514
Total		14239	28054	9514	20914	12664	28054	9514	20914

Figure 8-12. Typical Decoding External Memory Requirements (Unit is 1024 bytes), and with enabling rotation, mirror, and de-ringing

8.4.11 VPU Integration into SoC

Figure 8-13 shows the method of connecting the VPU core in the i.MX53 processor.

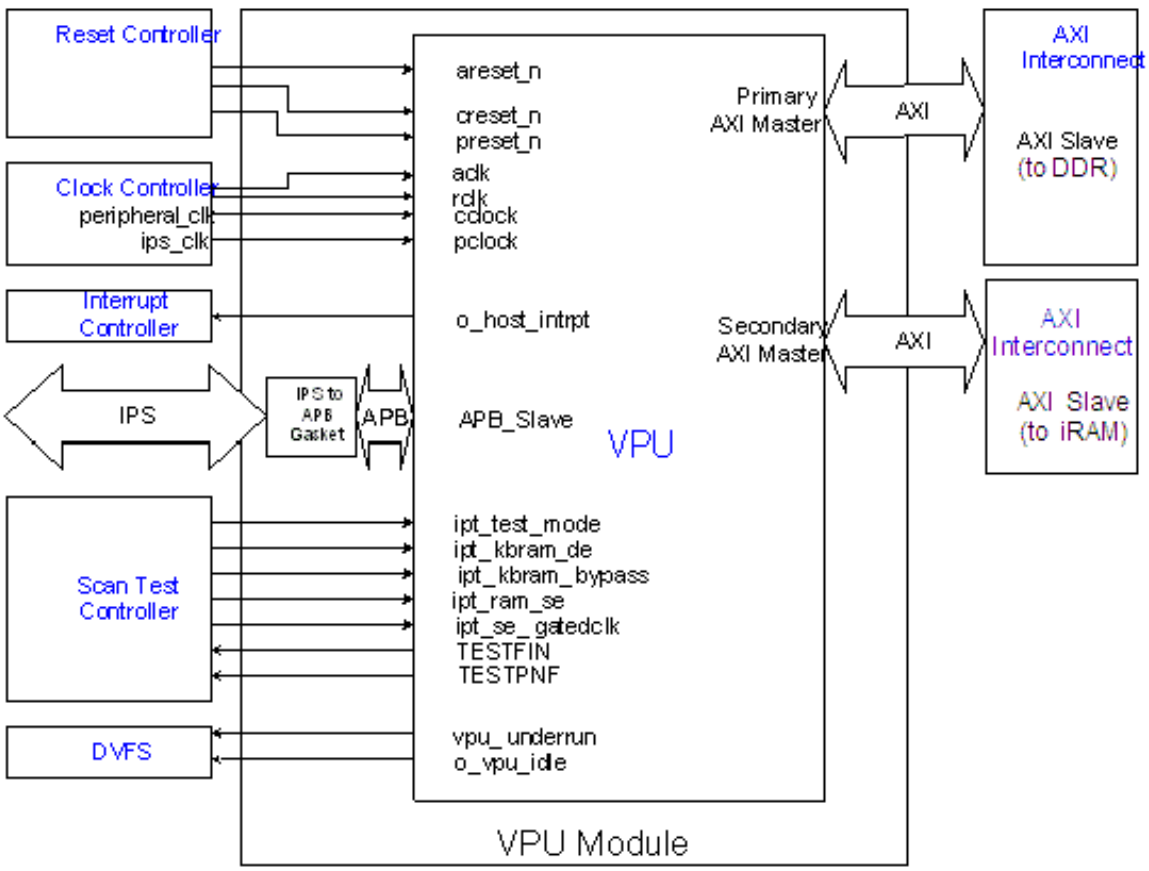


Figure 8-13. VPU System Connectivity

8.4.12 External Bus Connection

The VPU has the following interfaces:

- The first one is the Integrated Peripheral Bus (IP Bus) interface slave port. It serves as a configuration and programming interface. This interface has a maximum clock frequency of 66 MHz.
- There are two (2) AXI interfaces. The first one allows the VPU to access the main system memory (DDR). The second AXI interface allows the VPU to be connected directly to internal low-latency memory (iRAM) for bandwidth reduction purposes.

8.4.13 Other Signals Connection

Table 8-11 describes the other miscellaneous signals of the VPU.

Table 8-11. VPU Miscellaneous Signals

Name	Type	Source/ destination	Description
VPUNTRn	Output	Interrupt controller	Interrupt output from the VPU core. Active LOW.
VPUCLK	Input	Clock controller	Main input clock to the VPU core. That signal has a maximal frequency of 200 MHz
VPUCLKGATE	Output	Clock/power control logic	Signal permits gate the VPUCLK clock input.
ACLK	Input	Clock controller	AXI BUS clock. That signal has a maximal frequency of 200 MHz
ACLKGATE		Clock/power control logic	Signal permits gate the ACLK clock input of VPU.
SCANMODE	Input	Scan test controller	Test scan mode. A 1 on this pin forces scan mode.
SCANEN	Input	Scan test controller	Test scan enable. Active HIGH.
TESTSTRT	Input	Test controller	Embedded memory BIST test start/resume: 1 start/resume 0 normal operation.
TESTFIN	Output	Test controller	Embedded memory BIST test finished: 1 finished 0 test in progress or idle.
TESTPNF	Input	Test controller	Embedded memory BIST test pass/fail: 1 pass 0 fail
VPU_IDLE	Output	DVFS controller	Indicates Idle state of VPU. Could be used by DVFS controller.
VPU_UNDERRUN	Output	DVFS controller	Indicates that the VPU has taken longer to process a frame than the deadline established by rclock.
RCLCK	Input	Clock Controller	Low frequency clock used to determine the underrun condition

8.4.14 Clocking Architecture

The VPU uses four clock sources:

- AXI bus clock domain (aclk)
- Decoding core clock domain (cclk)
- IP bus clock domain (ipg_clk_s)
- Reference clock domain (rclk)

The VPU is designed following a Globally Asynchronous Locally Synchronous (GALS) architecture. As a result, there are no requirements to these clocks in terms of coherency, frequency or phase. Only the rising edges of the clocks are used. These clocks are not gated inside the VPU.

In i.MX53, the core clock (cclk) and axi (aclk) clocks are not tied together to the peripheral domain clock. The ipg_clk_s used in order to control VPU registers read/write function, ipg_clk_s is a gated clock of IP green-line clock (ipg_clk) with ips_module_en, it is turned off when there is no registers read/write, for power saving. The reference clock (rclk) is only used as reference clock for vpu_idle related logic.

8.4.15 Power Management

The Video Processing Unit (VPU)'s power is managed by two mechanisms: clock gating and frequency/voltage adjustments. The VPU core is connected to the peripheral power domain, therefore its voltage and frequency are controlled by the Dynamic Voltage Frequency Scaling for Peripherals (DVFSP) together with rest of the peripheral domain.

The DVFSP does not monitor the VPU directly to decide on entering the power saving mode. It monitors the signal VPU_UNDERRUN as an alarm signal that indicates the need to increase the clock frequency. VPU_UNDERRUN is generated as a count of rclock cycles and indicates that the processing of a frame has taken longer than the programmed time, usually 30 ms.

The ACLK_VPU and CCLK_VPU clocks are gated at the Clock Control Module (CCM) as soon as VPU_IDLE goes high to indicate that the current frame has been completed. To continue to work on the next frame the host re-enables the clockx by writing to the CCM directly.

8.4.16 Interrupt

The VPU has an interrupt request signal, o_host_intrpt, that is connected to the interrupt controller. The o_host_intrpt is synchronized to the rising edge of the cclk and goes to high when its asserted. The interrupt is cleared by system SW writing a '1' to the interrupt clear register of the VPU. The o_host_intrpt is HIGH until it is cleared.

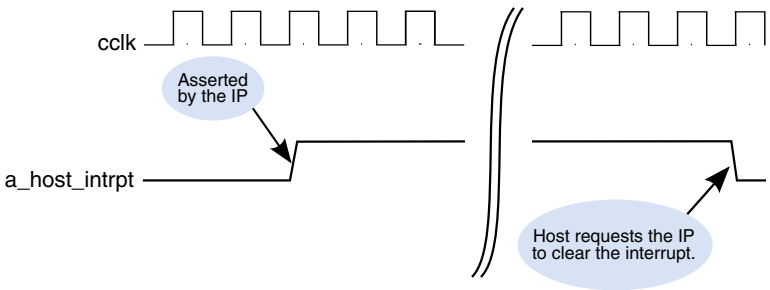


Figure 8-14. Interrupt Timing Diagram

8.4.17 Reset

The VPU has three reset signals areset_n, creset_n, preset_n that have to be synchronous to the corresponding clock. areset_n and creset_n are tied together to the peripheral domain reset. areset_n is tied to the ips bus reset.

In addition to these resets, the VPU can be reset under software control by writing to registers in the VPU memory-mapped register area. Three logical blocks are reset within the design:

- IP to APB gasket
- AXI interfaces
- Core logic

In SCANMODE the soft reset controls are bypassed and the system reset is connected directly to block resets to ensure observability.

All pending transactions are discarded after VPU core stop or reset in the AXI.

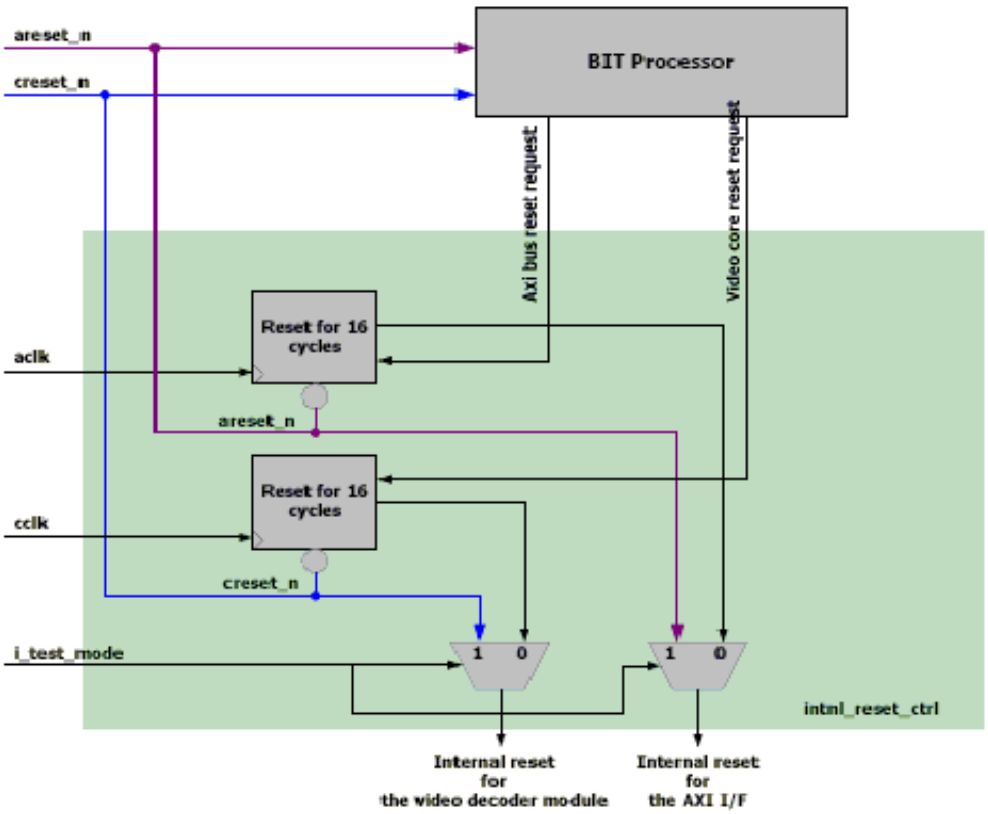


Figure 8-15. VPU Reset

8.5 OpenGL/ES Graphics Processing Unit 3D (GPU3D)

8.5.1 GPU3D Overview

The 3D Graphics Processing Unit (GPU3D) is based on the AMD Z430 IP core (also known as ATI Yamato DX). The core is an embedded engine capable of DirectX9 Shader Model 3.0+ program execution. The design is based on the ATI Xenos core, which is found in the Microsoft XBox 360 and has been extrapolated into recent PC video card designs. The block is focused on accelerating user level graphics APIs such as OpenGL ES 2.0 & 1.1, OpenVG 1.0, and Direct3D Mobile.

8.5.2 GPU3D Features

- Built to accelerate OpenGL ES 2.0

- Uses dynamically shared shader ALU/memory resources between vertex and pixel shaders.
  - Unified Shading Architecture
- Command Processor:
  - Advanced packet based command stream manager allowing for efficient and flexible transfer of graphics commands and host data from the host system to the graphics processor core
- Integrated Power Management
  - Block-level clock gating managed automatically in the IP.
- Graphics Memory Controller and Graphics Memory
  - Customer configurable on-chip memory used to accelerate 3D rendering using a binning architecture significantly reducing external system bandwidth requirements.

8.5.2.1 Capabilities

- Supports 2 & 4-sample MSAA at full rendering speed.
- Compressed Texture Support: ETC, DXTC, and ATI_TC
- Supports general purpose exports to system memory from Vertex and Pixel shaders.
- Uses indexed vertex data fetches from the Vertex Shader as a high-bandwidth vertex data path.
- Latency hiding via FIFOs and multi-threading
- Sophisticated Shader support
  - 512 4-Component constants
  - 1024 Shader Instructions
  - 16 Textures
  - 16 4-Component Interpolants
  - 64 General Purpose Registers
- Rich texture format and type
  - Compressed
  - 16 bit Floating point
  - Cube map
  - Volume textures
  - Non power-of-2
  - Anisotropic
- 1 Primitive (Triangle/Line/Point) every 6 clocks
- 1 Vector (4 component) and 1 Scalar (single component) ALU instruction per clock
- 2 Control Flow Instructions (Jumps, loops) per clock
- 1 Export per clock
- 1 4 component Pixel Shader input interpolant per clock
- Early-Z testing at up to 4 pixels per clock

- 1 pixel (at 4 sample MSAA) per clock with alpha blending and depth test
- 32 bit FP Internal Shader Precision

8.5.3 GPU3D Block Diagram

The Z430 contains three primary blocks:

- Graphics Core (GC)
  - Programmable graphics engine that performs all of the data processing and memory transactions
- Graphics Arbiter (GARB)
  - Controls the GMEM and arbitrates between requests from the GC and the system
- Graphics Memory (GMEM)
  - Local SRAM buffers (four 32KB buffers) that are used to data for the tile / bin of pixels the GC is working on

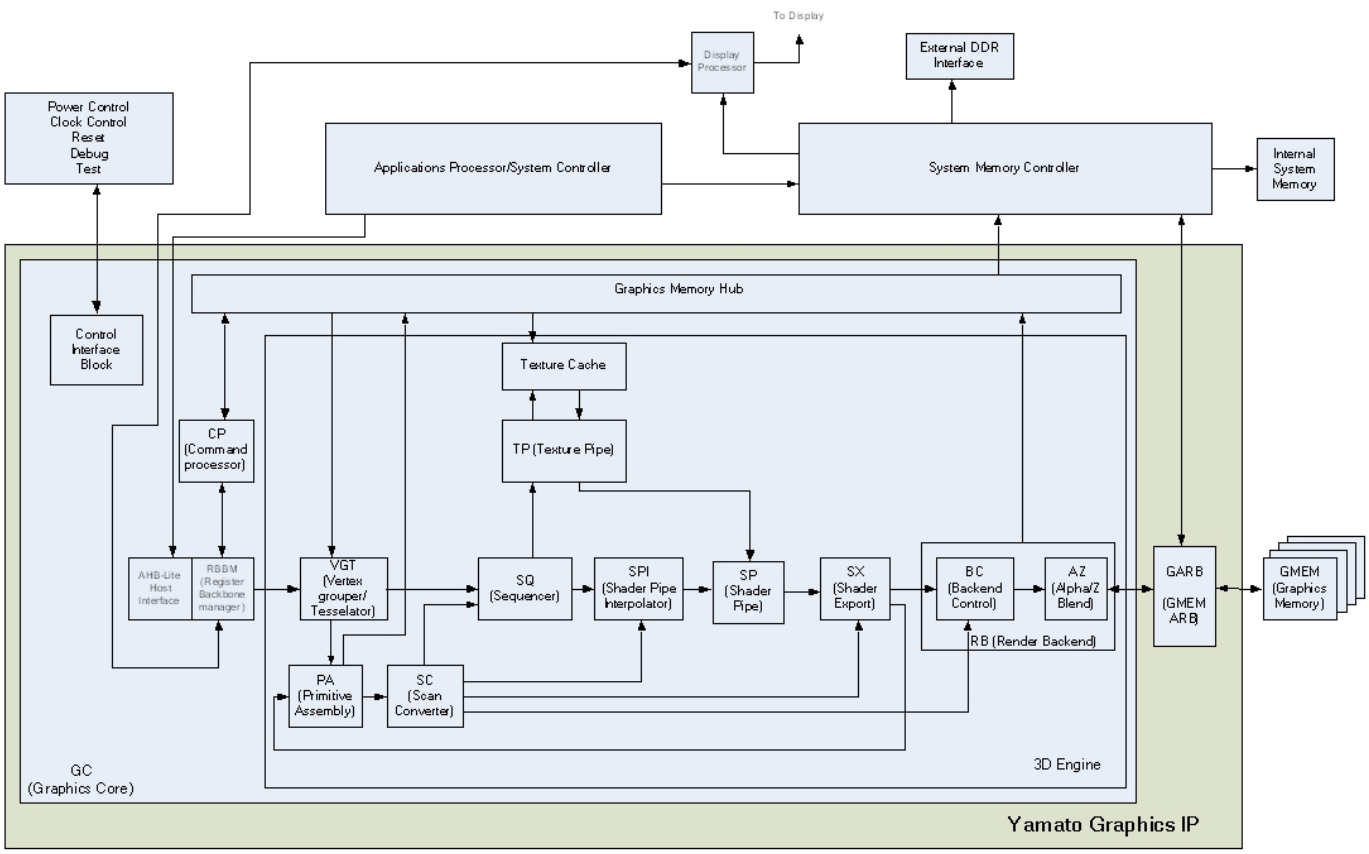


Figure 8-16. GPU3D (Z430) Internal Block Diagram

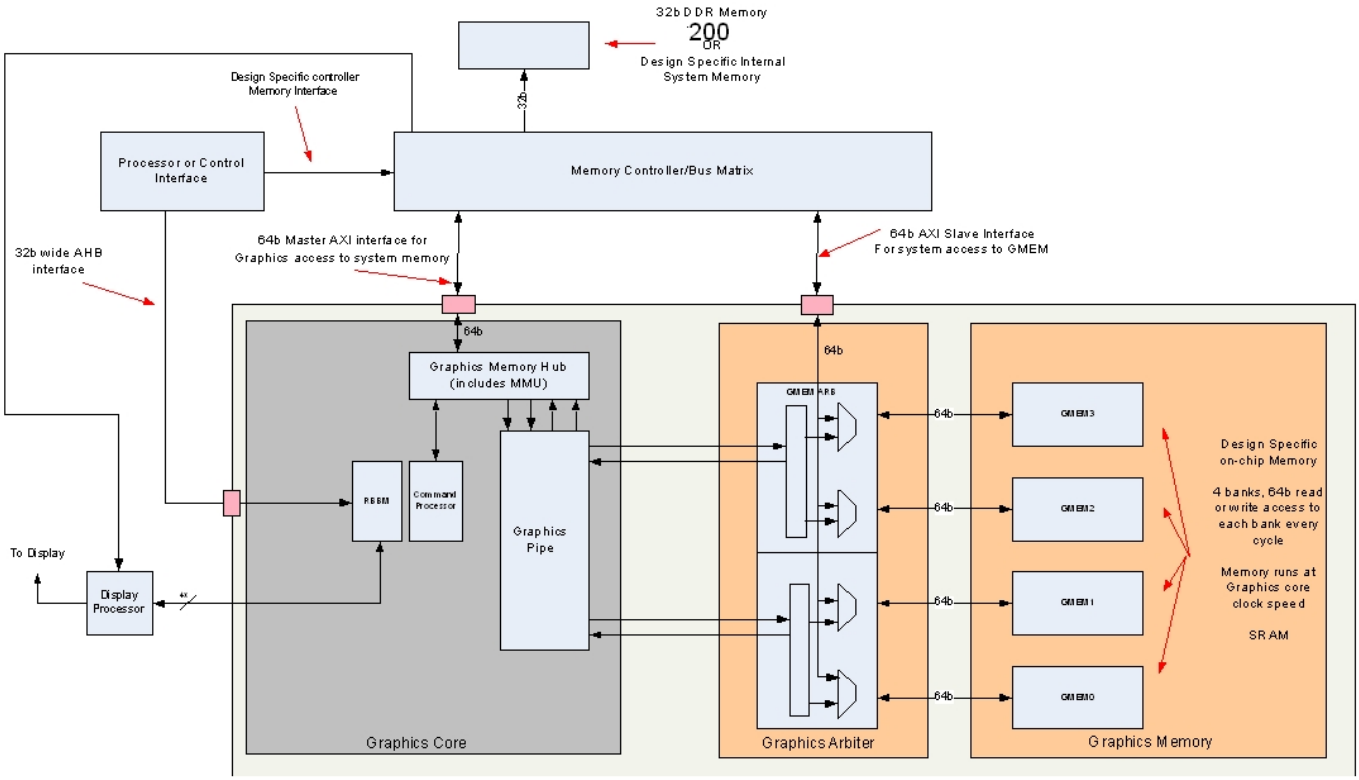


Figure 8-17. GPU3D (Z430) Interface Diagram

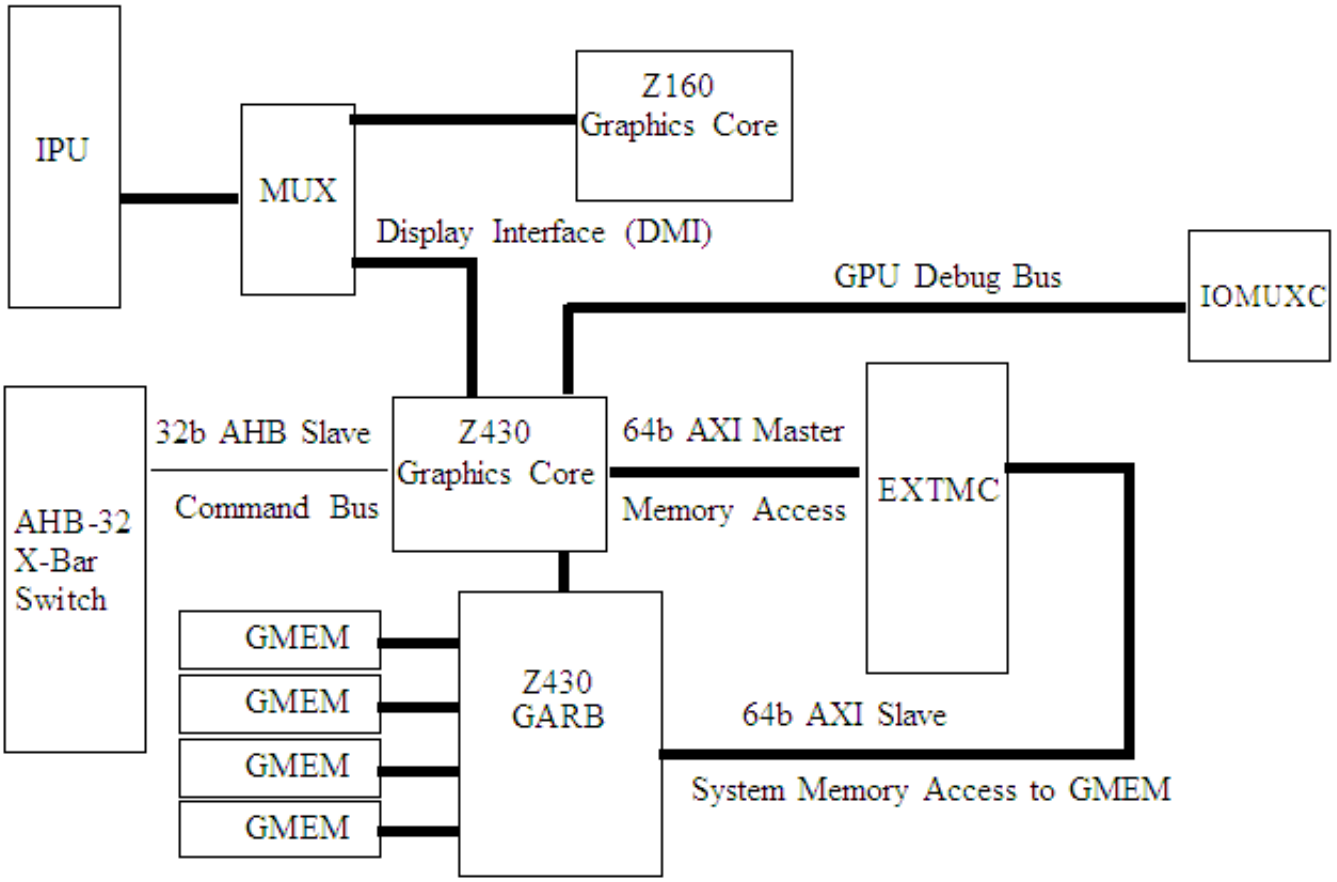


Figure 8-18. GPU3D System Integration Diagram

Interface Summary:

- Slave Interface to ARM platform or controller interface
  - Peripheral AMBA 2.0 AHB-Lite Host Interface
    - Provides host access to graphics core from the system processor.
    - 32b wide, synchronous with core.
    - Support for clock gating for phase removal on AHB clock.
- Memory Master Interface
  - AXI Master Interface into memory system (AXI V1.0)
  - 64b wide, synchronous
    - Provides access to system memory from the graphics core.
  - Programmable number of read and write requests to the memory system
    - Number of outstanding requests is limited by system resources (i.MX53 can support one request).
- Memory Slave Interface (GARB)
  - AXI slave interface for system access to graphics memory (AXI V1.0)
  - 64b wide, synchronous interface
  - 1 outstanding read and write request
- Display Module Interface (DMI)

- For synchronization with the display processor.
- Provides indication to display processor from graphics core that a newly rendered frame is ready and that the display must perform a buffer flip at the next vertical blank.

Table 8-12. DMI Output Clocking Signals

Signal Name	Type	Width	Description
USE_BUFID0	Output	1	When active, indicates that frame buffer 0 should be displayed next
USE_BUFID1	Output	1	When active, indicates that frame buffer 1 should be displayed next
USE_BUFID2	Output	1	When active, indicates that frame buffer 2 should be displayed next

- Provides indication to graphics core from the display processor that the buffer identified by buffer ID has been displayed at least once.

Table 8-13. DMI Input Clocking Signals

Signal Name	Type	Width	Description
VSYNC_VALID	Input	1	Vertical sync signal from display controller for indicating vertical blanking period
ACTIVE_BUFID0	Input	1	When active, indicates that frame buffer 0 is currently being displayed by the display controller
ACTIVE_BUFID1	Input	1	When active, indicates that frame buffer 1is currently being displayed by the display controller
ACTIVE_BUFID2	Input	1	When active, indicates that frame buffer 2is currently being displayed by the display controller

- The DISP signals support up to 4 display channels. For each channel, there is a set of signals for the GPU3D to signal the newest available buffer (supporting up to 3 frame buffers), a set of signals for the system display controller (IPU) to signal the buffer currently being displayed, and a vertical sync.
- The vertical sync (VSYNC) from the display controller occurs after each frame is displayed, during the vertical blanking period. One-hot encoded USE_BUFID signals are updated by Z430 whenever there is a new rendered buffer available (and the previous value of USE_BUFID has already been sampled by the IPU/

display controller), so it can change anywhere in the frame. It is resynchronized and sampled a few cycles before the start of the vertical blanking period (rising edge of VSYNC) in display controller, while the currently displayed buffer (ACTIVE_BUFID signals) is updated by display controller at the rising edge of VSYNC with these values that were just sampled from Z430 a few clock cycles before the VSYNC. Signals ACTIVE_BUFF are then re synchronized and sampled in Z430 after synchronizing to VSYNC rising edge.
    - At least one of the display channel interfaces must be directly connected to the system display controller (IPU) to achieve the lowest delay between frame renderings.
    - The IPU must have controls to enable the interface, to utilize the double and triple buffering mechanisms, and to lock and unlock the rendering to the display's vertical sync.
- Control and Power Management
    - Control Interface Block (CIB)
        - Holding block for clock gating, reset etc.
        - Interrupt Interface (INT)
        - Error and status interrupts to the controller.
        - Activity Halt Interface (AHI)
        - Debug bus outputs
        - GPIO bus for miscellaneous control functions
    - Design For Test Interface (DFT)
        - Memory test collar access.
        - Scan/BIST access.
- Debug
    - Set of signals OR chained between internal masters for debug purposes
    - 32b bus multiplexed to 16b

8.5.4 GPU3D Performance

Graphics primitive rendering speed

- 1 Primitive (Triangle/Line/Point) every 6 clocks.
- Up to 1 Vertex Shader Instruction per clock (shared with Pixel Shader)
- Up to 4 Pixel per clock Z only processing (in Single Sample). 2 Pixels/clk in 2x MSAA mode.
    - 1 pixel (at 4 sample MSAA) per clock with alpha blending and depth test
- 1 Pixel Shader Input Interpolant (4-D IEEE 32-bit float) per clock + 4 Z samples generated/clock
- 1 Pixel Shader Instructions per clock (shared with Vertex Shader)

- 1 Bilinear-filtered Texel fetches per clock.
- 1 Pixel per clock with alpha blending and depth test at up to 4 samples/clk

8.5.4.1 GPU3D Memory Accesses

The Z430 generates 128 and 256 bit bursts to memory from its internal masters, which are arbitrated by a Memory Hub (MH). The MH supports up to 64 outstanding transactions, 8 AXI IDs, and contains a MMU.

Typical graphics use cases (usually games) will utilize approximately 400 to 500 MB/s and require a memory latency of less that 90ns (70ns is preferred).

The Memory Hub selects between the following clients in the graphics core:

- CP - (Command Processor) Read/Write
- VGT - (Vertex Grouper and Tesselator) Read Only
- TC - (Texture Cache) Read only
- RB - (Render Block) Write Only

Sub-Client	Sub-Client Designation	AXI ID	Operation	Transaction Size (Bytes)	Sub-Client Outstanding Transaction Limit
Ring Buffer	CPr0	0	Read	32	Programmable (Note 1)
Indirect Buffer #1	CPr1	1	Read	32	Programmable (Note 1)
Indirect Buffer #2	CPr2	2	Read	32	Programmable (Note 1)
State Sub-Block, Constant, & Shader Instruction Data	CPr3	3	Read	32	Programmable (Note 1)
Micro-Engine & Write Pointer Polling	CPr4	4	Read	32	1 Micro-engine read & 1 write pointer read
VGT Indices	VGTr0	5	Read	32	5 (Note 2)
VGT Bin ID	VGTr1	6	Read	32	8
Texture/Vertex Read	TCr	7	Read	16/32	9
Synchronization Semaphores, Micro-Engine Semaphores, Constant State Data	CPw	Programmable	Write	16/32	8 Unconfirmed writes, unlimited writes past point of confirmation
RB Copy	RBw	Programmable	Write	32	No client limit
PA	PAw	Programmable	Write	32	1
MMU TLB Miss	MMUr	Programmable	Read	32	1

All transactions are incrementing address bursts. The starting address is always aligned to the burst size, for example all 16 Byte bursts start at a 16-Byte aligned address; all 32 Byte bursts start at a 32-Byte aligned address.

Command stream and vertex data is read in bursts of 256 bits.

The Burst cache combines color accesses from several smaller objects to 4x4 pixel (16-bit), or 4x2 pixel (32-bit), which are transferred to memory in bursts of 256 bits.

Color writes use byte enables when incomplete pixel blocks are written to the memory

8.5.5 GPU3D Clocking

Yamato design uses a single synchronous clock SYS_GC_sclk. There is a single time domain, and no multicycle paths, latches or negative edge flops are used in the design. (The CKGATER uses a negative latch, but that is a special cell).

The SYS_GC_sclk should be implemented with programmable dividers to run at full speed, synchronous bus speed, and half bus speed to meet system power use cases.

The Control Interface Block (CIB) provides clocks, internal clock gating and reset the rest of the internal blocks. The Graphics Core (GC) and Graphics Arbiter (GARB) are provided with two versions of the core clock (SYS_GC_sclk and SYS_GA_sclk from the CIB). Both clocks are synchronous and in phase with core clock. Two versions are supplied so that clocks to the GC can be removed independently of the GARB.

8.5.5.1 GPU3D Clock Gating

Yamato supports three levels of clock gating:

- Explicit removal of clocks to the GC core by the system under software control
- Designer controlled removal of clocks to logic blocks
- Power compiler instantiated clock gating.

Explicit clock gating is invoked under system control and as such is transparent to the GC. All clocks (GC, AXI, AHB) are removed. Prior to clock removal, the system should ensure that the GC is idle (no outstanding bus traffic, no outstanding command activity in command buffer). Removing the clocks to the GC does not impact GC state (unless power is also removed).

Designers can chose to remove clocks to blocks or sections of blocks based on those functions becoming idle. To facilitate a clean mechanism for transition to idle state, all block clock control signals are propagated to the top level where they are combined and flopped in the CIB. Outputs from the CIB at the top level are used to control clock gaters which are used to remove clocks in a glitch free manner. All CIB clock-gater control signals come directly from a flop.

Blocks may chose to implement a gated and a non-gated version of the clock. Non-gated versions of the clock are fed through clock gaters which are always enabled. This will help with PD balancing of gated vs. non-gated clocks.

8.5.6 GPU3D Resets

The Yamato core uses an asynchronous reset strategy. This signal is propagated throughout the block by the Control Interface Block. The GPU3D takes 64 clock cycles to complete a reset.

8.5.7 GPU3D Interrupts

The graphics core provides a single interrupt pin to the system. Once asserted the interrupt pin remains asserted until the ARM platform clears the IRQ bit in the RBBM_IRQ register.

In general interrupts are used to signal error conditions or to support frame buffer swap.

8.5.8 Debug

The GPU3D has a 32b debug bus that is controlled by a 16b GPIO bus. The GPU3D GPIO bus should be memory mapped (as a single register) and provides byte enables to select debug information and soft reset of blocks. The 32b debug bus should be mapped to system GPIO. There is also an option of using a 32b debug input bus that is logically ORed with the debug output.

- 00 - output lower 16 bits
- 01 - output upper 16 bits
- 10 - toggle between outputing lower and upper 16 bits per 83MHz clock
- 11 - unused

In addition, the interface is limited to a 83MHz clock due to the max pin frequency.

8.5.9 Software

This figure shows the general software stack utilized by applications using the Z430 GPU3D (shown as Yamato GPU3D in the diagram). The stack is here for background purposes and details will be expanded in a separate section.

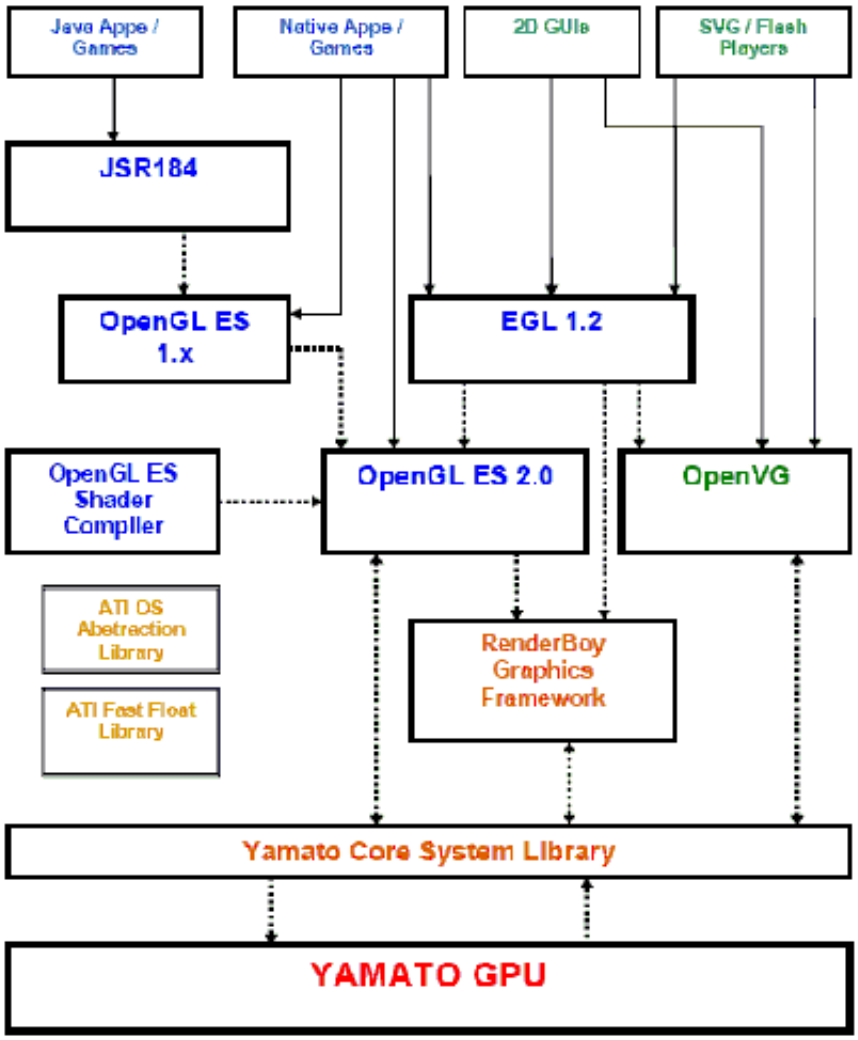


Figure 8-19. Graphics Software Stack

8.6 Graphics Processing Unit 2D (GPU2D)

8.6.1 GPU2D Overview

The 2D graphics processing unit is based on the AMD Z160 IP core (also known as ATI / Bitboys G12). The core is an embedded, 2D and vector graphics accelerator targeting the OpenVG 1.0 graphics API and feature set. The core is divided into two segments. The first accelerates 2D bitmap graphics operations, such as BitBlt, fill and raster operations using a separate 2D graphics acceleration unit. Vector graphics rendering is accelerated by a separate anti-aliasing polygon rasterizer. The core has a rich, but well-chosen set of features, with emphasis being on very high image quality and low memory bandwidth consumption.

8.6.2 GPU2D Features

- Frame buffer sizes supported up-to 2048x2048
- ARGB4444, RGB565, ARGB1555, ARGB5551, ARGB8888 frame buffer modes
- Configurable ARGB order in frame buffer: ARGB, BGRA, ABGR, RGBA
- Linear and block-based (4x4 pixels) frame buffer modes
- Fast buffer clears
- Support for OpenVG render to Image

8.6.2.1 2D Bitmap Graphics (Separate 2D unit)

- BitBlt (surface-to-surface copy)
  - Format conversion from monochrome/ARGB/YUV to ARGB during BitBlt
- Block fill
- Internal 32-bit color precision
- Source bitmap format:
  - 1/4/8-bit monochrome
  - ARGB4444, RGB565, ARGB1555, RGBA5551, and ARGB8888
  - Configurable ARGB order: ARGB, BGRA, ABGR, RGBA
  - Packed YUV 4:2:2 formats (FOURCC codes YUY2, UYVY, YVYU), two pixels per 32 bits of data
  - 1-bit bitmap maps to foreground and background colors
  - 4-bit bitmap is optionally gamma corrected to 8-bit alpha values and can be combined with foreground color to draw anti-aliased fonts
- Destination bitmap format:
  - ARGB4444, RGB565, ARGB1555, RGBA5551, and ARGB8888
  - Configurable ARGB order: ARGB, BGRA, ABGR, RGBA
- Supports three source bitmaps for separate mask/pattern/alpha bitmap support plus reading destination for ROP, blend and color key operations
- Supports masking source coordinates for wrapping patterns
- Supports ROP4 (ROP3 with separate ROPs for masked and unmasked pixels) logical operations
- Supports inverting mask and alpha values from source
- Supports destination rotation by 0/90/180/270 degrees
- Supports programmable blending with optional alpha un-premultiply
- Supports per pixel and constant alpha with optional modulation by source color alpha for OpenVG alpha masking
- Supports color keying by source and destination colors, with optional ignoring of alpha channel

- Supports one scissor rectangle for destination coordinates
- Dithering (ordered)
- Color component masking
- sRGB reads and writes
- Non-power of two source and destination bitmap sizes supported (stride must be a multiple of 32-bits)
- BitBlt with scaling, bilinear filtering with texture lookups, programmable filter kernels possible with the programmable Pixel processor

8.6.2.2 Vector Graphics

- Rasterization of convex and concave polygons with anti-aliasing
- Efficient native polygon rendering (no tessellation to triangles)
- Non-zero and odd-even fill rules
- Primitives supported:
  - Polygons
  - OpenVG path primitives (except Elliptical Arcs): Horizontal/vertical lines, generic lines, curves, smooth curves, moveto, path closing
  - Curve types supported: cubic and quadratic Bézier
  - Strokes with thickness, joints and end caps, unlimited stroke thickness
  - Special case handling of singularities for thick strokes
  - Supports paths with a maximum of 256 crossings along a horizontal or a vertical line
- Input coordinates:
  - Absolute and relative coordinate input in floating point
  - Fixed-point (byte, short, int) and floating-point coordinate input - 0.8, 0.16, 16.16 formats
- Geometry
  - User to surface transform for vertices and stroke shape
  - Hardware curve tessellation
  - Adjustable accuracy for curve and round cap splitting
  - OpenVG/SVG join types: Miter (with miter limit), round, bevel
  - OpenVG/SVG cap types: Butt, round, square
- Pixel processing:
  - Programmable gradient and texturing processor
  - Linear and radial gradients (with focal point)
  - Perspective texture mapping with filtering
  - Two textures supported
  - sRGB and pre-multiply support for textures
  - 16-sample anti-aliasing

- Per-pixel alpha-masking
- Maximum texture size: 1024x1024 pixels

- Vector graphics rendering system AR M platform load:
  - Display list generation during path creation - commands and vertices are stored to an internal format/buffer, no format conversion is performed
  - Filling or stroking a path only requires a few register writes to start the operation in hardware
  - Display lists are transferred to the vector graphics rasterizer using DMA without ARM platform interaction

8.6.3 GPU2D Block Diagram

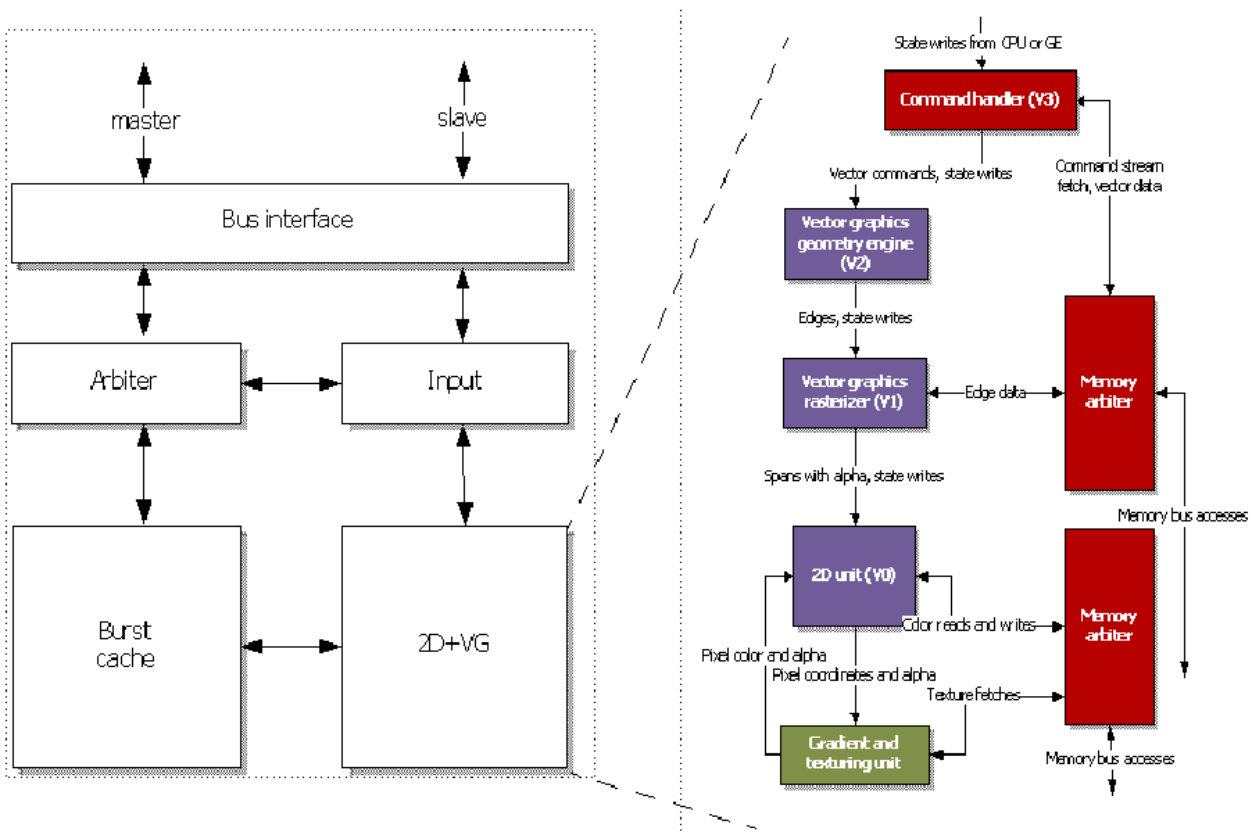


Figure 8-20. GPU2D (Z160) Internal Block Diagram

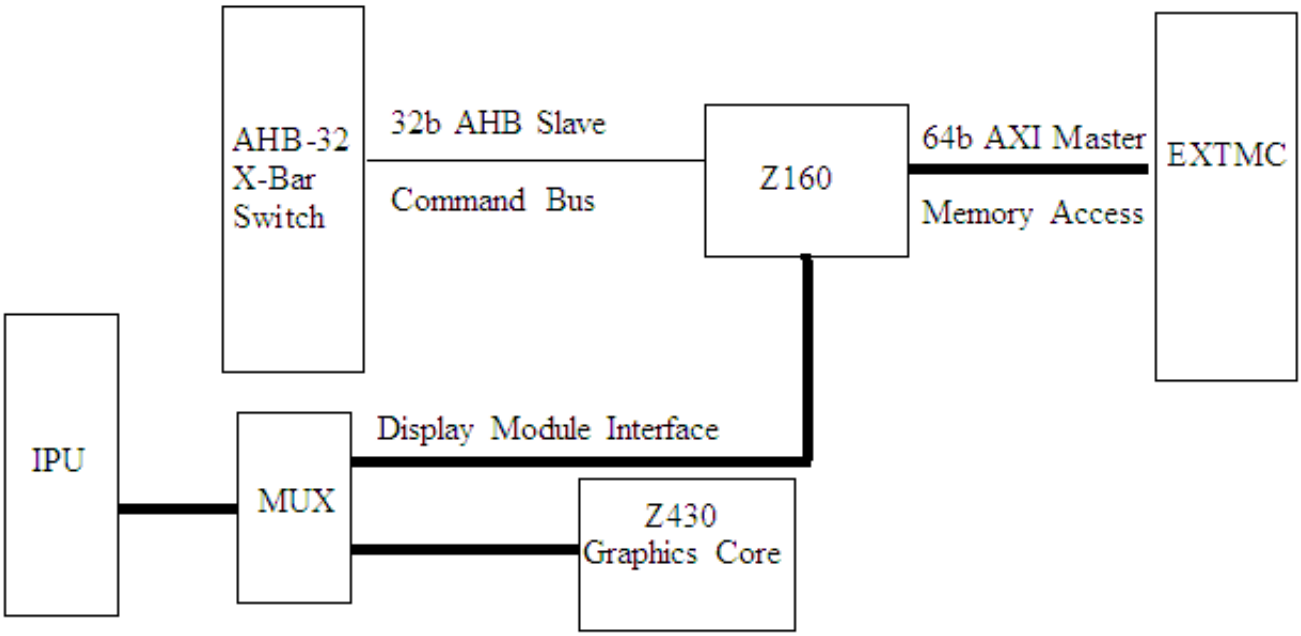


Figure 8-21. GPU2D (Z160) Interface Diagram

Details on the Display Module Interface (DMI) can be found in the GPU3D chapter.

8.6.4 GPU2D Performance

Graphics primitive rendering speed

- 5 clock cycles per vertex (includes transformation)
- 1 pixel / clock (solid color or linear gradient)
- 2 clocks / pixel (affine texture mapping)
- 4 clocks / pixel (with texturing with perspective transformation and radial gradients without separate focal point)
- 8 clocks / pixel (radial gradient with separate focal point)

8.6.4.1 GPU2D Memory Accesses

Command stream and vertex data is read in bursts of 256 bits.

The Burst cache combines color accesses from several smaller objects to 4x4 pixel (16-bit), or 4x2 pixel (32-bit), which are transferred to memory in bursts of 256 bits.

Color writes use byte enables when incomplete pixel blocks are written to the memory

8.6.5 GPU2D Clocking

Table 8-14. Z160 Clocking Signals

Signal Name	Description
busclk	BusIF clock signal
clk_0	Continuos core clock, arbiter and input (not gated)
clk_1	Bcache clock signal
clk_2	2d and VG_V3 clock signal
clk_3	VG_V2 and VG_V1 clock signal
clock_1_ena_o	clock enable for clk_1 (0=clock disabled, 1=clock activated)
clock_2_ena_o	clock enable for clk_2 (0=clock disabled, 1=clock activated)
clock_3_ena_o	clock enable for clk_3 (0=clock disabled, 1=clock activated)

8.6.5.1 GPU2D Clock Gating

There are two different clock gating modes in the GPU2D:

- G12_POWER_OFF - In this mode all the clocks to the G12 are switched off. This mode is only used when the graphics core is totally idle.
- G12_POWER_SW - In this mode the clock gating is controlled by the driver. Appropriate clock gating commands are interleaved into the command stream between render state changes. This is the default state after the core has been reset.

8.6.6 GPU2D Resets

The Z160 has two reset signals:

- *busrst_n* for the bus reset signal
- *rst_n* for reset of all clocks

8.6.7 GPU2D Interrupts

The core has a single general purpose programmable interrupt called *irq_o*.

8.7 Audio Subsystem

8.7.1 Overview

The blocks that belong to the audio subsystem are the SSI-1, SSI-2, SSI-3, AUDMUX, ESAI, SPDIF and ASRC. In addition, the IOMUXC must be appropriately configured to get signals in and out of the chip.

SSI-1,2,3 are synchronous serial interfaces used to transfer audio data. SSI-1,3 are on the peripheral bus and SSI-2 is on the shared peripheral bus. Instead of connecting to the IOMUXC directly, their serial lines connect to the Digital Audio Multiplexer (AUDMUX).

The AUDMUX provides flexible, programmable routing of the serial interfaces (SSI-1 or SSI-2) to and from off-chip devices. The AUDMUX routes audio data (and even splices together multiple time-multiplexed audio streams) but does not decode or process audio data itself. The AUDMUX is controlled by the ARM but can route data even when the ARM is in a low-power mode.

The Enhanced Serial Audio Interface (ESAI) provides a full-duplex serial port for serial communication with a variety of serial devices, including industry-standard codecs, SPDIF transceivers, and other processors. The ESAI consists of independent transmitter and receiver sections, each section with its own clock generator. The ESAI is connected to the IOMUXC and to the ESAI_BIFIFO sub-block.

The ESAI Bus Interface and FIFO (ESAI_BIFIFO) is the interface between the ESAI block and the shared peripheral bus. It contains the FIFOs used to buffer data to/from the ESAI, as well as providing the data word alignment and padding necessary to match the 24-bit data bus of the ESAI to the 32-bit data bus of the shared peripheral bus.

The Sony/Philips Digital Interface (SPDIF) audio block is a stereo transceiver that allows the processor to receive and transmit digital audio over it. The SPDIF receiver section includes a frequency measurement block that allows the precise measurement of an incoming sampling frequency. A recovered clock is provided by the SPDIF receiver section and may be used to drive both internal and external components in the system. The SPDIF is connected to the shared peripheral bus.

The Asynchronous Sample Rate Converter (ASRC) converts the sampling rate of a signal associated to an input clock into a signal associated to a different output clock.The ASRC supports concurrent sample rate conversion of up to 10 channels of over 120dB THD+N. The sample rate conversion of each channel is associated to a pair of incoming and outgoing sampling rates.The ASRC supports up to 3 sampling rate pairs. The ASRC is connected to the shared peripheral bus.

8.7.2 Audio Subsystem Block Diagram

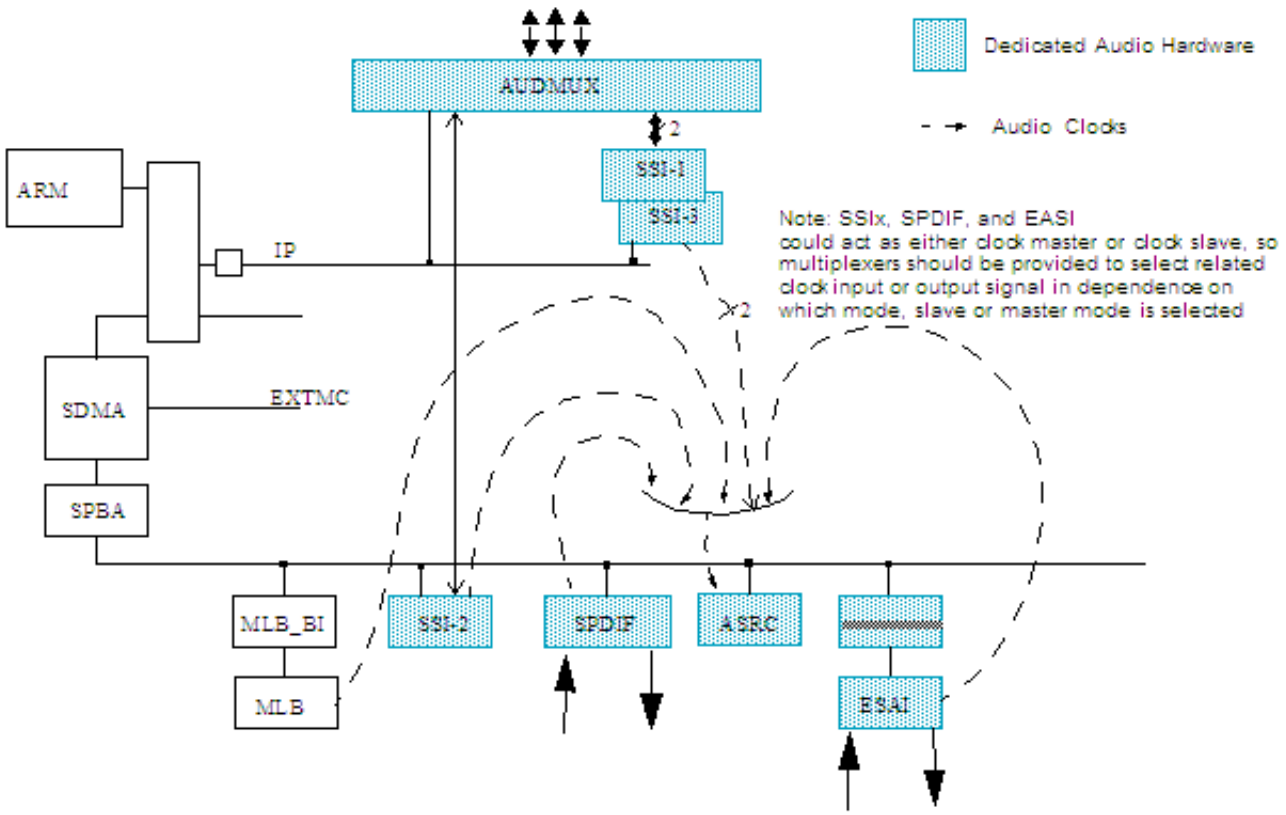


Figure 8-22. Audio Subsystem Block Diagram

8.7.2.1 Standard Serial Interface Controller (SSI)

The Standard Serial Interface Controller (SSI) is a full-duplex serial port that allows communication with external devices using a variety of serial protocols. The SSI supports a wide variety of protocols (SSI normal, SSI network, I2S, and AC-97), bit depths (up to 24 bits per word), and clock/frame sync options.

The SSI has two pairs of 8x24 FIFOs and hardware support for an external DMA controller in order to minimize its impact on system performance. The second pair of FIFOs provides hardware interleaving of a second audio stream which reduces ARM platform overhead in use cases where two timeslots are being used simultaneously.

The three SSIs may support three audio streams (possibly at different sample rates) simultaneously. SSI-1 and SSI-3 are located on the IP Bus, while SSI-2 is located on the Shared Peripheral Bus. Because the SDMA can directly access SSI-2 (being on the Shared Peripheral Bus), it is recommended that SSI-2 be used for high-bandwidth data transfers in order to optimize bus bandwidth consumption. SSI-1 and SSI-3 can then be used for lower bandwidth applications.

8.7.2.2 Digital Audio MUX (AUDMUX)

The Digital Audio Multiplexer (AUDMUX) provides a programmable interconnect fabric for voice, audio, and synchronous data routing between host serial interfaces, such as SSI, and peripheral serial interfaces-that is, audio and voice codecs.

The AUDMUX includes two types of interfaces. Host ports connect to the processor serial interfaces, and peripheral ports connect to off-chip audio devices. A desired connectivity is achieved by configuring the appropriate host and peripheral ports.

The AUDMUX provides flexible, programmable routing of the on-chip serial interfaces to and from off-chip audio devices. The AUDMUX routes audio data (and even splices together multiple time-multiplexed audio streams) but does not decode or process audio data itself.

Figure 8-23 illustrates how the AUDMUX is connected in the system.

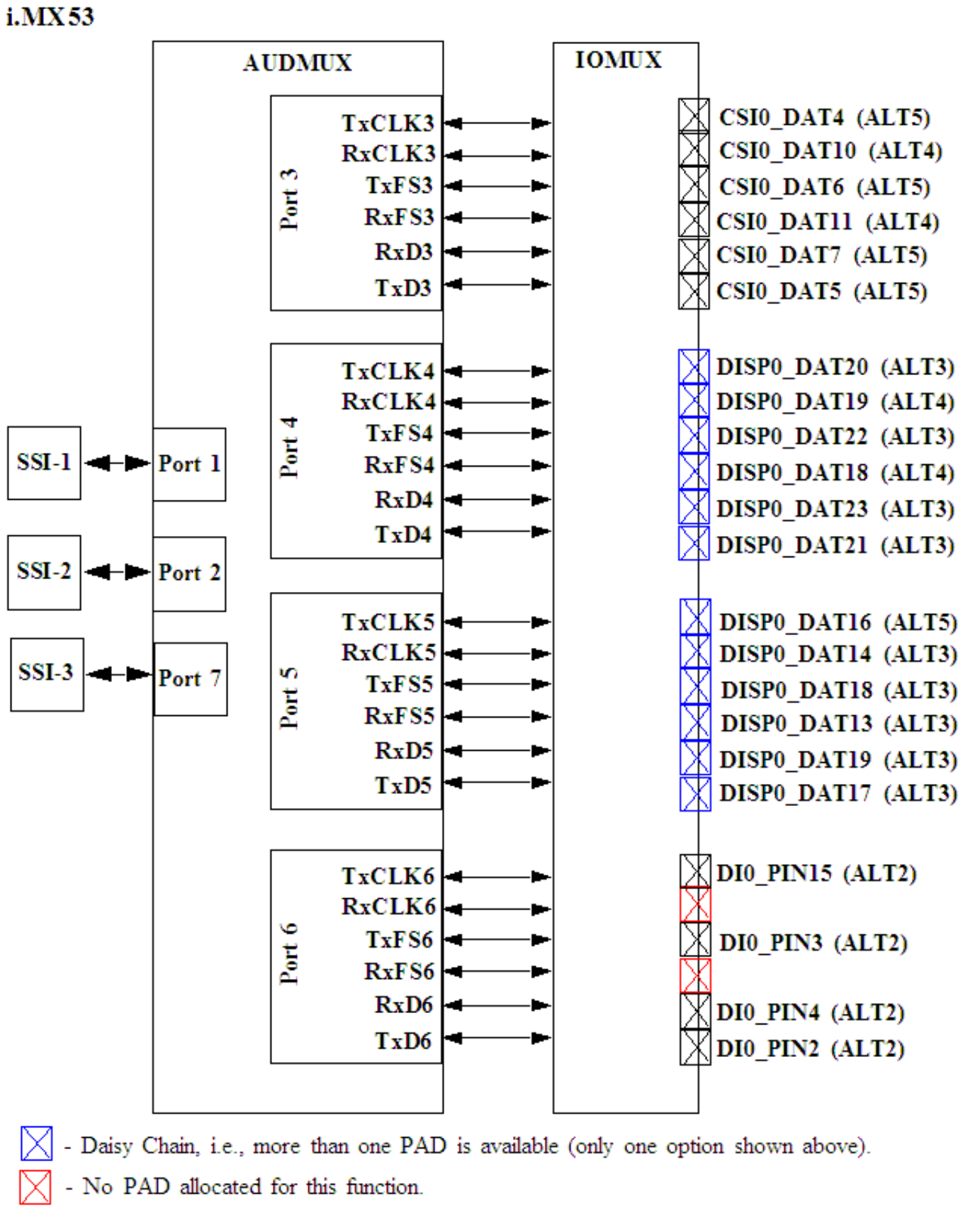


Figure 8-23. AUDMUX System Block Diagram

8.7.3 Enhanced Serial Audio Interface (ESAI)

The Enhanced Serial Audio Interface (ESAI) provides a full-duplex serial port for serial communication with a variety of serial devices, including industry-standard codecs, SPDIF transceivers, and other processors.

The ESAI consists of independent transmitter and receiver sections, each section with its own clock generator. All serial transfers are synchronized to a clock. Additional synchronization signals are used to delineate the word frames. The normal mode of operation is used to transfer data at a periodic rate, one word per period. The network mode is also intended for periodic transfers; however, it supports up to 32 words (time slots) per period. This mode can be used to build time division multiplexed (TDM) networks. In contrast, the on-demand mode is intended for non-periodic transfers of data and to transfer data serially at high speed when the data becomes available.

The ESAI has 12 pins for data and clocking connection to external devices. The ESAI is internally connected to the ESAI Bus Interface and FIFO (ESAI_BIFIFO), and does not connect directly to the shared peripheral bus. The ESAI interface is designed for a 24-bit data bus, while the shared peripheral data bus is 32-bit wide. Also, the ESAI data paths are only double buffered, not allowing efficient DMA service in the applications processor environment. The ESAI_BIFIFO allows increasing the data buffering and data width matching to the shared peripheral bus.

8.7.4 Sony/Philips Digital Interface (SPDIF)

The Sony/Philips Digital Interface (SPDIF) block is a stereo that allows the processor to transmit digital audio over it using the IEC60958 standard, consumer format. i.MX53 provides one SPDIF transmitter with one output. The SPDIF allows the handling of both SPDIF channel status (CS) and User (U) data. Although i.MX53 includes SPDIF receiver part, it is likely not be used for the intended use-cases of i.MX53.

The SPDIF transmit clock is generated by the SPDIF internal clock generator sub-block and the clock sources are from outside of the SPDIF block. The ESAI, SSI and Media Local Bus (MediaLB) Controller (MLB) clock sources should provide a clock that is at least 64 x Fs, where Fs is the sampling frequency. The external clock source should provide at least 128 x Fs. Clocks of higher frequency may be provided as long as the multiplication factor is a power of 2 (for example, 128x, 256x or 512x). Also, clock frequency precision of 100ppm or better should be provided. The SPDIF transmit clock sources should come from the ESAI HCKT block I/O, SSI-1 TXCLK signal, SSI-2

TXCLK signal and MLB clock, as well as from a dedicated audio clock source (either a quartz oscillator or an external clock input). The SPDIF transmit clock can be one of the ASRC clock output sources.

Figure 8-24 shows the clock structure of the SPDIF transceiver.

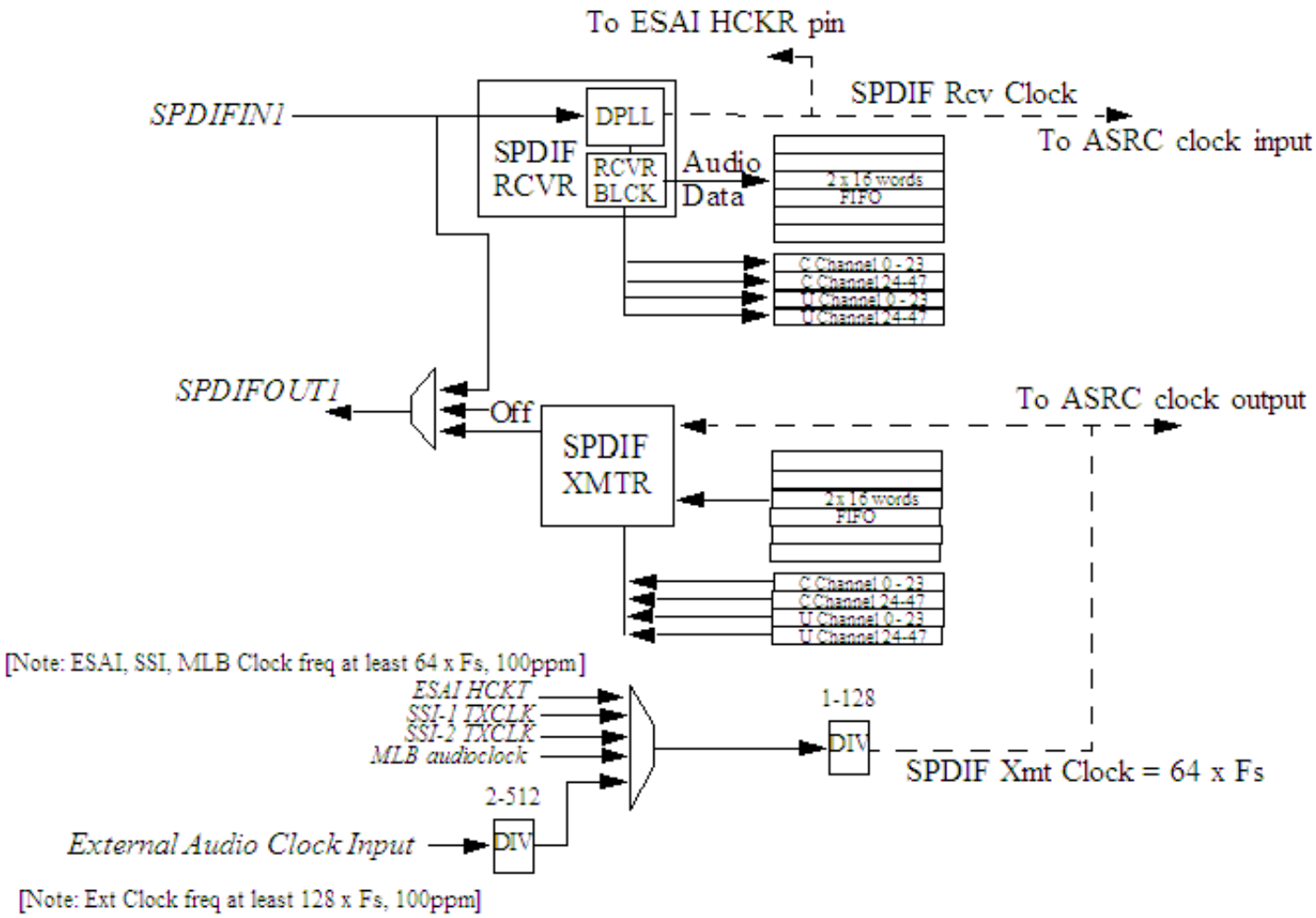


Figure 8-24. SPDIF Transceiver Clock Diagram

8.7.5 Asynchronous Sample Rate Converter (ASRC)

The incoming audio data may be received from various sources at different sampling rates. Also the outgoing audio data may have different sampling rates and it can also be associated to output clocks that are asynchronous to the input clocks.

The Asynchronous Sample Rate Converter (ASRC) converts the sampling rate of a signal associated to an input clock into a signal associated to a different output clock. The ASRC supports concurrent sample rate conversion of up to 10 channels of about -120dB THD+N. The sampling rate conversion of each channel is associated to a pair of incoming and outgoing sampling rates. The ASRC supports up to three sampling rate pairs.

In the real-time audio use case, both input/output sampling rate clocks are activated. Both sampling rate clocks are directly connected to the ASRC, and the ratio estimation of the input clocks to output clocks is used to perform the sample rate conversion in ASRC hardware.

Several audio blocks (namely SSI-1, SSI-2, SSI-3, S/PDIF and ESAI) can act as either clock master or clock slave. Multiplexers are provided on the SoC level to select which of the block's clock signals (clock in or clock out) can be connected to the ASRC input. This depends on the block's operational mode as shown in Figure 8-25 and Table 8-15. Figure 8-25 presents the multiplexer cell used for all ASRC input clocks. Table 8-15 shows the detailed clocks, multiplexer control and data bits for each ASRC input clock. For the audio block clocks that are directly connected to ASRC (without the multiplexer scheme) see Table 8-16.

In the non-real-time streaming audio use case, the input sampling rate clock does not need to be provided. Instead, the ideal-ratio value conversion is set in the ASRC interface registers. In this case only the output sampling rate clock must be provided, and the fixed ratio of the input to the output in the register is used to perform the sample rate conversion.

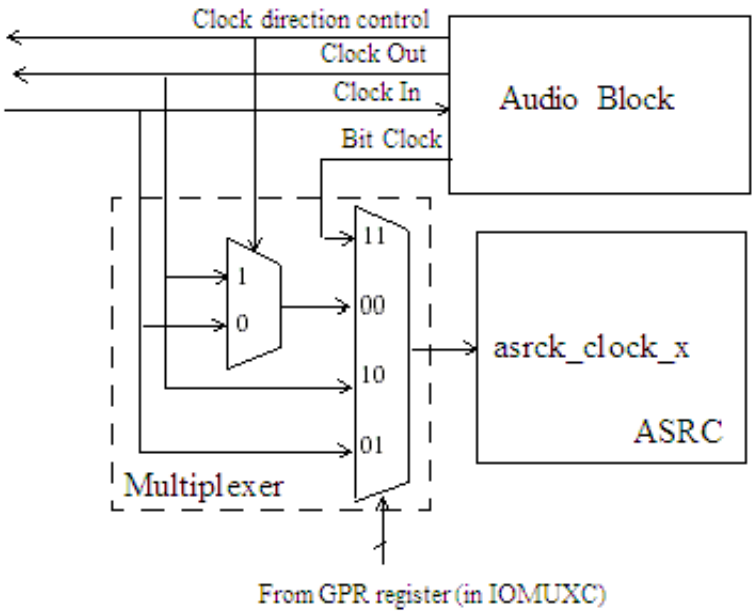


Figure 8-25. Muxing scheme template for ASRC input clocks

Table 8-15. ASRC Muxed Input Clocks

ASRC Clock Input	Audio block source	MUX2:1 Control	MUX2:1 Input 0	MUX2:1 Input 1	MUX4:1 Control	MUX4:1 Input 00	MUX4:1 Input 01	MUX4:1 Input 10	MUX4:1 Input 11
asrck_clock_1	SSI-1 (Rx)	SRCR[RXDIR][1]	External SRCK	InternalSRCK	GPR0[17:16][2]	MUX2:1 output	External SRCK	InternalSRCK	RX bit clock
asrck_clock_9	SSI-1 (Tx)	STCR[TXDIR]	External STCK	InternalSTCK	GPR0[17:16]	MUX2:1 output	External STCK	InternalSTCK	TX bit clock
asrck_clock_2	SSI-2 (Rx)	SRCR[RXDIR]	External SRCK	InternalSRCK	GPR0[19:18]	MUX2:1 output	External SRCK	InternalSRCK	RX bit clock
asrck_clock_a	SSI-2 (Tx)	STCR[TXDIR]	External STCK	InternalSTCK	GPR0[19:18]	MUX2:1 output	External STCK	InternalSTCK	TX bit clock
asrck_clock_3	SSI-3 (Rx)	SRCR[RXDIR]	External SRCK	InternalSRCK	GPR0[21:20]	MUX2:1 output	External SRCK	InternalSRCK	RX bit clock
asrck_clock_b	SSI-3 (Tx)	STCR[TXDIR]	External STCK	InternalSTCK	GPR0[21:20]	MUX2:1 output	External STCK	InternalSTCK	TX bit clock
asrck_clock_0	ESAI (Rx)	RCCR[RCKD]	External SCKR	InternalSCKR	GPR0[23:22]	MUX2:1 output	External SCKR	InternalSCKR	N/A[3]
asrck_clock_8	ESAI (Tx)	TCCR[TCKD]	External SCKT	InternalSCKT	GPR0[23:22]	MUX2:1 output	External SCKT	InternalSCKT	N/A

1. Register[bit] format, SRCR is the register name in the block specified in the "Audio block source" column, while RXDIR is name field/bit in the register.
2. GPR0 is the General Register 0 in the IOMUX control (IOMUXC). See IOMUXC chapter for more details.
3. Not in used, the clock value is "0".

Table 8-16. ASRC Direct Clocks

ASRC Clock Input	Block driving clock source	Block output clock
asrck_clock_4	SPDIF (Rx)	SPDIF Rx Clock
asrck_clock_c	SPDIF (Tx)	SPDIF Tx Clock
asrck_clock_5	MLB	MLB Input clock (from device)
asrck_clock_6	Reserved	Reserved[1]
asrck_clock_7	Reserved	Reserved
asrck_clock_d	LPCG	ASRC root clock

1. Not in used, the clock value is "0".

Chapter 9
Power Management

9.1 Overview

The i.MX53 has several voltage domains.

- Peripheral domain: The peripheral domain consists of the i.MX53 peripherals and ARM asynchronous interface. The nominal voltage is 1.3V. This domain has Dynamic Voltage and Frequency Scaling (DVFS). When the chip is in STOP mode, the voltage can be reduced to LP STOP voltage.
- ARM domain: The ARM domain consists of the ARM Cortex A8. The nominal voltage is 1V. This domain has DVFS. When the chip is in STOP, the voltage can be reduced to GP STOP voltage.
- Memory array domain: The memory array domain consists of all the memory arrays, including the L1 and L2 caches. The nominal voltage is 1.3V.
- Secure Real Time Clock (SRTC) domain: The SRTC domain consists of the SRTC. The operating voltage is 1.3V. The voltage to this domain must always remain on. This domain is not affected by STOP mode.
- PLL domain: This domain consists of 2 supplies for the PLLs of i.MX53. The first supply is for the digital domain of the DPLLs' nominal voltage, 1.3V. The second supply is for the analog domain of the PLLs and its nominal voltage is 1.8V. This domain is not affected by STOP mode.
- FUSE program domain: this domain consists of the supply needed to program the e-fuses. The nominal voltage of this domain is 3.3 V. This supply should be connected to ground once programming has been completed. This domain is not affected by STOP mode. Table 9-1 shows the power domains.

Table 9-1. Power domains

Domain name	Description	Nominal Voltage	DVFS	Comments
VCC	i.MX53 peripheral including Cortex A8 LP side	1.3V	1V	

Table continues on the next page...

Table 9-1. Power domains (continued)

Domain name	Description	Nominal Voltage	DVFS	Comments
VCCGP	Cortex A8 cmos065gp side	1V	0.8V	
VDDA/VDDAL1	memory arrays including Cortex A8 L2 memory arrays and L1 memories.	1.3V	NA	See restrictions in Dynamic Voltage and Frequency Scaling (DVFS)
SRTC_POW	SRTC	1.3V	NA	
PLL_DIG	DPLL digital side and FPM	1.3V	NA	
PLL_ANA	PLL analog side	1.8V	NA	

NOTE

* Please note that the power values defined in this table are for reference only. For the validated numbers please see i.MX53 datasheet.

9.2 Power Saving Methodology

Several power management features are supported in i.MX53 to reduce active and static power consumption.

9.2.1 Active Power Savings

The main active power savings techniques applied to i.MX53 are listed below:

- Clock gating - i.MX53 implements three levels of clock gating. These levels are:
  - Clock tree roots (in the CCM)
  - Clock tree branches (in the LPCG)
  - Clock tree leaf nodes (in the blocks)
- DVFS - Dynamic Frequency Voltage Scaling. There are 2 DVFS engines. One for the peripheral domain and one for the ARM domain.

9.2.2 Leakage Power Savings

The main leakage control mechanisms on i.MX53 are reduced voltage in STOP mode and SRPG power gating.

- SRPG - State Retention Power Gating. The state of the flip flops are preserved with a continuously powered on supply, while the combinatorial logic be

are powered down, using internal switch cells. This is available on the Cortex A8 platform. There is also a possibility to SRPG only the NEON floating point inside the Cortex A8.

The power gating sequencing is controlled by the Global Power Controller (GPC) block on i.MX53. The following power gating options exist:

- SRPG of the Cortex A8 platform in WAIT mode.
- SRPG of the Cortex A8 platform in STOP mode.
- SRPG of the NEON floating unit when it's not needed, by SW control.

9.3 Block Connectivity for Low Power Modes

Figure 9-1 shows the connectivity of the blocks for low power modes.

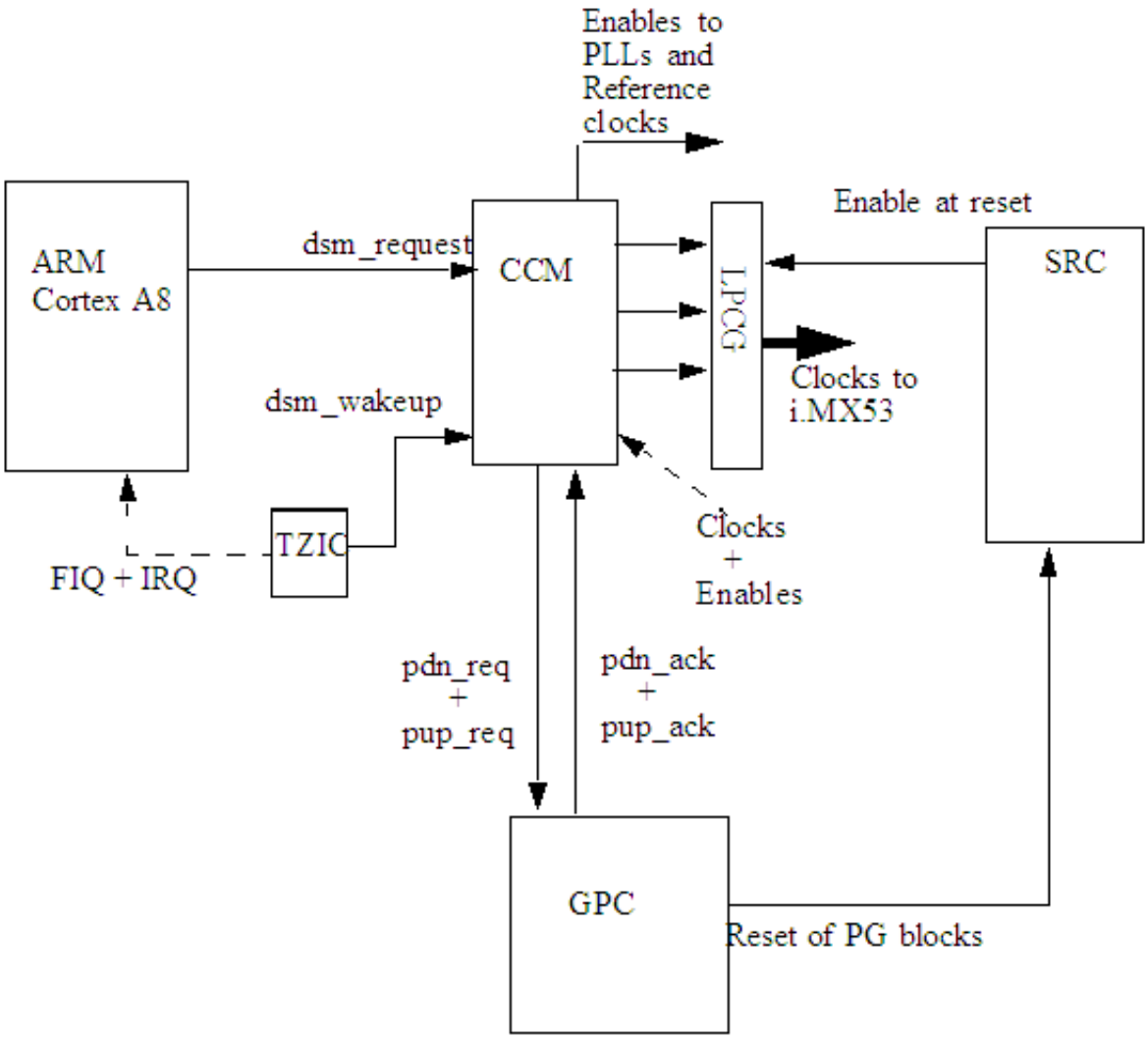


Figure 9-1. Connectivity of module in low power modes

9.4 Low Power Modes

i.MX53 can be put into different low power modes. The different power modes are:

- RUN - Core is active, clocks are on, the peripheral blocks required are active. SW can close clocks of blocks that are not in use.
- WAIT - Core is disabled and clock gated, bus clocks to peripherals can be on as required. SRPG can be applied to Cortex A8 as described on the section above.
- STOP - Core is disabled, peripherals are disabled, bus clocks are off, DPLLs are off. SRPG can be applied to Cortex A8 as described on the section above.

The different power modes of the individual blocks are:

- Active - Operational and doing work with clocks on.
- Idle - Not doing work, but clocks on.
- Disabled - Not doing work and clocks off.

The block modes can be mapped onto the domain modes as shown in Table 9-2.

Table 9-2. Low Power Modes

Mode	Core	Modules	PLL	CKIH	CKIL
RUN	Active	Active, Idle, or Disabled [1]	On	On	On
WAIT	Disabled	Active, Idle, or Disabled [1]	On	On	On
STOP	Disabled	Disabled [2]	Off	Off	On

[1] During RUN and WAIT modes, the peripherals can be active, but not all must be active. The blocks that are not required for the use case can be Idle or Disabled in the RUN and WAIT modes.

[2] Some blocks can operate from CKIL in STOP mode.

9.4.1 Low Power Mode Inputs to Blocks in i.MX53

Many of the peripherals have the following signals:

- ipg_stop - this signal indicates to the block that system is entering stop mode - will be connected to ipg_stop signal from the CCM
- ipg_wait - this signal indicates to the block that system is entering wait mode -will be connected to ipg_wait signal from CCM
- ipg_doze - legacy pin used for doze mode indication - there are a few blocks in i.MX53 that have this pin and they will be connected as follows:
  - uart: ipg_doze is connected to ipg_wait from CCM. Since uart does not have an ipg_wait signal.
  - sim: ipg_doze is connected to ipg_wait from CCM. Since sim does not have an ipg_wait signal.
  - epit: ipg_doze is grounded.
  - gpt: ipg_doze is grounded.
  - pwm: ipg_doze is grounded.
  - wdog: ipg_doze is grounded.

9.4.2 Power Down Sequence

1. SW prepares the LPM bits in the CCM see #clock_control_module_ccm as well as the ARM SRPG power gate bits in the GPC see #general_power_controller_gpc. The

information if the ARM platform can be SRPG power gated is stored in the GPC block.
2. ARM starts the LPM sequence. Resulting in the CCM raising request for power down for ARM and peripherals.
3. Power down sequence, following ARM sequence as described in the #state_retention_power_gating_controller_srpgc, PGC and EMPGC specs.
4. A combined power down acknowledge signal is issued to the CCM from the GPC block.

9.4.3 Power Up Sequence

1. An interrupt is received and as a result after returning clocks is needed then a power up request signal is issued from the CCM to the GPC.
2. The power up of the ARM.

9.5 RAM Memory Supply Connections

The memories in i.MX53 have 3 supply pins:

- VDD - The memory periphery supply.
- VDDA/VDDAL1 - The memory array supply.
- VDD_STATE - The supply to keep the value of the outputs during power down and the redundancy information for redundant memories.

Each memory periphery supply is connected to the supply of the block the memory is placed in.

All the memories' array supply (VDDA/VDDAL1) and VDD_STATE are connected to a dedicated supply pin on i.MX53 this is done due to the fact that the memories array supply does not support a voltage lower then 1.08V. So during DVFS the periphery supply is lowered along with the rest of the chip.

VDDAL1 is dedicated pin used for the memory arrays of the cortex_A8 L1 array. It can be tied to VDDA.

NOTE

The dedicated array supply needs to be lowered to support a maximal difference between VDDA/VDDAL1 and VDD of 300mV.

9.6 Dynamic Voltage and Frequency Scaling (DVFS)

i.MX53 IC has DVFS that is one of the active power saving methodologies. The i.MX53 will support the following voltages:

- Peripheral side (cmos065lp process): 1.3V to 1V.
- Cortex A8 side (cmos065gp process): 1V to 0.8V.

DVFS is supported for both the ARM and the peripherals using 2 different load monitors:

1. DVFSC - uses weighing signals relevant to the ARM platform (cmos065gp) and affecting only the ARM platform.
2. DVFSP - uses weighing signal relevant to the i.MX53 peripherals and affecting only the peripheral side.

While in DVFS each clock will be divided by 2, 3 or 4 all across the peripheral domain, depending on the definition in the CCM. For a restriction table please refer to the #clock_control_module_ccm.

Chapter 10
System Security

10.1 Introduction

Security is a common requirement for platforms built using the i.MX53, although the specific needs vary greatly depending on the platform and market. The type and cost of assets to be protected on a portable consumer device are very different from those to be protected on automotive or industrial platforms, and the same applies to the kind of attacks and level of resources threatening those assets. The platform designer must select an appropriate set of counter measures to meet the relevant platform security needs.

For the platform designer to meet the requirements for each market, the i.MX53 incorporates a range of security features which can be used individually or in concert to underpin the platform security architecture. Most of the i.MX53 security features provide protection against particular kinds of attack and can be configured at various levels according to the required degree of protection. These features are designed to work together and can be integrated with appropriate software to create defensive layers. In addition to protection features, the i.MX53 includes a general purpose accelerator to enhance the performance of selected industry standard cryptographic algorithms.

Main i.MX53 security features:

Secure High Assurance Boot

- Security library embedded in tamper-proof on-chip ROM
- Based on established cryptography: Public Key Infrastructure (PKI), SHA-256 and RSA 2048
- Run every time chip is reset
- Image Version Control (on-chip OTP-based)
- Secure signing services

HW Cryptographic Accelerators

- Symmetric: AES, DES/ 3DES, ARC4

- Asymmetric: RSA, ECC
- Message Digest and HMAC: SHA-1, SHA-224, SHA-256, MD-5

Run-time Integrity Checker and Security Controller (including Security Monitor)

- Run-time monitoring of the critical SW integrity

Secure Storage

- On-chip zeroizable Secure RAM
- Off-chip storage protected using AES-256 and chip's unique hardware-only key

True Random Number Generator Accelerator (RNGA)

Secure JTAG Controller (with electrical fuses)

- JTAG access can be permanently disabled by an on-chip OTP element
- Alternatively, JTAG access can be controlled (by secret keys)

Secure real-time clock

- On-chip, self-powered real-time clock
- Monotonic counter

Universal Unique ID

ARM TrustZone and hardware Firewall

- The Firewall controls access to all hardware resource from DMA bus masters

Physical Tamper Detection

- Clock and power supply tamper detection
- Device's cover seal break detection

The following are the i.MX53 security components.

- TrustZone (TZ) Architecture in the Cortex A8 Platform, TrustZone Aware Interrupt Controller (TZIC) and TrustZone Watchdog Timer (WDOG-2)
- High Assurance Boot (HAB) feature in the System Boot
- Security Controller (SCC) with 16 Kbyte of on-chip Secure RAM
- Symmetric/Asymmetric Hashing and Random Accelerator (SAHARA)
- Real Time Integrity Checker (RTIC)
- Secure Real Time Clock (SRTC)
- IC Identification Module (IIM) with on-chip electrical fuses
- Central Security Unit (CSU)
- Secure JTAG Controller (SJC)

- Watermark mechanism in the Multi Master Multi Memory Interface (M4IF)
- Locked mode in the Smart Direct Memory Access (SDMA) controller

Detailed descriptions of each component are found in *MCIMX53 Multimedia Applications Processor Security Reference Manual* (MCIMX53SRM).

Chapter 11
ARM Cortex A8 Platform (ARM Platform)

11.1 Introduction

This chapter discusses the architecture and design of the ARM™Cortex® A8 platform. Detailed design information of each block within the platform is not covered in this document.

11.2 Overview

A block diagram of the ARM platform is shown in Figure 11-1. The ARM Platform consists of the ARM Ltd. ARM processor which includes a NEON™ co-processor, L1 cache, L2 cache, Embedded Trace Macrocell (ETM) and a Cross Trigger Interface (CTI). The platform includes the essential sub-blocks: platform control, test logic and debug (Cross trigger Matrix (CTM), Embedded Trace Buffer (ETB), and a second CTI).

The ARM processor instruction and data read/write AXI master port is connected from the non-ARM Platform side of the Level 2 Cache. The L2 can access external L3 memory through this port.

The core platform supports static debug through the debug logic to SoC. This includes the capability of real time trace through ARM's Coresight ETM, ETB and CTM sub-blocks. The second CTI sub-block allows cross triggering of internal and external trigger sources.

The ARM platform has two power domains (LP & GP) which are separated by level shifters. The LP power domain serves as a interface to the rest of the SoC. The GP power domain is completely contained within the platform and allows the ARM core and subsystem to run at much higher frequencies than the rest of the SoC.

The boundary of the two power domains is also an asynchronous boundary between the ARM platform and the rest of the SoC. Synchronizers in both the GP and LP power domains of the platform allow the ARM core and subsystem to run asynchronously from the rest of the SoC.

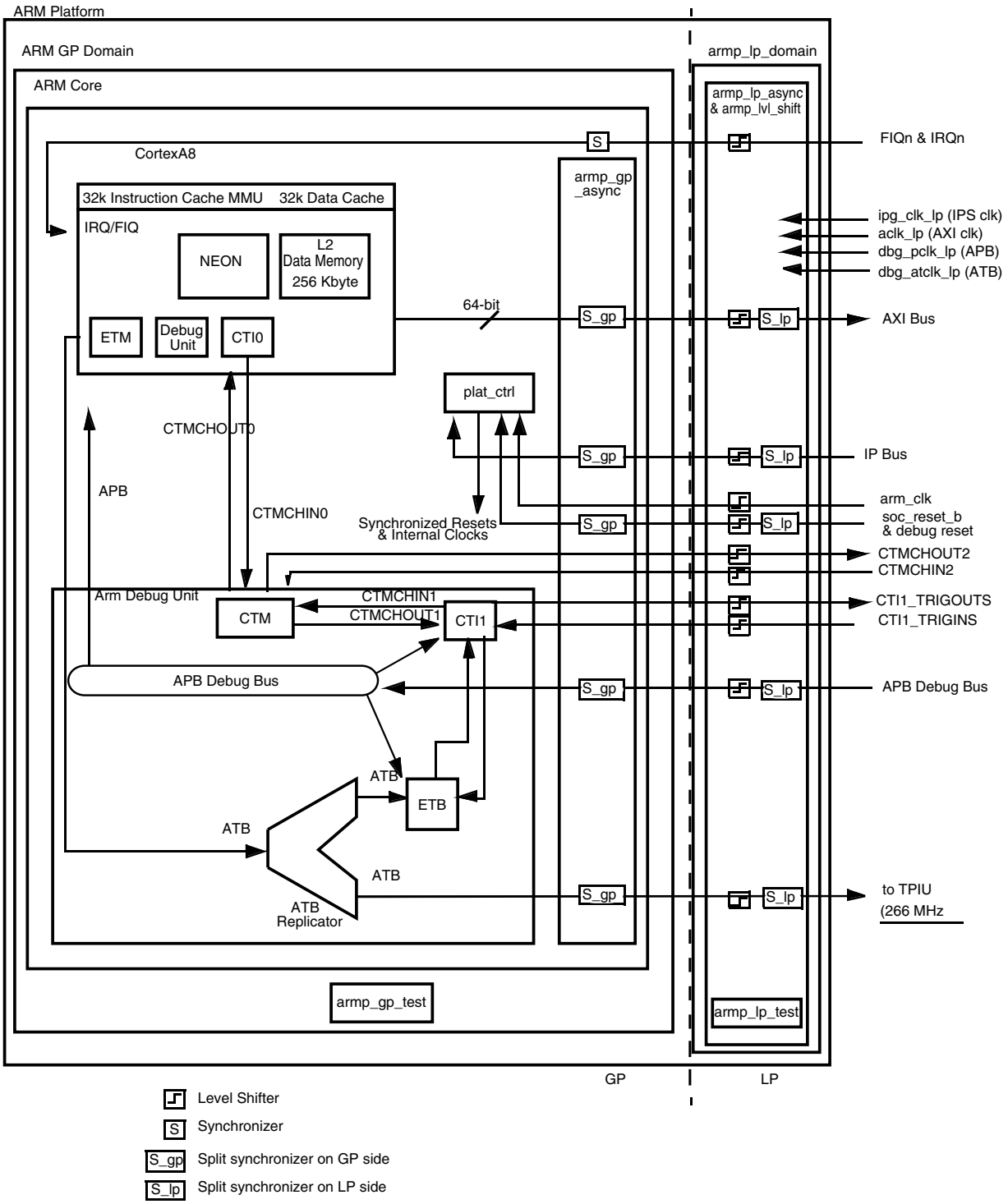


Figure 11-1. ARM platform Block Diagram

11.2.1 Core Platform Sub-Blocks

11.2.1.1 ARM platform

The information presented in this section focuses on design aspects of the ARM platform in the ARM platform subsystem. The ARM platform is ARM's first superscalar processor featuring technology for enhanced code density and performance, NEON™technology for multimedia and signal processing, and Jazelle®RCT (Runtime Compilation Target) technology for efficient support of ahead-of-time and just-in-time compilation of Java and other bytecode languages.

The ARM platform incorporates an integer core that implements the ARMv7-A architecture instruction set. It supports the ARMv7 and Thumb® -2 instruction sets. A NEON sub-block is included to accelerate the performance of multimedia applications. Included Vector Floating Point v3 architecture is IEEE 754 compliant. The processor includes an AMBA® 3 AXI high-performance 64-bit SoC interconnect.

The following are the features of ARM platform:

- The ARM platform's sophisticated pipeline architecture is based on dual, symmetric, in-order issue, 13-stage pipelines with advanced dynamic branch prediction achieving 2.0 DMIPS/MHz. The instruction execute unit consists of two symmetric Arithmetic Logical Unit (ALU) pipelines and the multiply pipeline.
  - In-order, dual-issue, superscalar microprocessor core
  - 13-stage main integer pipeline
  - 10-stage NEON media pipeline for executing NEON and VFP instruction sets
  - Dedicated L2 cache with programmable wait states
  - Global history based branch prediction
- Works in conjunction with a power optimized load store pipeline to deliver 2.0 DMIPS/MHz for power sensitive applications
- ARMv7 architecture compliant including:
  - Thumb-2 technology for greater performance, energy efficiency, and code density
  - NEON signal processing extensions to accelerate media codecs such as H.264 and MP3
  - Jazelle RCT Java-acceleration technology to optimize Just In Time (JIT) and Dynamic Adaptive Compilation (DAC), and reduce memory footprint by up to three times
- Integrated Level 2 Cache
  - Built using standard compiled LP RAMs

    - Sized at 256 Kb
    - Programmable delay
- Optimized Level 1 Caches
    - Customized design for performance and power optimization
    - Sized at 32 Kb Instruction and 32 Kb Data
    - Combine minimal access latency with hash way determination to maximize performance and minimize power consumption.
- Dynamic Branch Prediction
    - Enabled by branch target and global history buffers
    - Achieves 95% accuracy across industry benchmarks.
    - Replay mechanism minimizes miss-predict penalty
- Memory System
    - Single-cycle load-use penalty for access to the L1 cache
    - Hash array in the L1 cache limits activation of the memories to when they are likely to be needed.
    - Direct interface between the integrated, configurable L2 cache and the NEON media unit for data streaming
    - Banked L2 cache design that enables only one bank at a time
- Memory Management Unit (MMU) and separate instruction and data Translation Look-aside Buffers (TLBs) of 32 entries each
- Embedded Trace Macrocell (ETM) support for nonintrusive debug
- ARMv7 debug with watchpoint and breakpoint registers and a 32-bit Advanced Peripheral Bus (APB) interface to the CoreSight debug system.

11.2.1.2 Instruction Fetch

The instruction fetch unit predicts the instruction stream, fetches instructions from the L1 instruction cache, and places the fetched instructions into a buffer for consumption by the decode pipeline. The instruction fetch unit also includes the L1 instruction cache.

11.2.1.3 Instruction Decode

The instruction decode unit decodes and sequences all ARM and Thumb-2 instructions including the debug control coprocessor, CP14, and the system control coprocessor, CP15 instructions.

The instruction decode unit handles the sequencing of:

- Exceptions
- Debug Events
- Reset Initialization

- Memory Built-In Self Test (MBIST) for L1 cache
- Wait-for-interrupt
- Other Unusual Events
- Instruction Cycle Timing

11.2.1.4 Instruction Execute

The instruction execute unit consists of two symmetric Arithmetic Logical Unit (ALU) pipelines and the multiply pipeline. The execute pipelines also perform register write back.

The instruction execute unit:

- executes all integer ALU and multiply operations including flag generation
- generates the virtual addresses for loads and stores and the base write-back value, when required
- supplies formatted data for stores and also forwards data and flags
- processes branches and other changes of instruction stream and evaluates instruction condition codes.

11.2.1.5 Load/Store

The load/store unit encompasses the entire L1 data side memory system and the integer load/store pipeline. This includes the:

- L1 data cache.
- data side TLB.
- integer store buffer.
- NEON store buffer.
- integer load data alignment and formatting.
- integer store data alignment and formatting.

The pipeline accepts one load or store per cycle that can be present in either pipeline 0 or pipeline 1. This gives the processor flexibility when scheduling load and store instructions.

11.2.1.6 L2 Cache

The L2 cache unit includes the L2 cache and the Buffer Interface Unit (BIU). It services L1 cache misses from both the instruction fetch unit and the load/store unit.

11.2.1.7 NEON

The NEON unit includes the full 10-stage NEON pipeline that decodes and executes the NEON media instruction set. The NEON unit includes:

- the NEON instruction queue
- the NEON load data queue
- two pipelines of NEON decode logic
- three execution pipelines for NEON integer instructions
- two execution pipelines for NEON floating-point instructions
- one execution pipeline for NEON and VFP load/store instructions
- the VFP engine for full execution of the VFPv3 data-processing instruction set.

11.2.1.8 Processor Debug Unit

The processor debug unit assists in debugging software running on the processor. In combination with a software debugger program, the debug unit enables debugging the following:

- application software
- operating systems
- hardware systems based on an ARM processor

The debug unit enables:

- stopping program execution
- examining and altering processor and coprocessor states
- examining and altering memory and input/output peripheral states
- restarting the processor core

11.2.1.9 Embedded Trace Marcocell (ETM)

The ETMv3.3 unit is a nonintrusive trace macrocell that filters and compresses an instruction and data trace for use in system debugging and system profiling. The ETM unit has an external interface outside of the processor called the Advanced Trace Bus (ATB) interface.

The ARM platform ETM provides real time instruction trace for the ARM platform. It is designed to be used with the CoreSight Design Kit. Unlike previous versions of the ARM platforms, this ETM is embedded inside the Processor Core.

Real time tracing is controlled by specifying a set of filtering and triggering resources which include address and data comparators, counters and sequencers. Though data trace can not be enabled, the ETM can still trigger based on data values. Also, the ETM can trace data address values.

The ARM platform ETM contains the following main components:

- Core interface
- Trace generation
- Filtering and triggering resources
- Main FIFO
- AMBA 3 ATB interface
- AMBA 3 APB interface

11.2.1.10 Cross Trigger Interface 0 (CTI0)

This block controls the Trigger Interface (TI). The CTI combines and maps the trigger requests, and broadcasts them to all other interfaces on the embedded cross trigger (ECT) as channel events. When the CTI receives a channel event it maps this onto a trigger output. This enables subsystems to cross trigger with each other. The receiving and transmitting of triggers is performed through the TI.

Each CTI has 8 trigger inputs, 8 trigger outputs and 4 sets of 4 channel I/O(s).

11.2.2 Summary of Remaining Platform Components

11.2.2.1 Platform Controller

The Platform Control contains all the miscellaneous logic required for the platform. The main purpose of the Platform Control Module within the ARM platform is to implement a set of control and/or status registers and associated logic for the management of certain platform functions such as internally generated clocks, debug enable/status, low power control, platform version ID, etc.

The Platform Control Module provides these main features:

- Internal clock generation with the ability to programatically set the frequency (divide-by) for each clock independently.
- Ability to force (down-counter) preload for any one or more clocks at anytime independent of preload set.
- Provides a gated clock for the IP Bus asynchronous bridge for power savings

- Provides 16 bits of General Purpose register bits for routing out to Platform boundary.
- Provides 8 bits of Platform Internal Control register bits for external Platform (SoC) use.
- Provides a set of Low Power control and status bits.
- Control bit to enable Debug along with Debug active status bit.
- NEON activity monitor which can be used in determining when to disable NEON thus saving power.

11.2.2.2 Debug Sub-Blocks

The ARM platform debug blocks are part of the overall Coresight debug system which include the ETB, CTM, CTI1, ATB replicator and APB address decode. It is expected that a DAP block and one or more additional CTI will be included at the SoC level. This section gives a brief overview of the sub-blocks that are implemented within the ARM platform. For details of the full Coresight debug subsystem, please refer to the SoC debug section.

11.2.2.2.1 Embedded Trace Buffer (ETB)

The ETB provides on-chip storage of trace data using 32-bit RAM. The ETB accepts trace data from the ARM ETM through an ATB port (passing through a replicator in between). Providing an on-chip buffer alleviates the pin count, bandwidth, and pad design requirements associated with sending trace data to a debugger directly through package pins in a real-time fashion.

Features:

- 4Kbyte compiled memory for the trace buffer and optionally can be used as a general purpose memory
- Only 32-bit accesses to the 4Kbyte buffer is supported
- AMBA Peripheral Bus programming interface for configuration and memory access

11.2.2.2.2 AMBA Trace Bus (ATB) Replicator

The ATB Replicator enables two trace sinks (ETB and an off platform port generally connected to a Trace Port Interface Unit - TPIU) to be wired together and receive ATB trace data from the same trace source (ETM). There are no programable registers. It takes incoming trace data from a single source (ETM) and replicates it as multiple masters.

The CSREPLICATOR is part of the armp_gp_platform block. Two output master ports are required for this design - one for the on-platform ETB and one for the off-platform TPIU. Placing the TPIU off platform allows future debug trace sources to connect to the TPIU via a FUNNEL, without the need to modify the ARM platform.

11.2.2.2.3 Cross Trigger Interface 1 (CTI1)

This block controls the Trigger Interface (TI). The CTI combines and maps the trigger requests, and broadcasts them to all other interfaces on the ECT as channel events. When the CTI receives a channel event it maps this onto a trigger output. This enables subsystems to cross trigger with each other. The receiving and transmitting of triggers is performed through the TI.

Each CTI has 8 trigger inputs, 8 trigger outputs and 4 sets of 4 channel I/O(s).

11.2.2.2.4 Cross Trigger Matrix - CTM

This block controls the distribution of channel events. It provides channel interfaces to the CTIs. The CTM can also connect to another CTM via a channel interface. This allows multiple CTMs to be connected.

The ARM platform has one CTM inside the platform to handle events between the platform's CTI, the processor's CTI and the rest of the ECT system outside the platform. Placing a CoreSight cross trigger matrix (CSCTM) on platform minimizes the event cycle time between the ETB and the ETM. The handshaking between the CTIs on-platform and the on-platform CTM are not enabled. This requires the ARM's CTI, the on-platform CTI1 and the on-platform CTM to all be in the same clock domain. The on-platform CTM connects to the ECT system via an off-platform CTM and the respective channels require synchronizing and handshaking enabled.

11.2.2.2.5 Advanced Peripheral Bus - APB Debug Bus

The APB originates off platform generally from a Debug Access Port (DAP) block. This bus allows access to the registers in all of the debug blocks which have addressable registers.

11.2.2.3 Asynchronous Wrapper

The ARM platform hard macro contains synchronizers for most signals into and out of the platform where synchronization is required. This means the ARM platform can function asynchronously from the rest of the SoC.

11.2.3 Configuration

There are several configuration options associated with the ARM platform. These are determined at the platform level and selected as follows:

- The L1 cache size is 32 Kb instruction and 32 Kb data
- There is parity on the L1 cache
- There is no error correction on the L1 cache
- The L2 cache size is 256 Kb
- There is no parity on the L2 cache
- There is no error correction on the L2 cache
- There are 2 L2 tag banks and 4 data banks per tag bank
- The AXI data bus width out of the platform is 64-bit

11.2.4 Endian Modes

The ARM platform supports Little Endian mode only.

11.2.5 Bus Interfaces

This section will discuss the major bus interfaces of the ARM platform. The ARM platform has the following bus interfaces:

- AMBA AXI interface
- APB CoreSight interface
- ATB CoreSight interface
- Peripheral Interface (IP Bus)

11.2.5.1 AMBA AXI Interface

The AXI bus interface is the main interface to the system bus. It performs L2 cache fills and noncacheable accesses for both instructions and data. The AXI interface utilizes a 64-bit wide input and output data bus. It also supports multiple outstanding requests on the AXI bus. The AXI bus is allowed to be asynchronous between the platform and SoC domains. This provides maximum flexibility at the SoC level.

11.2.5.2 APB CoreSight Interface

The APB is an AMBA bus used for debugging. The CoreSight interface is the ARM architecture for multi-processor trace and debug. It defines what debug and trace components are required and how they are connected. The platform Debug APB should be connected to the SoC DAP APB mux. The APB is allowed to be asynchronous between the platform and SoC domains. This provides maximum flexibility at the SoC level.

11.2.5.3 ATB CoreSight Interface

The ATB is a trace output bus used for debugging. The CoreSight components are programmed with the DAP using the APB programming bus. Trace is output over the ATB trace bus. The ATB is allowed to be asynchronous between the platform and SoC domains. This provides maximum flexibility at the SoC level.

11.2.5.4 Peripheral Interface (IP Bus)

The IP Bus interface to the platform connects the on-chip registers to the memory map and internal buses at the SoC level. Location (addresses of the registers) is determined at the SoC level. The platform acts as a slave to an SoC based IP Bus controller.

11.3 Memory Map and Register Definition

The memory map for ARM platform specific registers is shown below. This does not include coprocessor registers contained in the ARM platform or the ARM Coresight Debug. For documentation of registers contained in those blocks, please refer to the appropriate technical reference manual.

This register block address space uses five address lines which are fully decoded. The space includes nine registers and 23 unimplemented locations. The unimplemented locations will return an error if they are accessed. Depending on how the SoC decodes the register block, the entire 32-word register block could alias in the larger memory space.

ARM memory map

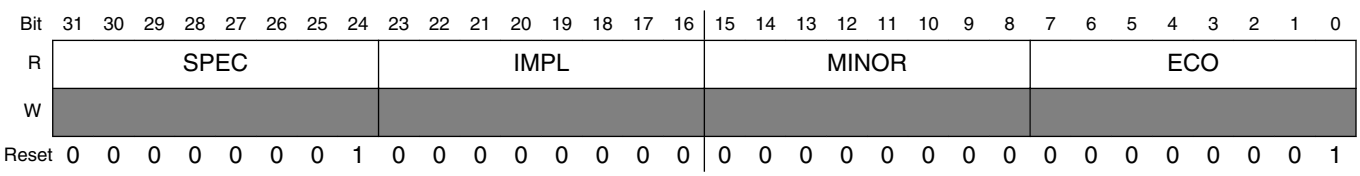
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FA_0000	Platform Version ID (ARM_PVID)	32	R	0100_0001h	11.3.1/ 623
63FA_0004	General Purpose Control (ARM_GPC)	32	R/W	0004_FF00h	11.3.2/ 624
63FA_000C	Low Power Control (ARM_LPC)	32	R/W	0000_0000h	11.3.3/ 625
63FA_0010	NEON Low Power Control (ARM_NLPC)	32	R/W	0000_0000h	11.3.4/ 626
63FA_0014	Internal Clock Generation Control (ARM_ICGC)	32	R/W	0000_7777h	11.3.5/ 627
63FA_0018	ARM Memory Configuration (ARM_AMC)	32	R/W	0000_0003h	11.3.6/ 629
63FA_0020	NEON Monitor Control (ARM_NMC)	32	R/W	000F_F000h	11.3.7/ 630
63FA_0024	NEON Monitor Status (ARM_NMS)	32	w1c	0000_0000h	11.3.8/ 631

11.3.1 Platform Version ID (ARM_PVID)

The platform version ID register (ARM_PVID) contains a 32-bit version ID which can be used to link the silicon to a specific version of the platform database. The version ID should match our release label applied to the platform database.

This register can be accessed by a 32-bit secure/non-secure, user/supervisor, read transaction. Writes return an error.

Address: ARM_PVID is 63FA_0000h base + 0h offset = 63FA_0000h



ARM_PVID field descriptions

Field	Description
31–24 SPEC	Major architectural or significant spec changes
23–16 IMPL	Implementation changes
15–8 MINOR	Minor changes (bug fixes, I/O changes)

Table continues on the next page...

ARM_PVID field descriptions (continued)

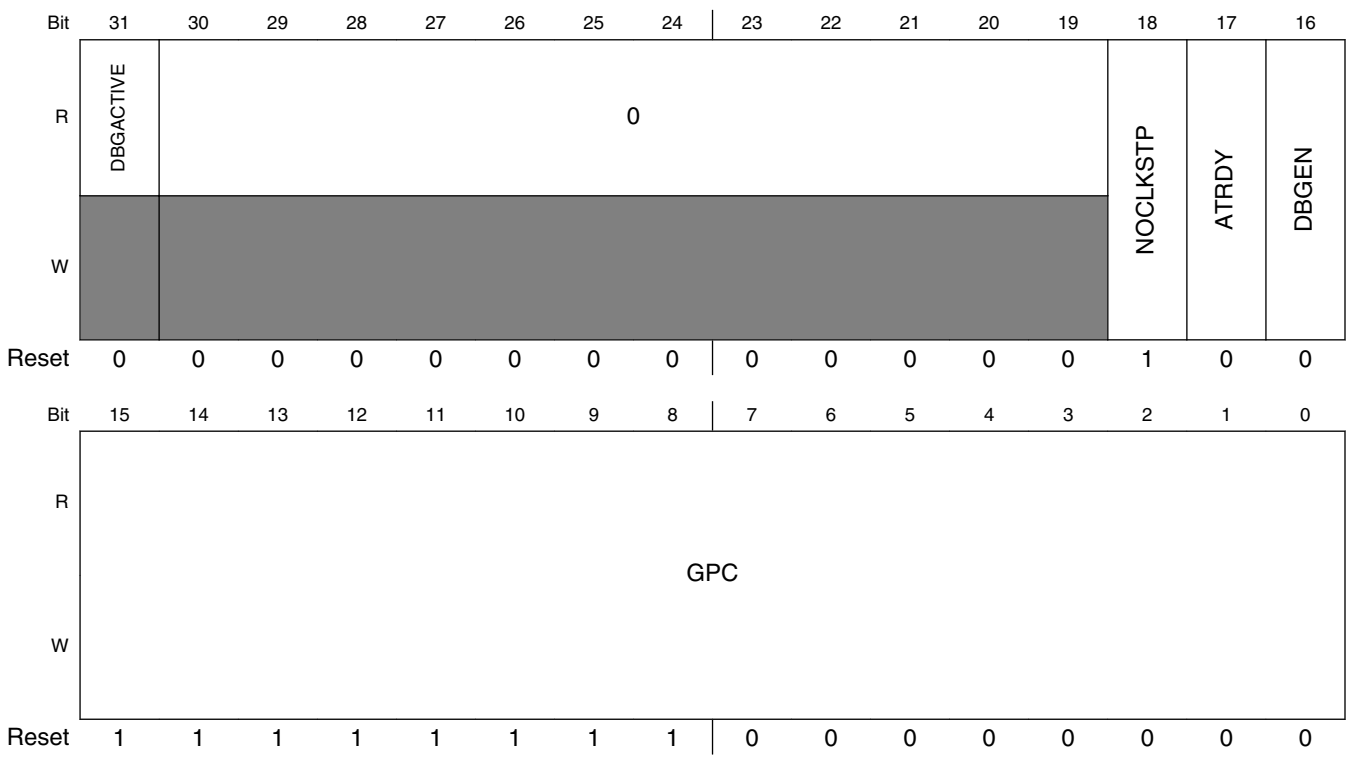
Field	Description
7–0 ECO	ECO changes

11.3.2 General Purpose Control (ARM_GPC)

This register contains 16-bits of general purpose control which drive core platform outputs that can be used within the SoC for general purpose control. This register also contains the DBG_ACTIVE status bit and DBG_EN control bit for entering Debug mode via a software access to this register.

This register can only be accessed by 32-bit secure supervisor transactions.

Address: ARM_GPC is 63FA_0000h base + 4h offset = 63FA_0004h



ARM_GPC field descriptions

Field	Description
31 DBGACTIVE	This bit indicates the status of debug. This allows the user to determine if debug has been enabled either from off platform, via the external DAP_SYS JTAG interface (dbgen_in platform input), or via software accesses to the DBGEN bit in this register.

Table continues on the next page...

ARM_GPC field descriptions (continued)

Field	Description
	0 debug is not enabled. Debug clocks are off and debug registers are inaccessible. 1 debug is enabled. Debug clocks are on and the debug system is ready to be used.
30–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18 NOCLKSTP	This bit is used to control clock-gating within the CORTEX-A8n. Note that this is distinctly different than the higher-level (platform-level) clock-gating/low-power control afforded by the other Plat_Ctrl low-power functionality (dsm_request along with the DSM and DBG_DSM control bits within the Low Power Control register). 1 All normal clock-gating activity (WFI, NEON, and so on) will be overridden and no clock gating will occur within the CORTEX-A8n. NOCLKSTP, however, has no affect on the higher-level platform clock-gating, and the low-power functions discussed in the LPC register will work as architected. 0 clock-gating within the CORTEX-A8n will be enabled. This is the recommended setting in order to reduce run mode power consumed by the clocking network within the Integer core as well as the Neon co-processor.
17 ATRDY	1 this bit disables the platform boundary ATB interface. In most SoCs this will be connected to a TPIU. This prevents traces to the ETB from stalling due to the ASYNCATB FIFO overflowing. 0 this bit allows the platform boundary ATB interface to be active. In this case, traces to the ETB could stall due to ASYNCATB FIFO becoming full. This may cause the ETM data loss.
16 DBGEN	Debug enable. This allows the user to manually activate clocks within the debug system. This register bit directly controls the platform's dbgen_out output signal which connects to the DAP_SYS to enable all debug clocks. Once enabled, the clocks cannot be disabled except by asserting the disable_trace input of the DAP_SYS.
15–0 GPC	General Purpose Control bits directly control the general_purpose_outs[15:0] output ports of the ARM Platform. These bits will be connected at the SoC level and therefore will control functions decided by the SoC architects, and must be documented by the SoC team.

11.3.3 Low Power Control (ARM_LPC)

This register is used to control entry to low power mode and can only be accessed by 32-bit secure supervisor transactions.

There exists a platform output, dsm_request, that when asserted, requests the SoC to turn off clocks to the platform, and optionally deactivate one or more power supplies. DSM can only be entered after the ARM Platform has executed a WFI instruction, completed all outstanding bus transactions, and all activity at the L2 level has completed. Assertion of either irq_b, fiq_b, or dbg_edbgrq platform inputs will cause the ARM platform to complete the WFI instruction and will result in the negation of dsm_request.

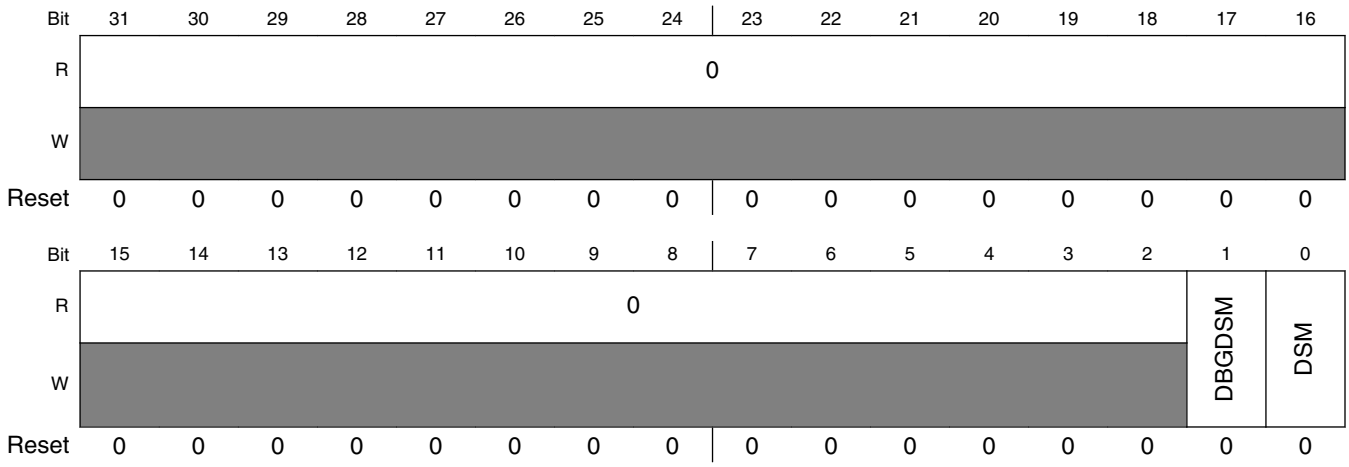
The DSM bit configures the platform to either: block deep sleep mode entry (default), or allow DSM entry when requested.

The DBG_DSM bit configures the platform to block deep sleep mode entry when debug mode is enabled (dbgen_in platform input is asserted high). This will allow the user to ensure deep sleep mode is not entered when debug is enabled.

The NEON_RST bit is used for placing the ARM NEON processor into, or releasing it from, reset.

This register can only be accessed by 32-bit secure supervisor transactions.

Address: ARM_LPC is 63FA_0000h base + Ch offset = 63FA_000Ch



ARM_LPC field descriptions

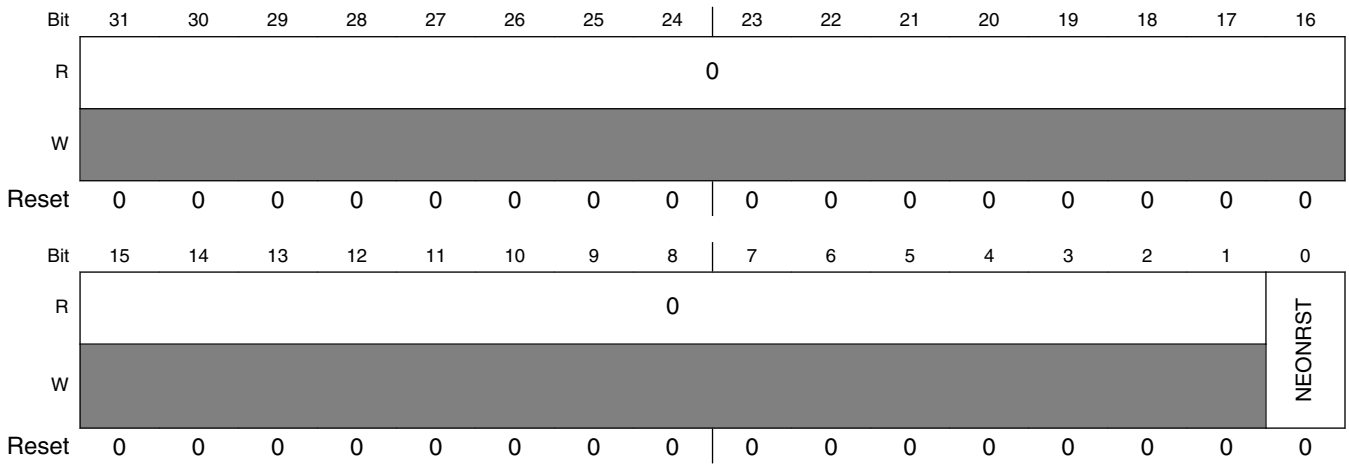
Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1 DBGDSM	Debug Deep Sleep Mode Enable 1 DSM can be entered (dsm_request will not be blocked) regardless if debug is enabled or not. 0 DSM can NOT be entered (dsm_request will be blocked) if debug is enabled
0 DSM	Deep Sleep Mode Enable is used to gate the platform's dsm_request signal, preventing the platform from issuing a deep sleep mode request. 1 DSM can be entered (dsm_request will not be blocked). 0 DSM can NOT be entered (dsm_request will be blocked).

11.3.4 NEON Low Power Control (ARM_NLPC)

This register is used to place the ARM NEON processor into, or releasing it from, reset. Refer to the table below for a description of this control bit.

This register can only be accessed by 32-bit secure supervisor transactions.

Address: ARM_NLPC is 63FA_0000h base + 10h offset = 63FA_0010h



ARM_NLPC field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 NEONRST	Hold the ARM NEON in its reset state in preparation for a low power mode. 1 ARESETNEONn input to the ARM is driven low - placing NEON into reset. 0 ARESETNEONn input to the ARM is driven high - releasing NEON from reset.

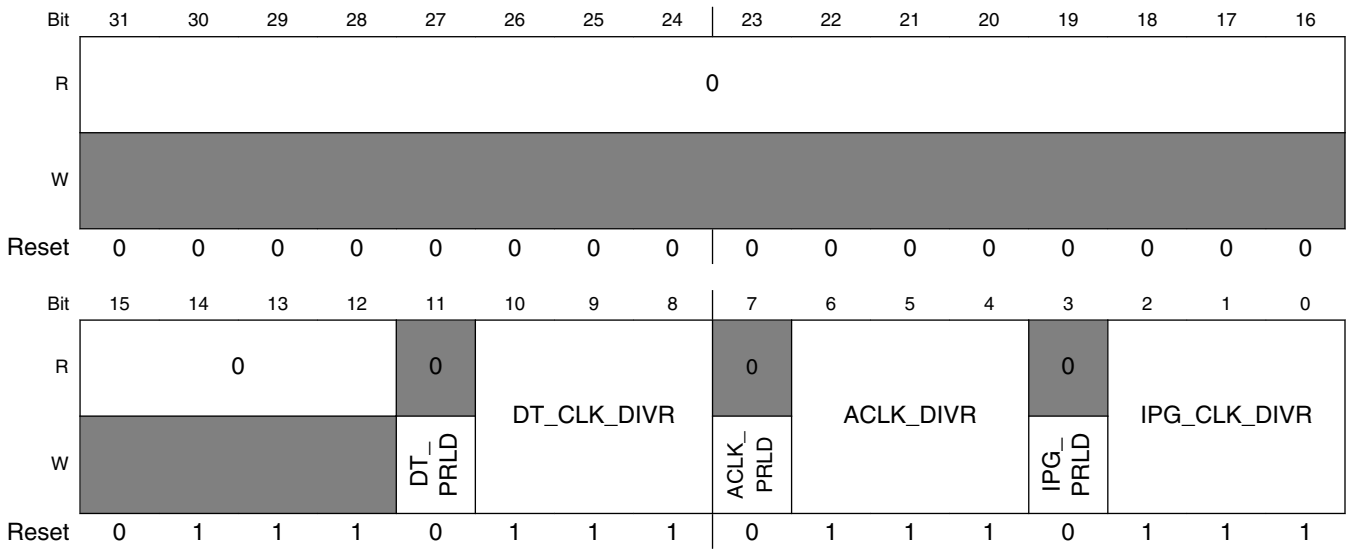
11.3.5 Internal Clock Generation Control (ARM_ICGC)

This register is used to control the clock generation circuitry for all derived clocks used within the ARM platform (ipg_clk, aclk, dbg_atclk, and dbg_pclk). The dbg_pclk clock is generated as a straight divide-by-2 of dbg_atclk.

Note: At reset, all generated clocks are edge aligned and are generated at 8:1. dbg_pclk will require 1 dbg_atclk clock cycle before it begins clocking as a divided dbg_atclk.

This register can only be accessed by 32-bit secure supervisor transactions.

Address: ARM_ICGC is 63FA_0000h base + 14h offset = 63FA_0014h



ARM_ICGC field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11 DT_PRLD	Debug AMBA Trace Bus Clock Down Counter Preload Reads always return 0 0 has no effect. If preload is 0 during a write of a new divide ratio, the down counter will not be loaded with the new value until it reaches 0 (rolls over). 1 A write of 1 will force the DBG_ATCLK down counter to preload on the next arm_clk.
10–8 DT_CLK_DIVR	Debug AMBA Trace Bus Clock Divide Ratio These bits control the clock divide ratio for the debug AMBA trace bus (ATB) and GP async interface clocks. The ATB bus is used to transfer trace messages from the ETM to the ETB or TPIU. When ETM trace data is being transferred over the ATB to the embedded trace buffer (ETB), the ATB is capable of running up to half the ARM Platform frequency. However, when ETM trace data is being transferred to the TPIU to be sent out the trace port, the ATB should be slowed to some appropriate SoC frequency. DT_CLK_DIVR[2:0] = 0b000 results in a 1:1 clock ratio DT_CLK_DIVR[2:0] = 0b001 results in a 2:1 clock ratio DT_CLK_DIVR[2:0] = 0b010 results in a 3:1 clock ratio DT_CLK_DIVR[2:0] = 0b011 results in a 4:1 clock ratio DT_CLK_DIVR[2:0] = 0b100 results in a 5:1 clock ratio DT_CLK_DIVR[2:0] = 0b101 results in a 6:1 clock ratio DT_CLK_DIVR[2:0] = 0b110 results in a 7:1 clock ratio DT_CLK_DIVR[2:0] = 0b111 results in a 8:1 clock ratio
7 ACLK_PRLD	AXI Master Port Clock Down Counter Preload Reads always return 0

Table continues on the next page...

ARM_ICGC field descriptions (continued)

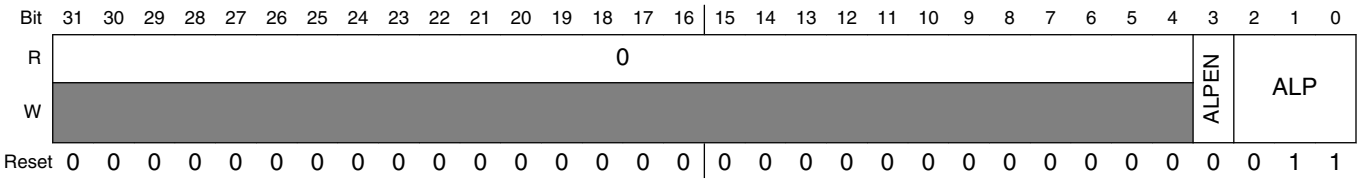
Field	Description
	0 has no effect. If preload is 0 during a write of a new divide ratio, the down counter will not be loaded with the new value until it reaches 0 (rolls over). 1 will force the ACLK down counter to preload on the next arm_clk
6–4 ACLK_DIVR	AXI Master Port Clock Divide Ratio Controls the clock divide ratio for the ARM Platform AXI bus and GP AXI async interface clocks and clock enable. The ratio of clock cycles generated for arm_clk to aclk is: ACLK[2:0] +1 : 1 ACLK_DIVR[2:0] = 0b000 results in a 1:1 clock ratio ACLK_DIVR[2:0] = 0b001 results in a 2:1 clock ratio ACLK_DIVR[2:0] = 0b010 results in a 3:1 clock ratio ACLK_DIVR[2:0] = 0b011 results in a 4:1 clock ratio ACLK_DIVR[2:0] = 0b100 results in a 5:1 clock ratio ACLK_DIVR[2:0] = 0b101 results in a 6:1 clock ratio ACLK_DIVR[2:0] = 0b110 results in a 7:1 clock ratio ACLK_DIVR[2:0] = 0b111 results in a 8:1 clock ratio
3 IPG_PRLD	Platform IP Bus Port Clock Down Counter Preload Reads always return 0 0 has no effect. If preload is 0 during a write of a new divide ratio, the down counter will not be loaded with the new value until it reaches 0 (rolls over). 1 will force the ACLK down counter to preload on the next arm_clk
2–0 IPG_CLK_DIVR	Platform IP Bus Port Clock Divide Ratio Controls the clock divide ratio for the Platform Controller and GP async IP Bus clock and clock enable. The ratio of clock cycles generated for arm_clk to ipg_clk is: IPG_CLK[2:0] +1 : 1 IPG_CLK_DIVR[2:0] = 0b000 results in a 1:1 clock ratio IPG_CLK_DIVR[2:0] = 0b001 results in a 2:1 clock ratio IPG_CLK_DIVR[2:0] = 0b010 results in a 3:1 clock ratio IPG_CLK_DIVR[2:0] = 0b011 results in a 4:1 clock ratio IPG_CLK_DIVR[2:0] = 0b100 results in a 5:1 clock ratio IPG_CLK_DIVR[2:0] = 0b101 results in a 6:1 clock ratio IPG_CLK_DIVR[2:0] = 0b110 results in a 7:1 clock ratio IPG_CLK_DIVR[2:0] = 0b111 results in a 8:1 clock ratio

11.3.6 ARM Memory Configuration (ARM_AMC)

This register is used to configure the ALP inputs to the ARM platform memories. These bits are used to set the leakage configuration bits on the memories.

This register can only be accessed by 32-bit secure supervisor transactions.

Address: ARM_AMC is 63FA_0000h base + 18h offset = 63FA_0018h



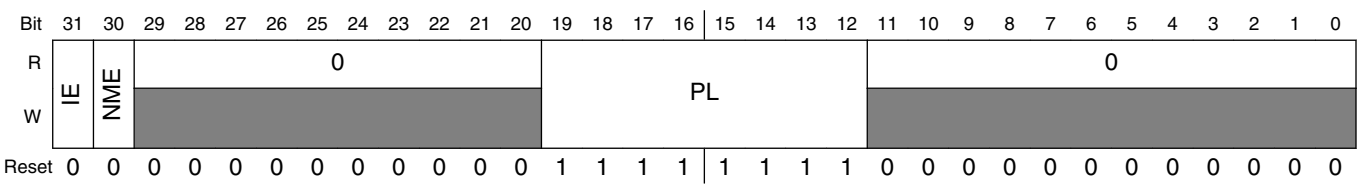
ARM_AMC field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value zero. Reserved
3 ALPEN	ALPEN - ALP Enable is used to activate the ALP bits in this register to override the default memory leakage configuration setting. 1 ALP bits of the memory are over-written with the ALP bits of this register. 0 ALP bits of the memory are driven to the default (reset) value of 3'b000.
2–0 ALP	ALP Memory leakage configuration bits

11.3.7 NEON Monitor Control (ARM_NMC)

This register is used to control the NEON monitor and can only be accessed by 32-bit secure/non-secure supervisor transactions.

Address: ARM_NMC is 63FA_0000h base + 20h offset = 63FA_0020h



ARM_NMC field descriptions

Field	Description
31 IE	Interrupt Enable 1 The monitor interrupt output is enabled and plat_ctrl_nm_irq_b is asserted when the monitor counter expires. 0 The monitor interrupt output is disabled.
30 NME	NEON Monitor Enable
29–20 Reserved	This read-only field is reserved and always has the value zero. Reserved
19–12 PL	Preload value for the upper 8 bits of the 16 bit NEON activity counter.

Table continues on the next page...

ARM_NMC field descriptions (continued)

Field	Description
11–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

11.3.8 NEON Monitor Status (ARM_NMS)

This register is used to read the status of the NEON monitor and can only be accessed by 32-bit secure/non-secure supervisor transactions.

Address: ARM_NMS is 63FA_0000h base + 24h offset = 63FA_0024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	NI	0														
W	w1c															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ARM_NMS field descriptions

Field	Description
31 NI	NEON Idle Status 1 Indicates that the NEON activity counter has expired. plat_ctrl_nm_irq_b will assert if the IE bit is set. 0 Indicates that the NEON activity counter has not expired.
30–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

11.4 Platform Clocks

The clocking strategy of the ARM platform can be summarized by the following bullets:

- The ARM platform will receive one functional clock from the clock control module (CCM), **arm_clk**, and several SoC clocks for boundary synchronization purposes (**arm_clk** can be completely asynchronous from any other SoC clocks).
- The ARM platform, in conjunction with an external CCM, will support dynamic clock frequency scaling.

- Divided clocks based on **arm_clk** will be generated within the platform for clocking buses which cannot operate at the high **arm_clk** frequencies. These divided clock ratios can be configured by on-platform registers.
  - Derived Clocks:
    - ipg_clk: This clock is used for IP Bus logic.
    - aclk: This clock is used for AXI bus logic.
    - dbg_atclk: This clock is used for the Debug AMBA Trace logic.
    - dbg_pclk: This clock is used for the Debug synchronizer logic.
- Two-level clock gating will be employed.
  - Block level clock gating will be used when possible. A specific block's clock will be gated off when it is not in use.
  - Register level clock gating will be used throughout the platform via power design tools.
- **arm_clk** may be turned off for various low-power use cases by an external CCM which will monitor the **DSM_request** output.

11.5 Platform Power Management

The ARM platform contains low voltage logic elements, an asynchronous and level shifted interface, state retention latches, low leakage power switches, and floating node isolation circuits. These elements each serves a specific purpose in the power management scheme implemented in the ARM platform. The power management capabilities of the ARM platform are summarized in this section.

11.5.1 Voltage and Frequency Scaling

The ARM platform contains structures necessary for the independent voltage scaling of the GP supply. Voltage and frequency scaling control systems are implemented at the chip level with the ARM platform acting as an object of the control. Asynchronous Interface Logic and Level-shifting between SoC-logic and ARM-logic have been implemented in the ARM platform to support the voltage and frequency scaling capabilities of the integrated system.

11.5.1.1 Asynchronous Interface Logic

Voltage and frequency scaling is most effective when there is a continuum of available frequencies at which the ARM platform may be clocked. Asynchronous Interface Logic removes the clock ratio dependencies between the ARM platform and the system in which it operates.

11.5.1.2 Level Shifting between SoC-logic and ARM-logic

Independent voltage scaling of the ARM platform requires a distinct power supply for the GP logic. Level shifters exist at the interface of the SoC logic and the ARM logic to allow independent voltage control of the SoC power supply and the ARM power supply.

11.5.2 Power Gating in the ARM platform

To reduce standby leakage power consumption, the ARM platform contains internal power supplies that may be completely power gated during wait for interrupt modes. Entry and exit sequences of power gating modes are completely controlled by programmed operation of the General Power Controller, so this section will only serve to describe what is possible in terms of how the ARM platform may be powered down during wait for interrupt modes.

11.5.2.1 Isolation Circuitry at the ARM platform Interface

In order to power down digital logic correctly, a special interface must exist which will protect the system from floating signals. Such isolation circuitry has been carefully designed in order to allow glitch-free power down modes of the ARM platform.

11.5.2.2 Power Gating the Memory Peripheries

Once the STANDBYWFI mode has been invoked, all of the ARM platform memory peripheries may be power gated by the assertion of both the l1_pwrdwn and the l2_pwrdwn signals. This will significantly reduce the amount of leakage from the GP supply.

11.5.2.3 Retaining the State of the ARM platform Registers

The ARM platform has been implemented using State Retention Power Gating (SRPG) register elements. In order to safely enter a power gated mode that includes power gating of the ARM platform high performance logic gates, a sequenced assertion of signals must occur. Similarly, a sequenced deassertion is necessary when emerging from power gated states. The required sequences are automatically provided to the ARM platform by the General Power Controller.

11.5.2.4 Power Gating the ARM platform High Performance Logic Gates

Once the STANDBYWFI mode has been invoked, the ARM platform high performance logic gates are power gated by assertion and negation of signals by internal Power management sub-blocks. This will turn off a portion of the leakage from the GP supply.

11.5.2.5 Power Gating the L2 Bit Arrays

If the L2 cache has been flushed, the L2 bit array may be power gated during the subsequent STANDBYWFI mode. Once the STANDBYWFI mode has been invoked, the L2 memory bit arrays may be power gated. This will turn off a portion of the leakage from the LP supply.

11.5.2.6 Controlling Power Gating using the General Power Controller (GPC)

Specific details concerning the power gating entry and exit sequence may be obtained from the block guide of the General Power Controller (GPC). This high level summary is intended to provide some overview.

Code must be formed that will take advantage of low workloads by putting the ARM platform into a STANDBYWFI state whenever possible. Prior to entering the STANDBYWFI state, the code should assess several things about the workload and the battery conservation requirements:

- What is the maximum acceptable interrupt service latency?
- What is the expected time to be spent in STANDBYWFI?

Once these time requirements are understood, then Table 11-10 can be used to determine the appropriate STANDBYWFI mode to invoke. The GPC registers can be programmed with the appropriate values to invoke the appropriate STANDBYWFI mode. After GPC programming is complete, the code should execute the STANDBYWFI instruction.

The GPC will enforce the appropriate sequence for entering and exiting the programmed power gated modes.

11.5.2.7 Power Gating the NEON Block

The NEON co-processor in the ARM platform can be put into a state retention mode and powered down to reduce current consumption when it is not needed. The sequence is shown in the flow diagram of Figure 11-10. Each box in the diagram is numbered to correspond to the following more thorough description:

1. User Selects NEON to be Powered Down

   The NEON Coprocessor is enabled out of reset and will remain powered up unless the user decides to power it down. This method will power down NEON after a user programmable time of NEON inactivity. The current state of the NEON is held in SRPG flops. To select this operation:

   The user code will program the amount of time that the NEON needs to be inactive before powerdown in the NEON Monitor Control Register. Effectively, the user programs the upper 8 bits of a 20-bit counter. This counter runs at the arm_clk rate and so the user can program a timeout from 4K to 1M arm_clk rising edges.

   In the same register, the user code enables the NEON Monitor and enables the NEON Monitor Interrupt.

2. Pre-Set Timer Count

   The timer is loaded (automatically by hardware) with the count value chosen by the user.

3. Decision: NEON Instruction Encountered?

   If a NEON instruction is encountered the sequence is aborted until the NEON is no longer in use. As long as NEON Instructions are executing, the NEON status bit will remain in the active state and the counter gets forced with the pre-set count.

4. Decrement Timer

   When no NEON instructions are encountered, the timer will count towards the timeout.

5. Decision: Timed Out?

   If the user selected time has lapsed without any NEON activity, then the hardware will generate an interrupt.

6. Exception Sequence: Power Down NEON

   The interrupt service routine disables the NEON sub-block with the following sequence:

   A. Software must disable access to the NEON unit using the Coprocessor Access Control Register, see c1, Coprocessor Access Control Register on page 3-64 of the ARM TRM. All outstanding NEON instructions retire and all subsequent NEON instruction cause an Undefined instruction exception.

```
MRC p15, 0, <Rd>, c1, c0, 2; Read Coprocessor Access Control Register
BIC <Rd>, <Rd>, #0xF00000; Disable access to CP10 and CP11
MCR p15, 0, <Rd>, c1, c0, 2; Write Coprocessor Access Control Register
```

   B. Activate the NEON output clamps.

   C. Place NEON into state retention.

   D. Prepare the NEON memory for power down keeping the state of its bits.

   E. Separate the NEON power supplies so that the ones that retain state can continue to have power.

   F. Remove power from the NEON power domain.

7. Decision: NEON Instruction Attempted

   The part will continue running regular ARM Platform instructions with the NEON in low power state as long as no NEON instructions are attempted. When a NEON instruction is attempted, an unimplemented instruction exception occurs.

8. Exception Sequence: Power Up NEON

   An exception is taken when a NEON instruction is attempted with the NEON is disabled. The exception routine will determine that this is indeed what occurred, enable the NEON block and wait for the NEON to power up. Returning from the exception will cause the NEON instruction to be restarted and run successfully. This is a more detailed description of the sequence:

   A. Power is turned on to the NEON power domain.

   B. Reconnect the NEON power supplies together.

   C. Restore the NEON memory power maintaining the state of it's saved bits.

   D. Reload the flops with the SRPG saved contents.

E. Release the NEON output clamps.

F. Software must enable access to the NEON unit using the Coprocessor Access Control Register, see c1, Coprocessor Access Control Register on page 3-64 of the ARM TRM.

```
MRC p15, 0, <Rd>, c1, c0, 2; Read Coprocessor Access Control Register
ORR <Rd>, <Rd>, #0xF00000; Enable access to CP10 and CP11
MCR p15, 0, <Rd>, c1, c0, 2; Write Coprocessor Access Control Register
```

G. Return from the exception causing the NEON instruction to be reloaded and executed.

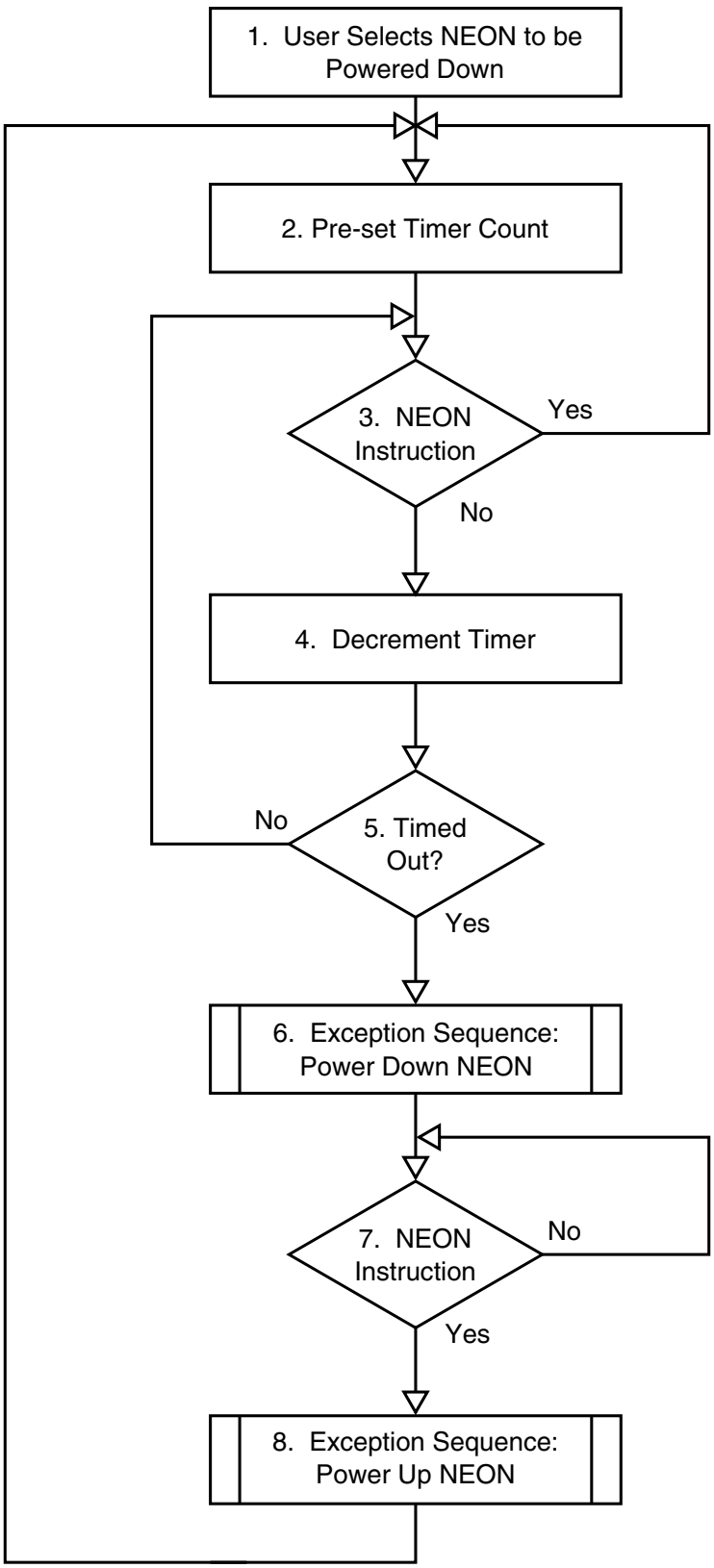


Figure 11-10. Flow Diagram of NEON Power Down Sequence

11.5.3 Modes of Operation

There are several basic low power modes of the ARM platform.

Table 11-10. Modes of Operation of the ARM platform

MODE	Description
RUN	ARM_CLK running, code executing
CLOCKED_WAIT	ARM_CLK running, STANDBYWFI=TRUE
UNCLOCKED_WAIT	ARM_CLK off, STANDBYWFI=TRUE
STOP1	UNCLOCKED_WAIT plus L1 and L2 peripheries power gated
STOP2	UNCLOCKED_WAIT plus L1 and L2 peripheries power gated, ARM_HiP_logic power gated
STOP3	UNCLOCKED_WAIT plus L1 and L2 peripheries power gated, ARM_HiP_logic power gated, L2 cache flushed, and L2 bit arrays power gated

There are several combinations of reduced power in which one of more portions of the platform are in a power saving mode.

Table 11-11. Platform Power Modes

Core and Platform	NEON	L1	L2
Powered Up	Powered Up	Powered Up	Powered Up
Powered Up	Reset (ARM Mode)	Powered Up	Powered Up
State Retention	State Retention	State Retention	State Retention
State Retention	State Retention	Powered Down	State Retention
State Retention	State Retention	State Retention	Powered Down
State Retention	State Retention	Powered Down	Powered Down

Chapter 12
ARM Platform Debug

12.1 Introduction

The purpose of this document is to provide an overview of the ARM platform debug. It will cover the sub-blocks inside the armp_gp_debug block and also the blocks that support the platform debug within the SOC, but external to the ARM platform.

Debug for the ARM platform uses ARM's DK11 CoreSight Debug Kit. The following CoreSight blocks are used: Debug Access Port (DAP) , Embedded Trace Macrocell v 3.3 (ETM), CoreSight replicator (CSREPLICATOR), CoreSight trace port interface unit (CSTPIU), CoreSight embedded trace buffer (CSETB), CoreSight cross trigger interface (CSCTI) and the CoreSight cross trigger matrix (CSCTM). In this introduction, a brief description of each block will be given. For more in depth information, please refer to the CoreSight_DK_TRM and other relative documentation from ARM.

12.1.1 Overview

ARM's CoreSight Design Kit (CSDK) provides a single solution for multi core and bus trace. The CSDK provides the following capabilities for system-wide trace:

- debug and trace visibility of the entire system
- cross triggering support between SOC subsystems
- multi-source trace in a single stream
- higher data compression than previous solutions
- standard programmer's models for standard tools solutions

The CoreSight Design Kit comprises the following main components: Control and access, sources, links and sinks.

- Control and access components configure, access and control the generation of trace. They do not generate, nor process trace data. Examples include the debug access port (DAP) and the embedded cross trigger (ECT) interface.

- Source components generate trace data. Example source components are the ETM and the future AXI trace macrocell (XTM).
- Link components provide connection, triggering and flow of trace data. Link examples include the ATB replicator and the CoreSight Trace funnel.
- Sink components are the end point for trace data on an SOC. Example sinks are the trace port interface unit (TPIU) and the embedded trace buffer (ETB).

The figure below shows the ARM Platform debug block diagram. Further detail of the sub-blocks will be provided in the sections that follow.

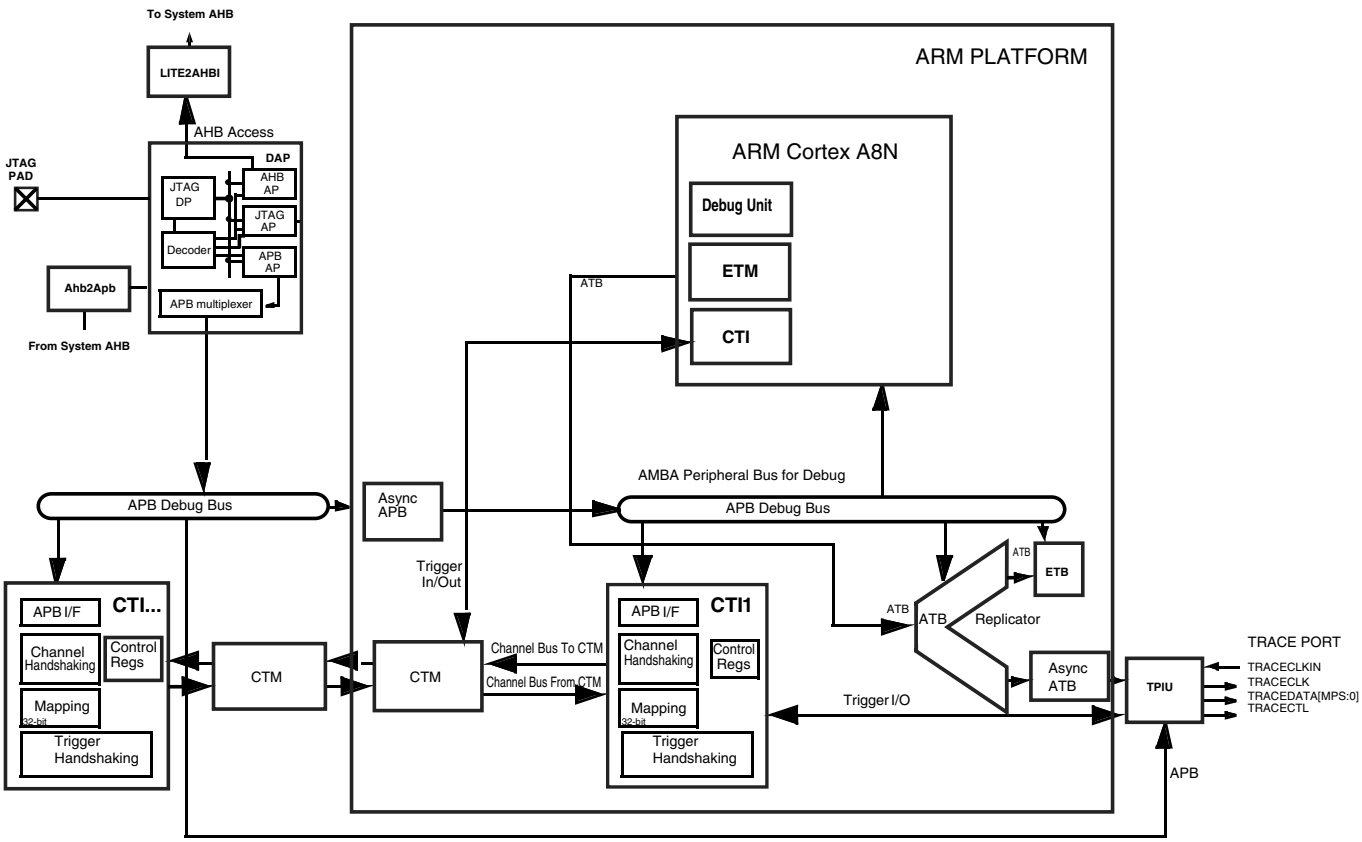


Figure 12-1. Block Diagram

12.1.2 ARM Debug Blocks

This section gives an overview of the Debug blocks used in the ARM platform and also the blocks outside the platform that are included in the CoreSight debug system.

The following blocks are included in the ARM platform debug block: CSETB, CSCTI, CSCTM, CSREPLICATOR and APB address decode logic.

The following blocks are part of the ARM platform, outside the debug block: ETM (embedded in the Cortex), CTI (also embedded in the Cortex), the Processor Debug Unit and two asynchronous bridges - the asyncapb and the asyncatb.

Also included in this block guide is an overview of the DAP and TPIU which reside outside the ARM platform, yet are key components to the debug system.

12.1.2.1 Processor Debug Unit

The processor debug unit assists in debugging software running on the processor. In combination with a software debugger program, the debug unit enables debugging the following:

- application software
- operating systems
- hardware systems based on an ARM processor

The debug unit enables:

- stopping program execution
- examining and altering processor and coprocessor states
- examining and altering memory and input/output peripheral states
- restarting the processor core

There are three ways to debug software running on the processor:

- Halt debug-mode debugging
- Monitor debug-mode debugging
- Trace debugging (ETM) - covered in a different section

12.1.2.1.1 Halting Debug-Mode Debugging

Halting debug-mode debugging is invasive. The processor halts when a debug event, such as a break point, occurs. An external debugger can examine and modify the processor state via the APB while the processor is halted.

12.1.2.1.2 Monitor Debug-Mode Debugging

When a debug event occurs during monitor debug-mode debugging, instead of halting the processor, the processor takes an exception. Special software can then take control to examine or alter the processor state. When execution of a monitor target starts, the state of the processor is preserved in the same manner as all ARM exceptions.

12.1.2.1.3 Programming the Debug Unit

The processor unit is programmed using the APB slave port. Features that can be accessed using the memory-mapped APB registers are:

- instruction address comparators for triggering break points
- data address comparators for triggering watchpoints
- a bidirectional Debug Communication Channel (DCC)
- all other state information associated with the debug unit

12.1.2.2 ARM embedded trace macrocell (ARM ETM)

The ARM ETM provides real time instruction trace for the Cortex A8N processor. It is designed to be used with the CoreSight Design Kit. Unlike previous versions of the ARM platforms, this ETM is embedded inside the ARM Core.

Real time tracing is controlled by specifying a set of filtering and triggering resources which include address and data comparators, counters and sequencers.

NOTE

Though data trace can not be enabled, the ETM can still trigger based on data values. The ETM can also trace data address values.

The ARM ETM contains the following main components:

- Core interface
- Trace generation
- Filtering and triggering resources
- Main FIFO
- AMBA 3 ATB interface
- AMBA 3 APB interface

The ETMv3.3 provides a number of configurations. The table below shows the options implemented in the ETM.

Table 12-1. ETMv3.3 Configurations

Resource Description	Configuration
Instruction trace	Yes
Data address trace	Yes
Data value trace	No
Jazelle trace	-

Table continues on the next page...

Table 12-1. ETMv3.3 Configurations (continued)

Resource Description	Configuration
Address comparator pairs	2
Data comparator	4
Context ID comparators	1
Sequencer	Yes
Start/stop sub-block	Yes
Embedded ICE comparators	0
External inputs	4
External outputs	2
Extended external inputs	49
Extended external input selectors	2
Instrumentation resources	4
FIFO full	No
FIFO full level setting	N/A
Branch broadcasting	Yes
ASIC control register (bits)	8
Data suppression	Yes
Software access to registers	Memory
Readable registers	Yes
FIFO size	128 bytes
Minimum port size	32
Maximum port size	32
Port modes	Dynamic
Asynchronous ATB	Yes
Load pc first	No
Fetch comparisons	No

12.1.2.3 CoreSight Embedded Trace Buffer (CSETB)

The ETB provides on chip storage of trace data. The ARM platform implements a 32-bit 4K RAM.

The ETB accepts trace data from a CoreSight replicator via an ATB write port. The ETB is configurable through an APB port.

Features:

- 4kB compiled memory for the trace buffer. Note: It can no longer be used as a general purpose memory.
- Only 32-bit accesses to the 4kB buffer is supported (Note: RAM size is 4K "bytes" - i.e. 1Kx32)
- Amba Peripheral Bus interface for configuration and memory access

12.1.2.4 CoreSight Replicator (CSREPLICATOR)

The ATB Replicator enables two trace sinks (TPIU and ETB) to be wired together and receive ATB trace data from the same trace source (ETM). There are no programable registers. It takes incoming trace data from a single source (ETM) and replicates it to two master ports.

The CSREPLICATOR is part of the armp_gp_debug block. Two output master ports are required for this design - one for the on-platform ETB and one for the off-platform TPIU. Placing the TPIU off platform allows future debug trace sources to connect to the TPIU through a FUNNEL, without the need to modify the ARM platform. The figure below shows a block diagram for the CSREPLICATOR.

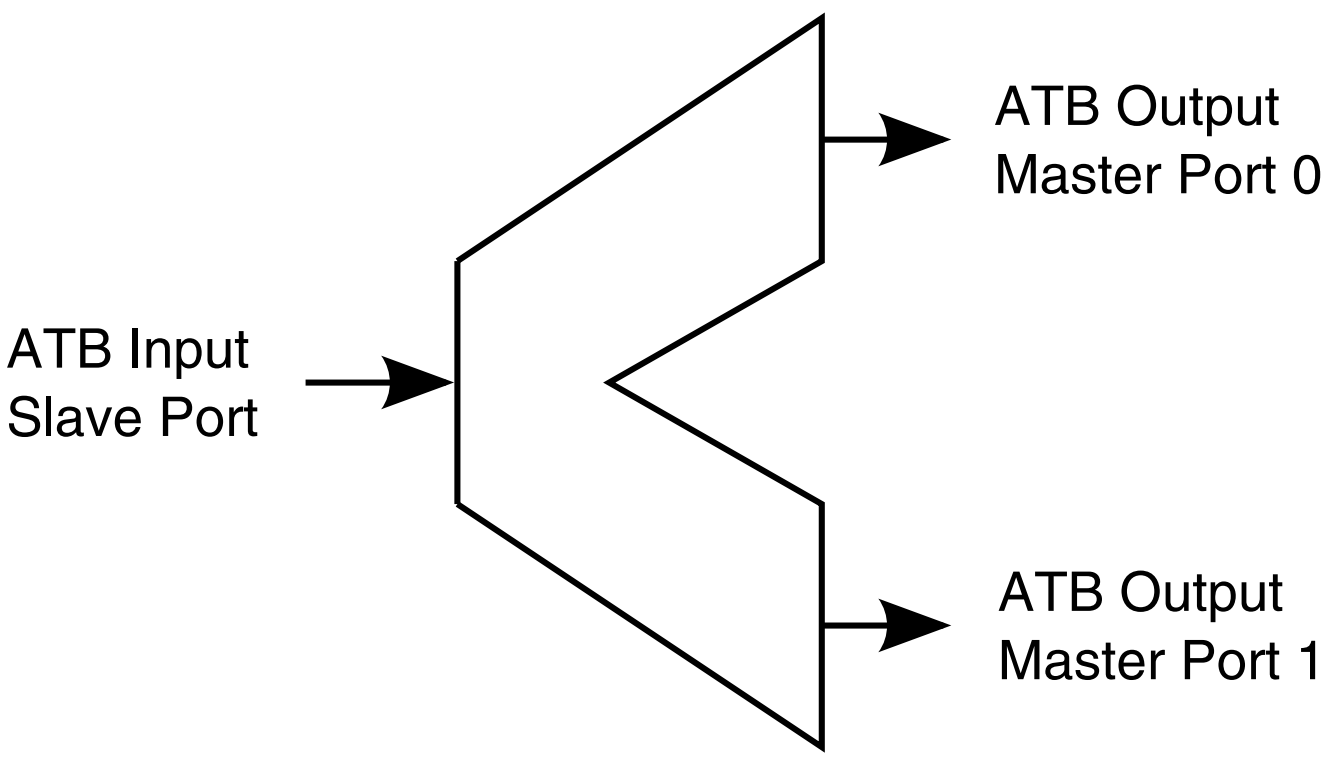


Figure 12-2. CSREPLICATOR Block Diagram

12.1.2.5 CoreSight trace port interface unit (CSTPIU)

The TPIU acts as a bridge between on-chip trace data, ID distinguishable, and a TPA. It receives ATB trace data and sends it off chip via ARM's standard trace signals TRACECLK, TRACEDATA and TRACECTL. The following sub-blocks are included in the TPIU (see the figure below): ATB interface, APB interface, Formatter, Asynchronous FIFO, Register bank, Trace out serializer and a Pattern generator. Further information on the TPIU can be found in ARM's technical reference manuals.

The TPIU is not on the ARM platform and is configurable via the APB.

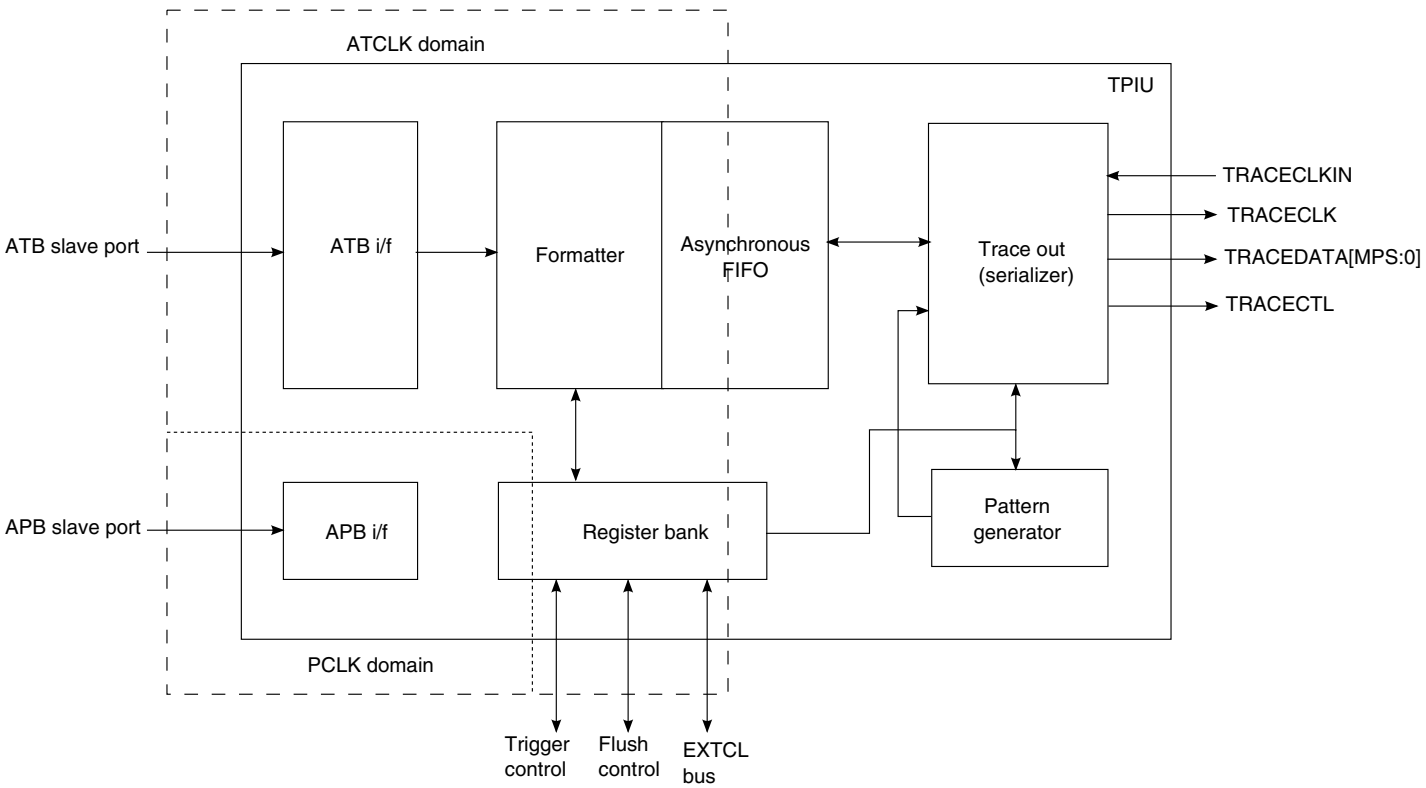


Figure 12-3. TPIU Block Diagram

All signal paths to the pads are subject to wire delays. Special consideration should be taken to re-balance the paths, removing the relative skews between the signals. An extra delay must be added on the TRACECLK path to ensure its rising and falling edge are during the stable part of TRACEDATA and TRACECTL.

12.1.2.6 CoreSight cross trigger interface (CSCTI)

The CSCTI is the CoreSight cross trigger interface component of an Embedded Cross Trigger (ECT) system. The CSCTI combines and maps trigger requests, broadcasting them to all other interfaces on the ECT as channel events. This enables subsystems to cross trigger with each other.

Each CTI has 8 trigger inputs, 8 trigger outputs and 4 sets of 4 channel I/Os. There are also acknowledge signals, configurable on/off, for the trigger and channel outputs. The channels connect to a CoreSight Cross Trigger Matrix (CSCTM). The CSCTM is off-platform, so special consideration should be taken when enabling the synchronizers and handshaking features on both the CSCTI and the CSCTM.

The platform requires one CTI block in addition to the embedded CTI in the ARM core. CTI0 resides in the Core and handles triggers/events from the CSETM and the CORE; CTI1 handles the triggers/events for the TPIU (off platform) and the CSETB. Since the TPIU is outside the platform (asynchronous), the TRIGIN synchronizers are enabled and handshaking is turned on.

The on-platform CTI1 is synchronous and at the same clock speed as the on-platform CTM. This allows the channel handshaking and channel synchronizers to be bypassed.

The figure below shows the recommended connections for CSCTI1.

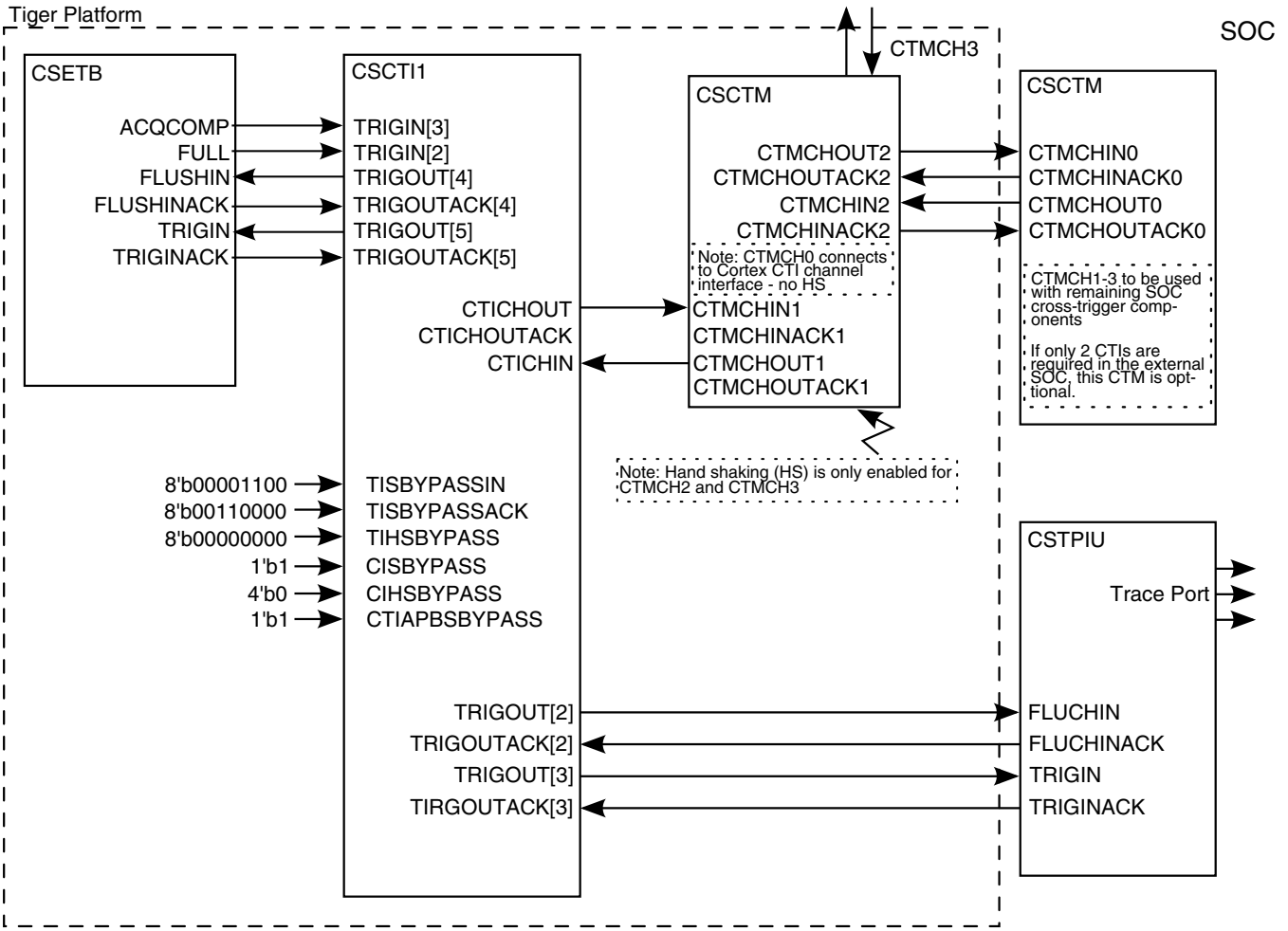


Figure 12-4. CSCTI1 Connections

The SOC CTIs require additional logic to support asynchronous trigger sources/ destinations. The cscti_extended sub-block includes logic, in addition to the CSCTI block, to support triggering across asynchronous boundaries where the source/destination might not have the necessary logic to support asynchronous boundaries.

12.1.2.7 CoreSight Cross Trigger Matrix (CSCTM)

This block controls the distribution of channel events. It provides channel interfaces to the CSCTIs. The CSCTM can also connect to another CSCTM via a channel interface. This allows multiple CSCTMs to be connected.

The ARM has one CSCTM inside the platform to handle events between the platform's CSCTIs and the rest of the ECT system outside the platform. Placing a CSCTM on platform minimizes the event cycle time between the ETB and the ETM. The handshaking between the CTIs on-platform and the on-platform CTM are not enabled.

This requires the Cortex's CTI, the on-platform CTI1 and the on-platform CTM to all be in the same clock domain. The on-platform CTM connects to the ECT system via an off-platform CTM and the respective channels require synchronizing and handshaking enabled.

If only two CTIs are required in the SOC, they can be connected directly to the on-platform CTM eliminating the need for a CTM outside the platform.

More information on the CoreSight ECT system can be found in ARM's Technical Reference manuals.

12.1.2.8 Debug Access Port (DAP)

The DAP provides multiple master driving ports, all accessible via a single external interface port. Components that access the DAP are called Debug Ports (DP) and components that access internal interfaces are called Access Ports.

The DAP provides real time access for the debugger without halting the core to:

- All debug configuration registers
- AMBA system memory and peripheral registers.

The DAP enables debug access to the complete SOC through a number of access ports. Access to the CoreSight Debug APB is enabled through the APB access port (APB-AP). System access can be accomplished via an AHB access port (AHB-AP). An APB multiplexer allows system accesses to debug CoreSight components connected to the APB.

There is also a JTAG access port (JTAG-AP) providing accesses to on-chip JTAG components. The DAP acts as a JTAG master. This feature will not be used for the ARM platform since the core's debug logic is now accessible through the APB.

The figure below shows a block diagram of the DAP.

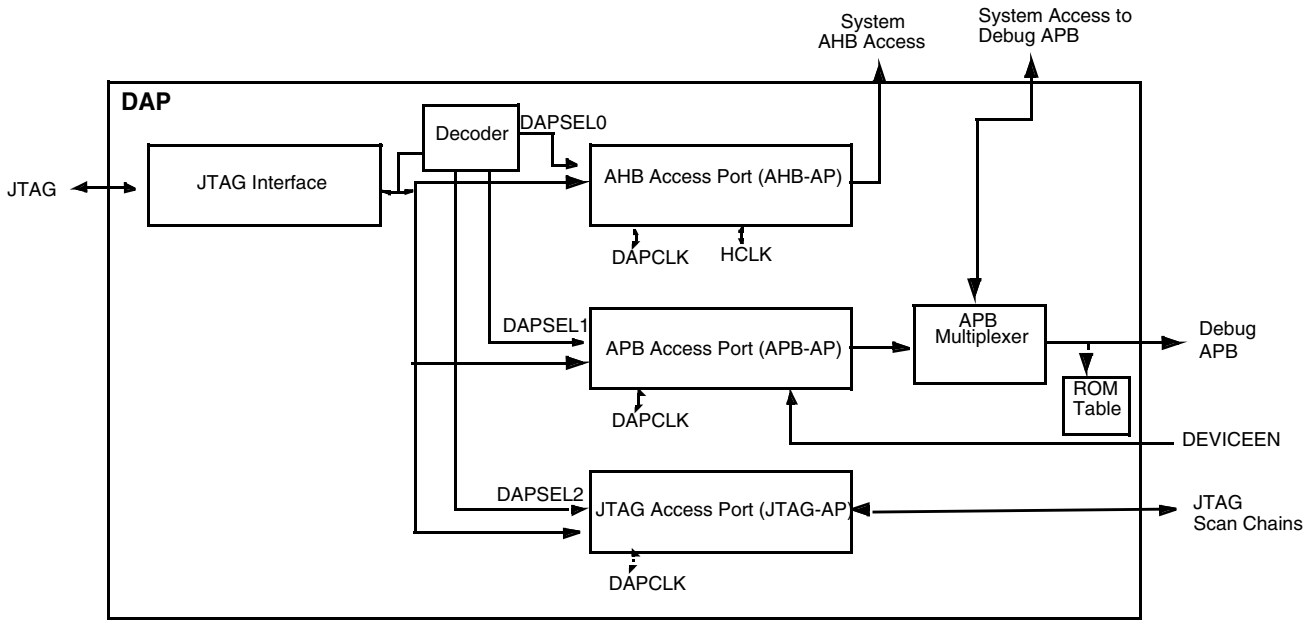


Figure 12-5. DAP Block Diagram

12.1.2.8.1 DAP_SYS

The DAP also requires decode logic for the CoreSight component PSEL signals, an AHB to APB bridge, an AHB lite to AHB bridge, detect logic to enable debug and reset synchronization logic. The dap_sys sub-block wraps these components with the DAP. The figure below shows a block diagram of the DAP_SYS.

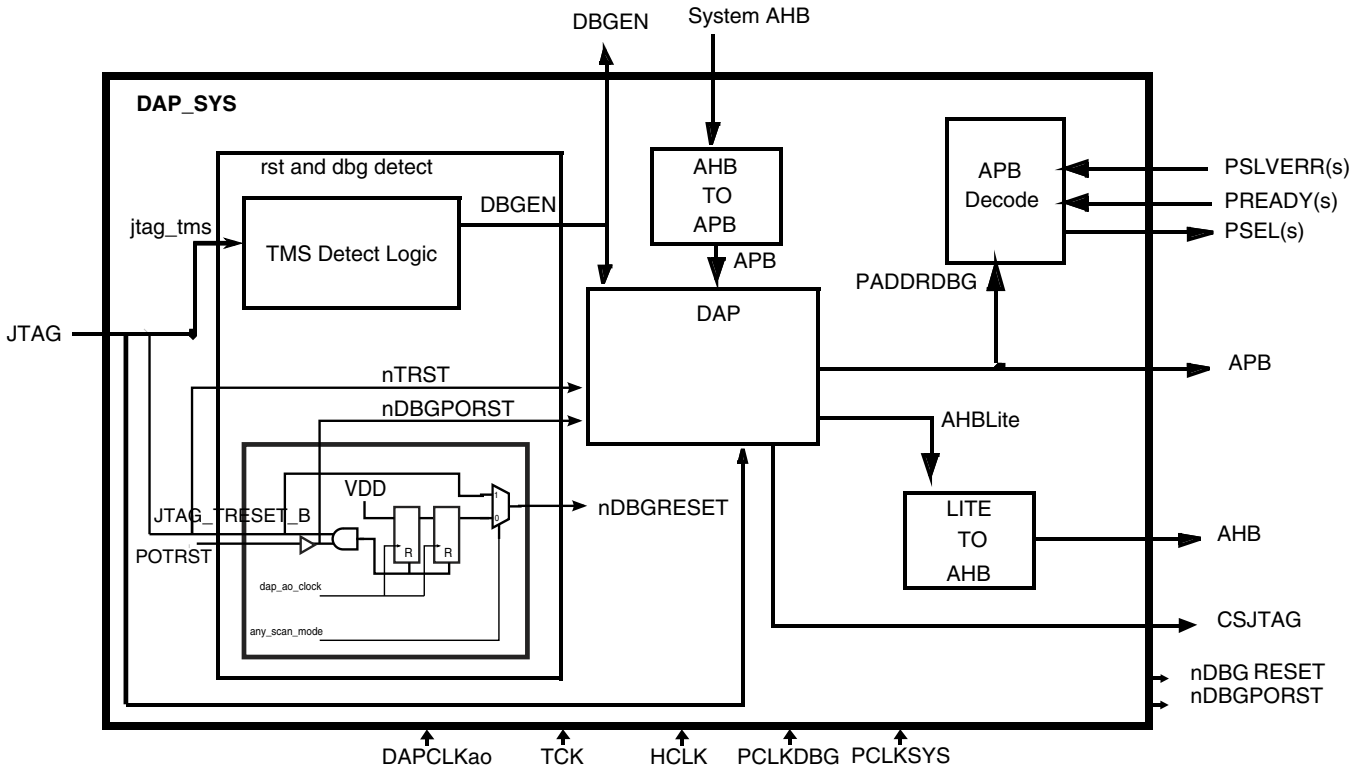


Figure 12-6. DAP_SYS Block Diagram

12.1.3 Modes of Operation

12.1.3.1 ARM Invasive Debug Mode

ARM Invasive Debug mode is usually entered through the JTAG port. Inside the dap_sys there is logic to assert DBGEN high when jtag_tms is active. The user can also invoke DBGEN by writing to a register inside the ARM plat_ctrl sub-block.

It is required that NIDEN is asserted while DBGEN is asserted. Otherwise, CTI functionality is limited when TINIDENSEL is tied to 0. Due to this requirement, NIDEN into the Core and the Debug block is a combination of dbg_niden ored with dbgen_in.

12.1.3.2 ARM Non-Invasive Debug Mode (Real-time Trace)

There are two methods to enter non-invasive debug mode - through JTAG or via memory mapped accesses. For memory mapped accesses, non-invasive debug can be enabled by writing to two secure supervisor registers, one within the Central Security Unit (CSU) and one within the ARM platform.

12.1.3.3 Normal Operating Modes

During normal operating mode, debug is not enabled.

12.1.3.4 Low Power Modes

During deep sleep mode (DSM) all debug is disabled. The platform_ctl sub-block has a bit allocated to overriding DSM when debug is enabled. This bit currently resets to a state disabling DSM while debug is enabled. For more information, refer to the platform control block guide.

The core has an input, DBGNOCLKSTOP (from ARM_GPC[NOCLKSTOP]), allowing the core's debug clocks to stay on while in WFI. The same bit used to override DSM should be connected to DBGNOCLKSTOP (inverted). Another solution is to allocated separate bit for this function in the platform_ctl sub-block. Make note that if the dsm override bit is not set and the platform goes into DSM, it doesn't matter what DBGNOCLKSTOP is set to.

12.2 Memory Map and Register Definition

Each CoreSight component has an allocated 4K. CoreSight components are part of either CoreSight Class or ROM Class. The figure below shows the CoreSight class layout for a 4KB block.

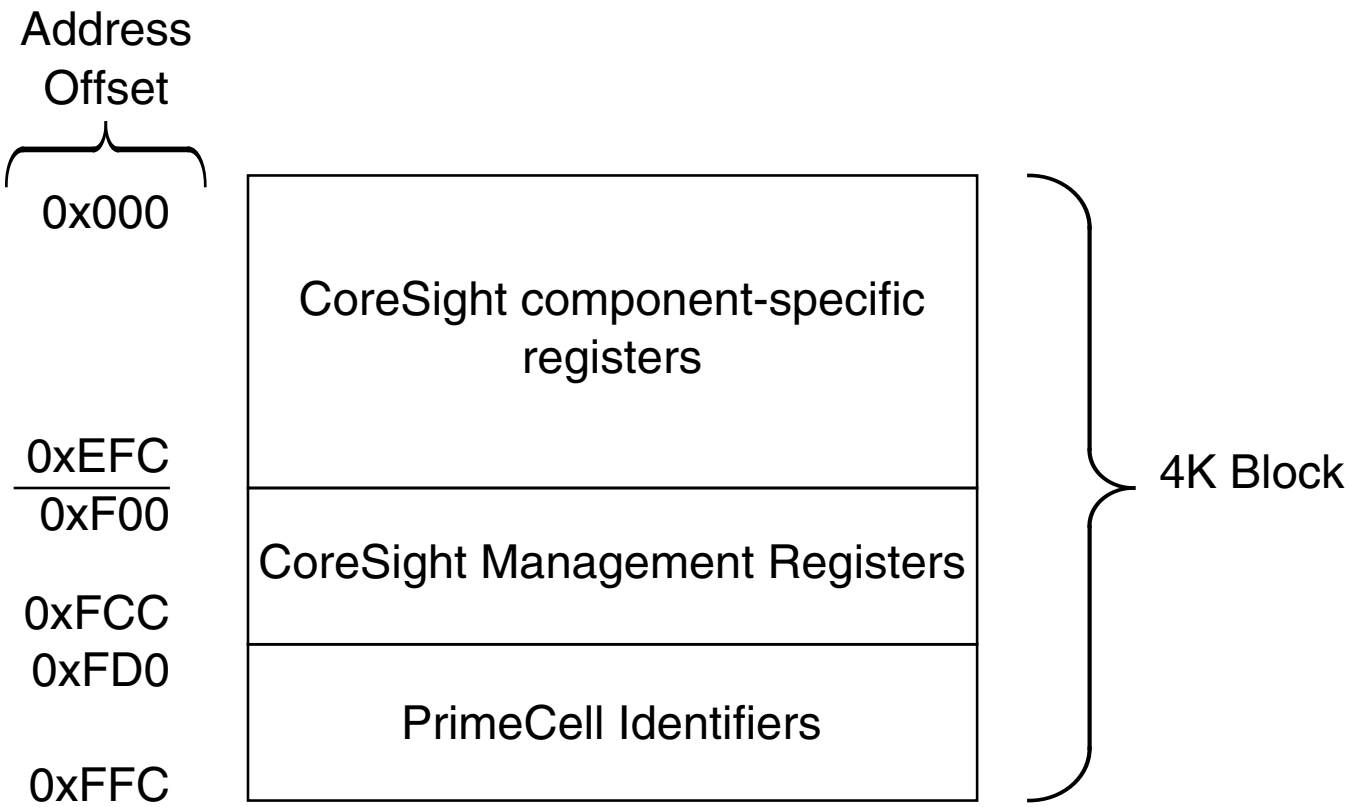


Figure 12-7. CoreSight Component Memory Map

The following table shows an example Debug memory map. The most significant half-word of the address will change based on the chip level memory map.

Table 12-2. Block Memory Map

Start Address	End Address	Size	Target IP	Comments
B0C00000	B0C00FFF	4K	Debug ROM	Located inside DAP
B0C01000	B0C01FFF	4K	ETB	
B0C02000	B0C02FFF	4K	ETM	
B0C03000	B0C03FFF	4K	TPIU	
B0C04000	B0C04FFF	4K	CTI0	On platform
B0C05000	B0C05FFF	4K	CTI1	On platform
B0C06000	B0C06FFF	4K	CTI2	Off platform
B0C07000	B0C07FFF	4K	CTI3	Off platform

12.2.1 Register Summary

This section summarizes the CoreSight Component registers in a table format. The tables are divided by Components. For a more detailed description of each register, refer to the CoreSight_DK_TRM.

The conventions in the table below serve as a key for the register summary.

Table 12-3. Register Conventions

Register Field Types	
RO	Read only. Writing this bit has no effect.
WO	Write only.
R/W	Standard read/write bit. Only software can change the bit's value (other than a hardware reset).
rwm	A read/write bit that may be modified by a hardware in some fashion other than by a reset.
w1c	Write one to clear. A status bit that can be read, and is cleared by writing a one.
RAZ	Read-as-zero
WI	Writes ignored
Self-clearing bit	Writing a one has some effect on the Block, but it always reads as zero. (Previously designated slfclr)

12.2.1.1 DAP JTAG DP Registers

Accessing a JTAG DP register depends on both the instruction register (IR) value of the DAP access and the address field of the DAP access.

The following table shows a register summary table for the DAP DP. Detailed register descriptions can be found inside the CoreSight_DK_TRM provided by ARM.

Table 12-4. DAP JTAG DP Registers

Address Field	IR Contents	Type	Register Name
1	IDCODE	RO	Identification Control Register
0x0	DAPACC	RAZ/WI	Reserved
0x4	DAPACC	R/W	DP Control/Status Register
0x8	DAPACC	R/W	Select Register
0x0	ABORT	WO	DAP Abort Register
0x4-0xC	ABORT	2	Reserved

1. There is no address associated with the ID code register. For more information, see the CoreSight_DK_TRM.
2. The value read on the abort scan chain is unpredictable. The result of accessing the abort scan chain without the address set to 0x0 is unpredictable.

12.2.1.2 DAP ROM Register Summary

The ROM table stores the locations of components on the debug APB. The DAP ROM is read only and configurable through the DAPROM.v and DapRomDefs.v RTL files. Writes are ignored.

The following table shows a register summary table for the DAP Rom. Detailed descriptions can be found inside the CoreSight_DK_TRM provided by ARM.

Table 12-5. DAP ROM Registers

Offset	Name	Type	Description
0xFD0	Peripheral ID4	RO	See CoreSight_DK_TRM
0xFD4	Peripheral ID5	RAZ	Reserved
0xFD8	Peripheral ID6	RAZ	Reserved
0xFDC	Peripheral ID7	RAZ	Reserved
0xFE0	Peripheral ID0	RO	See CoreSight_DK_TRM
0xFE4	Peripheral ID1	RO	See CoreSight_DK_TRM
0xFE8	Peripheral ID2	RO	See CoreSight_DK_TRM
0xFEC	Peripheral ID3	RO	See CoreSight_DK_TRM
0xFF0	Component ID0	RO	Set to 0x0D
0xFF4	Component ID1	RO	Set to 0x10
0xFF8	Component ID2	RO	Set to 0x05
0xFFC	Component ID3	RO	Set to 0xB1

12.2.1.3 Processor Debug Unit Register Summary

Most of the debug unit registers are accessible through the APB. There are some registers that can also be accessed through a coprocessor interface - CP14.

By default, CP14 registers can be accessed from a non-privileged mode. However, the processor can be programmed to disable user access modes using bit [12] of the Debug Status and Control Register (DSCR). For more information on the DSCR and access to CP14 registers, refer to the CortexA8 TRM document.

12.2.1.3.1 Coprocessor Registers Summary

The table below shows the valid debug instructions for accessing the debug registers.

Table 12-6. CP14 Debug Registers Summary

Instruction	Mnemonic	Description
MRC p14, 0, <Rd>, c0, c0, 0	DIDR	Debug Identification Register
MRC p14, 0, <Rd>, c1, c0, 0	DRAR	Debug ROM Address Register
MRC p14, 0, <Rd>, c2, c0, 0	DSAR	Debug Self Address Register
MRC p14, 0, <Rd>, c0, c5, 0 STC p14, c5, <addressing mode>	DTRRX	Data Transfer Register - Receive
MCR p14, 0, <Rd>, c0, c5, 0 LDC p14, c5, <addressing mode>	DTRTX	Data Transfer Register - Transmit
MRC p14, 0, <Rd>, c0, c1, 0 MRC p14, 0, PC, c0, c1, 0	DSCR	Debug Status and Control Register

12.2.1.3.2 Memory Mapped Registers Summary

The table below shows a complete list of memory mapped registers accessible using the APB slave port.

Table 12-7. Debug Unit APB Accessible Registers Summary

Offset	Register Number	Mnemonic	Type	Description
0x000	c0	DIDR	R	CP14 c0, Debug ID Register
0x004-0x014	c1-c5	-	R	RAZ
0x018	c6	WFAR	R/W	Watchpoint Fault Address Register
0x01C	c7	VCR	R/W	Vector Catch Register
0x020	c8	-	R	RAZ
0x024	c9	ECR	R/W	Event Catch Register
0x028	c10	DSCCR	R/W	Debug State Cache Control Register
0x02C	c11	-	R	RAZ
0x030-0x07C	c12-c31	-	R	RAZ
0x080	c32	DTRRX	R/W	Data Transfer Register - Receive
0x084	c33	ITR	W	Instruction Transfer Register
0x088	c34	DSCR	R/W	CP14 c1, Debug Status and Control Register
0x08C	c35	DTRTX	R/W	Data Transfer Register - Transmit
0x090	c36	DRCR	W	Debug Run Control Register
0x094-0x09C	c37-c63	-	R	RAZ
0x100-0x114	c64-c69	BVR	R/W	Breakpoint Value Registers
0x118-0x13c	c70-c79	-	R	RAZ
0x140-0x154	c80-c85	BCR	R/W	Breakpoint Control Register

Table continues on the next page...

Table 12-7. Debug Unit APB Accessible Registers Summary (continued)

Offset	Register Number	Mnemonic	Type	Description
0x158-0x17C	c86-c95	-	R	RAZ
0x180-0x184	c96-c97	WVR	R/W	Watchpoint Value Register
0x188-0x1BC	c97-c111	-	R	RAZ
0x1C0-0x1C4	c112-c113	WCR	R/W	Watchpoint Control Registers
0x1C8-0x1FC	c114-c127	-	R	RAZ
0x200-0x2FC	c128-c191	-	R	RAZ
0x300	c192	OSLAR	W	Operating System Lock Access Register
0x304	c193	OSLSR	R	Operating System Lock Status Register
0x308	c194	OSSRR	R/W	Operating System Save and Restore Register
0x30C	c195	-	R	RAZ
0x310	c196	PRCR	R/W	Device Power Down and Reset Control Register
0x314	c197	PRSR	R	Device Power Down and Reset Status Register
0x318-0x7FC	c198-c511	-	R	RAZ
0x800-0x8FC	c512-575	-	R	RAZ
0x900-0xCFC	c576-c831	-	R	RAZ
0xD00-0xFFC	c832-c1023	-	-	Management Register

12.2.1.4 ETB Register Summary

The ETB registers are summarized in the table below. Detailed information can be found in ARM's CoreSight_DK_TRM document.

Table 12-8. ETB Registers

Offset	Name	Type	Description
0x004	RAM Depth Register, RDP	RO	See CoreSight_DK_TRM
0x010	RAM Read Data Register, RRD	RO	See CoreSight_DK_TRM
0x014	RAM Read Pointer Register, RRP	R/W	See CoreSight_DK_TRM
0x00C	Status Register, STS	RO	See CoreSight_DK_TRM
0x018	RAM Write Pointer Register, RWP	R/W	See CoreSight_DK_TRM
0x01C	Trigger Counter, TRG	R/W	See CoreSight_DK_TRM
0x020	Control Register, CTL	R/W	See CoreSight_DK_TRM
0x024	RAM Write Data, RWD	WO	See CoreSight_DK_TRM

Table continues on the next page...

Table 12-8. ETB Registers (continued)

Offset	Name	Type	Description
0x300	Formatter and Flush Status FFSR	RO	See CoreSight_DK_TRM
0x304	Formatter and Flush Control FFCR	R/W	See CoreSight_DK_TRM
0xEE0	Integration Register ITMISCOP0	WO	See CoreSight_DK_TRM
0xEE4	Integration Register ITTRFLINACK	WO	See CoreSight_DK_TRM
0xEE8	Integration Register ITTRFLIN	RO	See CoreSight_DK_TRM
0xEEC	Integration Register ITATBDATA0	RO	See CoreSight_DK_TRM
0xEF0	Integration Register ITATBCTR2	WO	See CoreSight_DK_TRM
0xEF4	Integration Register ITATBCTR1	RO	See CoreSight_DK_TRM
0xEF8	Integration Register ITATBCTR0	RO	See CoreSight_DK_TRM
0xF00	Integration Mode Control Register	R/W	See CoreSight_DK_TRM
0xFA0	Claim Tag Set Register	R/W	See CoreSight_DK_TRM
0xFA4	Claim Tag Clear Register	R/W	See CoreSight_DK_TRM
0xFB0	Lock Access Register	WO	See CoreSight_DK_TRM
0xFB4	Lock Status Register	RO	See CoreSight_DK_TRM
0xFB8	Authentication Status Register	RO	See CoreSight_DK_TRM
0xFC8	Device ID	RO	See CoreSight_DK_TRM
0xFCC	Device Type Identifier Register	RO	See CoreSight_DK_TRM
0xFD0	Peripheral ID4	RO	See CoreSight_DK_TRM
0xFD4	Peripheral ID5	RAZ	Reserved
0xFD8	Peripheral ID6	RAZ	Reserved
0xFDC	Peripheral ID7	RAZ	Reserved
0xFE0	Peripheral ID0	RO	See CoreSight_DK_TRM
0xFE4	Peripheral ID1	RO	See CoreSight_DK_TRM
0xFE8	Peripheral ID2	RO	See CoreSight_DK_TRM

Table continues on the next page...

Table 12-8. ETB Registers (continued)

Offset	Name	Type	Description
0xFEC	Peripheral ID3	RO	See CoreSight_DK_TRM
0xFF0	Component ID0	RO	Set to 0x0D
0xFF4	Component ID1	RO	Set to 0x10
0xFF8	Component ID2	RO	Set to 0x05
0xFFC	Component ID3	RO	Set to 0xB1

12.2.1.5 ETM Register Summary

The ETM registers are described in the table below. A more detailed description of the ETM registers can be found in ARM's ETM_ARCHITECTURE_SPEC document.

Table 12-9. ETM Register Summary

Offset	Name	Type	Description
0x00	ETM Control	R/W	See ETMv3.3 Architecture Specification
0x04	ETM Configuration Control	RO	See ETMv3.3 Architecture Specification
0x08	Trigger Event	WO [1]	See ETMv3.3 Architecture Specification
0x0C	ASIC Control	WO[1]	See ETMv3.3 Architecture Specification
0x10	ETM Status	R/W	See ETMv3.3 Architecture Specification
0x14	System Configuration	RO	See ETMv3.3 Architecture Specification
0x18	Trace Start/Stop Resource Control	WO[1]	See ETMv3.3 Architecture Specification
0x1C	Trace Enable Control 2	WO[1]	See ETMv3.3 Architecture Specification
0x20	TraceEnable	WO[1]	See ETMv3.3 Architecture Specification
0x24	TraceEnable Control 1	WO[1]	See ETMv3.3 Architecture Specification
0x30	ViewData Event	WO[1]	See ETMv3.3 Architecture Specification
0x34	ViewData Control 1	WO[1]	See ETMv3.3 Architecture Specification
0x38	ViewData Control 2	WO[1]	See ETMv3.3 Architecture Specification
0x3C	ViewData Control 3	WO[1]	See ETMv3.3 Architecture Specification
0x40-0x7C	Address Comparator Value 1-16	WO[1]	See ETMv3.3 Architecture Specification
0x80-0xBC	Address Access Type 1-16	WO[1]	See ETMv3.3 Architecture Specification
0xC0-0xFC	Data Comparator Value 1-16	WO[1]	See ETMv3.3 Architecture Specification
0x40-0x4F	Data Comparator Mask 1-16	WO[1]	See ETMv3.3 Architecture Specification

Table continues on the next page...

Table 12-9. ETM Register Summary (continued)

Offset	Name	Type	Description
0x50-0x53	Counter Reload Value 1-4	WO[1]	See ETMv3.3 Architecture Specification
0x54-0x57	Counter Enable 1-4	WO[1]	See ETMv3.3 Architecture Specification
0x58-0x5B	Counter Reload Event 1-4	WO[1]	See ETMv3.3 Architecture Specification
0x5C-0x5F	Counter Value 1-4	R/W	See ETMv3.3 Architecture Specification
0x60-0x65	Sequencer State Transition Event	WO[1]	See ETMv3.3 Architecture Specification
0x66	-	-	Reserved
0x67	Current Sequencer State	R/W	See ETMv3.3 Architecture Specification
0x68-0x6B	External Output Event 1-4	WO[1]	See ETMv3.3 Architecture Specification
0x6C-0x6E	Context ID Comparator Value	WO[1]	See ETMv3.3 Architecture Specification
0x6F	Context ID Comparator Mask	WO[1]	See ETMv3.3 Architecture Specification
0x70-0x77	Implementation specific	WO[1]	See ETMv3.3 Architecture Specification
0x78	Synchronization Frequency	WO[1]	See ETMv3.3 Architecture Specification
0x79	ETM ID	RO	See ETMv3.3 Architecture Specification
0x7A	Configuration Code Extension	RO	See ETMv3.3 Architecture Specification
0x7B	Extended External Input Selection	WO[1]	See ETMv3.3 Architecture Specification
0x7C	Trace Start/Stop Embedded ICE Control	WO[1]	See ETMv3.3 Architecture Specification
0x7D	Embedded ICE Behavior Control	WO[1]	See ETMv3.3 Architecture Specification
0x7E-0x7F	-	-	Reserved
0x080	CoreSight Trace ID	R/W	See ETMv3.3 Architecture Specification
0x81-0xBF	-	-	Reserved
0xC0	OS Lock Access	WO	See ETMv3.3 Architecture Specification
0xC1	OS Lock Status	RO	See ETMv3.3 Architecture Specification
0xC2	OS Save/Restore	R/W	See ETMv3.3 Architecture Specification

Table continues on the next page...

Table 12-9. ETM Register Summary (continued)

Offset	Name	Type	Description
0xC3-0xCF	-	-	Reserved
0x380- 0x3BF	Integration registers	-	Reserved for Implementation-defined topology detection and integration registers.
0xF00	Integration Mode Control	R/W	See ETMv3.3 Architecture Specification
0xFA0	Claim Tag Set	R/W	See ETMv3.3 Architecture Specification
0xFA4	Claim Tag Clear	R/W	See ETMv3.3 Architecture Specification
0xFB0	Lock Access	WO	See ETMv3.3 Architecture Specification
0xFB4	Lock Status	RO	See ETMv3.3 Architecture Specification
0xFB8	Authentication Status	RO	See ETMv3.3 Architecture Specification
0xFC8	Device Configuration	RO	See ETMv3.3 Architecture Specification
0xFCC	Device Type	RO	See ETMv3.3 Architecture Specification
0xFD0	Peripheral ID4	RO	See ETMv3.3 Architecture Specification
0xFD4	Peripheral ID5	RO	See ETMv3.3 Architecture Specification
0xFD8	Peripheral ID6	RO	See ETMv3.3 Architecture Specification
0xFDC	Peripheral ID7	RO	See ETMv3.3 Architecture Specification
0xFE0	Peripheral ID0	RO	See ETMv3.3 Architecture Specification
0xFE4	Peripheral ID1	RO	See ETMv3.3 Architecture Specification
0xFE8	Peripheral ID2	RO	See ETMv3.3 Architecture Specification
0xFEC	Peripheral ID3	RO	See ETMv3.3 Architecture Specification
0xFF0	Component ID0	RO	See ETMv3.3 Architecture Specification
0xFF4	Component ID1	RO	See ETMv3.3 Architecture Specification
0xFF8	Component ID2	RO	See ETMv3.3 Architecture Specification
0xFFC	Component ID3	RO	See ETMv3.3 Architecture Specification

1. In ETMv3.1 and later, register is read/write if bit [11] of the ETM Configuration Code Extension Register (0x7A) is set to b1.

12.2.1.6 CSCTI Register Summary

The CSCTI registers are described in the table below. A more detailed description of the CSCTI registers can be found in ARM's CoreSight_DK_TRM document.

The table describes the location by an offset.

Table 12-10. CSCTI Register Summary

Offset	Name	Type	Width	Reset value	Description
0x000	CTICONTROL	R/W	1	0x0	See ARM CoreSight_DK_TRM
0x010	CTIINTACK	WO	8	-	See ARM CoreSight_DK_TRM
0x014	CTIAPPSET	R/W	4	0x0	See ARM CoreSight_DK_TRM
0x018	CTIAPPCLEAR	WO	?	0x0	See ARM CoreSight_DK_TRM
0x01C	CTIAPPPULSE	WO	4	0x0	See ARM CoreSight_DK_TRM
0x020-0x03C	CTIINEN	R/W	4	0x00	See ARM CoreSight_DK_TRM
0x0A0-0x0BC	CTIOUTEN	R/W	4	0x00	See ARM CoreSight_DK_TRM
0x130	CTITRIGINSTATUS	RO	8	-	See ARM CoreSight_DK_TRM
0x134	CTITRIGOUTSTATUS	RO	8	0x00	See ARM CoreSight_DK_TRM
0x138	CTICHINSTATUS	RO	4	-	See ARM CoreSight_DK_TRM
0x13C	CTICHOUTSTATUS	RO	4	0x0	See ARM CoreSight_DK_TRM
0x140	Channel gate	R/W	4	0xF	See ARM CoreSight_DK_TRM
0x144	External multiplexer control	R/W	8	0x00	See ARM CoreSight_DK_TRM
0xEDC	ITCHINACK	WO	4	0x0	See ARM CoreSight_DK_TRM
0xEE0	ITTRIGINACK	WO	8	0x00	See ARM CoreSight_DK_TRM
0xEE4	ITCHOUT	WO	4	0x0	See ARM CoreSight_DK_TRM
0xEE8	ITTRIGOUT	WO	8	0x00	See ARM CoreSight_DK_TRM
0xEEC	ITCHOUTACK	RO	4	0x0	See ARM CoreSight_DK_TRM
0xEF0	ITTRIGOUTACK	RO	8	0x00	See ARM CoreSight_DK_TRM
0xEF4	ITCHIN	RO	4	0x0	See ARM CoreSight_DK_TRM
0xEF8	ITTRIGIN	RO	8	0x00	See ARM CoreSight_DK_TRM
0xEFC-0xF7C	-	-	-	-	See ARM CoreSight_DK_TRM
0xF00	ITCTRL	R/W	1	0x0	See ARM CoreSight_DK_TRM
0xFA0	Claim Tag Set	R/W	4	0xF	See ARM CoreSight_DK_TRM
0xFA4	Claim Tag Clear	R/W	4	0x0	See ARM CoreSight_DK_TRM
0xFB0	Lock Access Register	WO	32	-	See ARM CoreSight_DK_TRM
0xFB4	Lock Status Register	RO	2	0x3	See ARM CoreSight_DK_TRM
0xFB8	Authentication Status	RO	4	0xA	See ARM CoreSight_DK_TRM
0xFC0-0xFC4	-	-	-	-	See ARM CoreSight_DK_TRM
0xFC8	Device ID	RO	20	0x40800	See ARM CoreSight_DK_TRM

Table continues on the next page...

Table 12-10. CSCTI Register Summary (continued)

Offset	Name	Type	Width	Reset value	Description
0xFCC	Device Type Identifier	RO	8	0x14	See ARM CoreSight_DK_TRM
0xFD0	PeripheralID4	RO	8	0x04	See ARM CoreSight_DK_TRM
0xFD4	PeripheralID5	-	-	-	See ARM CoreSight_DK_TRM
0xFD8	PeripheralID6	-	-	-	See ARM CoreSight_DK_TRM
0xFDC	PeripheralID7	-	-	-	See ARM CoreSight_DK_TRM
0xFE0	PeripheralID0	RO	8	0x06	See ARM CoreSight_DK_TRM
0xFE4	PeripheralID1	RO	8	0xB9	See ARM CoreSight_DK_TRM
0xFE8	PeripheralID2	RO	8	0x0B	See ARM CoreSight_DK_TRM
0xFEC	PeripheralID3	RO	8	0x00	See ARM CoreSight_DK_TRM
0xFF0	Component ID0	RO	8	0x0D	See ARM CoreSight_DK_TRM
0xFF4	Component ID1	RO	8	0x90	See ARM CoreSight_DK_TRM
0xFF8	Component ID2	RO	8	0x05	See ARM CoreSight_DK_TRM
0xFFC	Component ID3	RO	8	0xB1	See ARM CoreSight_DK_TRM

12.2.1.7 TPIU Register Summary

The TPIU (Trace Port Interface Unit) registers are described in the following table. A more detailed description of the TPIU registers can be found in ARM's CoreSight_DK_TRM document. These registers can be accessed via the APB port and are mem map accessible.

Table 12-11. TPIU Register Summary

Offset	Name	Type	Width	Reset value	Description
0x000	Supported port sizes	RO	32	0xFFFFFFFF	See ARM CoreSight_DK_TRM
0x004	Current port size	R/W	32	0x00000001	See ARM CoreSight_DK_TRM
0x100	Supported trigger modes	RO	18	0x11F	See ARM CoreSight_DK_TRM
0x104	Trigger counter value	R/W	8	0x00	See ARM CoreSight_DK_TRM
0x108	Trigger multiplier	R/W	5	0x00	See ARM CoreSight_DK_TRM

Table continues on the next page...

Table 12-11. TPIU Register Summary (continued)

Offset	Name	Type	Width	Reset value	Description
0x200	Supported test pattern/ modes	RO	18	0x3000F	See ARM CoreSight_DK_TRM
0x204	Current test pattern/ mode	RO	18	0x00000	See ARM CoreSight_DK_TRM
0x208	Test pattern repeat counter	R/W	8	0x00	See ARM CoreSight_DK_TRM
0x300	Formatter flush and status	RO	3	0x6	See ARM CoreSight_DK_TRM
0x304	Fomatter flush and control	R/W	14	0x1000	See ARM CoreSight_DK_TRM
0x308	Formatter synchroniz ation counter	R/W	12	0x040	See ARM CoreSight_DK_TRM
0x400	EXTCTL In Port	RO	8	Undefined	See ARM CoreSight_DK_TRM
0x408	EXTCTL Out Port	R/W	8	0x00	See ARM CoreSight_DK_TRM
0xEE4	Integration Register, ITTRFLIN ACK	WO	2	-	See ARM CoreSight_DK_TRM
0xEE8	Integration Register, ITTRFLIN	WO	2	Undefined	See ARM CoreSight_DK_TRM
0xEEC	Integration Register, ITATBDAT A0	WO	5	Undefined	See ARM CoreSight_DK_TRM
0xEF0	Integration Register, ITATBCT R2	WO	2	-	See ARM CoreSight_DK_TRM
0xEF4	Integration Register, ITATBCT R1	RO	7	Undefined	See ARM CoreSight_DK_TRM

Table continues on the next page...

Table 12-11. TPIU Register Summary (continued)

Offset	Name	Type	Width	Reset value	Description
0xEF8	Integration Register, ITATBCTR0	RO	10	Undefined	See ARM CoreSight_DK_TRM
0xF00	Integration Register, ITATBCTR0	R/W	1	0x0	See ARM CoreSight_DK_TRM
0xFA0	Claim Tag Set Register	R/W	4	0xF	See ARM CoreSight_DK_TRM
0xFA4	Claim Tag Clear Register	R/W	4	0x0	See ARM CoreSight_DK_TRM
0xFB0	Lock Access Register	WO	32	-	See ARM CoreSight_DK_TRM
0xFB4	Lock Status Register	RO	3	0x3	See ARM CoreSight_DK_TRM
0xFB8	Authentication Status Register	RO	8	0x00	See ARM CoreSight_DK_TRM
0xFC8	Device ID	RO	32	0x00	See ARM CoreSight_DK_TRM
0xFCC	Device Type Identifier Register	RO	8	0x21	See ARM CoreSight_DK_TRM
0xFD0	Peripheral ID4	RO	8	0x04	See ARM CoreSight_DK_TRM
0xFD4	Peripheral ID5	RO	8	0x00 (reserved)	See ARM CoreSight_DK_TRM
0xFD*	Peripheral ID6	RO	8	0x00 (reserved)	See ARM CoreSight_DK_TRM
0xFDC	Peripheral ID7	RO	8	0x00 (reserved)	See ARM CoreSight_DK_TRM
0xFE0	Peripheral ID0	RO	8	0x07	See ARM CoreSight_DK_TRM
0xFE4	Peripheral ID1	RO	8	0xB9	See ARM CoreSight_DK_TRM
0xFE8	Peripheral ID2	RO	8	0x0B	See ARM CoreSight_DK_TRM

Table continues on the next page...

Table 12-11. TPIU Register Summary (continued)

Offset	Name	Type	Width	Reset value	Description
0xFEC	Peripheral ID3	RO	8	0x00	See ARM CoreSight_DK_TRM
0xFF0	Component ID0	RO	8	0x0D	See ARM CoreSight_DK_TRM
0xFF4	Component ID1	RO	8	0x90	See ARM CoreSight_DK_TRM
0xFF8	Component ID2	RO	8	0x05	See ARM CoreSight_DK_TRM
0xFFC	Component ID3	RO	8	0xB1	See ARM CoreSight_DK_TRM

12.2.2 Clocks

The list below describes the Debug clocks within the ARM platform.

- ATCLK - This is the AMBA Trace Bus (ATB) clock. This clock is also used to clock the on-platform CTICLK and on-platform CTMCLK. ATCLK needs to be synchronous (equivalent or faster) to PCLKDBG.
- CTICLK - This is the main clock for the CTI block. The current ARM configuration requires this clock to be synchronous to the on-platform CTM clock.
- PCLKDBG - This is the Debug APB clock. It must be synchronous (equivalent or slower) to the ATCLK.
- TRACECLKIN - This is the Trace Port Interface Unit trace clock input. Since the trace port pads are limited to 133 mhz, TRACECLKIN is pad limited to less than or equal to 266 mhz.
- TRACECLK - This is the clock for the external trace port. It's a DDR clock and runs at 1/2 the speed of TRACECLKIN.

The platform control block generates the debug clocks from the arm clock. The requirement above states that the PCLKDBG needs to be synchronous to the ATCLK. When configuring the divide by settings, make sure this requirement is followed. PCLKDBG is a divide by 2 of ATCLK.

12.2.3 Reset

All debug resets are derived from JTAG_TRST_B and POR. These two resets go into the DAP_SYS to be combined and de-asserted synchronously to the DAPCLKao (always on clock). They also go into the ARM platform, being synchronously de-asserted inside the plat_cntrl block, generating resets for the debug logic inside the platform. Debug resets

that do not require synchronous de-assertion are routed directly to their destination, such as por for the Core and ETM. The figure below shows a diagram of the ARM platform debug reset strategy.

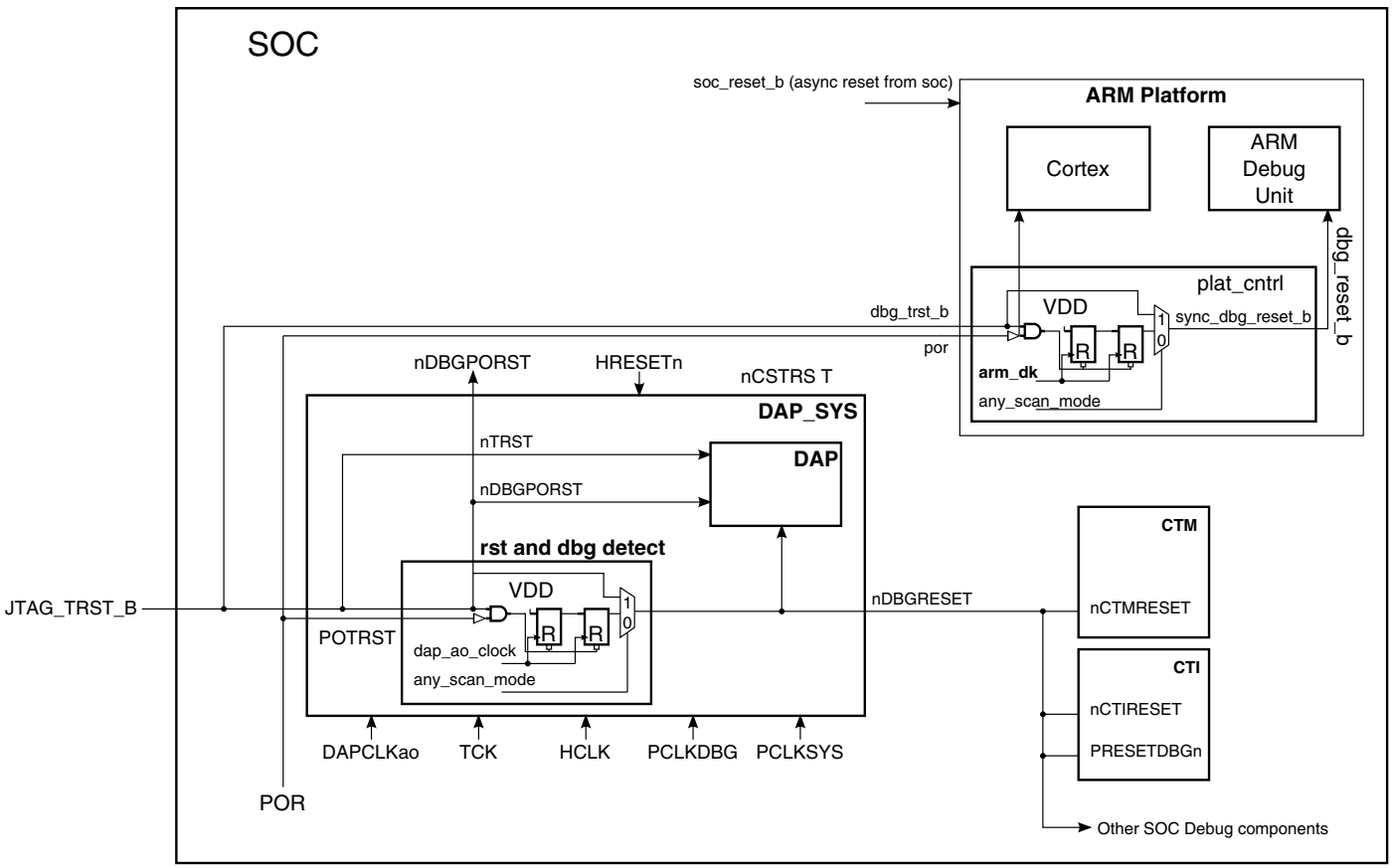


Figure 12-8. ARM Debug Reset Strategy

Chapter 13
Multi-Layer AHB Crossbar Switch (AHBMAX)

13.1 Overview

This section provides an overview of the Multi-Layer AHB Crossbar Switch (AHBMAX). The purpose of the AHBMAX is to concurrently support up to four simultaneous connections between master ports and slave ports. The AHBMAX supports a 32-bit address bus width and a 32-bit data bus width at all master and slave ports. A simplified block diagram is shown in Figure 13-1.

NOTE

The AHBMAX implements a 7 master by 4 slave configuration.

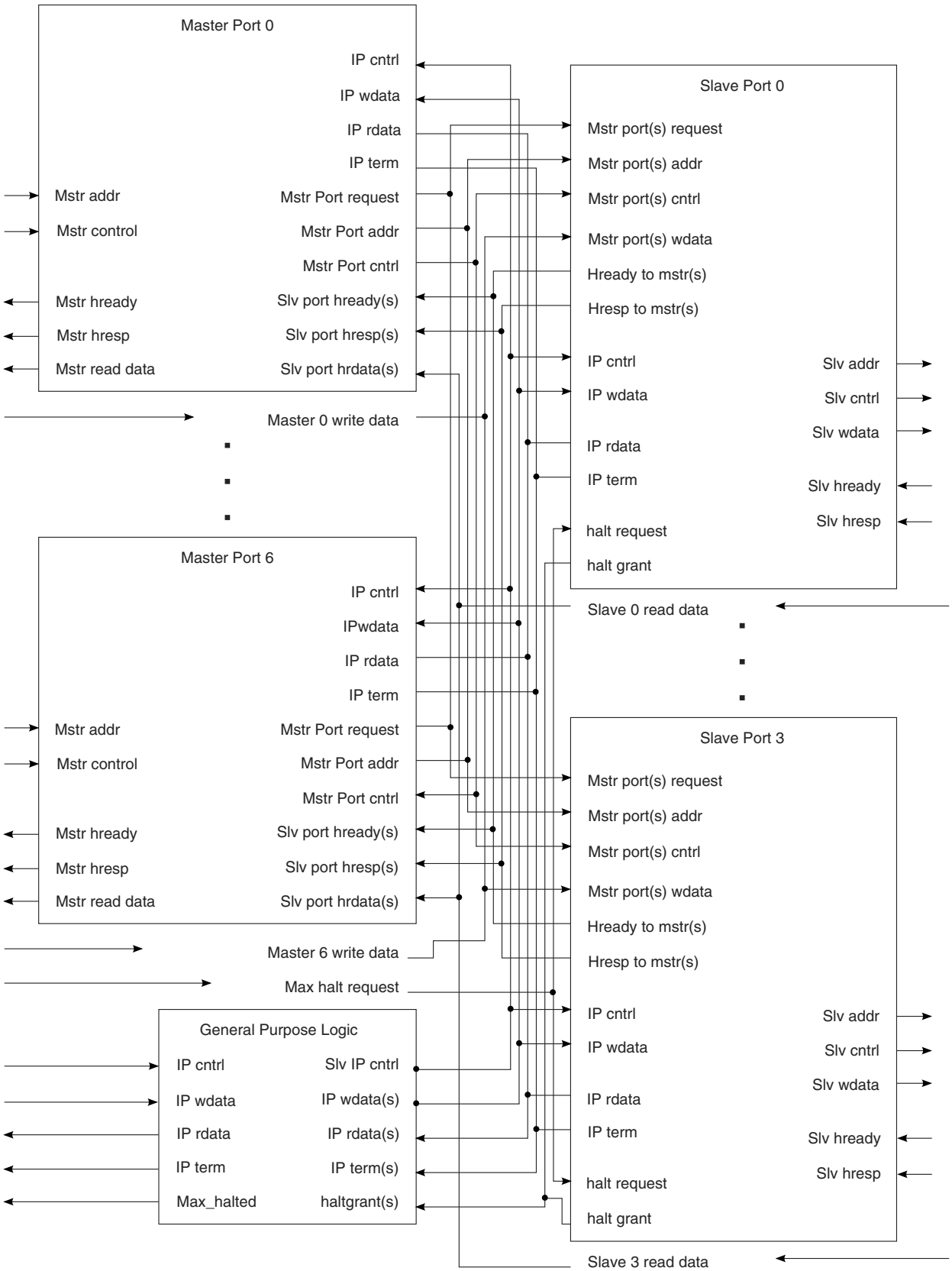


Figure 13-1. AHBMAX Block Diagram

13.2 System Connectivity

This section provides information on system connectivity of AHBMAX.

Table 13-1. Masters of AHBMAX

M0	DAP, SAHARA, SSC
M1	RTIC
M2	Pre-DMA port of SDMA
M3	eSDHC-3
M4	eSDHC-4, P-ATA
M5	eSDHC-1, eSDHC-2
M6	ARB3 port of EMI

Table 13-2. Slaves of AHBMAX

S0	DAP, IPU, OpenVG, GPU3D, SATA
S1	IPMUX2
S2	IPMUX1
s3	Output to EMI (M4IF) port #2

13.3 Features

The AHBMAX has the ability to gain control of all the slave ports and prevent any masters from making accesses to the slave ports. This feature is useful when the user wishes to turn off the clocks to the system and needs to ensure that no bus activity will be interrupted.

The AHBMAX can put each slave port into a low power park mode so that slave port will not dissipate any power transitioning address, control or data signals when not being actively accessed by a master port.

Each slave port can also support multiple master priority schemes. Each slave port has a hardware input which selects the master priority scheme so the user can dynamically change master priority levels on a slave port by slave port basis.

The AHBMAX will allow for concurrent transactions to occur from any master port to any slave port. It is possible for four master ports and all slave ports to be in use at the same time as a result of independent master requests. If a slave port is simultaneously

requested by more than one master port, arbitration logic will select the higher priority master and grant it ownership of the slave port. All other masters requesting that slave port will stalled until the higher priority master completes its transactions.

13.3.1 Limitations

The AHBMAX routes bus transactions initiated on the master ports to the appropriate slave ports. There is no provision included to route transactions initiated on the slave ports to other slave ports or to master ports. Simply put, the slave ports do not support the bus request/bus grant protocol, the AHBMAX assumes it is the sole master of each slave port.

Because the AHBMAX does not support the bus request/bus grant protocol, if multiple masters are to be connected to a single master port an external arbiter will need to be used. In the case of a single master connecting to a master port the single master's bus grant signal must be tied off in the asserted state.

Each master and slave port is fully AHB-Lite + AMBA V6 extensions compliant. The ports are not fully AHB compliant because the AHBMAX does not support SPLITs or RETRYs.

13.3.2 General Operation

When a master makes an access to the AHBMAX the access will be immediately taken by the AHBMAX. If the targeted slave port of the access is available then the access will be immediately presented on the slave port. It is possible to make single clock (zero wait state) accesses through the AHBMAX. If the targeted slave port of the access is busy or parked on a different master port the requesting master will simply see wait states inserted (hready held negated) until the targeted slave port can service the master's request. The latency in servicing the request will depend on each master's priority level and the responding peripheral's access time.

Because the AHBMAX appears to be just another slave to the master device, the master device will have no knowledge of whether or not it actually owns the slave port it is targeting. While the master does not have control of the slave port it is targeting it will simply be wait stated.

A master will be given control of the targeted slave port only after a previous access to a different slave port has completed, regardless of its priority on the newly targeted slave port. This prevents deadlock from occurring when a master has an outstanding request to

one slave port that has a long response time, has a pending access to a different slave port, and a lower priority master is also making a request to the same slave port as the pending access of the higher priority master.

Once the master has control of the slave port it is targeting the master will remain in control of that slave port until it gives up the slave port by running an IDLE cycle or by leaving that slave port for its next access. The master could also lose control of the slave port if another higher priority master makes a request to the slave port; however, if the master is running a locked or fixed length burst transfer it will retain control of the slave port until that transfer is completed. Based on the AULB bit in the MGPCR (Master General Purpose Control Register) the master will either retain control of the slave port when doing undefined length incrementing burst transfers or will lose the bus to a higher priority master.

The AHBMAX will terminate all master IDLE transfers (as opposed to allowing the termination to come from one of the slave busses). Additionally, when no master is requesting access to a slave port the AHBMAX will drive IDLE transfers onto the slave bus, even though a default master may be granted access to the slave port. When the AHBMAX is controlling the slave bus (i.e. during low power park or halt mode) the hmaster field will indicate 4'b0000.

When a slave bus is being IDLEd by the AHBMAX it can park the slave port on the master port indicated by the PARK bits in the AHBMAX_SGPCR (Slave General Purpose Control Register). This can be done in an attempt to save the initial clock of arbitration delay that would otherwise be seen if the master had to arbitrate to gain control of the slave port. The slave port can also be put into low power park mode in attempt to save power.

13.4 AHBMAX Interface Signals

This section provides information on AHBMAX interface signals, including AHB master and slave interface signals as well as IP bus interface signals.

13.4.1 AHBMAX Signal Descriptions

Please reference the AMBA Specification Rev 2.0 for a description of the AHB signals in the AHBMAX and the IP Bus Specification Rev 2.0 for a description of the IP Bus signals in the AHBMAX.

13.4.1.1 max_halt_request

This input signal is a request to halt all slave port bus activity (run AHBMAX originated IDLE cycles on each slave port bus, blocking all master port accesses). This signal can be used to gracefully shut down the AHBMAX so the system clock can be stopped for low power mode. This signal is captured by a flop inside the AHBMAX before use.

Once the AHBMAX is halted it will remain halted until max_halt_request is negated.

13.4.1.2 max_halted

This output is asserted once the AHBMAX is in control and running IDLE cycles on each slave port.

13.5 Coherency

Since the content of the registers has a real time effect on the operation of the AHBMAX it is important for the user to understand that any register modifications take effect as soon as the register is written. The values of the registers do not track with slave port related AHB accesses but instead track only with IP bus accesses.

The exception to this rule are the AULB bits in the AHBMAX_MGPCR*n*. The update of these bits is only recognized when the master on that master port runs an IDLE cycle, even though the IP bus cycle to write them will have long since terminated successfully. If the AULB bits in the AHBMAX_MGPCR*n* are written in between two burst accesses the new AULB encodings will not take effect until an IDLE cycle has been initiated by the master on that master port.

13.6 Detailed Functional Description

This section describes the functionality of the AHBMAX in greater detail.

13.6.1 Arbitration

The AHBMAX supports two arbitration schemes; a simple fixed-priority comparison algorithm, and a simple round-robin fairness algorithm. The arbitration scheme is independently programmable for each slave port.

13.6.1.1 Arbitration During Undefined Length Bursts

Arbitration points during an undefined length burst are defined by the current master's AHBMAX_MGPCR*n*[AULB] field setting. When a defined length is imposed on the burst via the AULB bits the undefined length burst will be treated as a single or series of single back to back fixed length burst accesses.

Example: A master runs an undefined length burst and the AULB bits in the AHBMAX_MGPCR*n* indicate arbitration will occur after the fourth beat of the burst. The master runs two sequential beats and then starts what will be an 12 beat undefined length burst access to a new address within the same slave port region as the previous access. The AHBMAX will not allow an arbitration point until the fourth overall access (second beat of the second burst). At that point all remaining accesses will be open for arbitration until the master loses control of the slave port.

Assume the master loses control of the slave port after the fifth beat of the second burst. Once the master regains control of the slave port no arbitration point will be available until after the master has run four more beats of its burst. After the fourth beat of the (now continued) burst (ninth beat of the second burst from the master's perspective) is taken all beats of the burst will once again be open for arbitration until the master loses control of the slave port.

Assume the master again loses control of the slave port on the fifth beat of the third (now continued) burst (10th beat of the second burst from the master's perspective). Once the master regains control of the slave port it will be allowed to complete its final two beats of its burst without facing arbitration.

Note that fixed length burst accesses will not be affected by the AULB bits. All fixed length burst accesses will lock out arbitration until the last beat of the fixed length burst.

13.6.1.2 Fixed Priority Operation

When operating in fixed-priority mode, each master is assigned a unique priority level in the AHBMAX_MPR*n* (Master Priority Register). If two masters both request access to a slave port the master with the highest priority in the selected priority register will gain control over the slave port.

Any time a master makes a request to a slave port the slave port checks to see if the new requesting master's priority level is higher than that of the master that currently has control over the slave port (unless the slave port is in a parked state). The slave port does an arbitration check at every clock edge to ensure that the proper master (if any) has control of the slave port.

If the new requesting master's priority level is higher than that of the master that currently has control of the slave port the new requesting master will be granted control over the slave port at the next clock edge. The exception to this rule is if the master that currently has control over the slave port is running a fixed length burst transfer or a locked transfer. In this case the new requesting master will have to wait until the end of the burst transfer or locked transfer before it will be granted control of the slave port. If the master is running an undefined length burst transfer the new requesting master must wait until an arbitration point for the undefined length burst transfer before it will be granted control of the slave port. Arbitration points for an undefined length burst are defined in the MGPCR for each master.

If the new requesting master's priority level is lower than that of the master that currently has control of the slave port the new requesting master will be forced to wait until the master that currently has control of the slave port either runs an IDLE cycle or runs a non IDLE cycle to a location other than the current slave port.

13.6.1.3 Round-Robin Priority Operation

When operating in round-robin mode, each master is assigned a relative priority based on the master number. This relative priority is compared to the ID of the last master to perform a transfer on the slave bus. The highest priority requesting master will become owner of the slave bus as the next transfer boundary (accounting for locked and fixed-length burst transfers). Priority is based on how far ahead the ID of the requesting master is to the ID of the last master (ID is defined by master port number, not the hmaster field).

Once granted access to a slave port, a master may perform as many transfers as desired to that port until another master makes a request to the same slave port. The next master in line will be granted access to the slave port at the next assertion of slave "n" hready, or possibly on the next clock cycle if the current master has no pending access request.

As an example of arbitration in round-robin mode, assume the AHBMAX is implemented with master ports 0, 1, 2, 3, 4 and 5. If the last master of the slave port was master 1, and master 0, 4 and 5 make simultaneous requests, (master ports 2 and 3 make no requests), they will be serviced in the order 4, 5 and then 0.

Parking may still be used in a round-robin mode, but will not affect the round-robin pointer unless the parked master actually performs a transfer. Handoff will occur to the next master in line after one cycle of arbitration. If the slave port is put into low power park mode the round-robin pointer will be reset to point at master port 0, giving it the highest priority.

13.6.2 Priority Assignment

Each master port needs to be assigned a unique 3 bit priority level. If an attempt is made to program multiple master ports with the same priority level within a register (MPR) the AHBMAX will respond with an error and the registers will not be updated.

13.6.3 Master Port Functionality

13.6.3.1 Master Port General Information

Each master port consists of two decoders, a capture unit, a register slice, a mux and a small state machine.

The first decoder is used to decode the haddr and control signals coming directly from the master, telling the state machine where the master's next access will be and if it is in fact a legal access. The second decoder gets its input from the capture unit, so it may be looking directly at the signals coming from the master or it may be looking at captured signals coming from the master, depending entirely on the state of the targeted slave port. The second decoder is then used to generate the access requests that go to the slave ports.

The capture unit is used to capture the address and control information coming from the master in the event that the targeted slave port cannot immediately service the master. The capture unit is controlled by outputs from the state machine which tell it to either pass through the original master signals or the captured signals.

The register slice contains the registers associated with the specific master port. The registers have a quasi-IP bus interface at this level for reads and writes and the outputs feed directly into the state machine.

The mux is used simply to select which slave's read data is sent back to the master. The mux is controlled by the state machine.

The state machine controls all aspects of the master port. It knows which slave port the master wants to make a request to and controls when that request is made. It also has knowledge of each slave port, knowing whether or not the slave port is ready to accept an access from the master port. This will determine whether or not the master may immediately have its request taken by the slave port or whether the master port will have to capture the master's request and queue it at the slave port boundary.

For a block diagram of a master port see Figure 13-2.

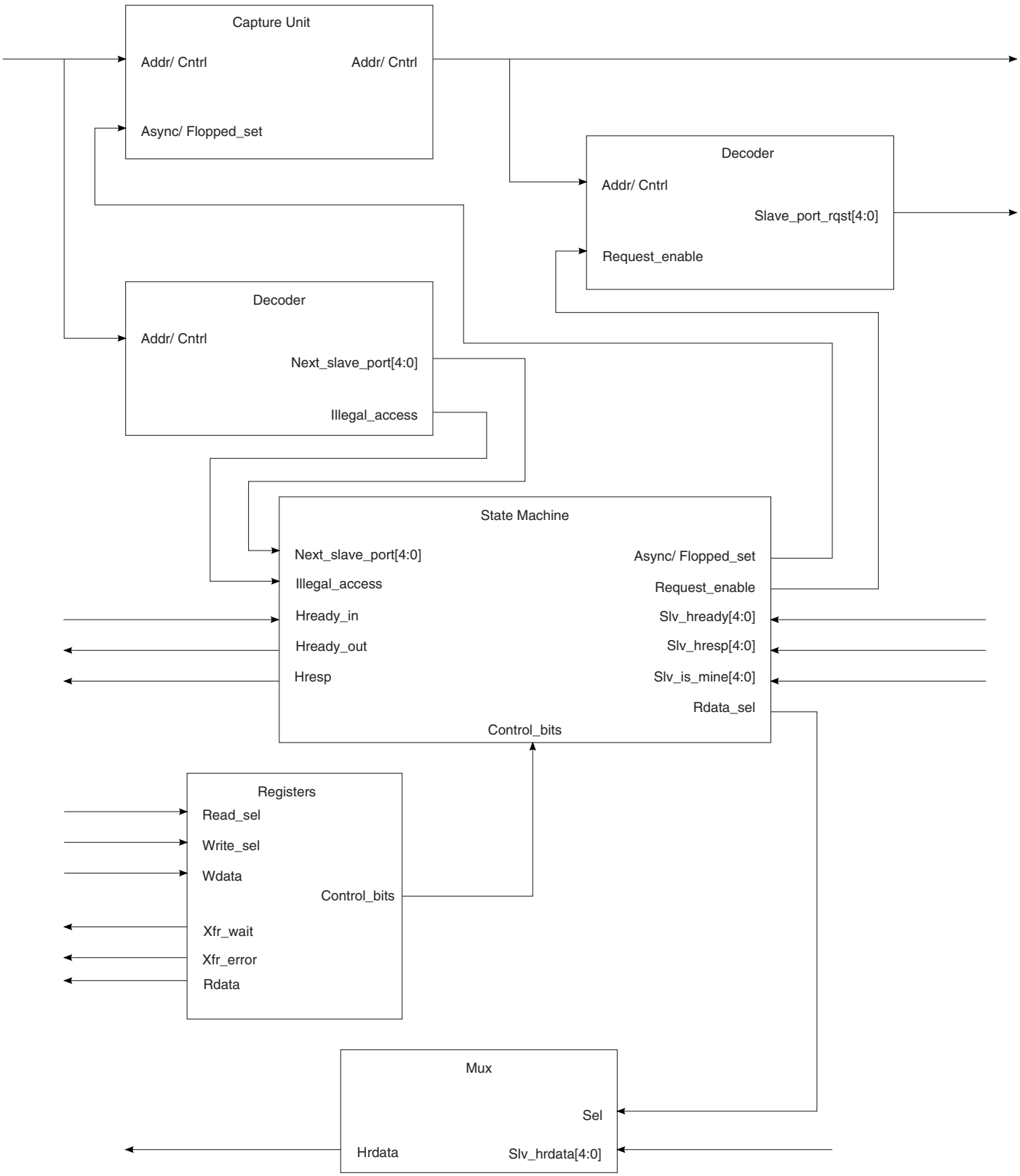


Figure 13-2. AHBMAX Master Port Block Diagram

13.6.3.2 Master Port Decoders

The decoders are very simple as they ensure an access request is allowed to be made and that the slave port targeted is actually present in the design. The decoders feeding the state machine are always enabled. The decoders that select the slave are enabled only when the master port controlling state machine wants to make a request to a slave port. This is necessary so that if a master port is making an access to a slave port and is being wait stated, and its next access is to a different slave port, the request to the second slave port can be held off until the access to the first slave port is terminated.

The decoders also output a "hole decode" or illegal access signal which tells the state machine that the master is trying to access a slave port that does not exist.

13.6.3.3 Master Port Capture Unit

The capture unit simply captures the state of the master's address and control signals if the AHBMAX cannot immediately pass the master's request through to the proper slave port. The capture unit consists of a set of flops and a mux which selects either the asynchronous path from address and control or the flopped (captured) address and control information.

13.6.3.4 Master Port Registers

The registers in the master port are only those registers associated with this particular master port. The read and write interface for the registers is a quasi-IP bus interface. It is not a full IP bus interface at this level because not all the IP bus signals are routed this deep in the design.

There is a register control block at the same level of the master port and slave port instantiations in the AHBMAX. This control block ensures that all accesses are 32-bit supervisor accesses before passing them on to the master ports.

The register outputs are connected directly to the state machine.

13.6.3.5 Master Port State Machine

13.6.3.5.1 Master Port State Machine States

The master side state machine's main function is to monitor the activities of the master port. The state machine has six states: busy, idle, stalled, steady state, first cycle error response and second cycle error response.

The busy state is used when the master runs a BUSY cycle to the master port. The master port maintains its request to the slave port if it currently owns the slave port; however, if it loses control of the slave port it will no longer maintain its request. If the master port loses control of the slave port it will not be allowed to make another request to the slave port until it runs a NSEQ or SEQ cycle.

The idle state is used when the master runs a valid IDLE cycle to the master port. The master port makes no requests to the slave ports (disables the slave port decoder) and terminates the IDLE cycle.

The stalled state is used when the master makes a request to a slave port that is not immediately ready to receive the request. In this case the state machine will direct the capture unit to send out the captured address and control signals and will enable the slave port decoder to indicate a pending request to the appropriate slave port.

The steady state is used when the master port and slave port are in fully asynchronous mode, making the AHBMAX completely transparent in the access. The state machine selects the appropriate slave's hresp0, hready and hrdata to pass back to the master.

The first cycle error response and second cycle error response states are self explanatory. The AHBMAX will respond with an error response to the master if the master tries to access an unimplemented memory location through the AHBMAX (i.e. a slave port that does not exist).

13.6.3.5.2 Master Port State Machine Slave Swapping

The design of the master side state machine is fairly straight forward. The one real decision to be made is how to handle the master moving from one slave port access to another slave port access. The approach that was taken was to minimize or eliminate, when possible, any "bubbles" that would get inserted into the access due to switching slave ports.

The state machine will not allow the master to request access to another slave port until the current access being made is terminated. This prevents a single master from owning two slave ports at the same time (the slave port it is currently accessing and the slave port it wishes to access next).

The state machine also maintains watch on the slave port the master is accessing as well as the slave port the master wishes to switch to. If the new slave port is parked on the master then the master will be able to make the switch without incurring any delays. The termination of the current access will also act as the launch of the new access on the new slave port. If the new slave port is not parked on the master then the master will incur a minimum one clock delay before it can launch its access on the new slave port.

This is the same for switching from the busy or idle state to actively accessing a slave port. If the slave port is parked on the master the state machine will go to the steady state and the access will begin immediately. If the slave port is not parked on the master (serving another master, parked on another master or in low power park mode) then the state machine will transition to the stalled state and at least a one clock penalty will be paid.

13.6.4 Slave Port Functionality

13.6.4.1 Slave Port General Information

Each slave port consists of a register slice, a bank of muxes and a state machine.

The register slice contains the registers associated with the specific slave port. The registers have a quasi-IP bus interface at this level for reads and writes and the outputs feed directly into the state machine.

The muxes are a series of 6 to 1 multiplexers that take in all the address, control and write data information from each of the master ports and then pass the correct master's signals to the slave port. The state machine controls all the muxes.

The state machine is where the main slave port arbitration occurs, it decides which master is in control of the slave port and which master will be in control of the slave port in the next bus cycle.

For a block diagram of a slave port see Figure 13-3.

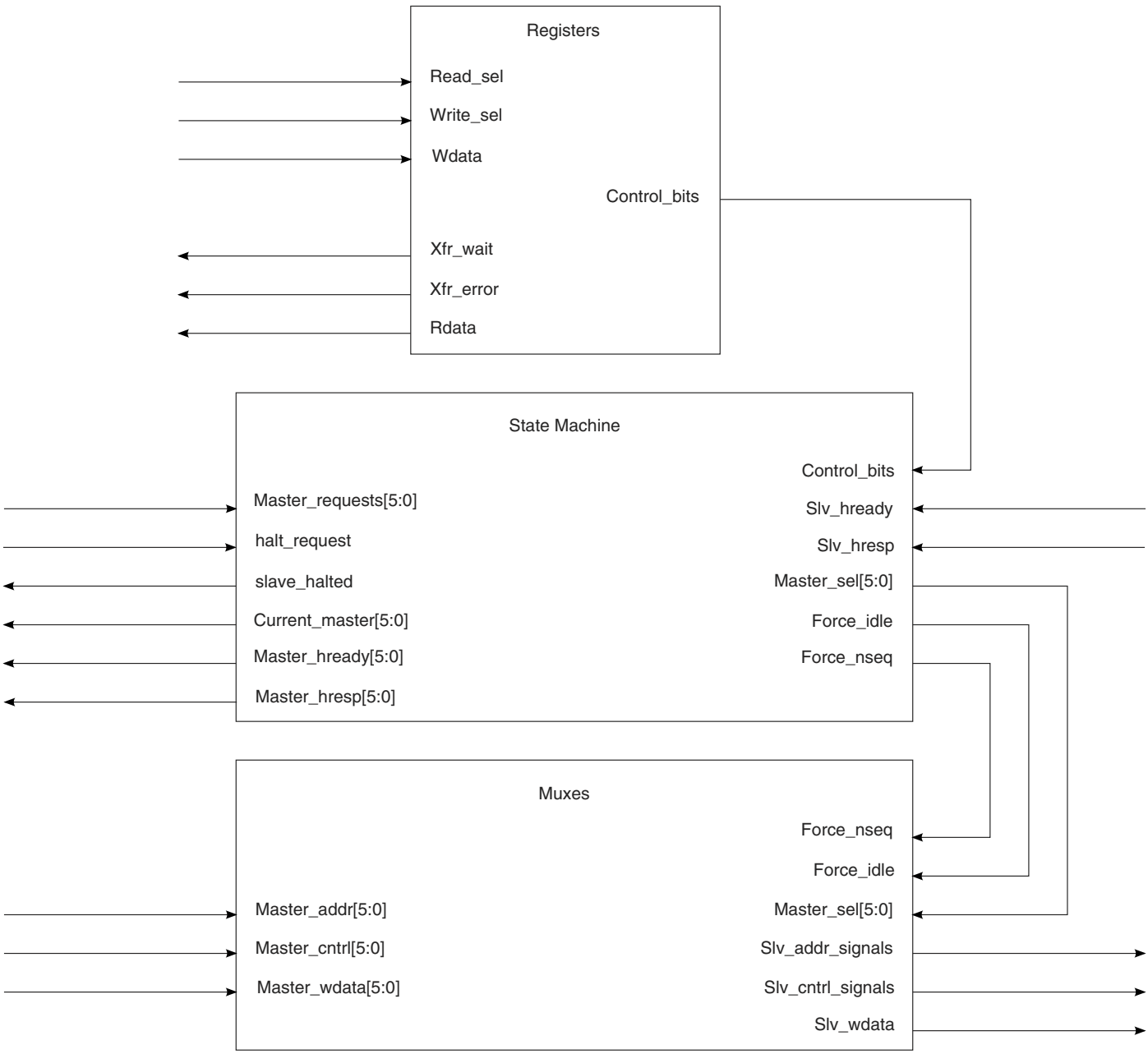


Figure 13-3. AHBMAX Slave Port Block Diagram

13.6.4.2 Slave Port Muxes

The block diagram (Figure 13-3) shows only one block for all the muxes. In reality that block instantiates many 6 to 1 muxes, one for each master-to-slave signal in fact. All the muxes are designed in an AND - OR fashion, so that if no master is selected the output of the muxes will be zero. (This is an important feature for low power park mode.)

The muxes also have an override signal which is used by the slave port to asynchronously force IDLE cycles onto the slave bus. When the state machine forces an IDLE cycle it zeros out htrans and hmastlock, making sure the slave bus sees a valid IDLE cycle being run by the AHBMAX.

The enable to the mux controlling htrans also contains an additional control signal from the state machine so that a NSEQ transaction can be forced. This is done any time the slave port switches masters to ensure that no IDLE-SEQ, BUSY-SEQ or NSEQ-SEQ transactions are seen on the slave port when they shouldn't be. If the state machine indicates to run both an IDLE and an NSEQ cycle, the IDLE directive will have priority.

NOTE

IDLE-SEQ is in fact an illegal access, but a possible scenario given the multi-master environment in the AHBMAX unless corrected by the AHBMAX.

13.6.4.3 Slave Port Registers

There is a register control block at the same level of the master port and slave port instantiations in the AHBMAX. This control block ensures that all accesses are 32-bit supervisor accesses before passing them on to the master and slave ports.

The registers in the slave port are only those registers associated with this particular slave port. The read and write interface for the registers is a quasi-IP bus interface. It is not a full IP bus interface at this level because not all the IP bus signals are routed this deep in the design.

The register outputs are connected directly to the slave state machine. The registers can be read from an unlimited number of times. The registers can only be written to as long as the RO bit is written to 0 in the AHBMAX_SGPCRn, once it is written to a 1 only a hardware reset will allow the registers to be written again.

13.6.4.4 Slave Port State Machine

13.6.4.4.1 Slave Port State Machine States

At the heart of the slave port is the state machine. The state machine is simplicity itself, requiring only four states - steady state, transition state, transition hold state and hold state. Either the slave port is owned by the same master it was in the last clock cycle (either by active use or by parking), it is transitioning to a new master (either for active use or parking), it is transitioning to a new master during wait states or it is being held on the same master pending a transition to a new master.

13.6.4.4.2 Slave Port State Machine Arbitration

The real work in the state machine is determining which master port will be in control of the slave port in the next clock cycle, the arbitration. Each master is programmed with a fixed 3 bit priority level. The AHBMAX uses these bits in determining priority levels when programmed for fixed priority mode of operation.

Arbitration always occurs on a clock edge, but only occurs on edges when a change in mastership will not violate AHB-Lite protocols. Valid arbitrations points include any clock cycle in which slave n hready is asserted (provide the master is not performing a burst or locked cycle) and any wait state in which the master owning the bus indicates a transfer type of IDLE (provided the master is not performing a locked cycle).

Since arbitration can occur on every clock cycle the slave port masks off all master requests if the current master is performing a locked transfer or a protected burst transfer, guaranteeing that no matter how low its priority level it will be allowed to finish its locked or protected portion of a burst sequence.

13.6.4.4.3 Slave Port State Machine Master Handoff

The only times the slave port will switch masters when programmed for fixed priority mode of operation is when a higher priority master makes a request or when the current master is the highest priority and it gives up the slave port by either running and IDLE cycle to the slave port or running a valid access to a location other than the slave port.

If the current master loses control of the slave port because a higher priority master takes it away the slave port will not incur any wasted cycles. The current master will get its current cycle terminated by the slave port at the same time the new master's address and control information will be recognized by the slave port. This will look like a seamless transition on the slave port.

If the current master is being wait stated when the higher priority master makes its request, then the current master will be allowed to make one more transaction on the slave bus before giving it up to the new master. Figure 13-4 illustrates the effect of a higher priority master taking control of the bus when the slave port is programmed for a fixed priority mode of operation.

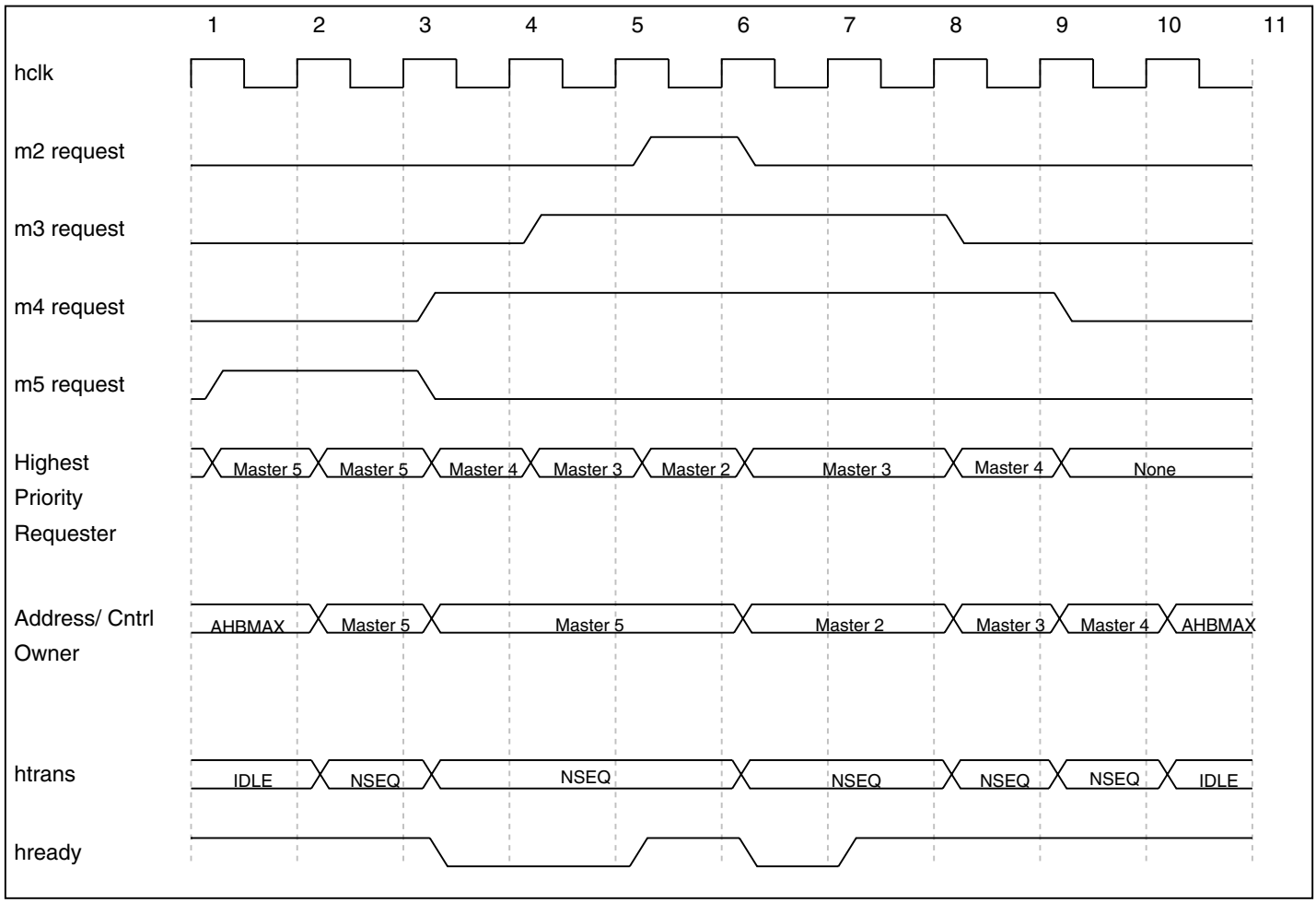


Figure 13-4. Low to High Priority Mastership Change

If the current master is the highest priority master and it gives up the slave port by running an IDLE cycle or by running a valid cycle to another location other than the slave port the next highest priority master will gain control of the slave port. If the current access incurs any wait states then the transition will be seamless and no bandwidth will be lost; however, if the current transaction is terminated without wait states then one IDLE cycle will be forced onto the slave bus by the AHBMAX before the new master will be able to take control of the slave port. If no other master is requesting the bus then IDLE cycles will be run by the AHBMAX but no bandwidth will truly be lost since no master is making a request.

Figure 13-5 illustrates the effect of a higher priority master giving up control of the bus.

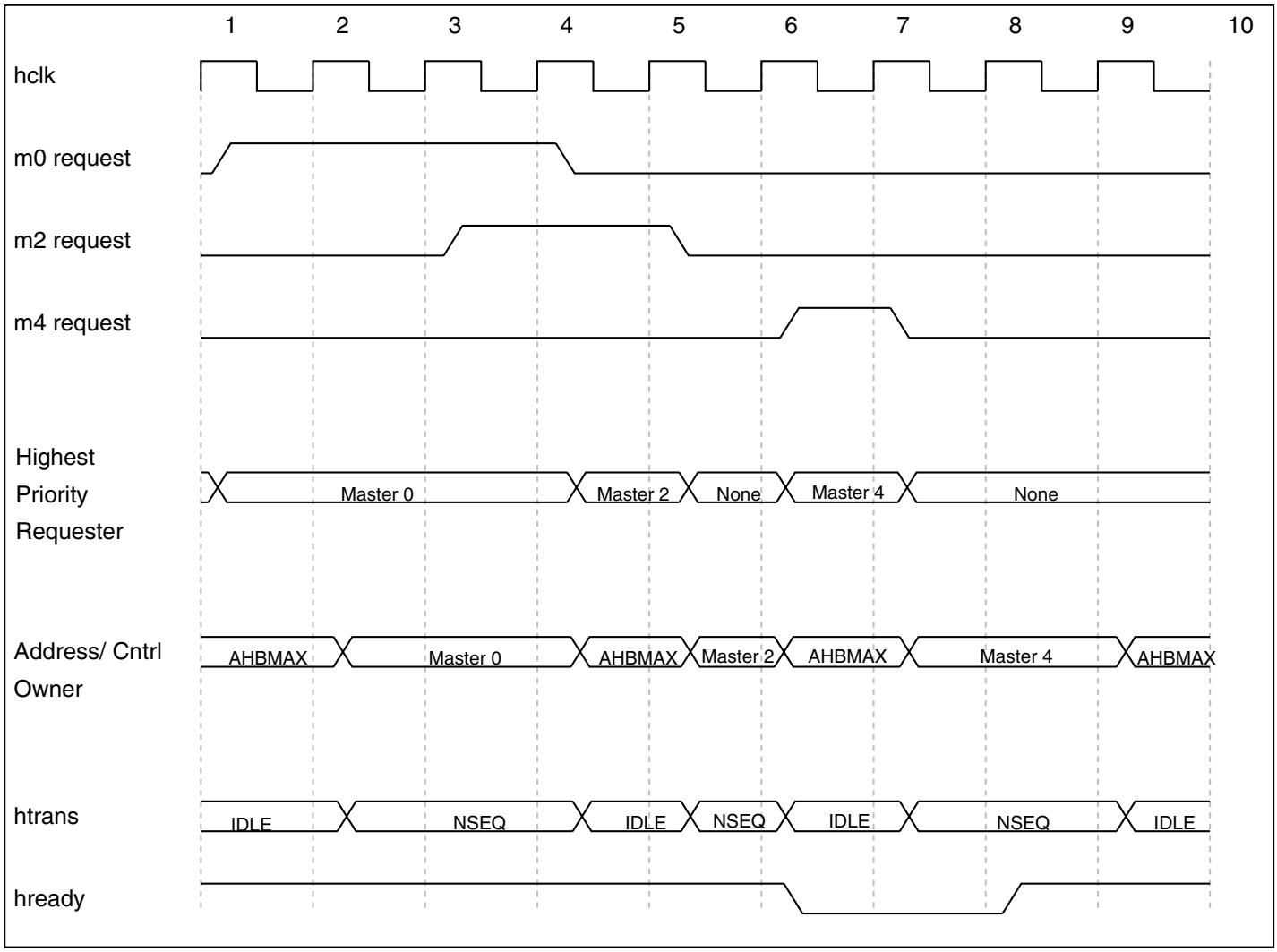


Figure 13-5. High to Low Priority Mastership Change

When the slave port is programmed for round-robin mode of arbitration then the slave port will switch masters any time there is more than one master actively making a request to the slave port. This will happen because any master other than the one which presently owns the bus will be considered to have higher priority.

Figure 13-6 shows an example of round-robin mode of operation.

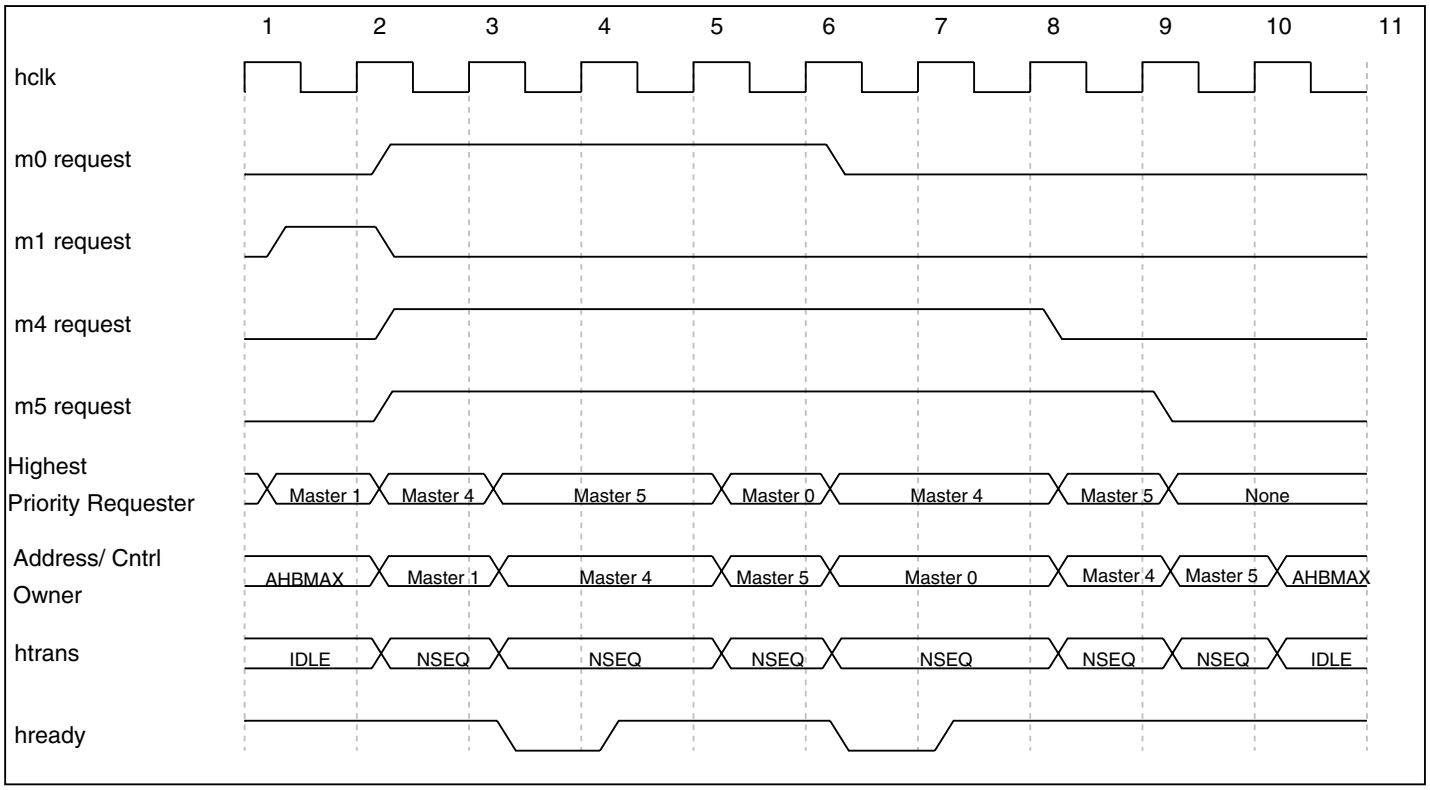


Figure 13-6. Round-robin Mastership Change

13.6.4.4.4 Slave Port State Machine Parking

If no master is currently making a request to the slave port then the slave port will be parked. It will park in one of four places, dictated by the PCTL and PARK bits in the AHBMAX_SGPCR and the locked state of the last master to access it.

If the last master to access the slave port ran a locked cycle and continues to run locked cycles even after leaving the slave port the slave port will park on that master without regard to the bit settings in the AHBMAX_SGPCR and without regard to pending requests from other masters. This is done so a master can run a locked transfer to the slave port, leave it, and return to it and be guaranteed that no other master has had access to it (provided the master maintains all transfers are locked transfers). If locking is not an issue for parking the AHBMAX_SGPCR bits will dictate the parking method.

If the PCTL bits are set for "low power park" mode then the slave port will enter low power park mode. It will not recognize any master as being in control of it and it will not select any master's signals to pass through to the slave bus. In this case all slave bus activity will effectively halt because all slave bus signals being driven from the AHBMAX will be 0. This of course can save quite a bit of power if the slave port will not

be in use for some time. The down side is that when a master does make a request to the slave port it will be delayed by one clock since it will have to arbitrate to acquire ownership of the slave port.

If the PCTL bits are set to "park on last" mode then the slave port will park on the last master to access it, passing all that masters signals through to the slave bus. The AHBMAX will asynchronously force htrans[1:0], hmaster[3:0], hburst[2:0] and hmastlock to 0 for all access that the master does not run to the slave port. When that master access the slave port again it will not pay any arbitration penalty; however, if any other master wishes to access the slave port a one clock arbitration penalty will be imposed.

If the PCTL bits are set to "use PARK" mode then the slave port will park on the master designated by the PARK bits. The behavior here is the same as for the "park on last" mode with the exception that a specific master will be parked on instead of the last master to access the slave port. If the master designated by the PARK bits tries to access the slave port it will not pay an arbitration penalty while any other master will pay a one clock penalty. Figure 13-7 illustrates parking on a specific master.

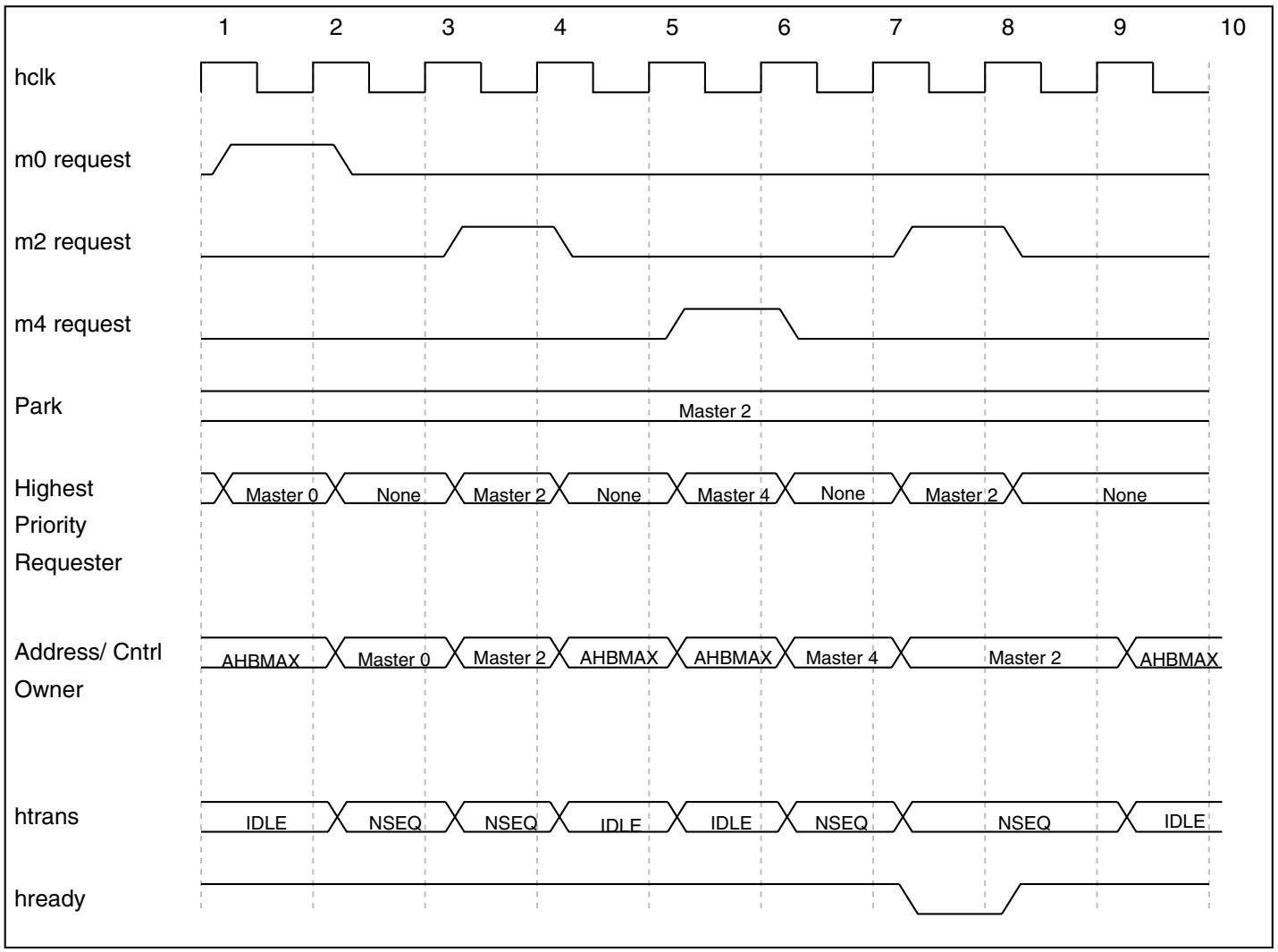


Figure 13-7. Parking on a Specific Master

Figure 13-8 illustrates parking on the last master. Note that in cycle 6 simultaneous requests are made by master 2 and master 4. Although master 2 has higher priority, the slave bus is parked on master 4 so master 4's access will be taken first. The slave port parks on master 2 once it has given control to master 2. This same situation can occur when parking on a specific master as well.

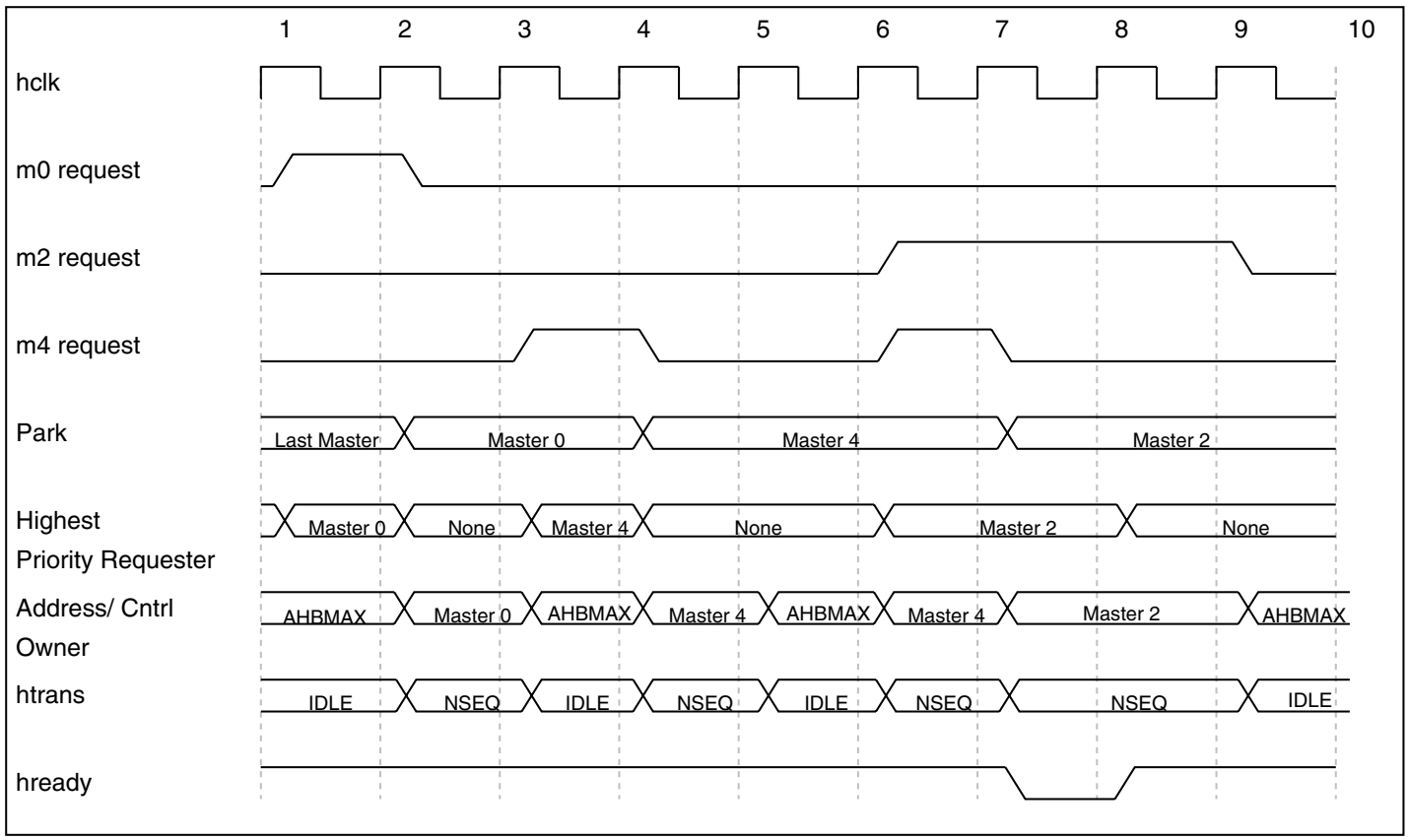


Figure 13-8. Parking on Last Master

13.6.4.4.5 Slave Port State Machine Halt Mode

If the max_halt_request input is asserted the slave port will eventually halt all slave bus activity and go into halt mode, which is almost identical to low power park mode. The HLP bit in the AHBMAX_SGPCR controls the priority level of the max_halt_request in the arbitration algorithm. If the HLP bit is cleared then the max_halt_request will have the highest priority of any master and will gain control of the slave port at the next arbitration point (most likely the next bus cycle, unless the current master is running a locked or fixed length burst transfer). If the HLP bit is set then the slave port will wait until no masters are actively making requests before moving to halt mode.

Regardless of the state of the HLP bit, once the slave port has gone into halt mode as a result of max_halt_request being asserted, it will remain in halt mode until max_halt_request is negated, regardless of the priority level of any masters that may make requests.

In halt mode no master is selected to own the slave port so all the outputs of the slave port are set to 0.

13.7 Initialization/Application Information

No initialization is required by or for the AHBMAX. Hardware reset ensures all the register bits used by the AHBMAX are properly initialized.

13.8 AHBMAX Interface

This section provides information on the AHBMAX interface.

13.8.1 AHBMAX Interface Overview

The main goal of the AHBMAX is to increase overall system performance by allowing multiple masters to communicate in parallel with multiple slaves. In order to maximize data throughput it is essential to keep arbitration delays to a minimum.

This section examines data throughput from the point of view of masters and slaves, detailing when the AHBMAX will stall the masters or insert bubbles on the slave side.

13.8.2 Master Ports-AHBMAX Interface

Master accesses will receive one of four responses from the AHBMAX. They will either be terminated, taken, stalled or responded to with an error.

13.8.2.1 Terminated Accesses

A master access will be terminated if the transfer type is IDLE. The AHBMAX will terminated the access and it will not be allowed to pass through the AHBMAX.

13.8.2.2 Taken Accesses

A master access will be taken if the transfer type is non IDLE and the slave port to which the access decodes is either currently servicing the master or is parked on the master. In this case the AHBMAX will be completely transparent and the master's access will be immediately seen on the slave bus and no arbitration delays will be incurred.

13.8.2.3 Stalled Accesses

A master access will be stalled if the transfer type is non IDLE and the access decodes to a slave port that is busy serving another master, parked on another master or is in low power park mode. The AHBMAX will indicate to the master that the address phase of the access has been taken but will then queue the access to the appropriate slave port to enter into arbitration for access to that slave port.

If the slave port is currently parked on another master or is in low power park mode and no other master is requesting access to the slave port then only one clock of arbitration will be incurred. If the slave port is currently serving another master of a lower priority and the master has a higher priority than all other requesting masters then the master will gain control over the slave port as soon as the data phase of the current access is completed (burst and locked transfers excluded). If the slave port is currently servicing another master of a higher priority then the master will gain control of the slave port once the other master releases control of the slave port if no other higher priority master is also waiting for the slave port.

13.8.2.4 Error Response Terminated Accesses

A master access will be responded to with an error if the transfer type is non IDLE and the access decodes to a location not occupied by a slave port.

13.8.3 Slave Ports-AHBMAX Interface

The goal of the AHBMAX with respect to the slave ports is to keep them 100% saturated when masters are actively making requests. In order to do this the AHBMAX must not insert any bubbles onto the slave bus unless absolutely necessary.

There is only one instance when the AHBMAX will force a bubble onto the slave bus when a master is actively making a request. This occurs when a higher priority master has control of the slave port and is running single clock (zero wait state) accesses while a lower priority master is stalled waiting for control of the slave port. When the higher priority master either leaves the slave port or runs an IDLE cycle to the slave port the AHBMAX will take control of the slave bus and run a single IDLE cycle before giving the slave port to the lower priority master that was waiting for control of the slave port.

The only other times the AHBMAX will have control of the slave port is when the AHBMAX is halting or when no masters are making access requests to the slave port and the AHBMAX is forced to either park the slave port on a specific master or put the slave port into low power park mode.

In most instances when the AHBMAX has control of the slave port it will indicate IDLE for the transfer type, negate all control signals and indicate ownership of the slave bus via the hmaster encoding of b0000. One exception to this rule is when a master running locked cycles has left the slave port but continues to run locked cycles. In this case the AHBMAX will control the slave port and will indicate IDLE for the transfer type but it will not affect any other signals.

NOTE

When a master runs a locked cycle through the AHBMAX, the master will be guaranteed ownership of all slave ports it accesses while running locked cycles for one cycle beyond when the master finishes running locked cycles.

13.9 Programmable Registers

There are four registers that reside in each slave port of the AHBMAX and one register that resides in each master port of the AHBMAX. These registers are IP bus compliant registers. Read and write transfers both require two IP bus clock cycles. The registers can only be read from and written to in supervisor mode. Additionally, these registers can only be read from or written to by 32-bit accesses.

The registers are fully decoded and an error response is returned if an unimplemented location is accessed within the AHBMAX.

The slave registers also feature a bit, which when written with a 1, will prevent the registers from being written to again. The registers will still be readable, but future write attempts will have no effect on the registers and will be terminated with an error response.

AHBMAX memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63F9_4000	Master Priority Register for Slave port n (AHBMAX_MPR0)	32	R/W	0054_3210h	13.9.1/ 694

Table continues on the next page...

AHBMAX memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63F9_4010	General Purpose Control Register for Slave port n (AHBMAX_SGPCR0)	32	R/W	0000_0000h	13.9.2/ 696
63F9_4100	Master Priority Register for Slave port n (AHBMAX_MPR1)	32	R/W	0054_3210h	13.9.1/ 694
63F9_4110	General Purpose Control Register for Slave port n (AHBMAX_SGPCR1)	32	R/W	0000_0000h	13.9.2/ 696
63F9_4200	Master Priority Register for Slave port n (AHBMAX_MPR2)	32	R/W	0054_3210h	13.9.1/ 694
63F9_4210	General Purpose Control Register for Slave port n (AHBMAX_SGPCR2)	32	R/W	0000_0000h	13.9.2/ 696
63F9_4300	Master Priority Register for Slave port n (AHBMAX_MPR3)	32	R/W	0054_3210h	13.9.1/ 694
63F9_4310	General Purpose Control Register for Slave port n (AHBMAX_SGPCR3)	32	R/W	0000_0000h	13.9.2/ 696
63F9_4800	General Purpose Control Register for Master port n (AHBMAX_MGPCR0)	32	R/W	0000_0000h	13.9.3/ 698
63F9_4900	General Purpose Control Register for Master port n (AHBMAX_MGPCR1)	32	R/W	0000_0000h	13.9.3/ 698
63F9_4A00	General Purpose Control Register for Master port n (AHBMAX_MGPCR2)	32	R/W	0000_0000h	13.9.3/ 698
63F9_4B00	General Purpose Control Register for Master port n (AHBMAX_MGPCR3)	32	R/W	0000_0000h	13.9.3/ 698
63F9_4C00	General Purpose Control Register for Master port n (AHBMAX_MGPCR4)	32	R/W	0000_0000h	13.9.3/ 698
63F9_4D00	General Purpose Control Register for Master port n (AHBMAX_MGPCR5)	32	R/W	0000_0000h	13.9.3/ 698
63F9_4E00	General Purpose Control Register for Master port n (AHBMAX_MGPCR6)	32	R/W	0000_0000h	13.9.3/ 698

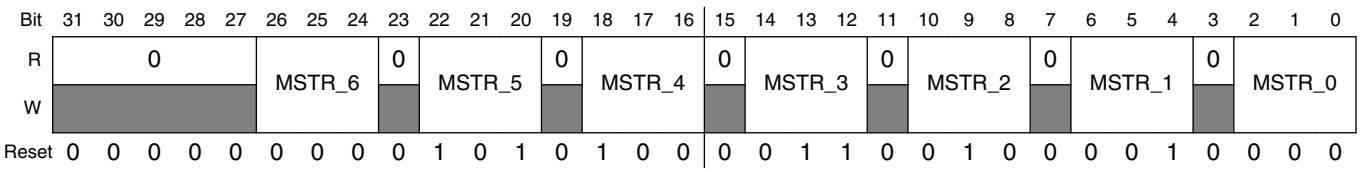
13.9.1 Master Priority Register for Slave port n (AHBMAX_MPR*n*)

The Master Priority Register (MPR) sets the priority of each master port on a per slave port basis and resides in each slave port.

The Master Priority Register can only be accessed in supervisor mode with 32-bit accesses. Once the RO (Read Only) bit has been set in the Slave General Purpose Control Register the Master Priority Register (MPR) can only be read from, attempts to write to it will have no effect on the MPR and result in an error response.

Additionally, no two available master ports may be programmed with the same priority level. Attempts to program two or more available masters with the same priority level will result in an error response and the MPR will not be updated.

Addresses: AHBMAX_MPR0 is 63F9_4000h base + 0h offset = 63F9_4000h
AHBMAX_MPR1 is 63F9_4000h base + 100h offset = 63F9_4100h
AHBMAX_MPR2 is 63F9_4000h base + 200h offset = 63F9_4200h
AHBMAX_MPR3 is 63F9_4000h base + 300h offset = 63F9_4300h



AHBMAX_MPR*n* field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved. They are read as zero and should be written with zero for upward compatibility.
26–24 MSTR_6	Master 6Priority. These bits set the arbitration priority for master port 6 on the associated slave port. These bits are initialized by hardware reset. 000 This master has the highest priority when accessing the slave port. 111 This master has the lowest priority when accessing the slave port.
23 Reserved	This read-only field is reserved and always has the value zero. Reserved. They are read as zero and should be written with zero for upward compatibility.
22–20 MSTR_5	Master 5 Priority. These bits set the arbitration priority for master port 5 on the associated slave port. These bits are initialized by hardware reset. 000 This master has the highest priority when accessing the slave port. 111 This master has the lowest priority when accessing the slave port.
19 Reserved	This read-only field is reserved and always has the value zero. Reserved. They are read as zero and should be written with zero for upward compatibility.
18–16 MSTR_4	Master 4 Priority. These bits set the arbitration priority for master port 4 on the associated slave port. These bits are initialized by hardware reset. 000 This master has the highest priority when accessing the slave port. 111 This master has the lowest priority when accessing the slave port.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved. They are read as zero and should be written with zero for upward compatibility.
14–12 MSTR_3	Master 3 Priority. These bits set the arbitration priority for master port 3 on the associated slave port. These bits are initialized by hardware reset. 000 This master has the highest priority when accessing the slave port. 111 This master has the lowest priority when accessing the slave port.
11 Reserved	This read-only field is reserved and always has the value zero. Reserved. They are read as zero and should be written with zero for upward compatibility.

Table continues on the next page...

AHBMAX_MPR*n* field descriptions (continued)

Field	Description
10–8 MSTR_2	Master 2 Priority. These bits set the arbitration priority for master port 2 on the associated slave port. These bits are initialized by hardware reset. 000 This master has the highest priority when accessing the slave port. 111 This master has the lowest priority when accessing the slave port.
7 Reserved	This read-only field is reserved and always has the value zero. Reserved. They are read as zero and should be written with zero for upward compatibility.
6–4 MSTR_1	Master 1 Priority. These bits set the arbitration priority for master port 1 on the associated slave port. These bits are initialized by hardware reset. 000 This master has the highest priority when accessing the slave port. 111 This master has the lowest priority when accessing the slave port.
3 Reserved	This read-only field is reserved and always has the value zero. Reserved. They are read as zero and should be written with zero for upward compatibility.
2–0 MSTR_0	Master 0 Priority. These bits set the arbitration priority for master port 0 on the associated slave port. These bits are initialized by hardware reset. 000 This master has the highest priority when accessing the slave port. 111 This master has the lowest priority when accessing the slave port.

13.9.2 General Purpose Control Register for Slave port n (AHBMAX_SGPCR*n*)

The Slave General Purpose Control Register (AHBMAX_SGPCR*n*) controls several features of each slave port.

The Read Only (RO) bit will prevent any registers associated with this slave port from being written to once set. This bit may be written with 0 as many times as the user desires, but once it is written to a 1 only a reset condition will allow it to be written again.

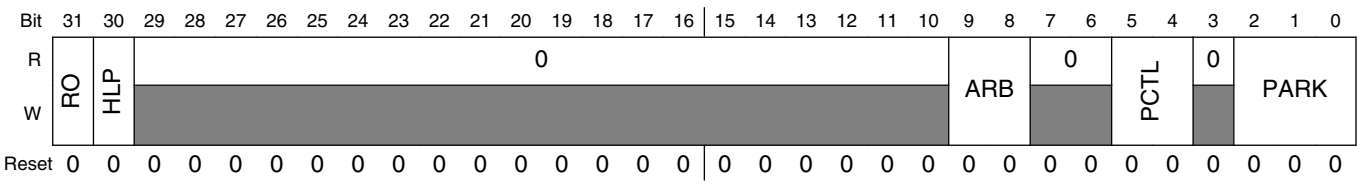
The Halt Low Priority (HLP) bit will set the priority of the max_halt_request input to the lowest possible priority for initial arbitration of the slave ports. By default it is the highest priority. Please note, setting this bit will not effect the max_halt_request from attaining highest priority once it has control of the slave ports.

The PCTL bits determine how the slave port will park when no master is actively making a request. The available options are to park on the master defined by the PARK bits, park on the last master to use the slave port, or go into a low power park mode which will force all the outputs of the slave port to inactive states when no master is requesting an access. The low power park feature can result in an overall power savings if a the slave port is not saturated; however, it will force an extra clock of latency whenever any master tries to access it when it is not in use because it will not be parked on any master

The PARK bits determine which master the slave will park on when no master is making an active request and the max_halt_request input is negated. Please use caution to only select master ports that are actually present in the design. If the user programs the PARK bits to a master not present in the current design implementation undefined behavior will result.

The AHBMAX_SGPCR*n* can only be accessed in supervisor mode with 32-bit accesses. Once the RO (Read Only) bit has been set in the AHBMAX_SGPCR*n* the AHBMAX_SGPCR*n* can only be read, attempts to write to it will have no effect on the AHBMAX_SGPCR*n* and result in an error response.

Addresses: AHBMAX_SGPCR0 is 63F9_4000h base + 10h offset = 63F9_4010h
AHBMAX_SGPCR1 is 63F9_4000h base + 110h offset = 63F9_4110h
AHBMAX_SGPCR2 is 63F9_4000h base + 210h offset = 63F9_4210h
AHBMAX_SGPCR3 is 63F9_4000h base + 310h offset = 63F9_4310h



AHBMAX_SGPCR*n* field descriptions

Field	Description
31 RO	Read Only. This bit is used to force all of a slave port's registers to be read only. Once written to 1 it can only be cleared by hardware reset. This bit is initialized by hardware reset. 0 All this slave port's registers can be written. 1 All this slave port's registers are read only and cannot be written (attempted writes have no effect and result in an error response).
30 HLP	Halt Low Priority. This bit is used to set the initial arbitration priority of the max_halt_request input. This bit is initialized by hardware reset. 0 The max_halt_request input has the highest priority for arbitration on this slave port 1 The max_halt_request input has the lowest initial priority for arbitration on this slave port.
29–10 Reserved	This read-only field is reserved and always has the value zero. Reserved. They read as zero and should be written with zero for upward compatibility.
9–8 ARB	Arbitration Mode. These bits are used to select the arbitration policy for the slave port. These bits are initialized by hardware reset. 00 Fixed Priority. 01 Round Robin (rotating) Priority. 10 Reserved 11 Reserved
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved. They read as zero and should be written with zero for upward compatibility.

Table continues on the next page...

AHBMAX_SGPCR*n* field descriptions (continued)

Field	Description
5–4 PCTL	Parking Control.These bits determine the parking control used by this slave port. These bits are initialized by hardware reset. 00 When no master is making a request the arbiter will park the slave port on the master port defined by the PARK bit field. 01 When no master is making a request the arbiter will park the slave port on the last master to be in control of the slave port. 10 When no master is making a request the arbiter will park the slave port on no master and will drive all outputs to a constant safe state. 11 Reserved
3 Reserved	This read-only field is reserved and always has the value zero. Reserved. They read as zero and should be written with zero for upward compatibility.
2–0 PARK	PARK. These bits are used to determine which master port this slave port parks on when no masters are actively making requests and the PCTL bits are set to 00. These bits are initialized by hardware reset. 000 Park on Master Port 0 001 Park on Master Port 1 010 Park on Master Port 2 011 Park on Master Port 3 100 Park on Master Port 4 101 Park on Master Port 5 110 Park on Master Port 6 111 Reserved

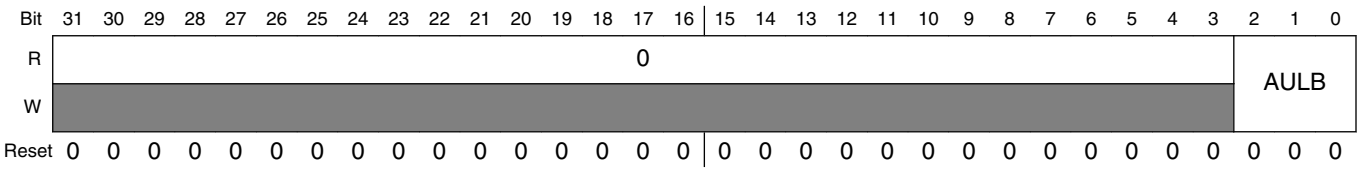
13.9.3 General Purpose Control Register for Master port n (AHBMAX_MGPCR*n*)

The Master General Purpose Control Register (AHBMAX_MGPCR*n*) presently controls only whether or not the master's undefined length burst accesses will be allowed to complete uninterrupted or whether they can be broken by requests from higher priority masters.

The AULB (Arbitrate on Undefined Length Bursts) bit field determines whether (and when) or not the AHBMAX will arbitrate away the slave port the master owns when the master is performing undefined length burst accesses.

The AHBMAX_MGPCR*n* can only be accessed in supervisor mode with 32-bit accesses.

Addresses: AHBMAX_MGPCR0 is 63F9_4000h base + 800h offset = 63F9_4800h
AHBMAX_MGPCR1 is 63F9_4000h base + 900h offset = 63F9_4900h
AHBMAX_MGPCR2 is 63F9_4000h base + A00h offset = 63F9_4A00h
AHBMAX_MGPCR3 is 63F9_4000h base + B00h offset = 63F9_4B00h
AHBMAX_MGPCR4 is 63F9_4000h base + C00h offset = 63F9_4C00h
AHBMAX_MGPCR5 is 63F9_4000h base + D00h offset = 63F9_4D00h
AHBMAX_MGPCR6 is 63F9_4000h base + E00h offset = 63F9_4E00h



AHBMAX_MGPCR*n* field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value zero. Reserved. They read as zero and should be written with zero for upward compatibility.
2–0 AULB	Arbitrate on Undefined Length Bursts. These bits are used to select the arbitration policy during undefined length bursts by this master. These bits are initialized by hardware reset. 000 No arbitration will be allowed during an undefined length burst. 001 Arbitration will be allowed at any time during an undefined length burst. 010 Arbitration will be allowed after four beats of an undefined length burst. 011 Arbitration will be allowed after eight beats of an undefined length burst. 100 Arbitration will be allowed after 16 beats of an undefined length burst. 101 Reserved 110 Reserved 111 Reserved

Chapter 14
AHB to IP Bridge (AIPSTZ)

14.1 Introduction

This section provides an overview of the AHB to IP Bridge (AIPSTZ). The peripheral bridge acts as an interface between the system bus and lower bandwidth IP Slave (IPS) bus peripherals.

14.1.1 Features

The following list summarizes the key features of the bridge:

- The bridge supports the IPS slave bus signals. This interface is only meant for slave peripherals.
- The bridge supports 8-, 16-, and 32-bit IPS peripherals. (Accesses larger than the size of a peripheral are not supported, except to 32-bit memory.)
- The bridge supports a pair of IPS accesses for 64-bit and certain misaligned AHB transfers to 32-bit memory in 64-bit platforms.
- The bridge directly supports up to 32 16-Kbyte external IPS peripherals, and 2 global external IPS peripheral spaces. The bridge occupies 64 MBytes of total address space.
- The bridge provides configurable per-block and per-master access protections.
- Peripheral read transactions require a minimum of 2 hclk clocks, and unbuffered write transactions require a minimum of 3 hclk clocks.
- The bridge uses one single asynchronous reset and one global clock.

14.2 General Operation

The AHB to IP bridge is the interface between the AHB and on-chip IPS peripherals, which are sub-blocks containing readable/writable control and status registers.

The AHB master reads and writes these registers through the AIPSTZ. The bridge generates block enables, the block address, transfer attributes, byte enables and write data as inputs to the IPS peripherals. The bridge captures read data from the IPS interface and drives it on the AHB.

It occupies a 64-Mbyte portion of the address space. 63.5 Mbytes are available for off-platform devices. The register maps of the IPS peripherals are located on 16-Kbyte boundaries. Each IPS peripheral is allocated one 16K-byte block of the memory map, and is activated by one of the block enables from the bridge. Up to thirty-two 16-Kbyte external IPS peripherals may be implemented, occupying contiguous blocks of 16 Kbytes. Two global external IPS block enables are available for the remaining 63 Mbytes of address space to allow for customization and expansion of addressed peripheral devices. In addition, a single "non-global" block enable is also asserted whenever any of the thirty-two non-global block enables is asserted.

The bridge is responsible for indicating to IPS peripherals if an access is in supervisor or user mode. It may block user mode accesses to certain IPS peripherals or it may allow the individual IPS peripherals to determine if user mode accesses are allowed. In addition, peripherals may be designated as write-protected.

The bridge supports the notion of "trusted" masters for security purposes. Masters may be individually designated as trusted for reads, trusted for writes, or trusted for both reads and writes, as well as being forced to look as though all accesses from a master are in user-mode privilege level. Refer to Control Registers for more information.

All peripheral devices are expected to only require aligned accesses equal to or smaller in size than the peripheral size. An exception to this rule is supported for 32-bit peripherals to allow memory to be placed on the IPS.

14.2.1 AIPSTZ Registers

This section provides information on the registers of the AIPS bridge.

In i.MX53 there are 2 blocks named AIPSTZ-1 and AIPSTZ-2. The registers shown in the parathyroids below are identical in both blocks, unless otherwise specified.

14.2.2 Overview

There are eleven registers that control the AIPS bridge. All registers are 32-bit registers and can only be accessed in supervisor mode by specifically configured bus masters such as the core processor. Additionally, these registers must only be read from or written to by a 32-bit aligned access. The bridge registers are mapped into the PACR0 address space.

Two system clocks are required for read accesses and three system clocks are required for write accesses to the bridge registers.

14.2.3 Control Registers

The memory map for the AIPS program-visible registers is shown in the table below.

The MPROT fields of the AIPSTZ_MPR and the AIPSTZ_PACR and AIPSTZ_OPACR registers are 4 bits in width. Some bits may be reserved depending on device.

Table 14-1. AIPSTZ Register Memory Map

PACR0_ Offset	[31:28]	[27:24]	[23:20]	[19:16]	[15:12]	[11:8]	[7:4]	[3:0]
0xBASE__0000	MPROT0	MPROT1	MPROT2	MPROT3	MPROT4	MPROT5	MPROT6	MPROT7
0xBASE_0004	MPROT8	MPROT9	MPROT10	MPROT11	MPROT12	MPROT13	MPROT14	MPROT15
0xBASE_0020	Reserved for on-platform Registers							
0xBASE_0024								
0xBASE_0028								
0xBASE_002c								
0xBASE_0040	OPACR0	OPACR1	OPACR2	OPACR3	OPACR4	OPACR5	OPACR6	OPACR7
0xBASE_0044	OPACR8	OPACR9	OPACR10	OPACR11	OPACR12	OPACR13	OPACR14	OPACR15
0xBASE_0048	OPACR16	OPACR17	OPACR18	OPACR19	OPACR20	OPACR21	OPACR22	OPACR23
0xBASE_004c	OPACR24	OPACR25	OPACR26	OPACR27	OPACR28	OPACR29	OPACR30	OPACR31
0xBASE_0050	OPACR32	OPACR33	Reserved					

14.3 Register Descriptions

14.3.1 Master Privilege Registers

Each AIPSTZ_MPR specifies eight 4-bit fields defining the access privilege level associated with a bus master in the platform, as well as specifying whether write accesses from this master are bufferable.

The registers provide one field per bus master, where field 15 corresponds to master 15, field 14 to master 14,... field 0 to master 0 (typically the processor core). The master index allocation is shown in Table 14-4.

Table 14-2. MPROT Field

Bit	3	2	1	0
Field	MBW	(MTR)	(MTW)	MPL
Reset	*	*	*	*
Read/Write	Read/Write			

Table 14-3. Master Protection Field Descriptions

Name	Description	Settings
3 (MBW)	**Master Buffer Writes** - This bit determines whether the AIPSTZ is enabled to buffer writes from this master.	0 Write accesses from this master are not bufferable 1 Write accesses from this master are allowed to be buffered
2 (MTR)	**Master Trusted for Reads**- This bit determines whether the master is trusted for read accesses.	0 This master is not trusted for read accesses. 1 This master is trusted for read accesses.
1 (MTW)	**Master Trusted for Writes**- This bit determines whether the master is trusted for write accesses.	0 This master is not trusted for write accesses. 1 This master is trusted for write accesses.
0 (MPL)	**Master Privilege Level** - This bit determines how the privilege level of the master is determined.	0 Accesses from this master are forced to user-mode (**ips_supervisor_access** is forced to zero) regardless of the **hprot[1]** access attribute. 1 Accesses from this master are not forced to user-mode. The **hprot[1]** access attribute is used directly to determine **ips_supervisor_access**.

NOTE

The reset value is set to 0000_0000_0700_0000, which makes master 1 (ARM CORE) the only trusted master. Trusted software can change the settings after reset.

Table 14-4. Master index allocation

Master index	Master name	Comments
Master 0	All masters excluding ARM core, SDMA and CAAM	Share the same number allocation.
Master 0	Reserved	
Master 1	ARM CORE	
Master 2	SDMA, SATA, FEC, MLB	Share the same number allocation.
Master 2	CAAM	
Master 3	GPU3D, GPU2D, VPU, IPU	Share the same number allocation.
Master 3	SDMA	
Master 4-15	Reserved	
Master 4	USB	
Master 5	PATA	
Master 6	Reserved	
Master 7	SAHARA	
Master 8	SCC	
Master 9	RTIC	
Master 10	ESDCHV2-4	
Master 11	Reserved	
Master 12	DAP	
Master 13	ESDHCV2-1	
Master 14	ESDHCV2-2	
Master 15	ESDHCV3-3	

14.3.2 Off-Platform Peripheral Access Control Registers (AIPSTZ_OPACRs)

Each of the off-platform peripherals have an Off-platform Peripheral Access Control Register (AIPSTZ_OPACR) which defines the access levels supported by the given block.

Each AIPSTZ_PACR has the following format:

Table 14-5. AIPSTZ_OPACR Fields

Bit	3	2	1	0
Field	BW	SP	WP	TP
Reset	0	1	0	0
Read/Write	Read/Write			

Table 14-6. Peripheral Access Control Register Field Descriptions

Name	Description	Settings
3 **(BW)**	**Buffer Writes** - This bit determines whether write accesses to this peripheral are allowed to be buffered.[1]	0 Write accesses to this peripheral are not bufferable by the AIPSTZ. 1 Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
2 **(SP)**	**Supervisor Protect** - This bit determines whether the peripheral requires supervisor privilege level for access.	0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus.
1 (WP)	**Write Protect** - This bit determines whether the peripheral allows write accesses	0 This peripheral allows write accesses. 1 This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus.
0 **(TP)**	**Trusted Protect** - This bit determines whether the peripheral allows accesses from an untrusted master.	0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus.

1. Buffered writes are not available for AIPSTZ. This bit should be set to '0'.

The AIPSTZ_OPACR index allocation for AIPSTZ-1 and AIPSTZ-2 are shown in Table 14-7 and Table 14-8 respectively.

Table 14-7. AIPSTZ_OPACR index allocation for AIPSTZ-1

AIPSTZ_OPACR index	Slave name	Comments
AIPSTZ_OPACR0	USB	
AIPSTZ_OPACR1	GPIO-1	
AIPSTZ_OPACR2	GPIO-2	
AIPSTZ_OPACR3	GPIO-3	
AIPSTZ_OPACR4	GPIO-4	
AIPSTZ_OPACR5	KPP	
AIPSTZ_OPACR6	WDOG-1	
AIPSTZ_OPACR7	WDOG-2	
AIPSTZ_OPACR8	GPT	
AIPSTZ_OPACR9	SRTC	
AIPSTZ_OPACR10	IOMUXC	
AIPSTZ_OPACR11	EPIT-1	

Table continues on the next page...

Table 14-7. AIPSTZ_OPACR index allocation for AIPSTZ-1 (continued)

AIPSTZ_OPACR index	Slave name	Comments
AIPSTZ_OPACR12	EPIT-2	
AIPSTZ_OPACR13	PWM-1	
AIPSTZ_OPACR14	PWM-2	
AIPSTZ_OPACR15	UART-1	
AIPSTZ_OPACR16	UART-2	
AIPSTZ_OPACR17	USB (PL301 port)	
AIPSTZ_OPACR18	CAN-1	
AIPSTZ_OPACR19	CAN-2	
AIPSTZ_OPACR20	SRC	
AIPSTZ_OPACR21	CCM	
AIPSTZ_OPACR22	GPC	
AIPSTZ_OPACR23	GPIO-5	
AIPSTZ_OPACR24	GPIO-6	
AIPSTZ_OPACR25	GPIO-7	
AIPSTZ_OPACR26	PATA (PORT PIO)	
AIPSTZ_OPACR27	I2C-3	
AIPSTZ_OPACR28	UART-4	
AIPSTZ_OPACR29	Reserved	
AIPSTZ_OPACR30	Reserved	
AIPSTZ_OPACR31	Reserved	
AIPSTZ_OPACR32	SPBA	
AIPSTZ_OPACR33	Reserved	

Table 14-8. AIPSTZ_OPACR index allocation for AIPSTZ-2

AIPSTZ_OPACR index	Slave name	Comments
AIPSTZ_OPACR0	DPLLIP-1	
AIPSTZ_OPACR1	DPLLIP-2	
AIPSTZ_OPACR2	DPLLIP-3	
AIPSTZ_OPACR3	DPLLIP-4	
AIPSTZ_OPACR4	UART-5	
AIPSTZ_OPACR5	AHBMAX	
AIPSTZ_OPACR6	IIM	
AIPSTZ_OPACR7	CSU	
AIPSTZ_OPACR8	ARM CORE	

Table continues on the next page...

Table 14-8. AIPSTZ_OPACR index allocation for AIPSTZ-2 (continued)

AIPSTZ_OPACR index	Slave name	Comments
AIPSTZ_OPACR9	OWIRE	
AIPSTZ_OPACR10	FIRI	
AIPSTZ_OPACR11	ECSPI-2	
AIPSTZ_OPACR12	SDMA (PORT IPS_HOST)	
AIPSTZ_OPACR13	SCC	
AIPSTZ_OPACR14	ROMC	
AIPSTZ_OPACR15	RTIC	
AIPSTZ_OPACR16	CSPI	
AIPSTZ_OPACR17	I2C-2	
AIPSTZ_OPACR18	I2C-1	
AIPSTZ_OPACR19	SSI-1	
AIPSTZ_OPACR20	AUDMUX	
AIPSTZ_OPACR21	RTC	
AIPSTZ_OPACR22	EXTMC	
AIPSTZ_OPACR23	PLARB2	
AIPSTZ_OPACR24	PLARB1	
AIPSTZ_OPACR25	MLB	
AIPSTZ_OPACR26	SSI-3	
AIPSTZ_OPACR27	FEC	
AIPSTZ_OPACR28	TVE	
AIPSTZ_OPACR29	VPU	
AIPSTZ_OPACR30	SAHARA	
AIPSTZ_OPACR31	PTP	
AIPSTZ_OPACR32	Reserved	
AIPSTZ_OPACR33	Reserved	

14.4 Functional Description

The AIPS bridge serves as a protocol translator between the AHB system bus and the IP bus. Support is provided for generating a pair of 32-bit IP bus accesses when targeted by a 64-bit system bus access, or a misaligned access which crosses a 32-bit boundary. No other bus-sizing access support is provided.

14.5 Access Protections

The AIPS bridge provides programmable access protections for both masters and peripherals. It allows the privilege level of a master to be overridden, forcing it to user-mode privilege, and allows masters to be designated as trusted or untrusted. Peripherals may require supervisor privilege level for access, may restrict access to a trusted master only, and may be write-protected.

14.6 Access Support

Aligned 64-bit accesses, aligned and misaligned word and half word accesses, as well as byte accesses are supported for 32-bit peripherals. Misaligned accesses are supported to allow memory to be placed on the IPS. Peripheral registers must not be misaligned, although no explicit checking is performed by the AIPS bridge.

The bridge will perform two IPS transfers for 64-bit accesses, word accesses with byte offsets of 1, 2, or 3, and for half word accesses with a byte offset of 3. All other accesses will be performed with a single IPS transfer.

Only aligned half word and byte accesses are supported for 16-bit peripherals. All other accesses types are unsupported, and results of such accesses are undefined. They are not terminated with an error response.

Only byte accesses are supported for 8-bit peripherals. All other accesses types are unsupported, and results of such accesses are undefined. They are not terminated with an error response.

14.7 Initialization Information

The AIPS bridge should be programmed before use. The following registers should be initialized: the Master Privilege Registers (AIPSTZ_MPRs), the Peripheral Access Control registers (AIPSTZ_PACRs), and the Off-platform Peripheral Access Control registers (AIPSTZ_OPACRs) described in 3.2 Control Registers.

Chapter 15
Asynchronous Sample Rate Converter (ASRC)

15.1 Introduction

The figure below is a system view of the connection between the ASRC block and other blocks.

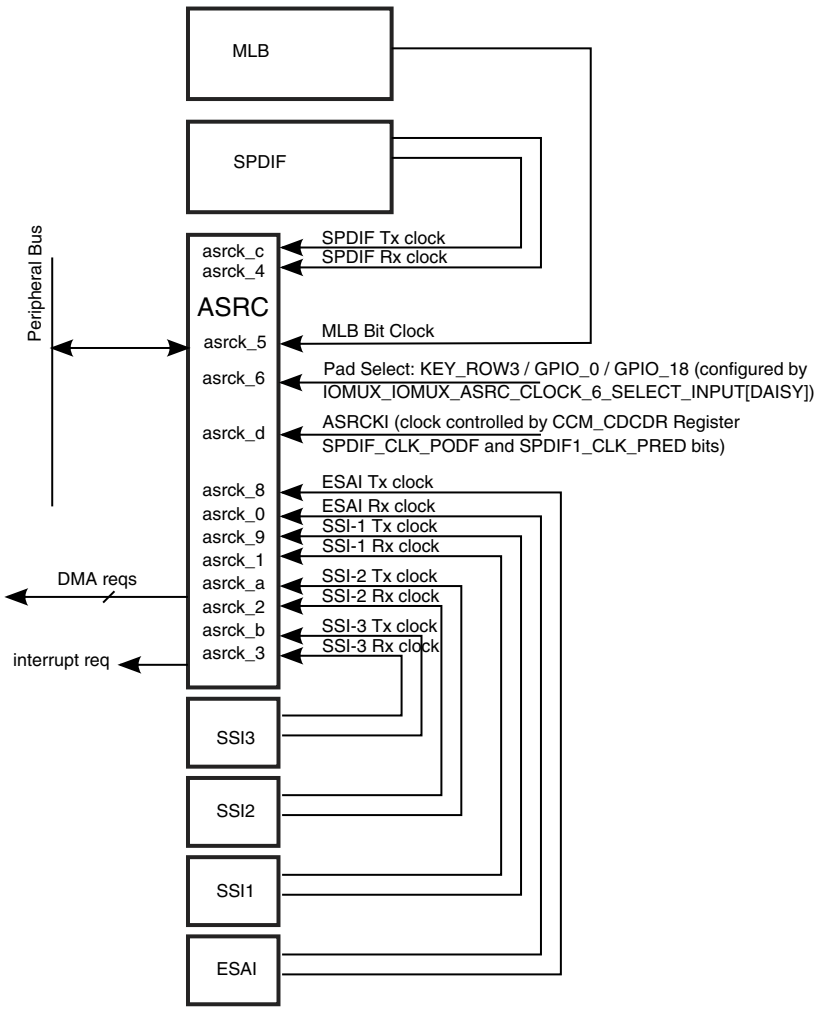


Figure 15-1. General System Overview

The following figure is the ASRC block diagram.

The red dotted line designates the ASRC block. Objects outside the dotted line represent SoC-level resources.

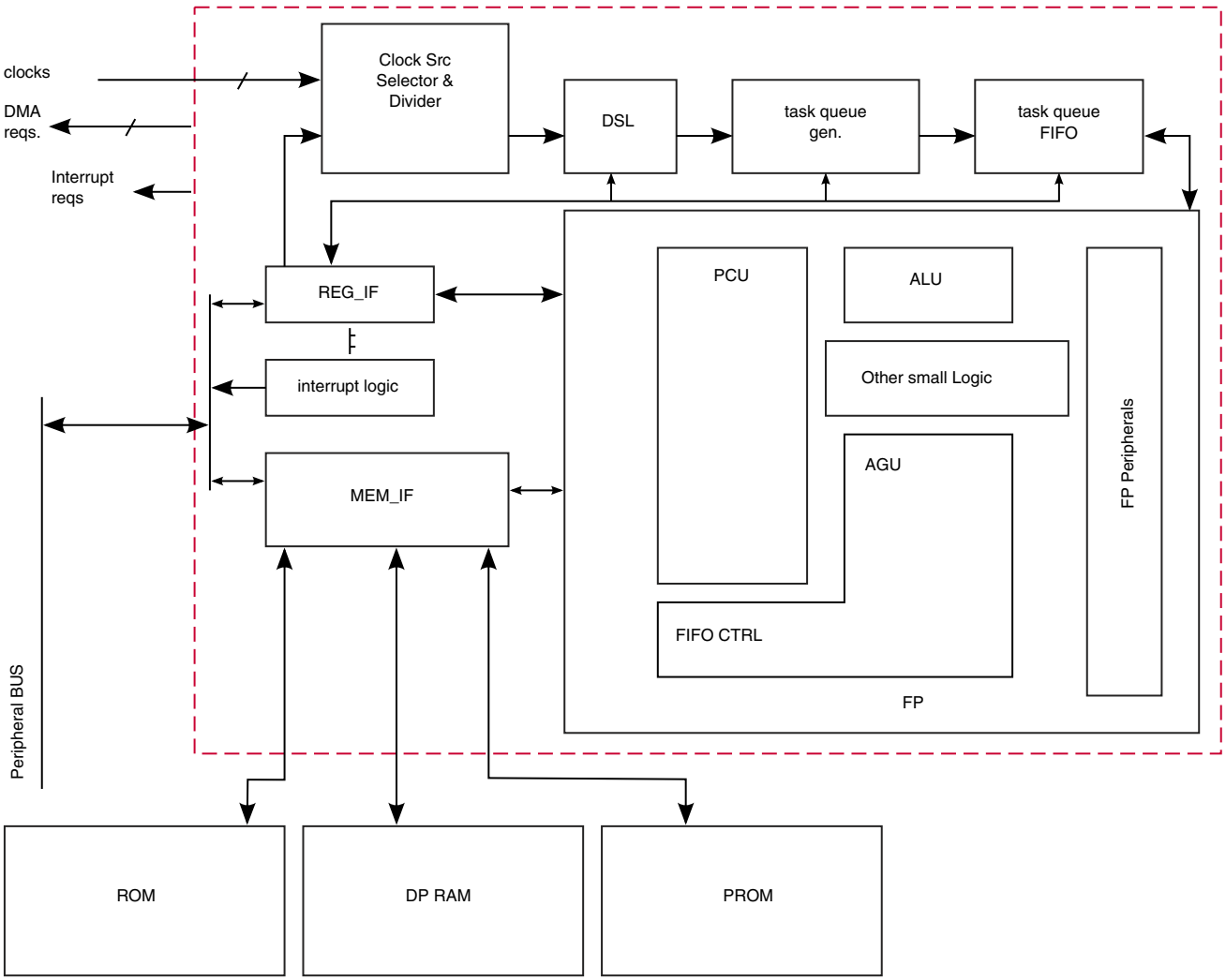


Figure 15-2. ASRC block diagram

15.1.1 Overview

The Asynchronous Sample Rate Converter (ASRC) converts the sampling rate of a signal associated with an input clock into a signal associated with a different output clock. The ASRC supports concurrent sample rate conversion of up to 10 channels of about -120dB THD+N. The ASRC supports up to three sampling rate pairs.

The incoming audio data to this chip may be received from various sources at different sampling rates. The outgoing audio data of this chip may have different sampling rates and it can also be associated with output clocks that are asynchronous to the input clocks.

The ASRC is implemented as a co-processor in hardware, with minimal ARM Platform intervention required.

15.1.2 Features

Table 15-1. ASRC Specifications

Parameters	Test Conditions	Minimum	Typical	Maximum	Unit
Channels Supported		01		10	
Pairs of Rate Conversion		1	-	3	
THD+N	120MHz < Fs_{ASRC}[2] <160MHz		-120		dB
Dynamic Range				144	dB
Settling Time			40		ms
Comment:					

1. When a pair has zero channels, the pair will be disabled, although the pair enable bit may be set in ASRCTR register.
2. Fs_{ASRC} is the processing clock of ASRC block.

Other Features:

- Support user-programmable threshold for the input/output FIFOs.
- Support flexible 8/16/24 bit width of input data, and 16/24 bit width of output data.
- Designed for rate conversion between 44.1kHz, 32kHz, 48kHz, 96kHz, and 192kHz. The useful signal bandwidth is below 24kHz.
- Other input sampling rates in the range of 8kHz to 200kHz is also supported, but possibly with less desirable bandwidth.
- Other output sampling rates in the range of 30kHz to 200kHz is also supported, but possibly with less desirable bandwidth.
- Automatic accommodation to slow variations in the incoming and outgoing sampling rates.
- Linear phase
- Tolerant to sample clock jitter

Clock/Data Connections

- The sampling rate clocks are directly connected to the ASRC block, the ratio estimation of the input clocks with output clocks are done in ASRC hardware.
- The clock signals come from the following blocks, for example:
  - ESAI, receiving bit clock and transmitting bit clock
  - SPDIF, receiving bit clock and transmitting bit clock
  - other audio peripherals etc.
- The exchange of audio data is done by the processor accessing ASRC block through registers defined on shared peripheral bus.

15.1.3 Modes of Operation

See the Programmable Registers section for a definition of the registers and parameters used in ASRC.

15.1.3.1 Data Transfer Schemes

15.1.3.1.1 Data Input Modes

The input mode for each of the three channel sets may be set independently. Three modes of supplying data to the ASRC input FIFOs are available:

- Polling
- Interrupt
- DMA

In all input-data transfer schemes, the ASRC fetches data from each enabled FIFO and processes the data sample-by-sample after each rising edge of the associated input sampling clock until the FIFO level reaches a threshold.

After the threshold is reached, the ASRC requests data. The FIFO size for each channel set is 64 samples and the threshold is set at 32 samples. The threshold can be defined by interface registers ASRMCRx, x=A, B or C.

If the ASRC attempts to fetch data from an empty FIFO, an error is generated and the ASRSTR_AOLE bit is set. If the ASRC overload interrupt is enabled (ASRIER_AOLIE bit is set), an interrupt is generated.

When writing data to an input FIFO, you must ensure that it is in a predefined sequence. For example, when writing to an input FIFO, the sequence should be: channel_0, channel_1, channel_2,..., channel_n, channel_0, channel_1, channel_2, etc. Here channel_n stands for the data intended for the n-th channel. The hardware will re-allocate each data to its corresponding channel FIFO. The channel being re-allocated is shown by ASRCCR_ACIA.

Mode 1 (Polling Mode)

Polling mode is the default mode following power-on or individual reset, and is selected by clearing the associated channel set A, B, or C data-input interrupt enable bit (ASRIER_ADIEx, where x=A, B or C). In this mode, data-input interrupts are disabled. When the FIFO level is below the threshold, the associated status bit (ASRSTR_AIDEx, where x=A, B, or C) is set. To clear the status bit, the FIFO must be written with enough data to raise the level above the threshold.

Mode 2 (Interrupt Mode)

The ASRC input FIFOs can also be serviced by interrupts. To enable interrupts, the corresponding data-input interrupt enable bits (ASRIER_ADIEx, where x=A, B, or C) should be set. An interrupt is automatically generated any time the input FIFO level is below the threshold. The interrupt is cleared when enough data is written to the FIFO to raise the level above the threshold.

Mode 3 (DMA Mode)

The ASRC input FIFOs can also be filled using DMA. In this mode, the data-input interrupt-enable bits (ASRIER_ADIEx, where x=A, B, or C) should be cleared and the DMA controller should be configured to use the ASRC as a request source.

15.1.3.1.2 Data Output Modes

The output mode for each of the 3 channel sets (A, B, and C) may be set independently.

Three modes of retrieving data from the ASRC output FIFOs are available:

- Polling
- Interrupt
- DMA

In all output-data transfer schemes, the ASRC places a processed sample into the associated output FIFO. After a threshold is reached, the ASRC requests that data be transferred out of the FIFO.

The FIFO size for each channel set is 64 samples and the threshold is set at 32 samples. The threshold can be defined by interface registers ASRMCRx, x=A, B or C.

If the ASRC attempts to place data into a FIFO that is already full, an error is generated and the ASRSTR_AOLE bit is set. If the ASRC overload interrupt is enabled (ASRIER_AOLIE bit is set), an interrupt is generated.

Each output FIFO is organized in the same channel order in which the associated input FIFO was written.

Two transfer modes are supported by Interface Block.

Mode 1 (Polling Mode)

The ASRC output FIFOs can be serviced by polling. In this mode, ensure the associated output-data interrupt enable bit (ASRIER_ADOEx, where x=A, B, or C) is cleared. In this mode, all output-data interrupts are disabled. Any time the output FIFO exceeds the threshold the associated status bit (ASRSTR_AODFx, where x=A, B, or C) is set. To clear the status bit, enough data must be read from the associated output FIFO to lower the level below the threshold.

Mode 2 (Interrupt Mode)

The ASRC output FIFOs may also be serviced using interrupts. To enable this mode, the corresponding output-data interrupt-enable bits (ASRIER_ADOEx, where x=A, B, or C) should be set. Any time the output FIFO level exceeds the threshold, an interrupt is automatically generated. The interrupt is cleared when enough data is read from the FIFO to lower the level below the threshold.

Mode 3 (DMA Mode)

The ASRC output FIFOs can also be read using DMA. In this mode, the output-data interrupt-enable bits (ASRIER_ADOEx, where x=A, B, or C) should be cleared and the DMA controller should be configured to use the ASRC as a request source.

15.1.3.2 Word Alignment Supported

15.1.3.2.1 Input Data Alignment Modes

The position and length of input data word to the input data FIFOs ASRDIA, ASRDIB, ASRDIC are programmable. The control bits are defined in ASRMCR1x {x=A, B, or C}. It supports the following modes.

Table 15-2. Input Data Alignment

Format	Bit Number																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8-bit LSB Aligned																									7	6	5	4	3	2	1	0
8-bit MSB Aligned																	7	6	5	4	3	2	1	0								
16-bit LSB Aligned																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit MSB Aligned	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
24-bit LSB Aligned									23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
24-bit MSB Aligned	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								

15.1.3.2.2 Output Data Alignment Modes

The position and length of output data word from the output data FIFOs ASRDOA, ASRDOB, ASRDOC are programmable. The control bits are defined in ASRMCR1x {x=A, B, or C}. It supports the following modes.

Table 15-3. Output Data Alignment

Format	Bit Number																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit LSB Aligned																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit LSB Aligned with Sign Extension	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit MSB Aligned	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
24-bit LSB Aligned									23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
24-bit LSB Aligned with Sign Extension	23	23	23	23	23	23	23	23	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
24-bit MSB Aligned	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								

15.2 Interrupts

ASRC has several interrupts events.

Priority	Description
lowest	ASRC Pair A input data needed
	ASRC Pair B input data needed
	ASRC Pair C input data needed
	ASRC Pair A output data ready
	ASRC Pair B output data ready
	ASRC Pair C output data ready
	ASRC Overload

15.3 DMA requests

ASRC has six DMA requests. They are directly connected to the lowest six status bits in the ASRSTR register.

Table 15-5. DMA requests

Type	Description
0	ASRC Pair A input data needed
1	ASRC Pair B input data needed
2	ASRC Pair C input data needed
3	ASRC Pair A output data ready
4	ASRC Pair B output data ready
5	ASRC Pair C output data ready

15.4 Functional Description

15.4.1 Algorithm Description

15.4.1.1 Signal Processing Flow

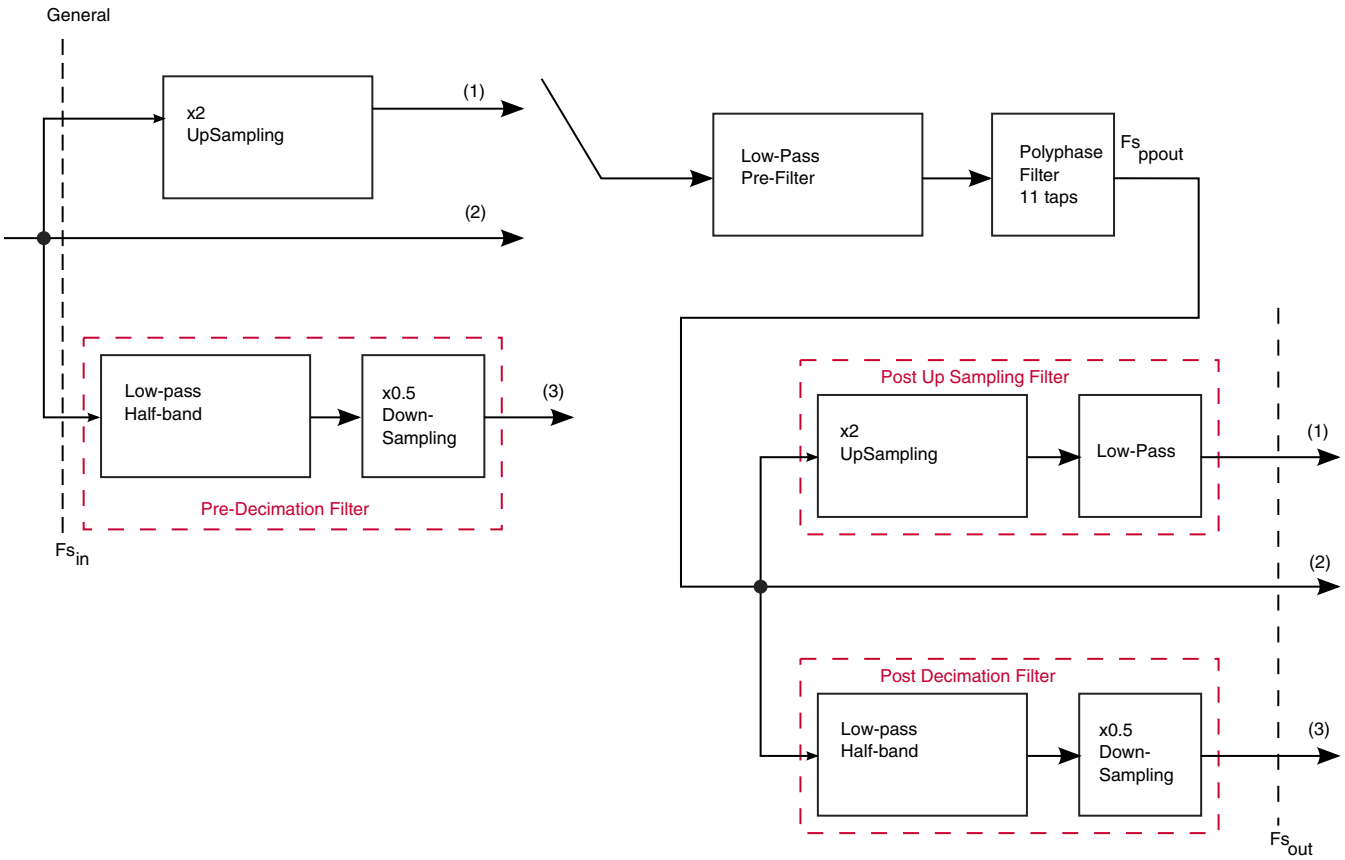


Figure 15-3. Signal processing configurations

The figure above shows the possible configurations of the ASRC. Each configuration consists of 2 to 4 stages.

- x2 up-sampling rate expander (zero insertion only) (Input Branch 1), or direct connection (Input Branch 2), or low-pass pre decimation filter (consisting of a low-pass half-band FIR filter with x0.5 downsampling rate decimator) (Input Branch 3),
- low-pass pre-filter, the low-pass bandwidth is at most 0.25 x Fs, where Fs is the sampling rate of the input signal to this low-pass pre-filter,
- polyphase filter,
- x2 post upsampling filter (consisting of a x2 up-sampling rate expander (zero insertion only) with low-pass half-band FIR filter) (Output Branch 1), or direct connection (Output Branch 2), or low-pass post decimation filter (consisting of a low-pass half-band FIR filter with x0.5 downsampling rate decimator) (Output Branch 3).

By flowing through different processing branches, and different setup of the pre-filter, this ASRC scheme can be used to handle different requirement of rate conversion.

Configuration (a): Input Branch 1+Output Branch 1: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/2$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}/2$.

Configuration (b): Input Branch 1+Output Branch 2: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/2$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}$.

Configuration (c): Input Branch 1+Output Branch 3: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/2$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = 2Fs_{out}$.

Configuration (d): Input Branch 2+Output Branch 1: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/4$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}/2$.

Configuration (e): Input Branch 2+Output Branch 2: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/4$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}$.

Configuration (f): Input Branch 2+Output Branch 3: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/4$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = 2Fs_{out}$.

Configuration (g): Input Branch 3+Output Branch 1: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/8$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}/2$.

Configuration (h): Input Branch 3+Output Branch 2: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/8$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}$.

Configuration (i): Input Branch 3+Output Branch 3: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/8$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = 2Fs_{out}$.

Table 15-6. Pre-Processing, Post-Processing Options

{Pre_Proc, Post_Proc}	Fsout (KHz)								
	8	32	44.1	48	64	88.2	96	128	192

Table continues on the next page...

Table 15-6. Pre-Processing, Post-Processing Options (continued)

Fsin (KHz)	8	{0,1}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}
	12	{0,2}	{0,1}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}
	16	{1,2}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}
	24	{1,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}	{0,0}	{0,0}
	32	{1,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}	{0,0}
	44.1	{2,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}
	48	{2,2}	{0,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}
	64	{2,2}	{0,2}	{0,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}
	88.2	NA	{1,2}	{1,2}	{1,2}	{1,1}	{1,1}	{1,1}	{1,1}	{1,1}
	96	NA	{1,2}	{1,2}	{1,2}	{1,1}	{1,1}	{1,1}	{1,1}	{1,1}
	128	NA	{1,2}	{1,2}	{1,2}	{1,1}	{1,1}	{1,1}	{1,1}	{1,1}
	192	NA	{2,2}	{2,2}	{2,2}	{2,1}	{2,1}	{2,1}	{2,1}	{2,1}

Comments:

In the {Pre_Proc, Post_Proc} pair, the meaning of the values are:

Pre_Proc:

- 0 --- Pre-processing Branch 1 as shown in Figure 15-3
- 1 --- Pre-processing Branch 2 as shown in Figure 15-3
- 2 --- Pre-processing Branch 3 as shown in Figure 15-3, decimation-by-2

Post_Proc:

- 0 --- Post-processing Branch 1 as shown in Figure 15-3
- 1 --- Post-processing Branch 2 as shown in Figure 15-3
- 2 --- Post-processing Branch 3 as shown in Figure 15-3

15.4.1.2 Operation of the Filter

15.4.1.2.1 Support of Physical Clocks

This design supports physical sampling clocks. The clocks can be provided by Sony/ Phillips digital interface (SPDIF), enhanced serial audio interface (ESAI), standard serial interface (media local bus controller (, Core master clock derivative as ASRCK1 .

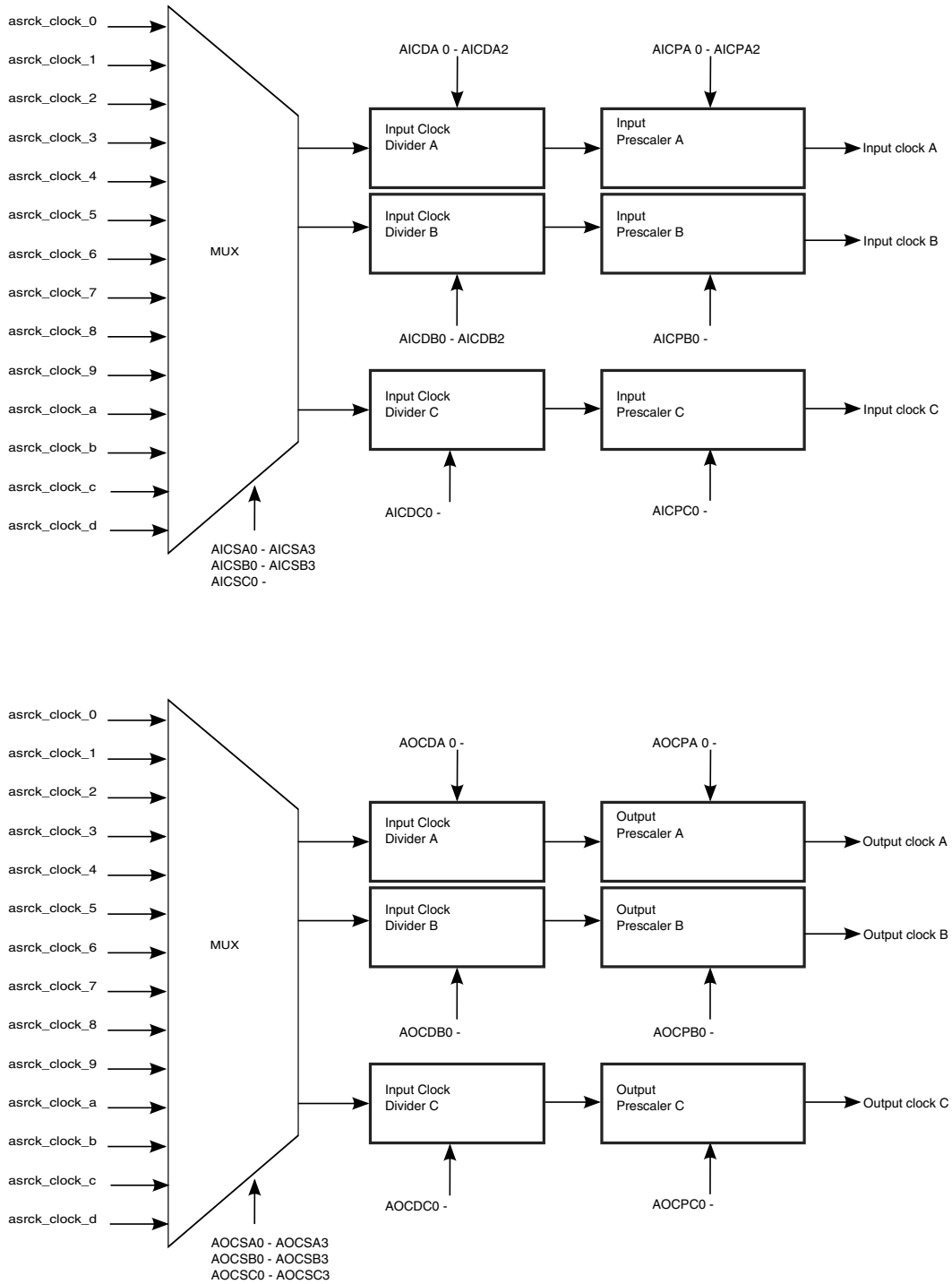


Figure 15-4. Clock Source Selector & Divider

Software can set the ASRC Clock Source Register (ASRCSR) and the Clock Divider Register to select the desired clock source and divide it to the needed sample rate clock for use by the ASRC. The clocks have the following restriction. If the prescaler is set to 1, the clock divider can only be set to 1 and the clock source must have a 50% duty cycle.

15.5 Startup Procedure

The following example shows the normal setup procedure for the ASRC block.

```
#include "asrc_common.h"

#include "stdio.h"

#include "soc_api.h"

int incnt=0;

int outcnt=0;

#include "wy_ideal_ratio_dataini_part.h"

WORD   IdealRatio_High=0x04; //

WORD   IdealRatio_Low=0x0; //

void asrc_config_alloc(WORD ASRCTR_VAL, WORD ASRIER_VAL, WORD ASRCNCR_VAL,
                           WORD ASRCFG_VAL, WORD ASRCDR1_VAL, WORD ASRCDR2_VAL,
                           WORD ASRCSR_VAL)

{  // Disable ASRC

    reg32_write(ASRC_ASRCTR,0x0);

    reg32_write(ASRC_ASRCTR,ASRCTR_VAL);

    reg32_write(ASRC_ASRIER,ASRIER_VAL);

    reg32_write(ASRC_ASRIEM,0x0);

    reg32_write(ASRC_ASRCNCR,ASRCNCR_VAL);

    reg32_write(ASRC_ASRCFG,ASRCFG_VAL);

    reg32_write(ASRC_ASRCDR1,ASRCDR1_VAL);

    reg32_write(ASRC_ASRCDR2,ASRCDR2_VAL);

    reg32_write(ASRC_ASRCSR, ASRCSR_VAL);

    reg32_write(ASRC_ASRPM1, 0x7fffff);

    reg32_write(ASRC_ASRPM2, 0x255555);

    reg32_write(ASRC_ASRPM3, 0xff7280);

    reg32_write(ASRC_ASRPM4, 0xff7280);

    reg32_write(ASRC_ASRPM5, 0xff7280);

    reg32_write(ASRC_ASRQFIFO1,0x001f00);

//  reg32_write(ASRC_ASRMCRA,0x001f00);

//  reg32_write(ASRC_ASRMCRB,0x001f00);

//  reg32_write(ASRC_ASRMCRC,0x001f00);
```

```
}
 void sim_ideal_ratio()
  {
    WORD tmp32bit;
    #define ASRSTR_AIDEA_MASK  0x1
    #define ASRSTR_AODFA_MASK  0x1 <<3
    #define ASRSTR_AOLE_MASK   0x1<<6
    #define ASRCTR_DBG_EN             1<<23
    #define ASRCTR_IDRA               1<<13
    #define ASRCTR_USRA               1<<14
    #define ASRC_CLK_PRED_RSTRICTED 0<<28
    #define ASRC_CLK_PRED_DFLT      1<<28 // default: 596MHz div by 2
    #define ASRC_CLK_PRED_DIV3      2<<28 // 596MHz div by 3
    #define ASRC_CLK_PRED_DIV4      3<<28 // 596MHz div by 4
    #define ASRC_CLK_PRED_DIV5      4<<28 // 596MHz div by 5
    #define ASRC_CLK_PRED_DIV6      5<<28 // 596MHz div by 6
    #define ASRC_CLK_PRED_DIV7      6<<28 // 596MHz div by 7
    #define ASRC_CLK_PRED_DIV8      7<<28 // 596MHz div by 8
    #define ECSPI_CLK_PRED_DFLT     1<<25
    #define ECSPI_CLK_PODF_DFLT     1<<19
    #define ASRC_CLK_PODF_DIV1      0<<9  // pred output divide by 1 again
    #define ASRC_CLK_PODF_DIV2      1<<9  // pred output divide by 2 again
    #define ASRC_CLK_PODF_DIV3      2<<9  // pred output divide by 3 again
    #define ASRC_CLK_PODF_DIV4      3<<9  // pred output divide by 4 again
    #define ASRC_CLK_PODF_DFLT      4<<9  // default: pred output divide by 5 again
    #define ASRC_CLK_PODF_DIV6      5<<9  // pred output divide by 6 again
    #define ASRC_CLK_PODF_DIV7      6<<9  // pred output divide by 7 again
    #define ASRC_CLK_PODF_DIV25     24<<9  // pred output divide by 7 again
    #define IEEE_CLK_PRED_DFLT      1<<6  //
    #define IEEE_CLK_PODF_DFLT      4     //
    #define ASR_HFA_HFB         0
    #define ASR_PREMODA_UP2         0<<6
    #define ASR_PREMODA_DIR         1<<6
```

```
      #define ASR_PREMODA_DN2         2<<6

      #define ASR_PREMODA_PAS         3<<6

      #define ASR_POSTMODA_UP2         0<<8

      #define ASR_POSTMODA_DIR         1<<8

      #define ASR_POSTMODA_DN2         2<<8

      // program CCM for ASRC core clocks

       reg32_write(CCM_CCGR7, 0xffffffff); // enable all perihperal clocks during all modes,
except stop mode

       reg32setbit(CCM_CSCMR2, 21); // Selector for asrc clock multiplexer

       //   0 ipp_asrc_ext

       //   1 pll4_sw_clk(Default): This one is choosen

       reg32_write(CCM_CSCDR2, ASRC_CLK_PRED_DIV8|ECSPI_CLK_PRED_DFLT|ECSPI_CLK_PODF_DFLT|
ASRC_CLK_PODF_DIV25|IEEE_CLK_PRED_DFLT|IEEE_CLK_PODF_DFLT);

       // Disable the ASRC

       reg32_write(ASRC_ASRCTR, 0x0);

       // program AHB clocks

       tmp32bit = reg32_read(CCM_CBCDR);

       tmp32bit = tmp32bit & (~0x00001C00);

       //tmp32bit = tmp32bit | (0x00000C00); // AHB 100MHz // divided-by-4

       //tmp32bit = tmp32bit | (0x00001000); // AHB 80MHz // divided-by-5

       //tmp32bit = tmp32bit | (0x00001400); // AHB 66MHz // divided-by-6

       //tmp32bit = tmp32bit | (0x00001800); // AHB 57MHz // divided-by-7

       tmp32bit = tmp32bit | (0x00001C00); // AHB 50MHz // divided-by-8

       reg32_write(CCM_CBCDR, tmp32bit); // enable all perihperal clocks during all modes,
except stop mode

       while ( (reg32_read(CCM_CDHIPR) & 0x00008) != 0);

              asrc_config_alloc ( 0x002 | ASRCTR_IDRA | ASRCTR_USRA,    // ASRCTR_VAL, Use
Ratio input, use ideal ratio, Enable Pair A,

              0x0, //0x09,                                              // ASRIER_VAL, Open
PairA input and output interrupt

              0x002,     // ASRCNCR_VAL, assign 2 channels to Pair A

              ASR_PREMODA_DIR | ASR_POSTMODA_DIR | ASR_HFA_HFB,         // ASRCFG_VAL,
POSTMODA=downsampling by 2 ; PREMODA=downsampling by 2

              0x03b03b , // ASRCDR1_VAL, AOCPA=3(FoutA/(2^3)); AICPA=3(FinA/(2^3));
AOCDA=7(div 8); AICDA=7(div 8);

              0x0  ,     // ASRCDR2_VAL,
```

```
            0x00d00d   // ASRCSR_VAL,  AOCSA=d: bit clock d: ASRCK1 clk from CCM; AICSA=d:
bit clock d: ASRCK1 clock from CCM;

            );

      reg32_write(ASRC_ASRIDRHA, 0x04); //

      reg32_write(ASRC_ASRIDRLA, 0x0); // Ideal Ratio is set to be 1.

      #define OUTFIFO_THRESH_0  8<<12

      #define INFIFO_THRESH_1   32

      reg32_write(ASRC_ASRMCRA, OUTFIFO_THRESH_0 | INFIFO_THRESH_1);

      reg32clrbit(ASRC_ASRMCRA, 23);   // zeroize Pair A buffers

      reg32setbit(ASRC_ASRMCRA, 21);   // stall conversion in case of near full/near empty
condition

      reg32clrbit(ASRC_ASRMCRA, 20); // Do not bypass polyA filter

      // Set ASRC Interrupt

      //CAPTURE_INTERRUPT(ASRC_INT_ROUTINE, asrc_handler);

      //enable_hdler(ASRC_INT_NUM);

      disable_hdler(ASRC_INT_NUM);

      incnt=0;

      outcnt=0;

      reg32setbit(ASRC_ASRCTR,0); // enable ASRC

      #define ASRCFG_INIA_FINISH  0x1<<21

      while ( (reg32_read(ASRC_ASRCFG) & ASRCFG_INIA_FINISH) == 0); // wait for ini finished.

      // Polling

          while (outcnt < 100) <

      {

              int ii;

              if ( (reg32_read(ASRC_ASRSTR) & ASRSTR_AIDEA_MASK) != 0 )

               {

                 for (ii=0;ii<2;ii++)</codeblock

                   {

                      reg32_write(ASRC_ASRDIA,asrc_input_array[incnt]);     // feed in input
data

                      reg32_write(ASRC_ASRDIA,asrc_input_array[incnt]);     // feed in input
data

                      incnt=(incnt+1)%128;

                   }
```

```
                }
            if ( (reg32_read(ASRC_ASRSTR) & ASRSTR_AODFA_MASK) != 0 )
             {
                for (ii=0;ii<2;ii++)<
                  {
                    WORD TempRdOut;
                    TempRdOut=reg32_read(ASRC_ASRDOA);  // get output data
                    TempRdOut=reg32_read(ASRC_ASRDOA);  // get output data
                    outcnt=outcnt+1;
                  }
               }
            if ( (reg32_read(ASRC_ASRSTR) & ASRSTR_AOLE_MASK) != 0 )
             {
                reg32_write(ASRC_ASRSTR,ASRSTR_AOLE_MASK);      // clear overloading
errors
              }
           }
        reg32clrbit(ASRC_ASRCTR,0); // disable ASRC
   }
```

15.6 Programmable Registers

All useful registers are listed in the memory map below. The access of undefined registers will behave as normal registers.

All the interface registers are LSB aligned except the input FIFOs and the output FIFOs, and each register has only 24 effective bits.

The input FIFO and output FIFO word alignment can be defined using ASRMCR1{A,B,C} registers in 32-bit interface system.

ASRC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5002_C000	ASRC Control Register (ASRC_ASRCTR)	32	R/W	0000_0000h	15.6.1/ 731
5002_C004	ASRC Interrupt Enable Register (ASRC_ASRIER)	32	R/W	0000_0000h	15.6.2/ 733
5002_C00C	ASRC Channel Number Configuration Register (ASRC_ASRCNCR)	32	R/W	0000_0000h	15.6.3/ 734
5002_C010	ASRC Filter Configuration Status Register (ASRC_ASRCFG)	32	R/W	0000_0000h	15.6.4/ 736
5002_C014	ASRC Clock Source Register (ASRC_ASRCSR)	32	R/W	0000_0000h	15.6.5/ 738
5002_C018	ASRC Clock Divider Register 1 (ASRC_ASRCDR1)	32	R/W	0000_0000h	15.6.6/ 742
5002_C01C	ASRC Clock Divider Register 2 (ASRC_ASRCDR2)	32	R/W	0000_0000h	15.6.7/ 743
5002_C020	ASRC Status Register (ASRC_ASRSTR)	32	R	0000_0000h	15.6.8/ 744
5002_C040	ASRC Parameter Register n (ASRC_ASRPMn1)	32	R/W	0000_0000h	15.6.9/ 747
5002_C044	ASRC Parameter Register n (ASRC_ASRPMn2)	32	R/W	0000_0000h	15.6.9/ 747
5002_C048	ASRC Parameter Register n (ASRC_ASRPMn3)	32	R/W	0000_0000h	15.6.9/ 747
5002_C04C	ASRC Parameter Register n (ASRC_ASRPMn4)	32	R/W	0000_0000h	15.6.9/ 747
5002_C050	ASRC Parameter Register n (ASRC_ASRPMn5)	32	R/W	0000_0000h	15.6.9/ 747
5002_C054	ASRC ASRC Task Queue FIFO Register 1 (ASRC_ASRTFR1)	32	R/W	0000_0000h	15.6.10/ 748
5002_C05C	ASRC Channel Counter Register (ASRC_ASRCCR)	32	R/W	0000_0000h	15.6.11/ 749
5002_C060	ASRC Data Input Register for Pair x (ASRC_ASRDIA)	32	W	0000_0000h	15.6.12/ 750
5002_C064	ASRC Data Output Register for Pair x (ASRC_ASRDOA)	32	R	0000_0000h	15.6.13/ 750
5002_C068	ASRC Data Input Register for Pair x (ASRC_ASRDIB)	32	W	0000_0000h	15.6.12/ 750
5002_C06C	ASRC Data Output Register for Pair x (ASRC_ASRDOB)	32	R	0000_0000h	15.6.13/ 750
5002_C070	ASRC Data Input Register for Pair x (ASRC_ASRDIC)	32	W	0000_0000h	15.6.12/ 750

Table continues on the next page...

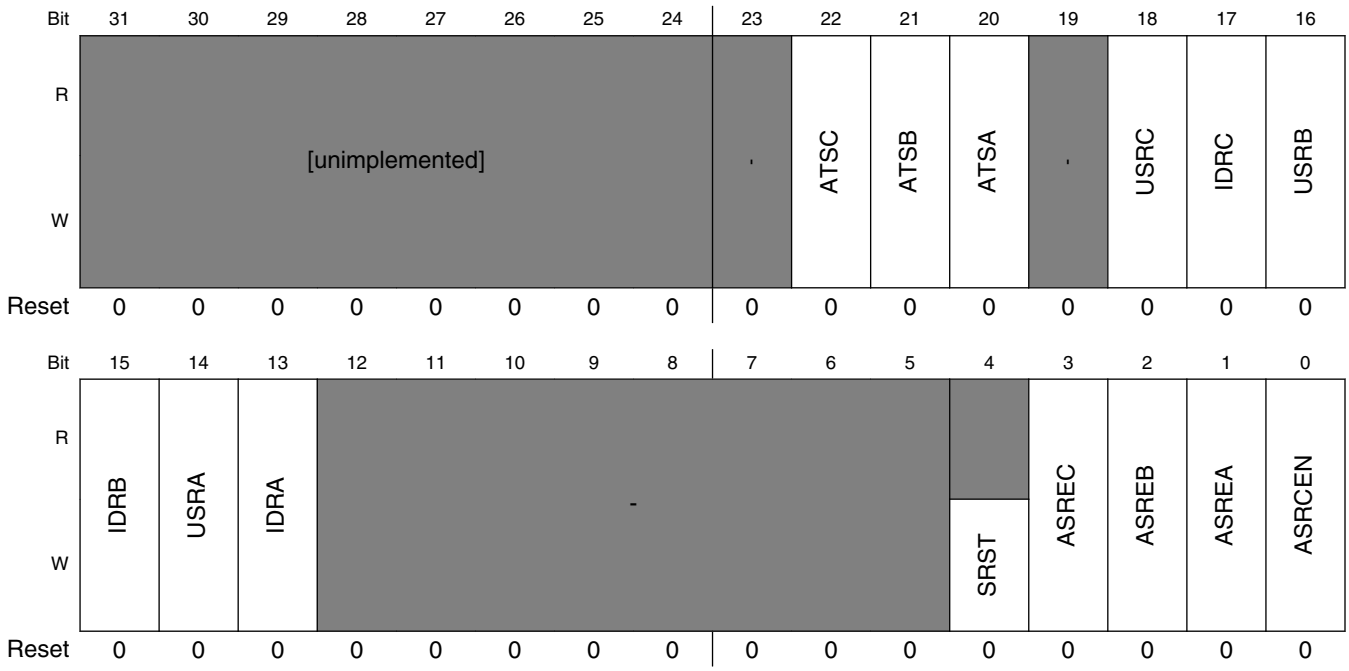
ASRC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5002_C074	ASRC Data Output Register for Pair x (ASRC_ASRDOC)	32	R	0000_0000h	15.6.13/ 750
5002_C080	ASRC Ideal Ratio for Pair A-High Part (ASRC_ASRIDRHA)	32	R/W	0000_0000h	15.6.14/ 751
5002_C084	ASRC Ideal Ratio for Pair A -Low Part (ASRC_ASRIDRLA)	32	R/W	0000_0000h	15.6.15/ 751
5002_C088	ASRC Ideal Ratio for Pair B-High Part (ASRC_ASRIDRHB)	32	R/W	0000_0000h	15.6.16/ 752
5002_C08C	ASRC Ideal Ratio for Pair B-Low Part (ASRC_ASRIDRLB)	32	R/W	0000_0000h	15.6.17/ 752
5002_C090	ASRC Ideal Ratio for Pair C-High Part (ASRC_ASRIDRHC)	32	R/W	0000_0000h	15.6.18/ 753
5002_C094	ASRC Ideal Ratio for Pair C-Low Part (ASRC_ASRIDRLC)	32	R/W	0000_0000h	15.6.19/ 753
5002_C098	ASRC 76kHz Period in terms of ASRC processing clock (ASRC_ASR76K)	32	R/W	0000_0A47h	15.6.20/ 754
5002_C09C	ASRC 56kHz Period in terms of ASRC processing clock (ASRC_ASR56K)	32	R/W	0000_0DF3h	15.6.21/ 755
5002_C0A0	ASRC Misc Control Register for Pair A (ASRC_ASRMCRA)	32	R/W	0000_0000h	15.6.22/ 756
5002_C0A4	ASRC FIFO Status Register for Pair A (ASRC_ASRFSTA)	32	R	0000_0000h	15.6.23/ 758
5002_C0A8	ASRC Misc Control Register for Pair B (ASRC_ASRMCRB)	32	R/W	0000_0000h	15.6.24/ 759
5002_C0AC	ASRC FIFO Status Register for Pair B (ASRC_ASRFSTB)	32	R	0000_0000h	15.6.25/ 761
5002_C0B0	ASRC Misc Control Register for Pair C (ASRC_ASRMCRC)	32	R/W	0000_0000h	15.6.26/ 762
5002_C0B4	ASRC FIFO Status Register for Pair C (ASRC_ASRFSTC)	32	R	0000_0000h	15.6.27/ 764
5002_C0C0	ASRC Misc Control Register 1 for Pair X (ASRC_ASRMCR1A)	32	R/W	0000_0000h	15.6.28/ 765
5002_C0C4	ASRC Misc Control Register 1 for Pair X (ASRC_ASRMCR1B)	32	R/W	0000_0000h	15.6.28/ 765
5002_C0C8	ASRC Misc Control Register 1 for Pair X (ASRC_ASRMCR1C)	32	R/W	0000_0000h	15.6.28/ 765

15.6.1 ASRC Control Register (ASRC_ASRCTR)

The ASRC control register (ASRCTR) is a 24-bit read/write register that controls the ASRC operations.

Address: ASRC_ASRCTR is 5002_C000h base + 0h offset = 5002_C000h



ASRC_ASRCTR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23 -	Reserved. Should be written as zero for compatibility.
22 ATSC	ASRC Pair C Automatic Selection For Processing Options When this bit is 1, pair C will automatic update its pre-processing and post-processing options (ASRCFG: PREMODC, ASRCFG:POSTMODC see ASRC Misc Control Register 1 for Pair CASRC Filter Configuration Status Register) based on the frequencies it detected. To use this option, the two parameter registers(ASR76K and ASR56K) should be set correctly (see ASRC Misc Control Register 1 for Pair CASRC 76kHz Period in terms of ASRC processing clock and ASRC Misc Control Register 1 for Pair CASRC 56kHz Period in terms of ASRC processing clock). When this bit is 0, the user is responsible for choosing the proper processing options for pair C. This bit should be disabled when {USRC, IDRC}={1,1}.
21 ATSB	ASRC Pair B Automatic Selection For Processing Options When this bit is 1, pair B will automatic update its pre-processing and post-processing options (ASRCFG: PREMODB, ASRCFG:POSTMODB see ASRC Misc Control Register 1 for Pair CASRC Filter Configuration Status Register) based on the frequencies it detected. To use this option, the two

Table continues on the next page...

ASRC_ASRCTR field descriptions (continued)

Field	Description
	parameter registers(ASR76K and ASR56K) should be set correctly (see ASRC Misc Control Register 1 for Pair CASRC 76kHz Period in terms of ASRC processing clock and ASRC Misc Control Register 1 for Pair CASRC 56kHz Period in terms of ASRC processing clock). When this bit is 0, the user is responsible for choosing the proper processing options for pair B. This bit should be disabled when {USRB, IDRB}={1,1}.
20 ATSA	ASRC Pair A Automatic Selection For Processing Options When this bit is 1, pair A will automatic update its pre-processing and post-processing options (ASRCFG: PREMODA, ASRCFG:POSTMODA see ASRC Misc Control Register 1 for Pair CASRC Filter Configuration Status Register) based on the frequencies it detected. To use this option, the two parameter registers(ASR76K and ASR56K) should be set correctly (see ASRC Misc Control Register 1 for Pair CASRC 76kHz Period in terms of ASRC processing clock and ASRC Misc Control Register 1 for Pair CASRC 56kHz Period in terms of ASRC processing clock). When this bit is 0, the user is responsible for choosing the proper processing options for pair A. This bit should be disabled when {USRA, IDRA}={1,1}.
19 -	Reserved. Should be written as zero for compatibility.
18 USRC	Use Ratio for Pair C Use ratio as the input to ASRC. This bit is used in conjunction with IDRC control bit.
17 IDRC	Use Ideal Ratio for Pair C When USRC=0, this bit has no usage. When USRC=1 and IDRC=0, ASRC internal measured ratio will be used. When USRC=1 and IDRC=1, the idea ratio from the interface register ASRIDRHC, ASRIDRLC will be used. It is suggested to manually set ASRCFG:POSTMODC, ASRCFG:PREMODC according to Table 15-6 in this case.
16 USRB	Use Ratio for Pair B Use ratio as the input to ASRC. This bit is used in conjunction with IDRB control bit.
15 IDRB	Use Ideal Ratio for Pair B When USRB=0, this bit has no usage. When USRB=1 and IDRB=0, ASRC internal measured ratio will be used. When USRB=1 and IDRB=1, the idea ratio from the interface register ASRIDRHB, ASRIDRLB will be used.It is suggested to manually set ASRCFG:POSTMODB, ASRCFG:PREMODB according to Table 15-6 in this case.
14 USRA	Use Ratio for Pair A Use ratio as the input to ASRC. This bit is used in conjunction with IDRA control bit.
13 IDRA	Use Ideal Ratio for Pair A When USRA=0, this bit has no usage. When USRA=1 and IDRA=0, ASRC internal measured ratio will be used. When USRA=1 and IDRA=1, the idea ratio from the interface register ASRIDRHA, ASRIDRLA will be used. It is suggested to manually set ASRCFG:POSTMODA, ASRCFG:PREMODA according to Table 15-6 in this case.
12–5 -	Reserved. Should be written as zero for compatibility.

Table continues on the next page...

ASRC_ASRCTR field descriptions (continued)

Field	Description
4 SRST	Software Reset This bit is self-clear bit. Once it is been written as 1, it will generate a software reset signal inside ASRC. After 9 cycles of the ASRC processing clock, this reset process will stop, and this bit will be cleared automatically.
3 ASREC	ASRC Enable C Enable the operation of the conversion C of ASRC. When ASREC is cleared, operation of conversion C is disabled.
2 ASREB	ASRC Enable B Enable the operation of the conversion B of ASRC. When ASREB is cleared, operation of conversion B is disabled.
1 ASREA	ASRC Enable A Enable the operation of the conversion A of ASRC. When ASREA is cleared, operation of conversion A is disabled.
0 ASRCEN	ASRC Enable Enable the operation of ASRC.

15.6.2 ASRC Interrupt Enable Register (ASRC_ASRIER)

Address: ASRC_ASRIER is 5002_C000h base + 4h offset = 5002_C004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	[unimplemented]								-																AFPWE	AOLIE	ADOEC	ADOEB	ADOEA	ADIEC	ADIEB	ADIEA
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRIER field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–8 -	Reserved. Should be written as zero for compatibility.
7 AFPWE	FP in Wait State Interrupt Enable Enables the FP in wait state interrupt. 1 interrupt enabled 0 interrupt disabled
6 AOLIE	Overload Interrupt Enable Enables the overload interrupt.

Table continues on the next page...

ASRC_ASRIER field descriptions (continued)

Field	Description
	1 interrupt enabled 0 interrupt disabled
5 ADOEC	Data Output C Interrupt Enable Enables the data output C interrupt. 1 interrupt enabled 0 interrupt disabled
4 ADOEB	Data Output B Interrupt Enable Enables the data output B interrupt. 1 interrupt enabled 0 interrupt disabled
3 ADOEA	Data Output A Interrupt Enable Enables the data output A interrupt. 1 interrupt enabled 0 interrupt disabled
2 ADIEC	Data Input C Interrupt Enable Enables the data input C interrupt. 1 interrupt enabled 0 interrupt disabled
1 ADIEB	Data Input B Interrupt Enable Enables the data input B interrupt. 1 interrupt enabled 0 interrupt disabled
0 ADIEA	Data Input A Interrupt Enable Enables the data input A Interrupt. 1 interrupt enabled 0 interrupt disabled

15.6.3 ASRC Channel Number Configuration Register (ASRC_ASRCNCR)

The ASRC channel number configuration register (ASRCNCR) is a 24-bit read/write register that sets the number of channels used by each ASRC conversion pair.

There are 10 channels available for distribution among 3 conversion pairs, they are ordered as 0,1,...,9. The bottom [0, ANCA-1] channels are used for pair A, the top [10-ANCC, 9] channels are used for pair C, and the [ANCA, ANCA+ANCB-1] channels are allocated for pair B. In case that ANCA=0, then the [0, ANCB-1] channels are assigned for pair B.

Address: ASRC_ASRCNCR is 5002_C000h base + Ch offset = 5002_C00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	[unimplemented]								-															ANCC			ANCB			ANCA		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRCNCR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–9 -	Reserved. Should be written as zero for compatibility.
8–6 ANCC	Number of C Channels[1] 0000 0 channels in C (Pair C is disabled) 0001 1 channel in C 0010 2 channels in C 0011 3 channels in C 0100 4 channels in C 0101 5 channels in C 0110 6 channels in C 0111 7 channels in C 1000 8 channels in C 1001 9 channels in C 1010 10 channels in C 1011-1111 Should not be used.
5–3 ANCB	Number of B Channels 0000 0 channels in B (Pair B is disabled) 0001 1 channel in B 0010 2 channels in B 0011 3 channels in B 0100 4 channels in B 0101 5 channels in B 0110 6 channels in B 0111 7 channels in B 1000 8 channels in B 1001 9 channels in B 1010 10 channels in B 1011-1111 Should not be used.

Table continues on the next page...

ASRC_ASRCNCR field descriptions (continued)

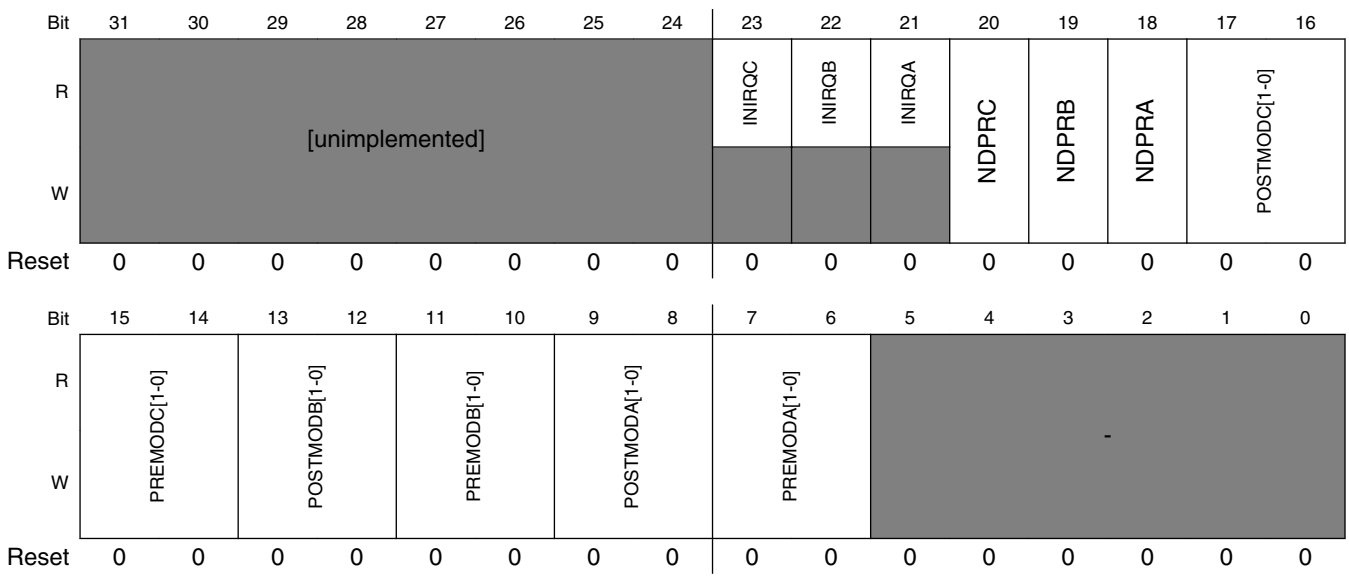
Field	Description
2–0 ANCA	Number of A Channels 0000 0 channels in A (Pair A is disabled) 0001 1 channel in A 0010 2 channels in A 0011 3 channels in A 0100 4 channels in A 0101 5 channels in A 0110 6 channels in A 0111 7 channels in A 1000 8 channels in A 1001 9 channels in A 1010 10 channels in A 1011-1111 Should not be used.

1. ANCC+ANCB+ANCA<=10. Hardware is not checking the constraint. Programmer should take the responsibility to ensure the constraint is satisfied.

15.6.4 ASRC Filter Configuration Status Register (ASRC_ASRCFG)

The ASRC configuration status register (ASRCFG) is a 24-bit read/write register that sets and/or automatically senses the ASRC operations.

Address: ASRC_ASRCFG is 5002_C000h base + 10h offset = 5002_C010h



ASRC_ASRCFG field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.

Table continues on the next page...

ASRC_ASRCFG field descriptions (continued)

Field	Description
23 INIRQC	Initialization for Conversion Pair C is served When this bit is 1, it means the initialization for conversion pair C is served. This bit is cleared by disabling the ASRC conversion pair (ASRCTR:ASREC=0 or ASRCTR:ASRCEN=0).
22 INIRQB	Initialization for Conversion Pair B is served When this bit is 1, it means the initialization for conversion pair B is served. This bit is cleared by disabling the ASRC conversion pair (ASRCTR:ASREB=0 or ASRCTR:ASRCEN=0).
21 INIRQA	Initialization for Conversion Pair A is served When this bit is 1, it means the initialization for conversion pair A is served. This bit is cleared by disabling the ASRC conversion pair (ASRCTR:ASREA=0 or ASRCTR:ASRCEN=0).
20 NDPRC	Not Use Default Parameters for RAM-stored Parameters For Conversion Pair C 0 Use default parameters for RAM-stored parameters. Override any parameters already in RAM. 1 Don't use default parameters for RAM-stored parameters. Use the parameters already stored in RAM.
19 NDPRB	Not Use Default Parameters for RAM-stored Parameters For Conversion Pair B 0 Use default parameters for RAM-stored parameters. Override any parameters already in RAM. 1 Don't use default parameters for RAM-stored parameter. Use the parameters already stored in RAM.
18 NDPRA	Not Use Default Parameters for RAM-stored Parameters For Conversion Pair A 0 Use default parameters for RAM-stored parameters. Override any parameters already in RAM. 1 Don't use default parameters for RAM-stored parameters. Use the parameters already stored in RAM.
17–16 POSTMODC[1-0]	Post-Processing Configuration for Conversion Pair C These bits will be read/write by user if ASRCTR:ATSC=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSC=1 (see ASRC Misc Control Register 1 for Pair CASRC Control Register). These bits set the selection of the post-processing configuration. 00 Select Upsampling-by-2 as defined in Signal Processing Flow. 01 Select Direct-Connection as defined in Signal Processing Flow. 10 Select Downsampling-by-2 as defined in Signal Processing Flow.
15–14 PREMODC[1-0]	Pre-Processing Configuration for Conversion Pair C These bits will be read/write by user if ASRCTR:ATSC=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSC=1 (see ASRC Misc Control Register 1 for Pair CASRC Control Register). These bits set the selection of the pre-processing configuration. 00 Select Upsampling-by-2 as defined in Signal Processing Flow 01 Select Direct-Connection as defined in Signal Processing Flow 10 Select Downsampling-by-2 as defined in Signal Processing Flow 11 Select passthrough mode. In this case, POSTMODC[1-0] have no use.
13–12 POSTMODB[1-0]	Post-Processing Configuration for Conversion Pair B These bits will be read/write by user if ASRCTR:ATSB=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSB=1 (see ASRC Misc Control Register 1 for Pair CASRC Control Register). These bits set the selection of the post-processing configuration.

Table continues on the next page...

ASRC_ASRCFG field descriptions (continued)

Field	Description
	00 Select Upsampling-by-2 as defined in Signal Processing Flow 01 Select Direct-Connection as defined in Signal Processing Flow 10 Select Downsampling-by-2 as defined in Signal Processing Flow
11–10 PREMODB[1-0]	Pre-Processing Configuration for Conversion Pair B These bits will be read/write by user if ASRCTR:ATSB=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSB=1 (see ASRC Misc Control Register 1 for Pair CASRC Control Register). These bits set the selection of the pre-processing configuration. 00 Select Upsampling-by-2 as defined in Signal Processing Flow 01 Select Direct-Connection as defined in Signal Processing Flow 10 Select Downsampling-by-2 as defined in Signal Processing Flow 11 Select passthrough mode. In this case, POSTMODB[1-0] have no use.
9–8 POSTMODA[1-0]	Post-Processing Configuration for Conversion Pair A These bits will be read/write by user if ASRCTR:ATSA=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSA=1 (see ASRC Misc Control Register 1 for Pair CASRC Control Register). These bits set the selection of the post-processing configuration. 00 Select Upsampling-by-2 as defined in Signal Processing Flow 01 Select Direct-Connection as defined in Signal Processing Flow 10 Select Downsampling-by-2 as defined in Signal Processing Flow
7–6 PREMODA[1-0]	Pre-Processing Configuration for Conversion Pair A These bits will be read/write by user if ASRCTR:ATSA=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSA=1 (see ASRC Misc Control Register 1 for Pair CASRC Control Register). These bits set the selection of the pre-processing configuration. 00 Select Upsampling-by-2 as defined in Signal Processing Flow 01 Select Direct-Connection as defined in Signal Processing Flow 10 Select Downsampling-by-2 as defined in Signal Processing Flow 11 Select passthrough mode. In this case, POSTMODA[1-0] have no use.
5–0 -	Reserved. Should be written as zero for compatibility.

15.6.5 ASRC Clock Source Register (ASRC_ASRCSR)

The ASRC clock source register (ASRCSR) is a 24-bit read/write register that controls the sources of the input and output clocks of the ASRC.

The clock connections are shown in ASRC Misc Control Register 1 for Pair CASRC Clock Source Register, also shown in Figure 15-1 :

Table 15-13. Bit Clock Definitions

Bit Clk Name	Definitions
0	ESAI RX clock

Table continues on the next page...

Table 15-13. Bit Clock Definitions (continued)

Bit Clk Name	Definitions
1	SSI-1 RX clock
2	SSI-2 RX clock
3	SSI-3 RX clock
4	SPDIF RX clock
5	MLB Bit clock
6	tied to zero
7	tied to zero
8	ESAI TX clock
9	SSI-1 TX clock
a	SSI-2 TX clock
b	SSI-3 TX clock
c	SPDIF TX clock
d	ASRCKI, clock controlled by CCM registers CSCMR2, CSCDR2

Address: ASRC_ASRCSR is 5002_C000h base + 14h offset = 5002_C014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	[unimplemented]								AOCSC				AOCSB				AOCSA				AICSC				AICSB				AICSA			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRCSR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–20 AOCSC	**Output Clock Source C** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2 0011 bit clock 3 0100 bit clock 4 0101 bit clock 5 0110 bit clock 6 0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D

Table continues on the next page...

ASRC_ASRCSR field descriptions (continued)

Field	Description
	1111 clock disabled, connected to zero any other value bit clock 0
19–16 AOCSB	**Output Clock Source B** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2 0011 bit clock 3 0100 bit clock 4 0101 bit clock 5 0110 bit clock 6 0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D 1111 clock disabled, connected to zero any other value bit clock 0
15–12 AOCSA	**Output Clock Source A** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2 0011 bit clock 3 0100 bit clock 4 0101 bit clock 5 0110 bit clock 6 0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D 1111 clock disabled, connected to zero any other value bit clock 0
11–8 AICSC	**Input Clock Source C** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2 0011 bit clock 3 0100 bit clock 4 0101 bit clock 5 0110 bit clock 6

Table continues on the next page...

ASRC_ASRCSR field descriptions (continued)

Field	Description
	0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D 1111 clock disabled, connected to zero any other value bit clock 0
7–4 AICSB	**Input Clock Source B** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2 0011 bit clock 3 0100 bit clock 4 0101 bit clock 5 0110 bit clock 6 0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D 1111 clock disabled, connected to zero any other value bit clock 0
3–0 AICSA	**Input Clock Source A** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2 0011 bit clock 3 0100 bit clock 4 0101 bit clock 5 0110 bit clock 6 0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D 1111 clock disabled, connected to zero any other value bit clock 0

15.6.6 ASRC Clock Divider Register 1 (ASRC_ASRCDR1)

The ASRC clock divider register (ASRCDR1) is a two 24-bit read/write register that controls the division factors of the ASRC input and output clock sources.

Address: ASRC_ASRCDR1 is 5002_C000h base + 18h offset = 5002_C018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	[unimplemented]								AOCDB			AOCPB			AOCDA		AOCPA				AICDB			AICPB			AICDA			AICPA		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

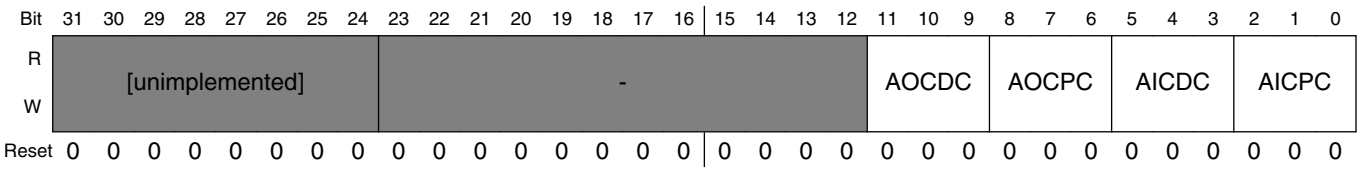
ASRC_ASRCDR1 field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–21 AOCDB	Output Clock Divider B Specify the divide ratio of the output clock divider B. The divide ratio may range from 1 to 8 (AOCDB[2:0] = 000 to 111).
20–18 AOCPB	Output Clock Prescaler B Specify the prescaling factor of the output prescaler B. The prescaling ratio may be any power of 2 from 1 to 128.
17–15 AOCDA	Output Clock Divider A Specify the divide ratio of the output clock divider A. The divide ratio may range from 1 to 8 (AOCDA[2:0] = 000 to 111).
14–12 AOCPA	Output Clock Prescaler A Specify the prescaling factor of the output prescaler A. The prescaling ratio may be any power of 2 from 1 to 128.
11–9 AICDB	Input Clock Divider B Specify the divide ratio of the input clock divider B. The divide ratio may range from 1 to 8 (AICDB[2:0] = 000 to 111).
8–6 AICPB	Input Clock Prescaler B Specify the prescaling factor of the input prescaler B. The prescaling ratio may be any power of 2 from 1 to 128.
5–3 AICDA	Input Clock Divider A Specify the divide ratio of the input clock divider A. The divide ratio may range from 1 to 8 (AICDA[2:0] = 000 to 111).
2–0 AICPA	Input Clock Prescaler A Specify the prescaling factor of the input prescaler A. The prescaling ratio may be any power of 2 from 1 to 128.

15.6.7 ASRC Clock Divider Register 2 (ASRC_ASRCDR2)

The ASRC clock divider register (ASRCDR2) is a two 24-bit read/write register that controls the division factors of the ASRC input and output clock sources.

Address: ASRC_ASRCDR2 is 5002_C000h base + 1Ch offset = 5002_C01Ch



ASRC_ASRCDR2 field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–12 -	Reserved. Should be written as zero for compatibility.
11–9 AOCDC	Output Clock Divider C Specify the divide ratio of the output clock divider C. The divide ratio may range from 1 to 8 (AOCDC[2:0] = 000 to 111).
8–6 AOCPC	Output Clock Prescaler C Specify the prescaling factor of the output prescaler C. The prescaling ratio may be any power of 2 from 1 to 128.
5–3 AICDC	Input Clock Divider C Specify the divide ratio of the input clock divider C. The divide ratio may range from 1 to 8 (AICDC[2:0] = 000 to 111).
2–0 AICPC	Input Clock Prescaler C Specify the prescaling factor of the input prescaler C. The prescaling ratio may be any power of 2 from 1 to 128.

15.6.8 ASRC Status Register (ASRC_ASRSTR)

The ASRC status register (ASRSTR) is a 24-bit read-write register used by the processor core to examine the status of the ASRC block and clear the overload interrupt request and AOLE flag bit. Read the status register will return the current state of ASRC.

Address: ASRC_ASRSTR is 5002_C000h base + 20h offset = 5002_C020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	[unimplemented]								-		DSLCNT	ATQOL	AOOLC	AOOLB	AOOLA	AIOLC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AIOLB	AIOLA	AODOC	AODOB	AODOA	AIDUC	AIDUB	AIDUA	FPWT	AOLE	AODFC	AODFB	AODFA	AIDEC	AIDEB	AIDEA
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRSTR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–22 -	Reserved. Should be written as zero for compatibility.
21 DSLCNT	DSL Counter Input to FIFO ready When set, this bit indicates that new DSL counter information is stored in the internal ASRC FIFO.When clear, this bit indicates that new DSL counter information is in the process of storage into the internal ASRC FIFO. When ASRIER:AFPWE=1, the rising edge of this signal will propose an interrupt request. Writing any value with this bit set will clear the interrupt request proposed by the rising edge of this bit.
20 ATQOL	Task Queue FIFO overload When set, this bit indicates that task queue FIFO logic is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRCTR:AOLIE as 1.

Table continues on the next page...

ASRC_ASRSTR field descriptions (continued)

Field	Description
19 AOOLC	Pair C Output Task Overload When set, this bit indicates that pair C output task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRCTR:AOLIE as 1.
18 AOOLB	Pair B Output Task Overload When set, this bit indicates that pair B output task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRCTR:AOLIE as 1.
17 AOOLA	Pair A Output Task Overload When set, this bit indicates that pair A output task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRCTR:AOLIE as 1.
16 AIOLC	Pair C Input Task Overload When set, this bit indicates that pair C input task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRCTR:AOLIE as 1.
15 AIOLB	Pair B Input Task Overload When set, this bit indicates that pair B input task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRCTR:AOLIE as 1.
14 AIOLA	Pair A Input Task Overload When set, this bit indicates that pair A input task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRCTR:AOLIE as 1.
13 AODOC	Output Data Buffer C has overflowed When set, this bit indicates that output data buffer C has overflowed. When clear, this bit indicates that output data buffer C has not overflowed
12 AODOB	Output Data Buffer B has overflowed When set, this bit indicates that output data buffer B has overflowed. When clear, this bit indicates that output data buffer B has not overflowed
11 AODOA	Output Data Buffer A has overflowed When set, this bit indicates that output data buffer A has overflowed. When clear, this bit indicates that output data buffer A has not overflowed
10 AIDUC	Input Data Buffer C has underflowed When set, this bit indicates that input data buffer C has underflowed. When clear, this bit indicates that input data buffer C has not underflowed.
9 AIDUB	Input Data Buffer B has underflowed When set, this bit indicates that input data buffer B has underflowed. When clear, this bit indicates that input data buffer B has not underflowed.

Table continues on the next page...

ASRC_ASRSTR field descriptions (continued)

Field	Description
8 AIDUA	Input Data Buffer A has underflowed When set, this bit indicates that input data buffer A has underflowed. When clear, this bit indicates that input data buffer A has not underflowed.
7 FPWT	FP is in wait states This bit is for debug only. When set, this bit indicates that ASRC is in wait states. When clear, this bit indicates that ASRC is not in wait states. If ASRCTR:AFPWE=1 and ASRCTR:ASDBG=1, an interrupt will be proposed when this bit is set.
6 AOLE	Overload Error Flag When set, this bit indicates that the task rate is too high for the ASRC to handle. The reasons for overload may be: - too high input clock frequency, - too high output clock frequency, - incorrect selection of the pre-filter, - low ASRC processing clock, - too many channels, - underrun, - or any combination of the reasons above. Since the ASRC uses the same hardware resources to perform various tasks, the real reason for the overload is not straight forward, and it should be carefully analyzed by the programmer. If ASRCTR:AOLIE=1, an interrupt will be proposed when this bit is set. Write any value with this bit set as one into the status register will clear this bit and the interrupt request proposed by this bit.
5 AODFC	Number of data in Output Data Buffer C is greater than threshold When set, this bit indicates that number of data already existing in ASRDORC is greater than threshold and the processor can read data from ASRDORC. When AODFC is set, the ASRC generates data output C interrupt request to the processor, if enabled (that is, ASRCTR:ADOEC = 1). A DMA request is always generated when the AODFC bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.
4 AODFB	Number of data in Output Data Buffer B is greater than threshold When set, this bit indicates that number of data already existing in ASRDORB is greater than threshold and the processor can read data from ASRDORB. When AODFB is set, the ASRC generates data output B interrupt request to the processor, if enabled (that is, ASRCTR:ADOEB = 1). A DMA request is always generated when the AODFB bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.
3 AODFA	Number of data in Output Data Buffer A is greater than threshold When set, this bit indicates that number of data already existing in ASRDORA is greater than threshold and the processor can read data from ASRDORA. When AODFA is set, the ASRC generates data output A interrupt request to the processor, if enabled (that is, ASRCTR:ADOEA = 1). A DMA request is always generated when the AODFA bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.

Table continues on the next page...

ASRC_ASRSTR field descriptions (continued)

Field	Description
2 AIDEC	Number of data in Input Data Buffer C is less than threshold When set, this bit indicates that number of data still available in ASRDIRC is less than threshold and the processor can write data to ASRDIRC. When AIDEC is set, the ASRC generates data input C interrupt request to the processor, if enabled (that is, ASRCTR:ADIEC = 1). A DMA request is always generated when the AIDEC bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.
1 AIDEB	Number of data in Input Data Buffer B is less than threshold When set, this bit indicates that number of data still available in ASRDIRB is less than threshold and the processor can write data to ASRDIRB. When AIDEB is set, the ASRC generates data input B interrupt request to the processor, if enabled (that is, ASRCTR:ADIEB = 1). A DMA request is always generated when the AIDEB bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.
0 AIDEA	Number of data in Input Data Buffer A is less than threshold When set, this bit indicates that number of data still available in ASRDIRA is less than threshold and the processor can write data to ASRDIRA. When AIDEA is set, the ASRC generates data input A interrupt request to the processor, if enabled (that is, ASRCTR:ADIEA = 1). A DMA request is always generated when the AIDEA bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.

15.6.9 ASRC Parameter Register n (ASRC_ASRPMn*n*)

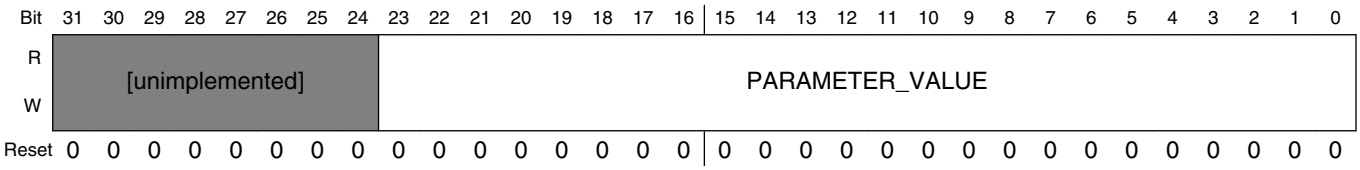
Parameter registers determine the performance of ASRC.

The parameter registers must be initialized by software before ASRC is enabled. Recommended values are given in ASRC Misc Control Register 1 for Pair CASRC Parameter Register n below,

Table 15-18. ASRC Parameter Registers (ASRPM1~ASRPM5)

Register	Offset	Access	Reset Value	Recommend Value
asrcpm1	0x40	R/W	0x00_0000	0x7fffff
asrcpm2	0x44	R/W	0x00_0000	0x255555
asrcpm3	0x48	R/W	0x00_0000	0xff7280
asrcpm4	0x4C	R/W	0x00_0000	0xff7280
asrcpm5	0x50	R/W	0x00_0000	0xff7280

Addresses: ASRC_ASRPMn1 is 5002_C000h base + 40h offset = 5002_C040h
ASRC_ASRPMn2 is 5002_C000h base + 44h offset = 5002_C044h
ASRC_ASRPMn3 is 5002_C000h base + 48h offset = 5002_C048h
ASRC_ASRPMn4 is 5002_C000h base + 4Ch offset = 5002_C04Ch
ASRC_ASRPMn5 is 5002_C000h base + 50h offset = 5002_C050h



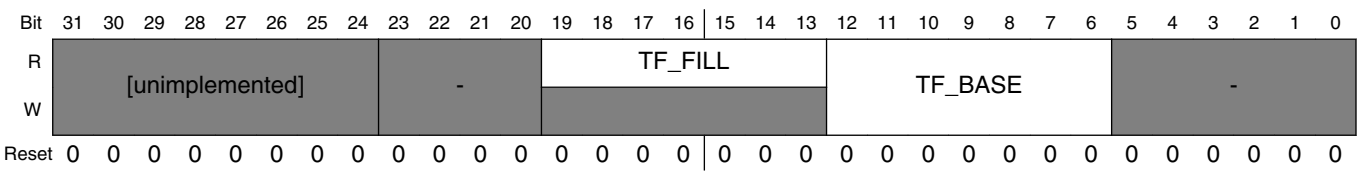
ASRC_ASRPMn*n* field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–0 PARAMETER_ VALUE	See recommended values table.

15.6.10 ASRC ASRC Task Queue FIFO Register 1 (ASRC_ASRTFR1)

The register defines and shows the parameters for ASRC inner task queue FIFOs.

Address: ASRC_ASRTFR1 is 5002_C000h base + 54h offset = 5002_C054h



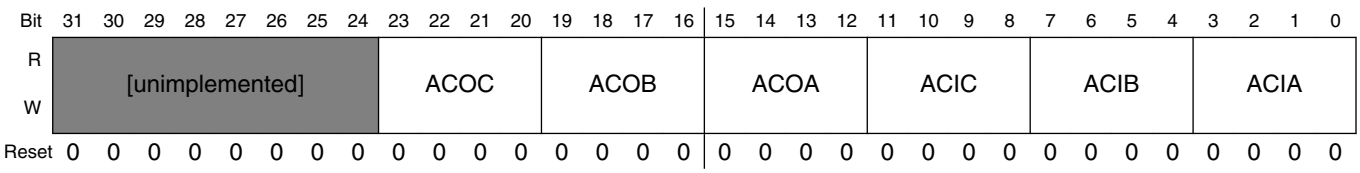
ASRC_ASRTFR1 field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–20 -	Reserved. Should be written as zero for compatibility.
19–13 TF_FILL	Current number of entries in task queue FIFO.
12–6 TF_BASE	Base address for task queue FIFO. Set to 0x7C.
5–0 -	Reserved. Should be written as zero for compatibility.

15.6.11 ASRC Channel Counter Register (ASRC_ASRCCR)

The ASRC channel counter register (ASRCCR) is a 24-bit read/write register that sets and reflects the current specific input/output FIFO being accessed through shared peripheral bus for each ASRC conversion pair.

Address: ASRC_ASRCCR is 5002_C000h base + 5Ch offset = 5002_C05Ch



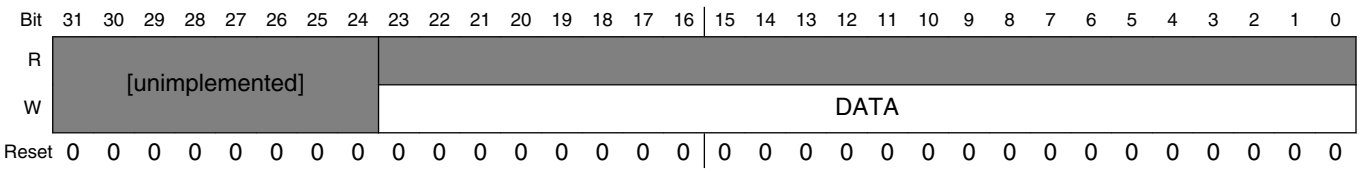
ASRC_ASRCCR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–20 ACOC	The channel counter for Pair C's output FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair C's output FIFO's usage. The value can be any value between [0, ANCC-1]
19–16 ACOB	The channel counter for Pair B's output FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair B's output FIFO's usage. The value can be any value between [0, ANCB-1]
15–12 ACOA	The channel counter for Pair A's output FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair A's output FIFO's usage. The value can be any value between [0, ANCA-1]
11–8 ACIC	The channel counter for Pair C's input FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair C's input FIFO's usage. The value can be any value between [0, ANCC-1]
7–4 ACIB	The channel counter for Pair B's input FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair B's input FIFO's usage. The value can be any value between [0, ANCB-1]
3–0 ACIA	The channel counter for Pair A's input FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair A's input FIFO's usage. The value can be any value between [0, ANCA-1]

15.6.12 ASRC Data Input Register for Pair *x* (ASRC_ASRDI*n*)

These registers are the interface registers for the audio data input of pair A,B,C respectively. They are backed by FIFOs.

Addresses: ASRC_ASRDIA is 5002_C000h base + 60h offset = 5002_C060h
ASRC_ASRDIB is 5002_C000h base + 68h offset = 5002_C068h
ASRC_ASRDIC is 5002_C000h base + 70h offset = 5002_C070h



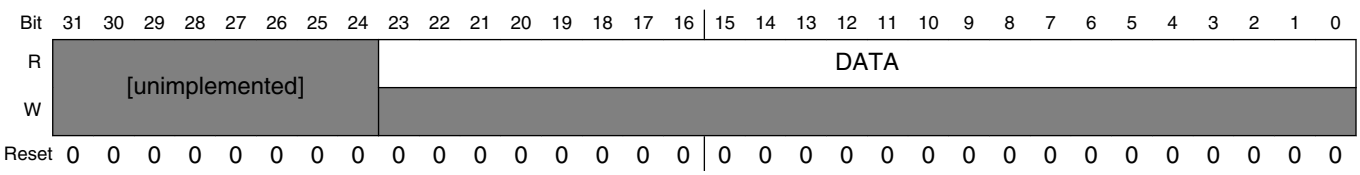
ASRC_ASRDI*n* field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–0 DATA	Audio data input

15.6.13 ASRC Data Output Register for Pair *x* (ASRC_ASRDO*n*)

These registers are the interface registers for the audio data output of pair A,B,C respectively. They are backed by FIFOs.

Addresses: ASRC_ASRDOA is 5002_C000h base + 64h offset = 5002_C064h
ASRC_ASRDOB is 5002_C000h base + 6Ch offset = 5002_C06Ch
ASRC_ASRDOC is 5002_C000h base + 74h offset = 5002_C074h



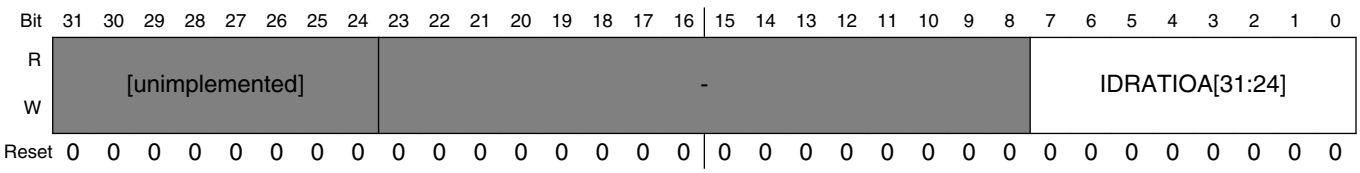
ASRC_ASRDO*n* field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–0 DATA	Audio data output

15.6.14 ASRC Ideal Ratio for Pair A-High Part (ASRC_ASRIDRHA)

The ideal ratio registers (ASRIDRHA, ASRIDRLA) hold the ratio value IDRATIOA. IDRATIOA = FsinA/FsoutA = TsoutA/TsinA is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRA, IDRA}=2'b11.

Address: ASRC_ASRIDRHA is 5002_C000h base + 80h offset = 5002_C080h



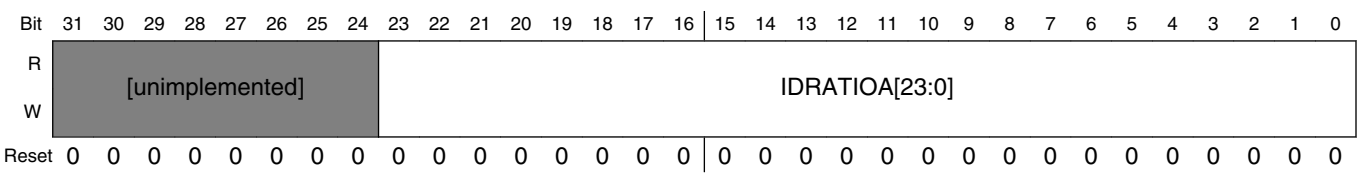
ASRC_ASRIDRHA field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–8 -	Reserved
7–0 IDRATIOA[31:24]	IDRATIOA[31:24]. High part of ideal ratio value for pair A

15.6.15 ASRC Ideal Ratio for Pair A -Low Part (ASRC_ASRIDRLA)

The ideal ratio registers (ASRIDRHA, ASRIDRLA) hold the ratio value IDRATIOA. IDRATIOA = FsinA/FsoutA = TsoutA/TsinA is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRA, IDRA}=2'b11.

Address: ASRC_ASRIDRLA is 5002_C000h base + 84h offset = 5002_C084h



ASRC_ASRIDRLA field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–0 IDRATIOA[23:0]	IDRATIOA[23:0]. Low part of ideal ratio value for pair A

15.6.16 ASRC Ideal Ratio for Pair B-High Part (ASRC_ASRIDRHB)

The ideal ratio registers (ASRIDRHB, ASRIDRLB) hold the ratio value IDRATIOB. IDRATIOB = FsinB/FsoutB = TsoutB/TsinB is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRB, IDRB}=2'b11.

Address: ASRC_ASRIDRHB is 5002_C000h base + 88h offset = 5002_C088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	[unimplemented]								-																IDRATIOB[31:24]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRIDRHB field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–8 -	Reserved
7–0 IDRATIOB[31:24]	IDRATIOB[31:24]. High part of ideal ratio value for pair B.

15.6.17 ASRC Ideal Ratio for Pair B-Low Part (ASRC_ASRIDRLB)

The ideal ratio registers (ASRIDRHB, ASRIDRLB) hold the ratio value IDRATIOB. IDRATIOB = FsinB/FsoutB = TsoutB/TsinB is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRB, IDRB}=2'b11.

Address: ASRC_ASRIDRLB is 5002_C000h base + 8Ch offset = 5002_C08Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	[unimplemented]								IDRATIOB[23:0]																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

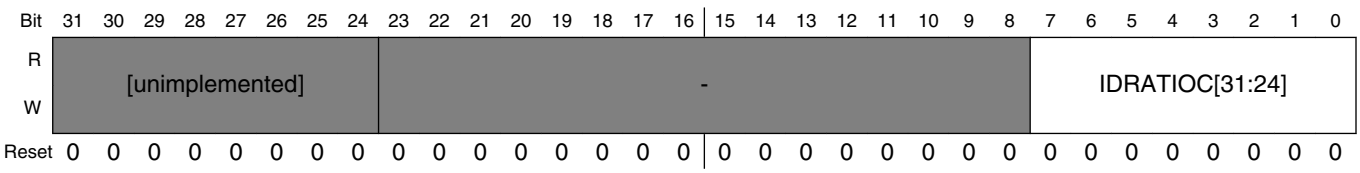
ASRC_ASRIDRLB field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–0 IDRATIOB[23:0]	IDRATIOB[23:0]. Low part of ideal ratio value for pair B.

15.6.18 ASRC Ideal Ratio for Pair C-High Part (ASRC_ASRIDRHC)

The ideal ratio registers (ASRIDRHC, ASRIDRLC) hold the ratio value IDRATIOC. IDRATIOC = FsinC/FsoutC = TsoutC/TsinC is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRC, IDRC}=2'b11.

Address: ASRC_ASRIDRHC is 5002_C000h base + 90h offset = 5002_C090h



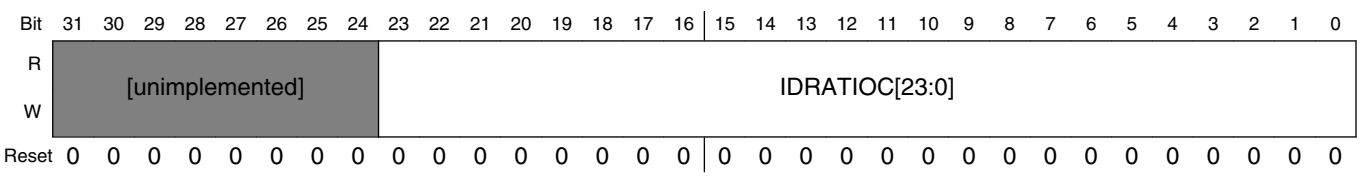
ASRC_ASRIDRHC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–8 -	Reserved
7–0 IDRATIOC[31:24]	IDRATIOC[31:24]. High part of ideal ratio value for pair C.

15.6.19 ASRC Ideal Ratio for Pair C-Low Part (ASRC_ASRIDRLC)

The ideal ratio registers (ASRIDRHC, ASRIDRLC) hold the ratio value IDRATIOC. IDRATIOC = FsinC/FsoutC = TsoutC/TsinC is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRC, IDRC}=2'b11.

Address: ASRC_ASRIDRLC is 5002_C000h base + 94h offset = 5002_C094h



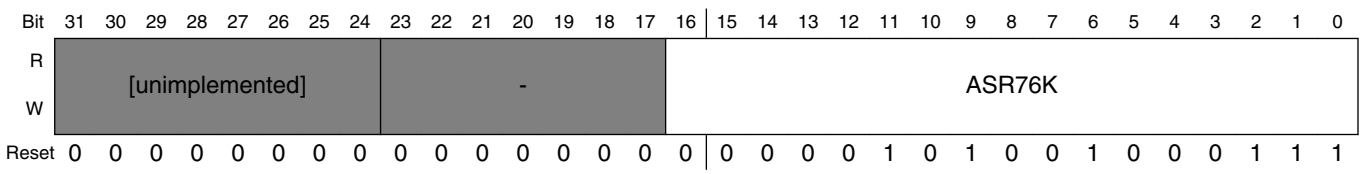
ASRC_ASRIDRLC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–0 IDRATIOC[23:0]	IDRATIOC[23:0]. Low part of ideal ratio value for pair C.

15.6.20 ASRC 76kHz Period in terms of ASRC processing clock (ASRC_ASR76K)

The register (ASR76K) holds the period of the 76kHz sampling clock in terms of the ASRC processing clock with frequency FsASRC. ASR76K = FsASRC/Fs76k. Reset value is 0x0A47 which assumes that FsASRC=200MHz. This register is used to help the ASRC internal logic to decide the pre-processing and the post-processing options automatically (see ASRC Misc Control Register 1 for Pair CASRC Control Register and ASRC Misc Control Register 1 for Pair CASRC Filter Configuration Status Register). In a system when FsASRC=133MHz, the value should be assigned explicitly as 0x06D6 in user application code.

Address: ASRC_ASR76K is 5002_C000h base + 98h offset = 5002_C098h



ASRC_ASR76K field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–17 -	Reserved
16–0 ASR76K	Value for the period of the 76kHz sampling clock.

15.6.21 ASRC 56kHz Period in terms of ASRC processing clock (ASRC_ASR56K)

The register (ASR56K) holds the period of the 56kHz sampling clock in terms of the ASRC processing clock with frequency FsASRC. ASR56K = FsASRC/Fs56k. Reset value is 0x0DF3 which assumes that FsASRC=200MHz. This register is used to help the ASRC internal logic to decide the pre-processing and the post-processing options automatically (see ASRC Misc Control Register 1 for Pair CASRC Control Register and ASRC Misc Control Register 1 for Pair CASRC Filter Configuration Status Register). In a system when FsASRC=133MHz, the value should be assigned explicitly as 0x0947 in user application code.

Address: ASRC_ASR56K is 5002_C000h base + 9Ch offset = 5002_C09Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	[unimplemented]								-								ASR56K															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	1	0	0	1	1

ASRC_ASR56K field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–17 -	Reserved
16–0 ASR56K	Value for the period of the 56kHz sampling clock

15.6.22 ASRC Misc Control Register for Pair A (ASRC_ASRMCRA)

The register (ASRMCRA) is used to control Pair A internal logic.

Address: ASRC_ASRMCRA is 5002_C000h base + A0h offset = 5002_C0A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	[unimplemented]								ZEROBUFA	EXTTHRSHA	BUFSTALLA	BYPASSPOLYA	-		OUTFIFO_THRESHOLDA[5:0]	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OUTFIFO_THRESHOLDA[5:0]				RSYNIFA	RSYNOFA	-				INFIFO_THRESHOLDA[5:0]					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRMCRA field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23 ZEROBUFA	Initialize buf of Pair A when pair A is enabled. Always clear option. This bit is used to control whether the buffer is to be zeroized when pair A is enabled. 1 Don't zeroize the buffer 0 Zeroize the buffer
22 EXTTHRSHA	Use external thresholds for FIFO control of Pair A This bit will determine whether the FIFO thresholds externally defined in this register is used to control ASRC internal FIFO logic for pair A. 1 Use external defined thresholds. 0 Use default thresholds.
21 BUFSTALLA	Stall Pair A conversion in case of Buffer Near Empty/Full Condition This bit will determine whether the near empty/full FIFO condition will stall the rate conversion for pair A. This option can only work when external ratio is used. Near empty condition is the condition when input FIFO has less than 4 useful samples per channel. Near full condition is the condition when the output FIFO has less than 4 vacant sample words to fill per channel. 1 Stall Pair A conversion in case of near empty/full FIFO conditions. 0 Don't stall Pair A conversion even in case of near empty/full FIFO conditions.

Table continues on the next page...

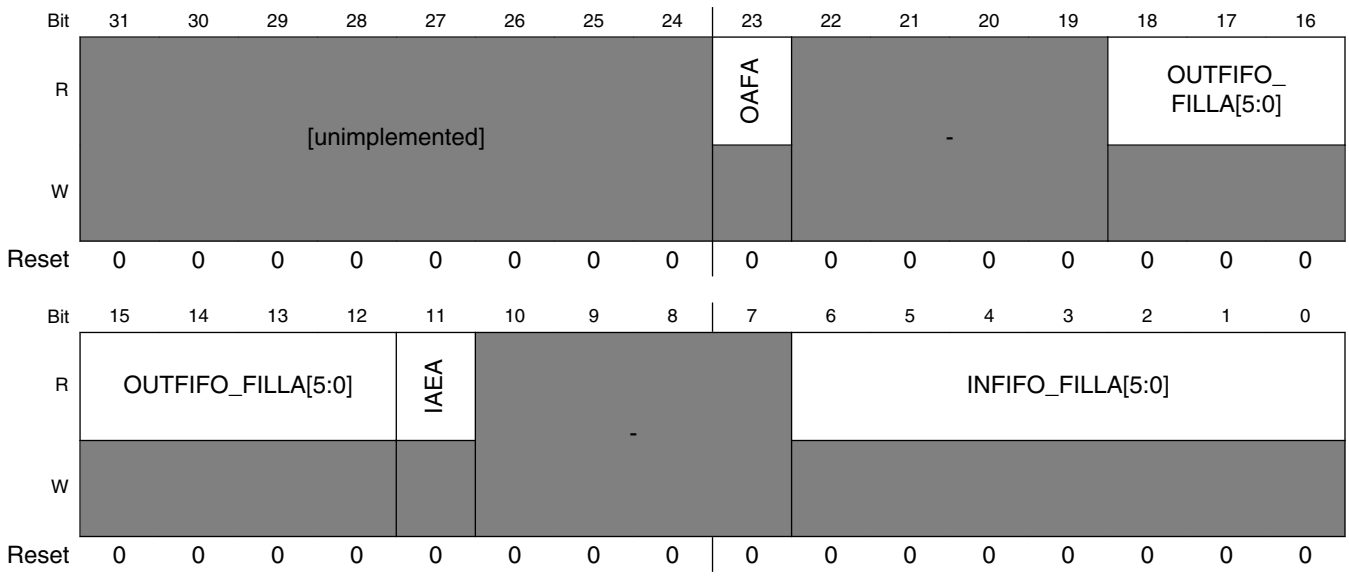
ASRC_ASRMCRA field descriptions (continued)

Field	Description
20 BYPASSPOLYA	Bypass Polyphase Filtering for Pair A This bit will determine whether the polyphase filtering part of Pair A conversion will be bypassed. 1 Bypass polyphase filtering. 0 Don't bypass polyphase filtering.
19–18 -	Reserved. Should be written as zero for future compatibility.
17–12 OUTFIFO_ THRESHOLDA[5:0]	The threshold for Pair A's output FIFO per channel These bits stand for the threshold for Pair A's output FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of output FIFO fillings of the pair is greater than n samples per channel, the output data ready flag is set; when the number of output FIFO fillings of the pair is less than or equal to n samples per channel, the output data ready flag is automatically cleared.
11 RSYNIFA	Re-sync Input FIFO Channel Counter If bit set, force ASRCCR:ACIA=0. If bit clear, untouch ASRCCR:ACIA.
10 RSYNOFA	Re-sync Output FIFO Channel Counter If bit set, force ASRCCR:ACOA=0. If bit clear, untouch ASRCCR:ACOA.
9–6 -	Reserved. Should be written as zero for future compatibility.
5–0 INFIFO_ THRESHOLDA[5:0]	The threshold for Pair A's input FIFO per channel These bits stand for the threshold for Pair A's input FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of input FIFO fillings of the pair is less than n samples per channel, the input data needed flag is set; when the number of input FIFO fillings of the pair is greater than or equal to n samples per channel, the input data needed flag is automatically cleared.

15.6.23 ASRC FIFO Status Register for Pair A (ASRC_ASRFSTA)

The register (ASRFSTA) is used to show Pair A internal FIFO conditions.

Address: ASRC_ASRFSTA is 5002_C000h base + A4h offset = 5002_C0A4h



ASRC_ASRFSTA field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23 OAFA	Output FIFO is near Full for Pair A This bit is to indicate whether the output FIFO of Pair A is near full.
22–19 -	Reserved. Should be written as zero for future compatibility.
18–12 OUTFIFO_ FILLA[5:0]	The fillings for Pair A's output FIFO per channel These bits stand for the fillings for Pair A's output FIFO per channel. Possible range is [0,64].
11 IAEA	Input FIFO is near Empty for Pair A This bit is to indicate whether the input FIFO of Pair A is near empty.
10–7 -	Reserved. Should be written as zero for future compatibility.
6–0 INFIFO_FILLA[5:0]	The fillings for Pair A's input FIFO per channel These bits stand for the fillings for Pair A's input FIFO per channel. Possible range is [0,64].

15.6.24 ASRC Misc Control Register for Pair B (ASRC_ASRMCRB)

The register (ASRMCRB) is used to control Pair B internal logic.

Address: ASRC_ASRMCRB is 5002_C000h base + A8h offset = 5002_C0A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	[unimplemented]								ZEROBUFB	EXTTHRSHB	BUFSTALLB	BYPASSPOLYB	-		OUTFIFO_THRESHOLDB[5:0]	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OUTFIFO_THRESHOLDB[5:0]				RSYNIFB	RSYNOFB	-				INFIFO_THRESHOLDB[5:0]					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRMCRB field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23 ZEROBUFB	Initialize buf of Pair B when pair B is enabled This bit is used to control whether the buffer is to be zeroized when pair B is enabled. 1 Don't zeroize the buffer 0 Zeroize the buffer
22 EXTTHRSHB	Use external thresholds for FIFO control of Pair B This bit will determine whether the FIFO thresholds externally defined in this register is used to control ASRC internal FIFO logic for pair B. 1 Use external defined thresholds. 0 Use default thresholds.
21 BUFSTALLB	Stall Pair B conversion in case of Buffer Near Empty/Full Condition This bit will determine whether the near empty/full FIFO condition will stall the rate conversion for pair B. This option can only work when external ratio is used. Near empty condition is the condition when input FIFO has less than 4 useful samples per channel. Near full condition is the condition when the output FIFO has less than 4 vacant sample words to fill per channel. 1 Stall Pair B conversion in case of near empty/full FIFO conditions. 0 Don't stall Pair B conversion even in case of near empty/full FIFO conditions.

Table continues on the next page...

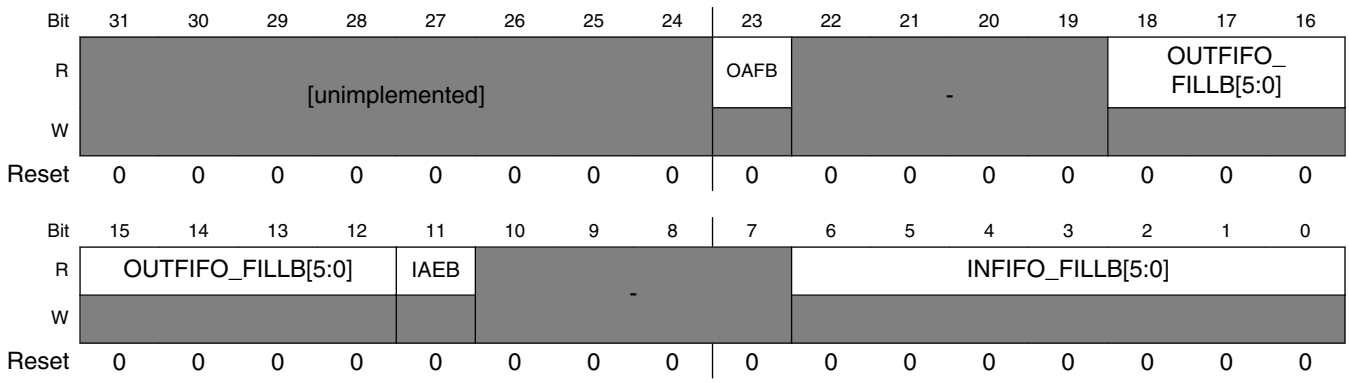
ASRC_ASRMCRB field descriptions (continued)

Field	Description
20 BYPASSPOLYB	Bypass Polyphase Filtering for Pair B This bit will determine whether the polyphase filtering part of Pair B conversion will be bypassed. 1 Bypass polyphase filtering. 0 Don't bypass polyphase filtering.
19–18 -	Reserved. Should be written as zero for future compatibility.
17–12 OUTFIFO_ THRESHOLDB[5:0]	The threshold for Pair B's output FIFO per channel These bits stand for the threshold for Pair B's output FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of output FIFO fillings of the pair is greater than n samples per channel, the output data ready flag is set; when the number of output FIFO fillings of the pair is less than or equal to n samples per channel, the output data ready flag is automatically cleared.
11 RSYNIFB	Re-sync Input FIFO Channel Counter If bit set, force ASRCCR:ACIB=0. If bit clear, untouch ASRCCR:ACIB.
10 RSYNOFB	Re-sync Output FIFO Channel Counter If bit set, force ASRCCR:ACOB=0. If bit clear, untouch ASRCCR:ACOB.
9–6 -	Reserved. Should be written as zero for future compatibility.
5–0 INFIFO_ THRESHOLDB[5:0]	The threshold for Pair B's input FIFO per channel These bits stand for the threshold for Pair B's input FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of input FIFO fillings of the pair is less than n samples per channel, the input data needed flag is set; when the number of input FIFO fillings of the pair is greater than or equal to n samples per channel, the input data needed flag is automatically cleared.

15.6.25 ASRC FIFO Status Register for Pair B (ASRC_ASRFSTB)

The register (ASRFSTB) is used to show Pair B internal FIFO conditions.

Address: ASRC_ASRFSTB is 5002_C000h base + ACh offset = 5002_C0ACh



ASRC_ASRFSTB field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23 OAFB	Output FIFO is near Full for Pair B This bit is to indicate whether the output FIFO of Pair B is near full.
22–19 -	Reserved. Should be written as zero for future compatibility.
18–12 OUTFIFO_ FILLB[5:0]	The fillings for Pair B's output FIFO per channel These bits stand for the fillings for Pair B's output FIFO per channel. Possible range is [0,64].
11 IAEB	Input FIFO is near Empty for Pair B This bit is to indicate whether the input FIFO of Pair B is near empty.
10–7 -	Reserved. Should be written as zero for future compatibility.
6–0 INFIFO_FILLB[5:0]	The fillings for Pair B's input FIFO per channel These bits stand for the fillings for Pair B's input FIFO per channel. Possible range is [0,64].

15.6.26 ASRC Misc Control Register for Pair C (ASRC_ASRMCRC)

The register (ASRMCRC) is used to control Pair C internal logic.

Address: ASRC_ASRMCRC is 5002_C000h base + B0h offset = 5002_C0B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	[unimplemented]								ZEROBUFC	EXTTHRSHC	BUFSTALLC	BYPASSPOLYC	-		OUTFIFO_THRESHOLDC[5:0]	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OUTFIFO_THRESHOLDC[5:0]				RSYNIFC	RSYNOFC	-				INFIFO_THRESHOLDC[5:0]					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRMCRC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23 ZEROBUFC	Initialize buf of Pair C when pair C is enabled This bit is used to control whether the buffer is to be zeroized when pair C is enabled. 1 Don't zeroize the buffer 0 Zeroize the buffer
22 EXTTHRSHC	Use external thresholds for FIFO control of Pair C This bit will determine whether the FIFO thresholds externally defined in this register is used to control ASRC internal FIFO logic for pair C. 1 Use external defined thresholds. 0 Use default thresholds.
21 BUFSTALLC	Stall Pair C conversion in case of Buffer Near Empty/Full Condition This bit will determine whether the near empty/full FIFO condition will stall the rate conversion for pair C. This option can only work when external ratio is used. Near empty condition is the condition when input FIFO has less than 4 useful samples per channel. Near full condition is the condition when the output FIFO has less than 4 vacant sample words to fill per channel. 1 Stall Pair C conversion in case of near empty/full FIFO conditions. 0 Don't stall Pair C conversion even in case of near empty/full FIFO conditions.

Table continues on the next page...

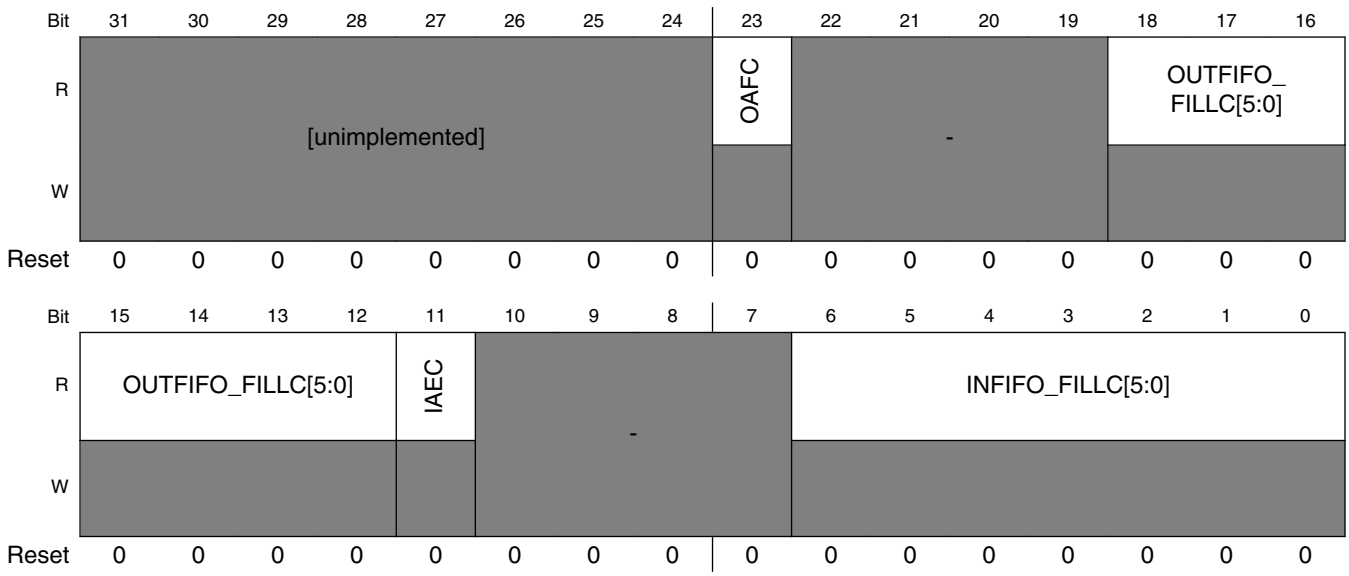
ASRC_ASRMCRC field descriptions (continued)

Field	Description
20 BYPASSPOLYC	Bypass Polyphase Filtering for Pair C This bit will determine whether the polyphase filtering part of Pair C conversion will be bypassed. 1 Bypass polyphase filtering. 0 Don't bypass polyphase filtering.
19–18 -	Reserved. Should be written as zero for future compatibility.
17–12 OUTFIFO_ THRESHOLDC[5:0]	The threshold for Pair C's output FIFO per channel These bits stand for the threshold for Pair C's output FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of output FIFO fillings of the pair is greater than n samples per channel, the output data ready flag is set; when the number of output FIFO fillings of the pair is less than or equal to n samples per channel, the output data ready flag is automatically cleared.
11 RSYNIFC	Re-sync Input FIFO Channel Counter If bit set, force ASRCCR:ACIC=0. If bit clear, untouch ASRCCR:ACIC.
10 RSYNOFC	Re-sync Output FIFO Channel Counter If bit set, force ASRCCR:ACOC=0. If bit clear, untouch ASRCCR:ACOC.
9–6 -	Reserved. Should be written as zero for future compatibility.
5–0 INFIFO_ THRESHOLDC[5:0]	The threshold for Pair C's input FIFO per channel These bits stand for the threshold for Pair C's input FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of input FIFO fillings of the pair is less than n samples per channel, the input data needed flag is set; when the number of input FIFO fillings of the pair is greater than or equal to n samples per channel, the input data needed flag is automatically cleared.

15.6.27 ASRC FIFO Status Register for Pair C (ASRC_ASRFSTC)

The register (ASRFSTC) is used to show Pair C internal FIFO conditions.

Address: ASRC_ASRFSTC is 5002_C000h base + B4h offset = 5002_C0B4h



ASRC_ASRFSTC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23 OAFC	Output FIFO is near Full for Pair C This bit is to indicate whether the output FIFO of Pair C is near full.
22–19 -	Reserved. Should be written as zero for future compatibility.
18–12 OUTFIFO_ FILLC[5:0]	The fillings for Pair C's output FIFO per channel These bits stand for the fillings for Pair C's output FIFO per channel. Possible range is [0,64].
11 IAEC	Input FIFO is near Empty for Pair C This bit is to indicate whether the input FIFO of Pair C is near empty.
10–7 -	Reserved. Should be written as zero for future compatibility.
6–0 INFIFO_FILLC[5:0]	The fillings for Pair C's input FIFO per channel These bits stand for the fillings for Pair C's input FIFO per channel. Possible range is [0,64].

15.6.28 ASRC Misc Control Register 1 for Pair X (ASRC_ASRMCR1*n*)

The register (ASRMCR1A) is used to control Pair *x* internal logic (for data alignment etc.).

The bit assignment for all the input data formats is the same as that supported by the SSI.

Addresses: ASRC_ASRMCR1A is 5002_C000h base + C0h offset = 5002_C0C0h
ASRC_ASRMCR1B is 5002_C000h base + C4h offset = 5002_C0C4h
ASRC_ASRMCR1C is 5002_C000h base + C8h offset = 5002_C0C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	[unimplemented]								-												IWD[2:0]			IMSB	-					OMSB	OSGN	OW16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRMCR1*n* field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–12 -	Reserved. Should be written as zero for future compatibility.
11–9 IWD[2:0]	Data Width of the input FIFO These three bits will determine the bitwidth for the audio data into ASRC All other settings not shown are reserved. 3'b000 24-bit audio data. 3'b001 16-bit audio data. 3'b010 8-bit audio data.
8 IMSB	Data Alignment of the input FIFO This bit will determine the data alignment of the input FIFO. 1 MSB aligned. 0 LSB aligned.
7–3 -	Reserved. Should be written as zero for future compatibility.
2 OMSB	Data Alignment of the output FIFO This bit will determine the data alignment of the output FIFO. 1 MSB aligned. 0 LSB aligned.
1 OSGN	Sign Extension Option of the output FIFO This bit will determine the sign extension option of the output FIFO.

Table continues on the next page...

ASRC_ASRMCR1*n* field descriptions (continued)

Field	Description
	1 Sign extension. 0 No sign extension.
0 OW16	Bit Width Option of the output FIFO This bit will determine the bit width option of the output FIFO. 1 16-bit output data 0 24-bit output data.

Chapter 16
Digital Audio Multiplexer (AUDMUX)

16.1 Overview

The Digital Audio Multiplexer (AUDMUX) provides a programmable interconnect device for voice, audio, and synchronous data routing between host serial interfaces such as the Synchronous Serial Interface Controller (SSI) and peripheral serial interfaces (audio and voice codecs, also known as coder-decoders).

This section includes a top level diagram that shows the functional organization of the block, including all off-chip signals.

AUDMUX allows the audio system connectivity to be modified through programming, as opposed to altering the PCB schematics of the system. The full description of the block is in Functional Description.

Figure 16-1 shows the block diagram.

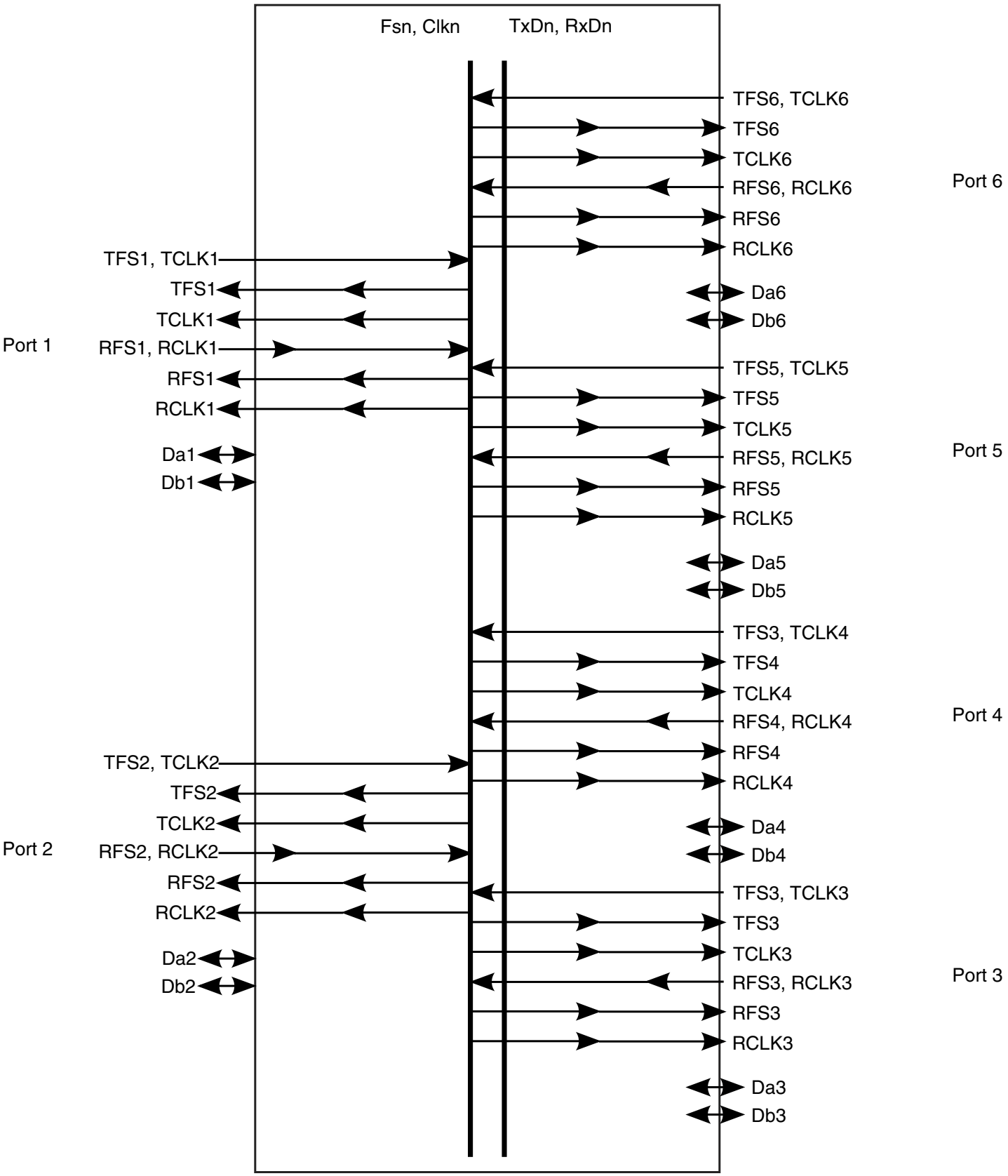


Figure 16-1. AUDMUX Block Diagram

With AUDMUX, resources do not need to be hard-wired and can be effectively shared in different configurations. AUDMUX interconnections allow multiple, simultaneous audio/voice/data flows between the ports in point-to-point or point-to-multipoint configurations.

AUDMUX includes two types of interfaces. Internal ports connect to the processor serial interfaces, and external ports connect to off-chip audio devices and serial interfaces of other processors. The desired connectivity is achieved by configuring the appropriate internal and external ports.

16.1.1 Features

Key features of the block include:

- Two internal ports
- Four external ports
- Three internal ports
- Four external ports
- Full 6-wire SSI interfaces for asynchronous receive and transmit
- Configurable 4-wire (synchronous) or 6-wire (asynchronous) peripheral interfaces
- Independent Tx/Rx Frame sync and clock direction selection for host or peripheral
- Each host interface's capability to connect to any other host or peripheral interface in a point-to-point or point-to-multipoint (network mode)
- Transmit and Receive Data switching to support external network mode

16.1.2 Modes and Operations

The AUDMUX supports the modes described in Operating Modes.

16.2 External Signal Description

The following table lists the off-chip signals of the AUDMUX for the external ports, where P*n* is P4 to P7 and m = (n-3).

Table 16-1. Off-Chip Block Signals

Name	Block Port	I/O	Function	Reset State	Pull-up
AUDm_TXD	P*n*	I/O	Transmit Data from Pn	1	Active
AUDm_RXD	P*n*	I/O	Receive Data at Pn	1	Active

Table continues on the next page...

Table 16-1. Off-Chip Block Signals (continued)

Name	Block Port	I/O	Function	Reset State	Pull-up
AUDm_TXC	P*n*	I/O	Transmit Clock input/output at Pn	1	-
AUDm_RXC	P*n*	I/O	Receive Clock input/output at Pn	1	-
AUDm_TXFS	P*n*	I/O	Transmit Frame sync input/output at Pn	1	-
AUDm_RXFS	P*n*	I/O	Receive Frame sync input/output at Pn	1	-

16.3 Default Register Configuration

There are two configuration registers for each port. Each pair of configuration registers is identical for each port; however, the default values following a reset differ as shown in the Memory Map.

- Default Port Configuration, describes the default configuration of the ports.

16.3.1 Default Port Configuration

After a reset, each port defaults to normal mode (PDCR*n*[MODE] = 0) with synchronous timing mode (PTCR*n*[SYN] = 1) enabled.

The default port-to-port connections are as follows:

- Port 1 to Port 6
  - Port 6 provides the clock and frame sync.
- Port 2 to Port 5
  - Port 5 provides the clock and frame sync.
- Port 3 to Port 4
  - Port 4 provides the clock and frame sync.
- Port 7 to Port 7 (in data loopback mode)
  - Clock and frame syncs are inputs.

16.4 Functional Description

This section provides a complete functional description of the AUDMUX.

16.4.1 Operating Modes

This section describes all functional operation modes of the AUDMUX.

Figure 16-1 shows the AUDMUX block diagram.

All of the ports are essentially identical; there is no functional difference among Ports 1 through 7. The main difference is whether a port is connected to an on-chip serial interface (SSI for example) or connected to the chip's pads to connect to off-chip serial devices (that is, any 4-wire or 6-wire external SSI, voice, I2S, or AC97 codec).

All ports can be configured as four- or six-wire interfaces. When configured as a six-wire interface, the additional RFS and RCLK signals of the interface enable the serial interface to be used in asynchronous mode with separate receive and transmit clocks.

All ports have a Tx/Rx switch to provide flexibility in supporting network mode configurations. The Tx/Rx switch enables the transmit and receive data lines to be swapped so that mastership of the serial bus can be passed among multiple external devices connected to a single port.

In addition to supporting the default external network mode, all ports support an internal network mode:

- With internal network mode, a point-to-multipoint network configuration with an arbitrary number of slaves can be supported if the external slaves are put into the high-impedance state (as defined in the SSI network mode protocol) and have pull-up resistors on their TxD pins. (Alternatively, this can be viewed as requiring a pull-up resistor on the corresponding AUDMUX RxD pin.)

Bit clock direction selection enables each port to be configured as a master or slave in the flow.

Possible scenarios include:

- SSI (internal port) drives a voice codec and a BT (Blue tooth) codec (both on external Port 6Port 5) and the Bottom Connector (on external Port 7Port 6) simultaneously using network mode. SSI is the master.
- An external processor (external port - Port 5Port 4) drives a voice codec and a BT codec (both on external Port 6Port 5) and the Bottom Connector (on Port 7Port 6) simultaneously using network mode. The external processor is the master.

16.4.1.1 Port Receive Data Modes

Each port has logic to select which data lines are used to create the RxD line for the corresponding host interface.

Figure 16-2 shows the logic used to create the RxD line for Port 1. This logic has the following modes of operation (as determined by MODE):

- Normal
- Internal network mode

The subsequent sections describe the various modes of the port receive data logic. The following terms are used to define the operation of the AUDMUX:

- Network mode- Time-Division Multiplexed protocol for sending unique data to multiple devices on a serial bus.
- Internal network mode-Physical bus configuration where multiple serial buses are effectively connected within the AUDMUX via digital logic to create point-to-multipoint connectivity. An arbitrary number of devices are supported. Devices must be put into the high-impedance state as specified by the network mode protocol. TxDATA lines of devices must be pulled high.
- External network mode-Physical bus configuration where multiple serial buses are electrically connected together on a printed circuit board (that is, external to the AUDMUX). Devices must put their TxDATA lines into the high-impedance state as specified by the network mode protocol.

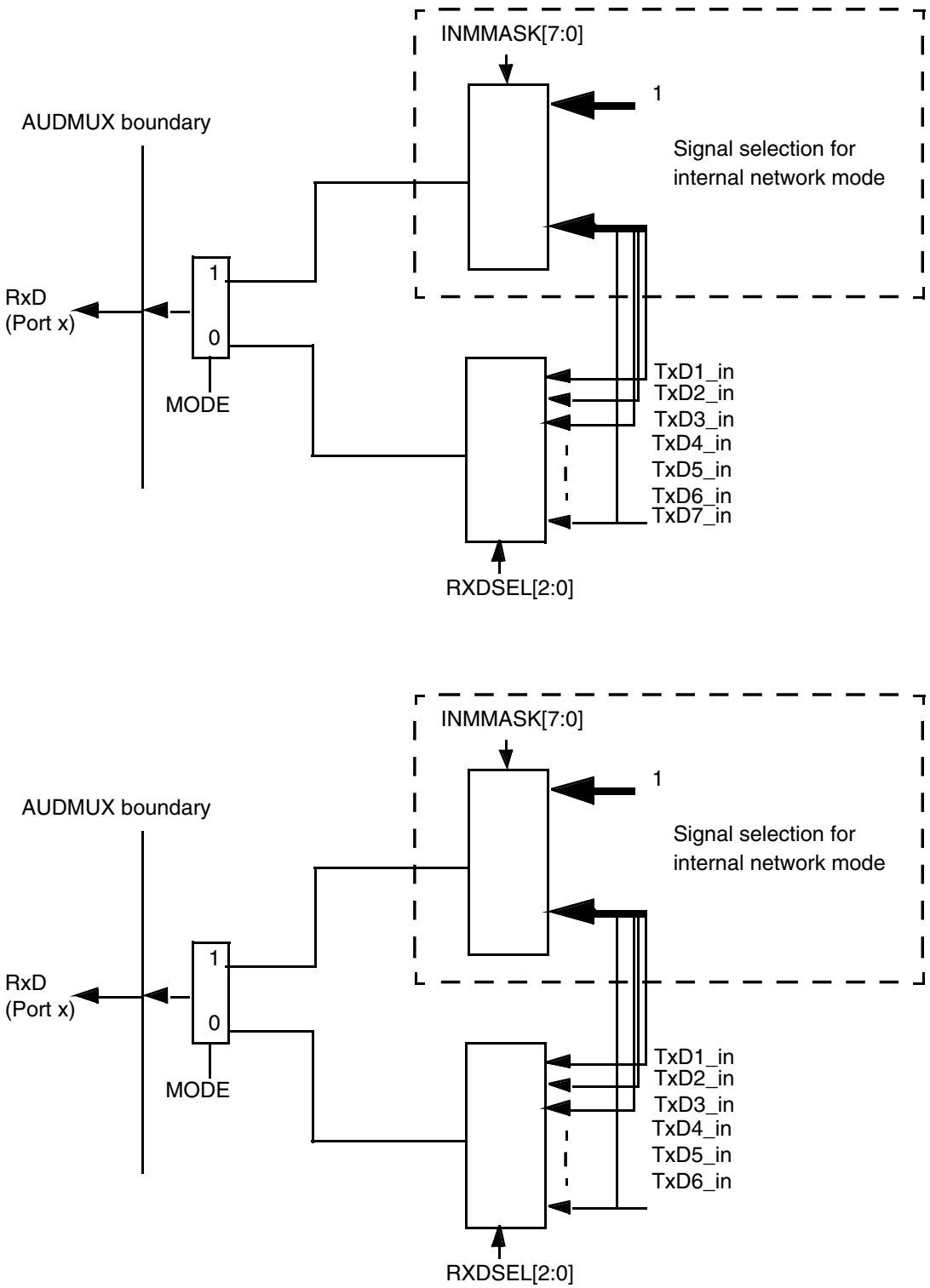


Figure 16-2. Receive Data Logic for Port x

16.4.1.1.1 Normal Mode

In normal mode (MODE = 0) , the port is connected in a point-to-point configuration (as a master or a slave) and the RXDSEL[2:0] setting selects the transmit signal from any port. In normal mode, any data format can be used (that is, SSI normal mode, SSI network mode, AC-97, and others.

16.4.1.1.2 Internal Network Mode

In internal network mode (MODE = 1) , the output of the AND gate is routed (via the output of the port) to the RxD signal of the corresponding host interface.

The INMMASK bit vector selects the transmit signals of the ports that are to be connected in network mode. The transmit signals received at the AUDMUX ports (TxDn_in) are ANDed together to form the output. In internal network mode, only one device can be transmitting in its predesignated timeslot and all other transmit signals must remain high (be in high-impedance state and pulled-up). Therefore, non-active signals in the selection will be high and do not influence the output of the AND gate.

Network mode is a protocol where a master SSI is connected to more than one slave SSI device and communication occurs on a time-slotted frame. Though network mode can allow master-slave and slave-slave communication, internal network mode supports only master-slave communication.

There are two scenarios where internal network mode can be used with external network mode:

1. Slave-only devices are attached to an external port.
2. A master device is attached to an external port and all slave devices connected to the same external port are disabled.

NOTE

When internal network mode is enabled at an external port, RXDSEL[3:0] for RxDn_obe selection is ignored and RxD_obe is always driven high (that is, asserted for all timeslots). All slave devices connected to the same port must be disabled.

Internal Network Mode Example 1

SSI_m and SSI_n are used with Port 4 in internal network mode as shown in Figure 16-3. No pull-up resistors are required because the interfaces combined in internal network mode are on-chip interfaces.

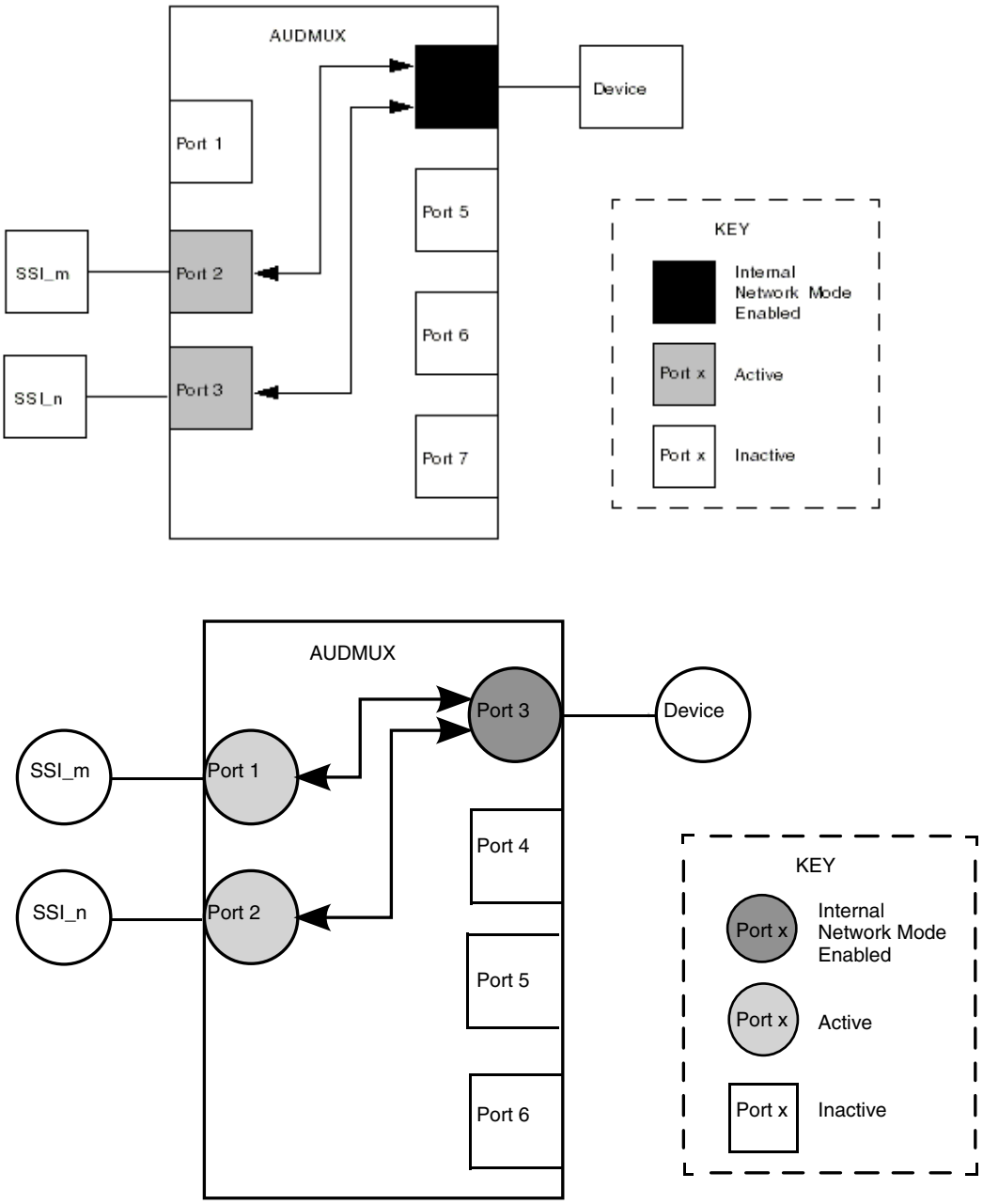


Figure 16-3. Block Diagram For Example 1

See Figure 16-4 for the timing diagram of Example 1. The clock and frame sync signals show the bit and frame timing for the serial bus. The vertical dashed lines divide the frame into four timeslots.

The data lines for SSI_m and SSI_n (as well as their output enables) are shown. Note that the on-chip interfaces drive a logic '1' when their output enables are logic '0'. The combined TxDATA line, which is the logical AND of the individual TxDATA lines, is used for Port 4Port3's TxDATA line.

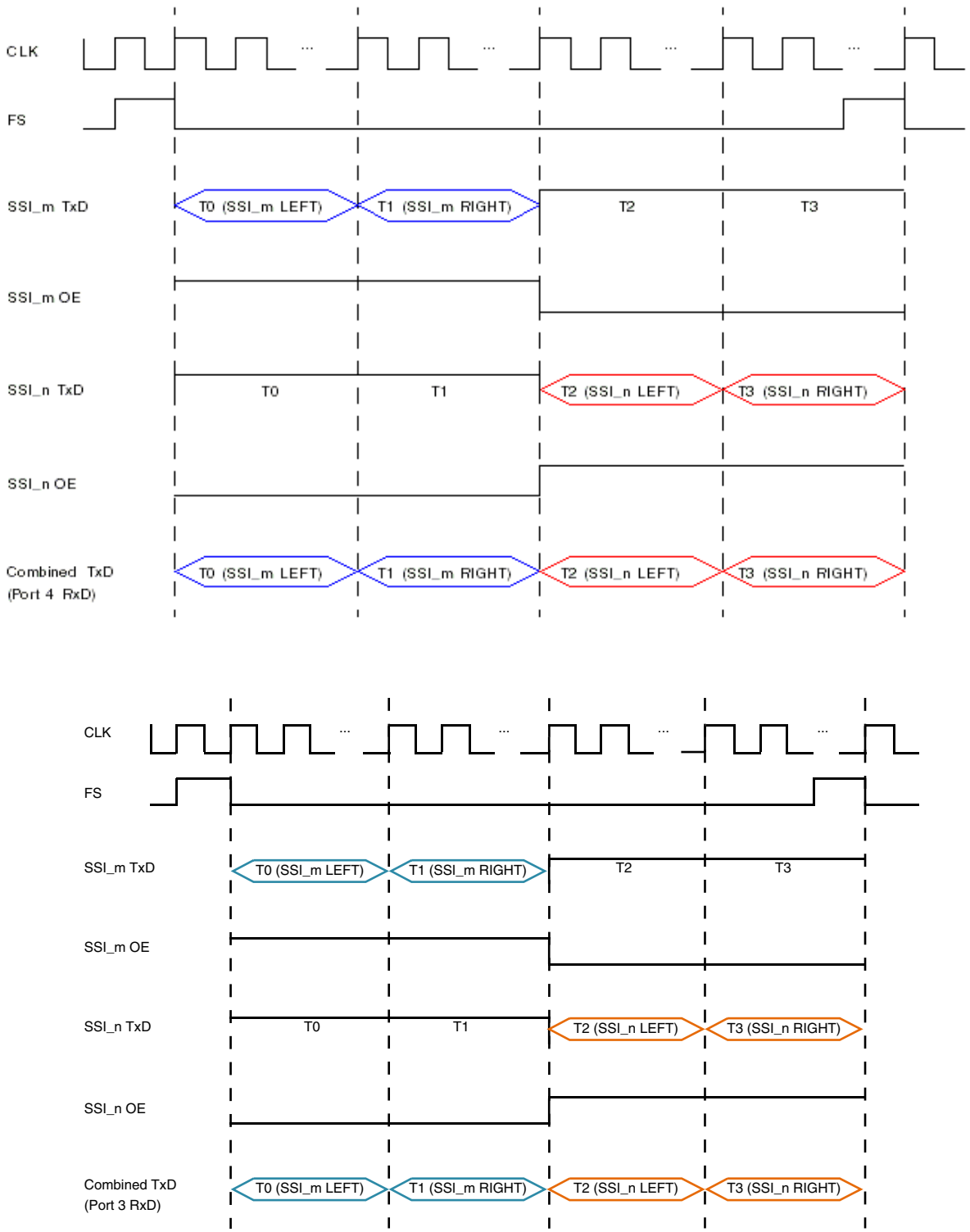


Figure 16-4. Example Using Internal Ports For Transmit Data

Internal Network Mode Example 2

The SSI, Port 4, and Port 5 are used with Port 6Port 3, and Port 4 are used with Port 5 in internal network mode, as shown in the following figure. Note that Port 4 and Port 5Port 3 and Port 4 are external ports. Therefore, pull-up resistors are required on the Port 4 RxDATA and Port 5 RxDATA pins. This example shows the timing associated with using adjacent timeslots for the SSI, Port 4 and Port 5Port 3 and Port 4.

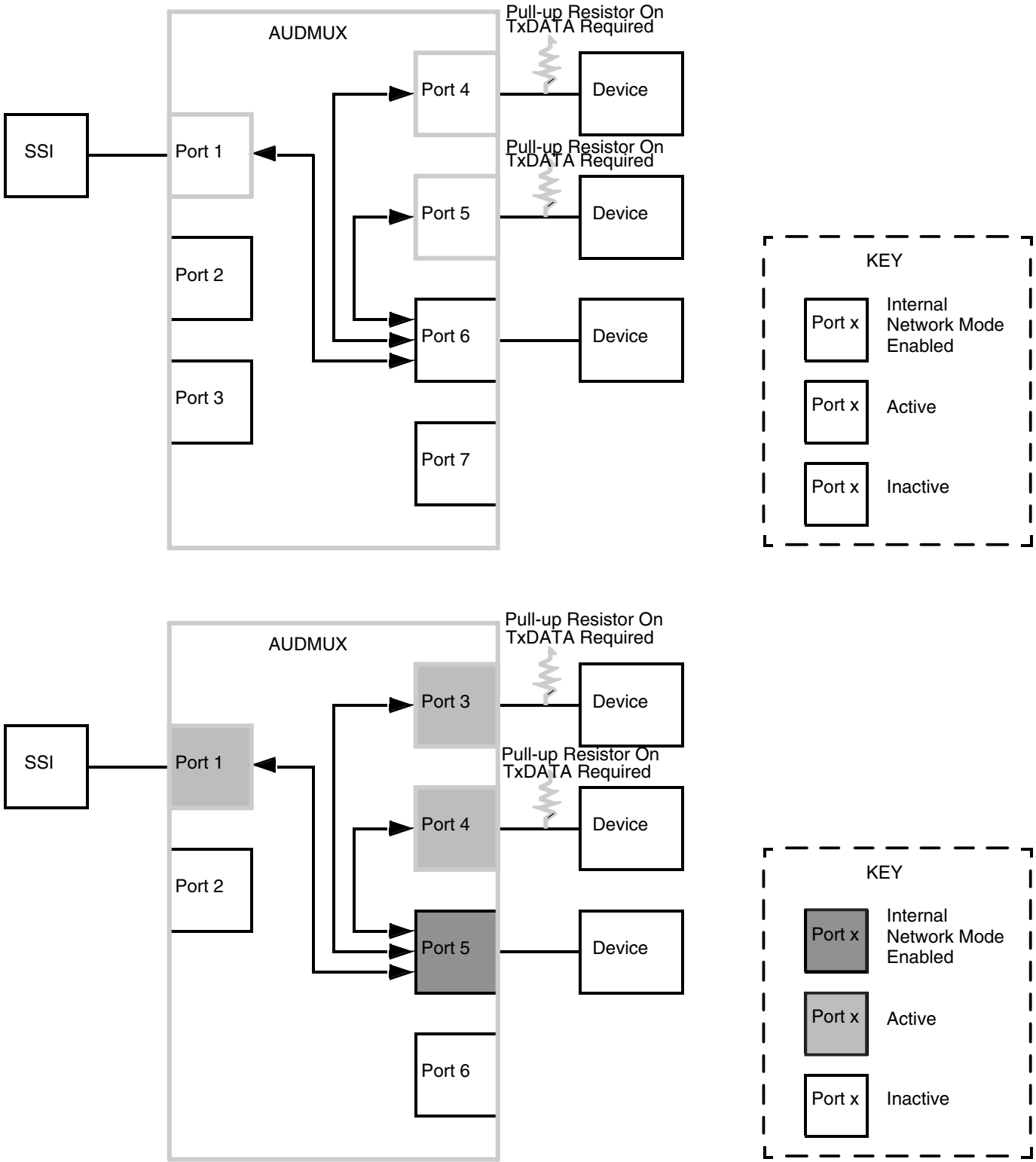


Figure 16-5. Block Diagram For Example 2

The resistance value of the pull-up resistors must be sufficiently high such that a value of '0' can be pulled up to logic '1' within half of a period of the bitclock. The required resistance must be no larger than:

Rmax = 1 / (2 * f_{bc} * C) where:

- f_{bc} is the frequency of the bitclock
- C is the total system capacitance (ICs, board traces, and so on)

The following figure shows the timing diagram for this example. The clock and frame sync signals show the bit and frame timing for the serial bus. The vertical dashed lines divide the frame into four timeslots.

The data lines for the SSI, Port 4 and Port 5Port 3 and Port 4 are shown. Note that the SSI transmits a logic '1' when its corresponding output enable is a logic '0'. The data lines from Port 4 and Port 5Port 3 and Port 4 at the pad are pulled high by pull-up resistors when they are in the high-impedance state. The data lines from Port 4 and Port 5Port 3 and Port 4 at the AUDMUX are pure digital signals and are constantly driven. The combined TxDATA line, which is the logical AND of the SSI, Port 4 and Port 5Port 3 and Port 4's TxDATA lines, is used for Port 6Port 5's TxDATA line.

Note the highlighted areas in the following figure. This shows the transition time that occurs while a TxDATA line is being pulled high. In this example, this transition time is a maximum of 1/2 the period of the serial bitclock. This prevents corruption of the first data bit of the next timeslot. It is critical that the pull-up resistance is sufficient for the given bitclock frequency and system capacitance.

Note that hysteresis should be enabled at Port 4's RxDATA pad and Port 5's RxDATA padPort 3's RxDATA pad and Port 4's RxDATA pad to prevent the digital signals created by the pad from toggling rapidly during the pull-up period. The pads typically require a transition within 25ns unless hysteresis is enabled. Instead of using hysteresis, one could select a pull-up resistor sufficiently high to pull-up the signal at the pad within 25 ns; however, that would result in a higher resistance value and higher current drain.

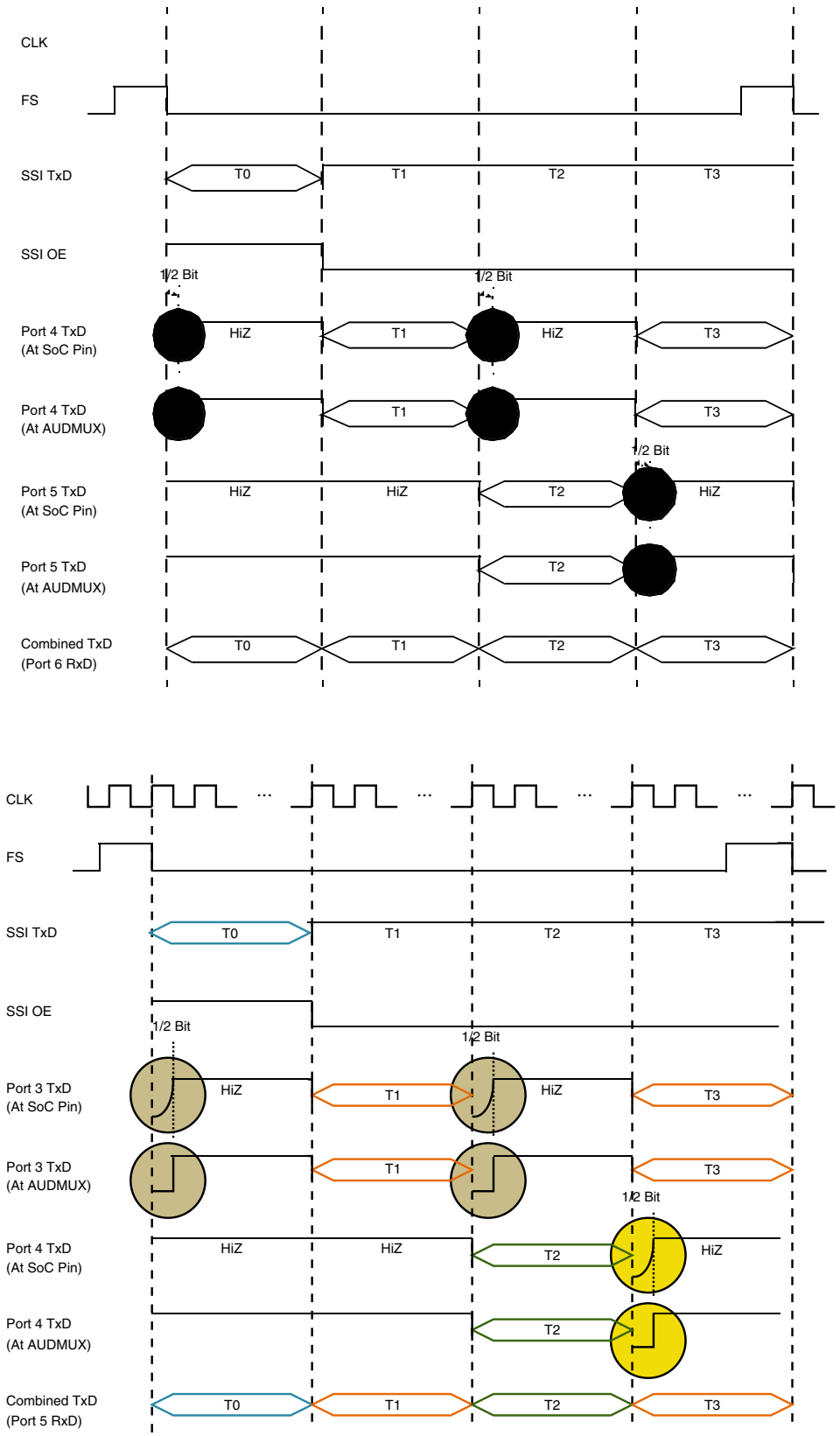


Figure 16-6. Example Using External Ports for Transmit Data in Cons

Internal Network Mode Example 3

The SSI and Port 4 are used with Port 6Port 3 are used with Port 5 in internal network mode as shown in the following figure. Note that Port 4Port 3 is an external port. Therefore, a pull-up resistor is required on the Port 4Port 3 TxDATA pin. This example shows the timing associated with inserting empty timeslots after the timeslots have been used by external ports.

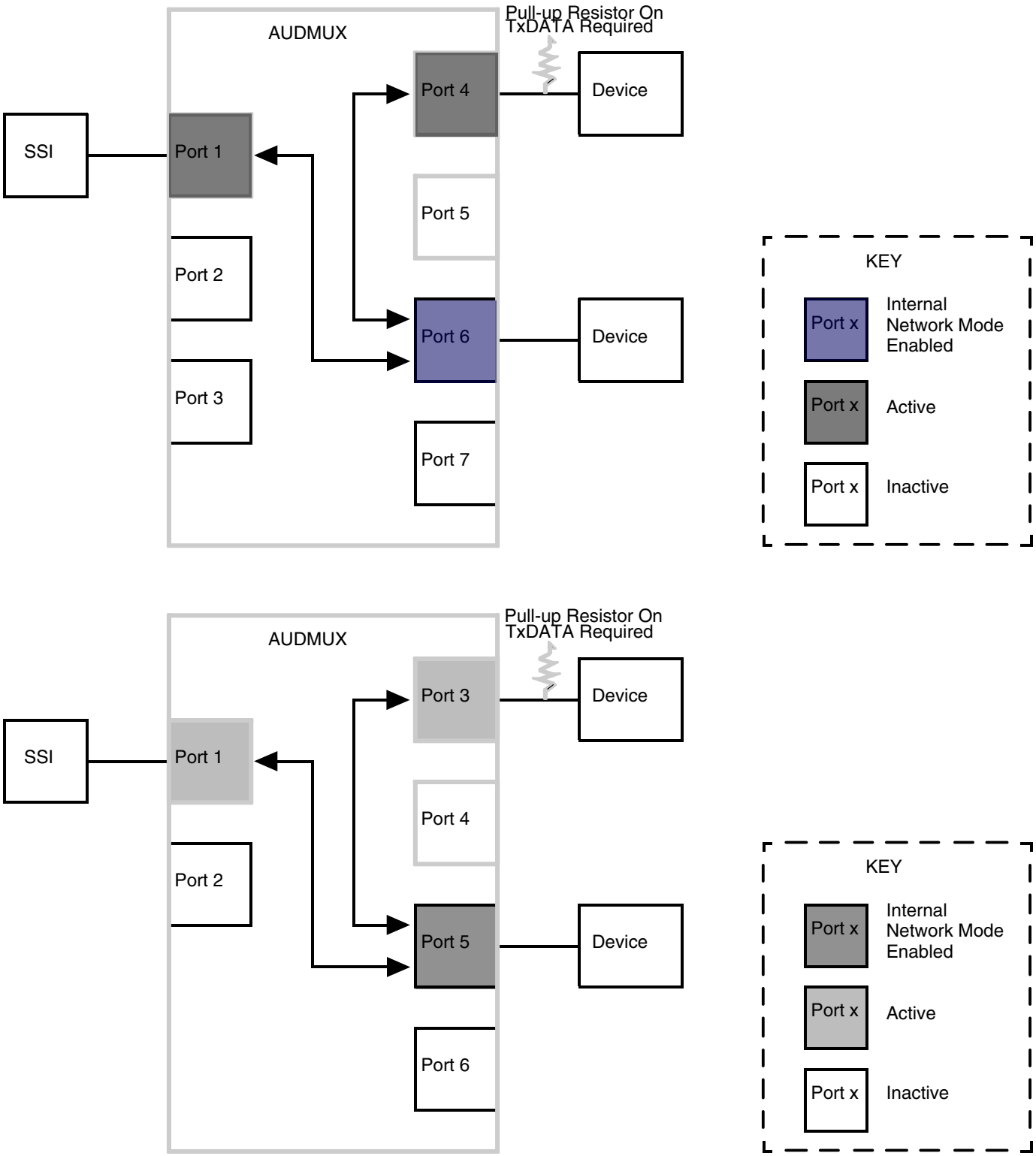


Figure 16-7. Block Diagram For Example 3

The resistance value of the pull-up resistors must be sufficiently high such that a value of '0' can be pulled up to logic '1' by the time that the next occupied timeslot occurs. This allows a much weaker pull-up to be used as compared to Example 2. The required resistance must be no larger than:

Rmax = (4 * n + 1) / (2* f_{bc} * C) where:

- n is the number of bits per timeslot
- f_{bc} is the frequency of the bitclock
- C is the total system capacitance (ICs, board traces, and so on)

The figure below shows the timing diagram for this example. The clock and frame sync signals show the bit and frame timing for the serial bus. The vertical dashed lines divide the frame into four timeslots.

The data lines for the SSI and Port 4Port 3 are shown. Note that the SSI transmits a logic '1' when its corresponding output enable is a logic '0'. The data line from Port 4Port 3 at the pad is pulled high by a pull-up resistor when they are in the high-impedance state. The data line from Port 4Port 3 at the AUDMUX is a pure digital signal and is constantly driven. The combined TxDATA line, which is the logical AND of the SSI and Port 4Port 3's TxDATA lines, is used for Port 6Port 5's RxDATA line.

Note the highlighted area in the following figure. This shows the transition time that occurs while Port 4Port 3's TxDATA line is being pulled high. In this example, this transition time is a maximum of two timeslots plus 1/2 the period of the serial bitclock. This prevents corruption of the first data bit of the next timeslot. It is critical that the pull-up resistance is sufficient for the given bitclock frequency and system capacitance.

Note that hysteresis must be enabled at Port 4Port 3's RxDATA pad to prevent the digital signal created by the pad from toggling rapidly during the extended pull-up period. The pads typically require a transition within 25 ns unless hysteresis is enabled.

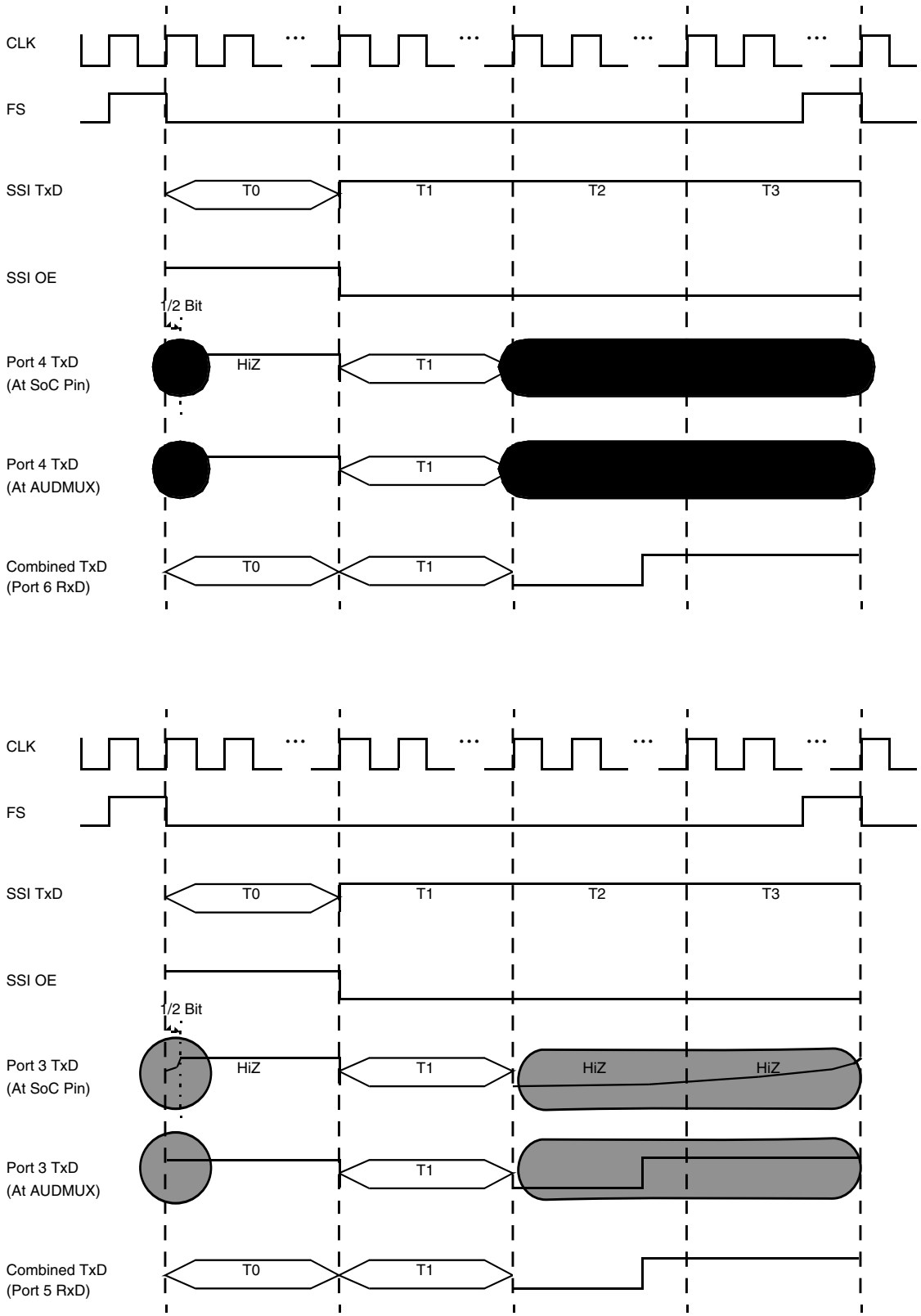


Figure 16-8. Example Using External Ports For Transmit Data In Nonconsecutive Timeslots

16.4.1.1.3 Transmit Data Output Enable Assertion

The TxDATA line from the internal network mode master (connected at any internal port) is put into the high-impedance state at the pad depending upon the assertion or deassertion of TxD_obe, its corresponding output enable generated by the network mode master.

In the case of an external network mode master (connected at an external port), the corresponding TxD_obe is always asserted after the port data register configuration.

16.4.1.2 Tx/Rx Switch and External Network Mode

External network mode is the traditional network mode connection. It is called external network mode to differentiate from the internal network mode. In external network mode, devices are connected to a single external port in a star or multi-drop configuration.

In network mode, there can be only one master (driving the frame sync and clock source) with the other devices configured in normal slave mode or network slave mode. Unlike internal network mode, both master-slave and slave-slave communication can take place in external network mode. Codec devices transmit on a single timeslot while processor serial interfaces (that is, SSI) can process more than one timeslot of data while in network master or slave mode.

The following figure shows the Tx/Rx data switch. RxD_obe is the output buffer enable signal and RxD_out is the data transmit signal from the serial interface. The TxD_in signal is the receive data signal going towards the RXDSEL muxes of all ports.

D_TxRx is the data pin which serves as the chip-level transmit data pin when the TxRx switch is not enabled. D_RxTx is the data pin which serves as the chip-level receive data pin when the TxRx switch is not enabled. The roles of these pins are reversed when the TxRx switch is enabled.

When TXRXEN is disabled (TXRXEN=0), RxD_out is routed to D_TxRx and D_RxTx is routed to TxD_in. The output buffer enable, selected by RXDSEL[2:0], is routed to Db_obe.

When the Tx/Rx switch is enabled (TXRXEN=1), RxD_out is routed to D_RxTx and D_TxRx is routed to TxD_in. The output buffer enable, selected by RXDSEL[2:0], is routed to Da_obe.

If the RXDSELn[2:0] field for any Port *n* is configured to select data from an internal port, the output buffer enable is selected by RXDSELn[2:0] and is routed to Da*n*_obe/ Db*n*_obe. In the case when the RXDSELn[2:0] field for Port *n* is configured to select data from an external port, the output buffer enable is always high and routed to Dan_obe/Dbn_obe, depending on the TXRXENn switch configuration.

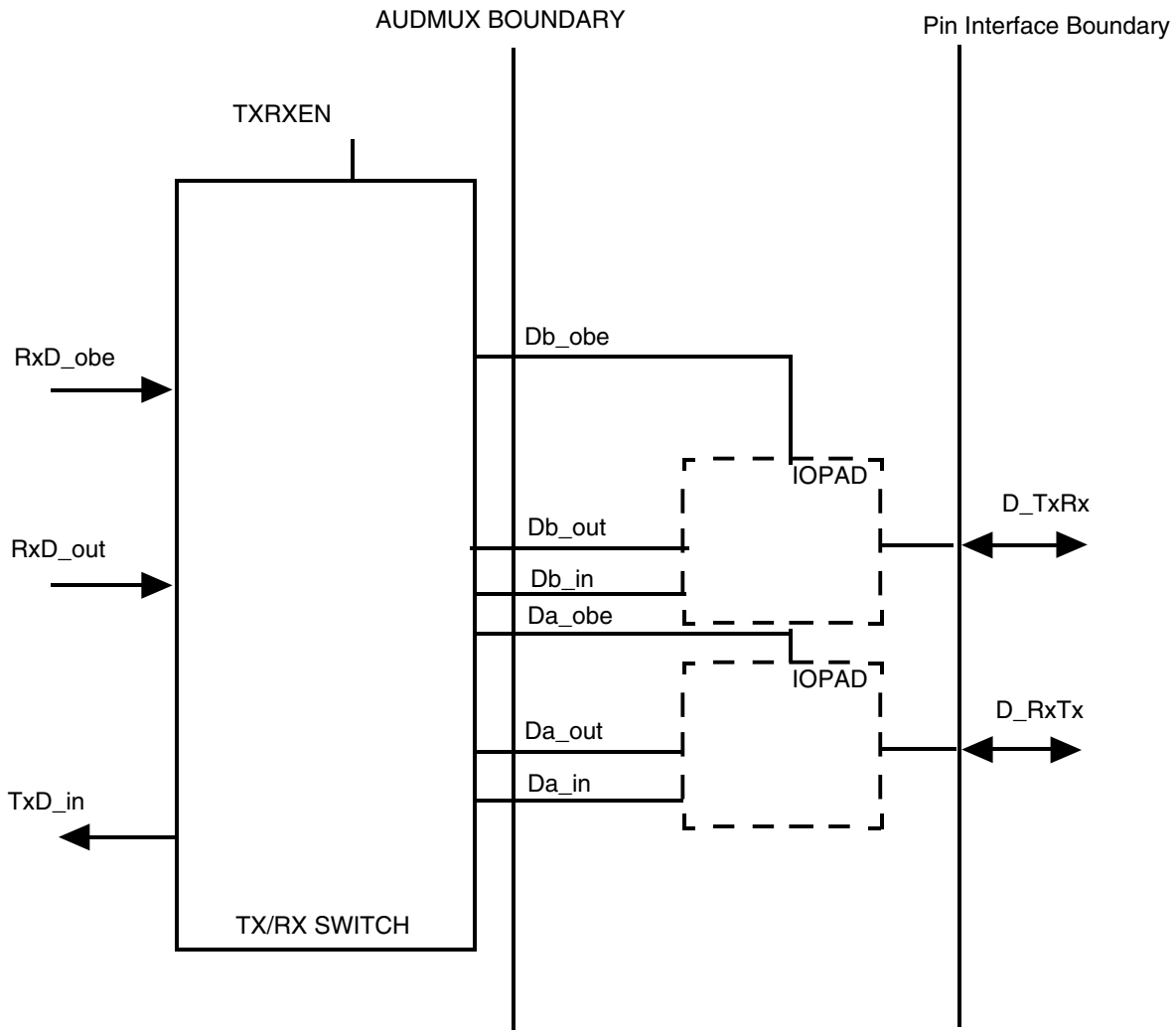


Figure 16-9. Tx/Rx Switch

16.4.1.3 Timing Modes

The AUDMUX ports are constructed as 6-wire interfaces. However, they can be used either in synchronous or asynchronous modes as determined by the SYN bit.

16.4.1.3.1 Synchronous Mode (4-Wire Interface)

In Synchronous mode, the port has a 4-wire interface (that is, RxD, TxD, TxCLK, TxFS). The receive data timing is determined by TxCLK and TxFS.

As shown in the following figure, Port x signals can be routed to Port y, producing 6-wire to 4-wire port connectivity.

TFS_in, RFS_in, TCLK_in, and RCLK_in are the input frame sync and bit clocks from the serial interface (Port x) with their corresponding output buffer enable signals (_obe). TFS_out, RFS_out, TCLK_out, and RCLK_out are the frame sync and bit clocks that are transmitted to the serial interface from the other ports.

The TFS_out and TCLK_out are selected at Port x by the TFSEL and TCSEL mux settings, respectively. RFS_out and RCLK_out are selected at Port x by the RFSEL and RCSEL mux settings, respectively. Similarly, in the external direction, Port y is configured as a 4-wire port; TFSEL selects the FS_obe and FS_out signals. In this mode, the configuration of RFSEL and RCSEL is not used, since the RFS_out and RCLK_out pins at Port y are not available.

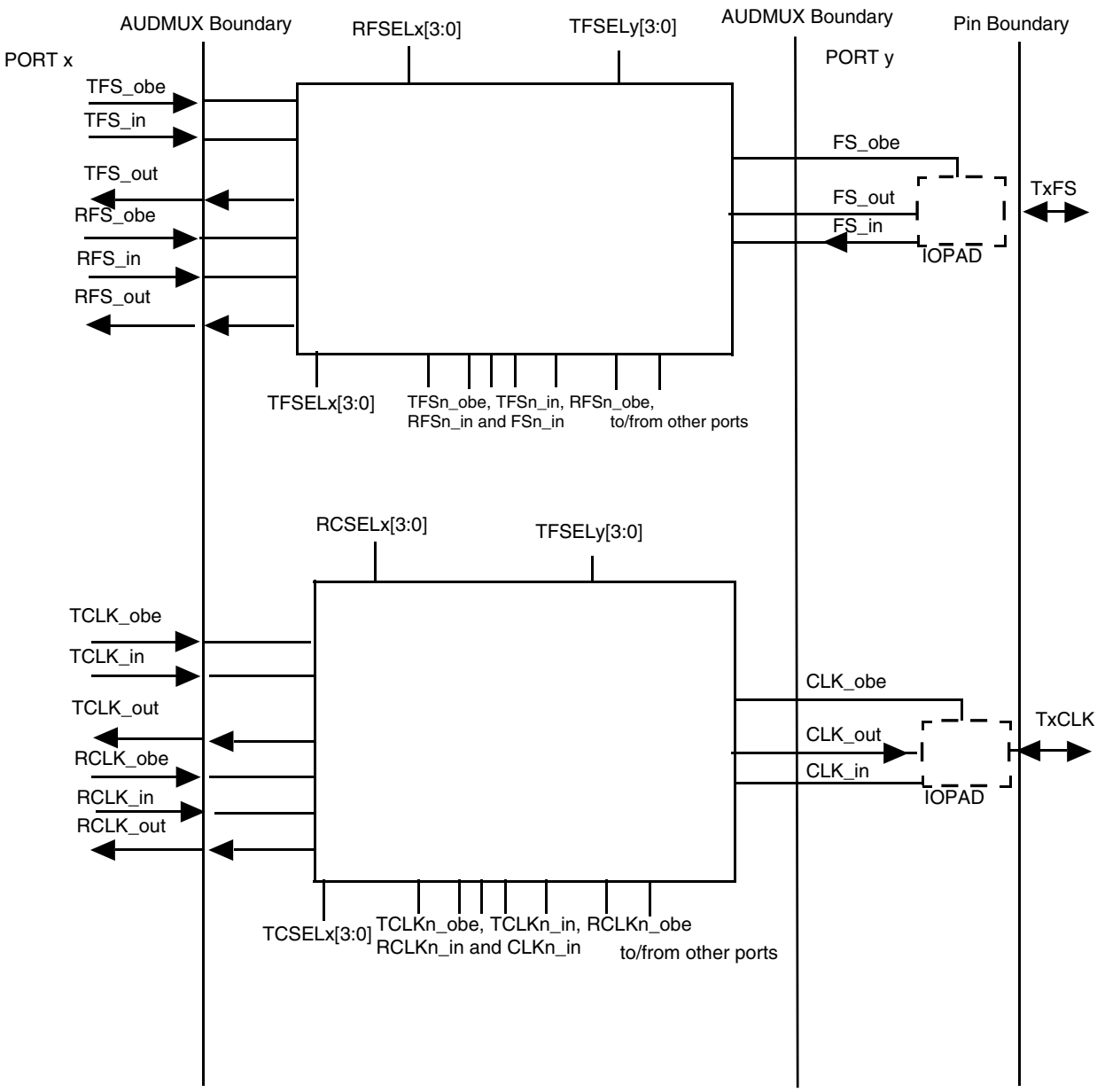


Figure 16-10. Frame Sync and Clock Routing When External Port Is 4-Wire

16.4.1.3.2 Asynchronous Mode (6-Wire Interface)

In Asynchronous mode, the port has a 6-wire interface (meaning RxD, TxD, TxCLK, TxFS, RxCLK, RxFS). This mode has additional receive clock (RxCLK) and frame sync (RxFS) signals as compared to the synchronous or 4-wire interface.

As shown in the figures below, Port x signals can be routed to Port y, producing 6-wire to 6-wire port connectivity.

TFS_in, RFS_in, TCLK_in, and RCLK_in are input frame sync and bit clocks from the serial interface (Port x) with their corresponding output buffer enable signals (_obe). TFS_out, RFS_out, TCLK_out, and RCLK_out are the frame sync and bit clocks that are transmitted to the serial interface from the other ports.

TFS_out and TCLK_out are selected by the TFSEL and TCSEL mux settings, respectively. RFS_out and RCLK_out are selected by the RFSEL and RCSEL mux settings, respectively. Similarly, in the external direction, the TFSEL selects the TxFS_obe and TxFS_out signals and TCSEL selects the TxCLK_obe and TxClk_out signals. The RFSEL selects the RxFS_obe and RxFS_out signals and RCSEL selects the RxCLK_obe and RxCLK_out signals.

NOTE

Because FS_in and CLK_in from external interfaces are also routed to the TFSEL and TCSEL muxes of the external ports, these signals do not have corresponding buffer enable signals. Consequently, their corresponding inputs to the TFSEL and TCSEL mux of the external ports have to be tied high.

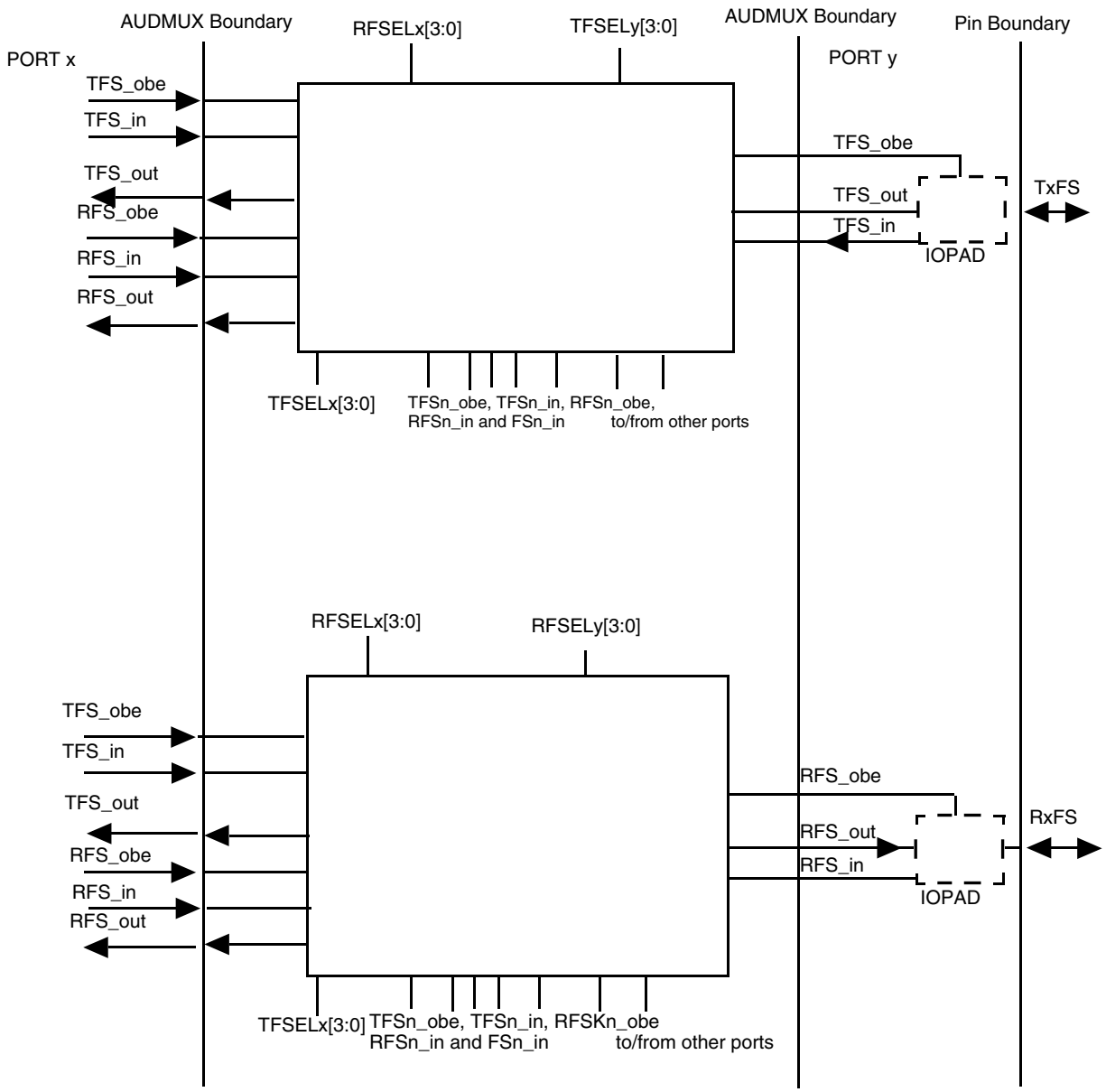


Figure 16-11. Frame Sync Routing When External Port Is 6-Wire

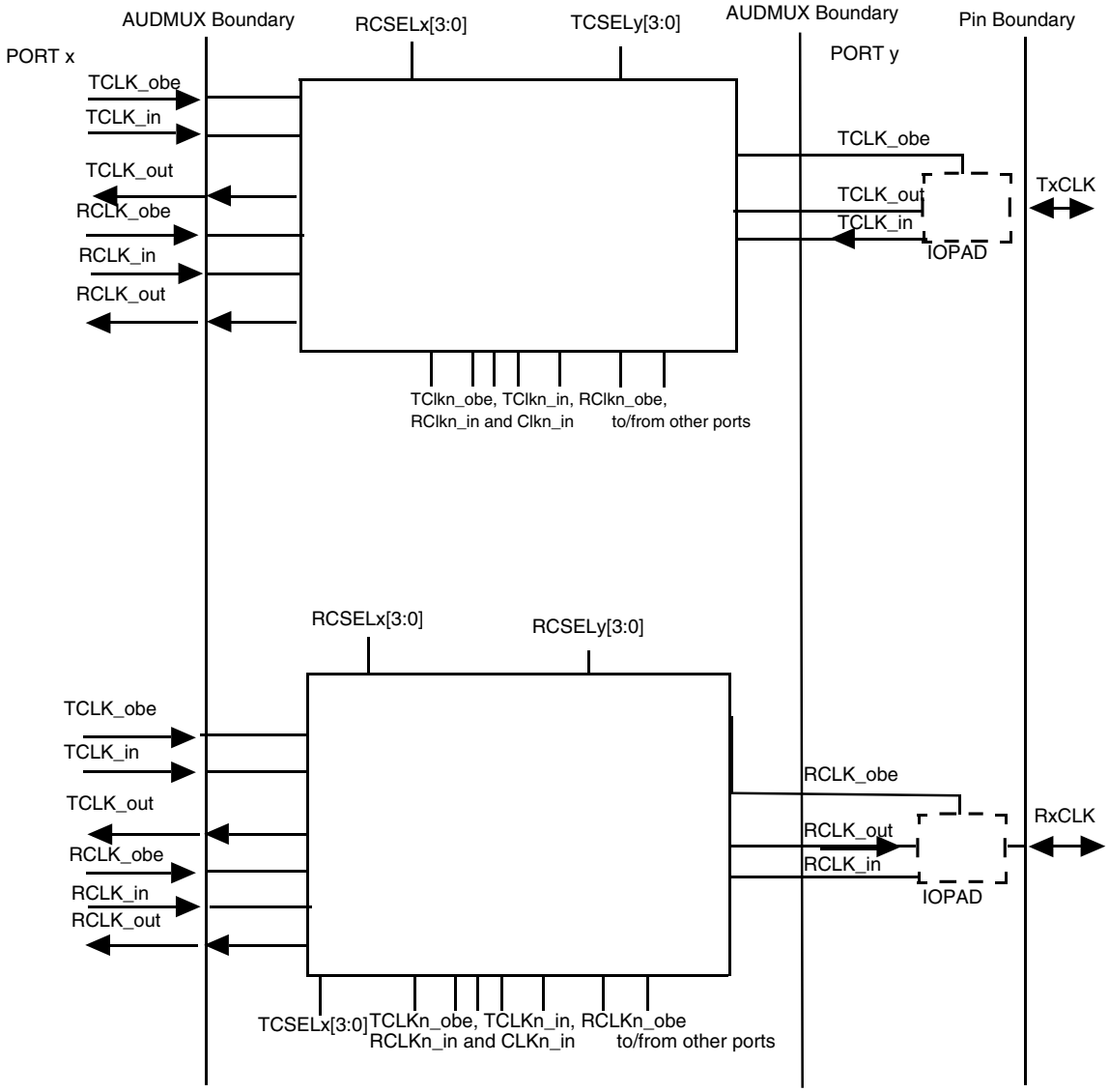


Figure 16-12. Clock Routing When External Port Is 6 Wire

16.4.2 Connectivity Between Ports

Four basic types of connections are provided by the AUDMUX:

- Internal port to external port
- External port to external port
- Internal port to internal port
- Loopback

The corresponding data connections are described in the following sections.

16.4.2.1 Internal Port to External Port Connectivity

The internal port is connected to a processor's serial interface. TxD_obe is the buffer enable signal from the serial interface, TxD_in is the input transmit data from the serial interface to the AUDMUX, and RxD_out is the receive data output from the AUDMUX to the serial interface.

RXDSEL[2:0] of the external port selects the buffer enable signal (TxD_obe) and transmit data output (TxD_out) signal from the TxD_obe and RxD_in signals. RXDSEL[2:0] is a common signal to both selection muxes.

NOTE

Because buffer TxD_in signals from external interfaces do not have corresponding buffer enable signals, their buffer enable signals into the selection mux are tied high. This will ensure that selection of TxD_in, as RxD_out will also drive the RxD_obe output high.

Transmit Data from the serial interface goes into the RXDSEL data mux and comes out as RxD_out. RxD_out is routed to Da_TxRx when TXRXEN is disabled and to D_RxTx when TXRXEN is enabled. Similarly, D_RxTx is routed to TxD_in when TXRXEN is disabled and D_TxRx is routed to TxD_in when TXRXEN is enabled. The routing of frame syncs is shown in Figure 16-11and the routing of interface clocks is shown in Figure 16-9.

If internal network mode is disabled, then RXDSEL selects the TxD_in, which is sent from the AUDMUX to the serial interface connected at Port x. When the internal network mode is selected, RxD_out is constructed by ANDing selected TxD_in signals from the ports (as determined by INMMASK).

If there is more than one device attached to the external port at D_TxRx and D_RxTx and one of the devices is a network master, then two conditions must be noted:

1. When the external master is enabled in network mode, then the serial interface at Port x must be configured as a slave (normal or network mode). No Tx/Rx switching is required.
2. When the external master is disabled and the serial interface at Port x and other slave devices must communicate, then the serial interface at Port x must be configured as a network mode master and the Tx/Rx switch at Port y must be enabled (TXRXEN=1). This will ensure that the transmit and receive paths are connected appropriately.

To communicate with more than one port, internal network mode can be enabled at Port x. In internal network mode, it is possible to communicate with any device attached to the other ports. Internal network mode shall be enabled at the port that is the SSI network mode master.

16.4.2.2 External Port to External Port Connectivity

External ports can communicate with external ports directly.

External ports can communicate together in three ways:

1. Each port's receive logic is configured in normal mode (MODE = 0) . Each port's RXDSEL[2:0] field is configured to select the other port's transmit data. Bit fields associated with clock/frame sync selection and direction are configured for each port. Either port can be the master.
2. One port is configured in internal network mode (MODE = 1) . All desired data lines are combined by the AND gate as determined by INMMASK[7:0]. Since an external port is being used as the internal network mode master, all other devices on the same AUDMUX port as the internal network mode master must be disabled. This configuration can be used with a combination of internal and external ports. All external ports must have a pull-up resistor on its RxDATA pin. Bit fields associated with clock/frame sync selection and direction are configured for each port. Any port can be the master.

16.4.2.3 Internal Port to Internal Port Connectivity

Internal ports can communicate with other internal ports directly, thereby providing a means for synchronous interprocessor communication.

Internal ports can communicate together in two ways:

1. Each port's receive logic is configured in normal mode (MODE = 0) . Each port's RXDSEL[2:0] field is configured to select the other port's transmit data. Bit fields associated with clock/frame sync selection and direction are configured for each port. Either port can be the master.
2. One port is configured in internal network mode (MODE = 1) . All desired data lines are combined by the AND gate as determined by INMMASK[7:0]. This configuration can be used with a combination of internal and external ports. All external ports must have a pull-up resistor on its RxDATA pin. Bit fields associated with clock/frame sync selection and direction are configured for each port. Any port can be the master.

16.4.2.4 Loopback Connectivity

AUDMUX ports can communicate with themselves in order to provide loopback functionality. Port x can route its TxDATA signal to its own RxD_out signal by setting RXDSELx[2:0] to its own port number. This is supported by all ports in the AUDMUX.

In addition, ports can provide loopback support in internal network mode. With internal network mode, the internal network mode master can loop its TxDATA signal (combined with those of other ports, if desired) back into its RxD_out signal. Port x's INMMASK should be set such that bit (x - 1) is clear in order to enable the loopback.

16.4.3 AUDMUX Clocking

This section provides information about AUDMUX clocking including clock inputs and the clock diagram.

16.4.3.1 AUDMUX Clock Inputs

The IP Bus read/write clock-peripheral clock-is an input to the AUDMUX. It is used for all AUDMUX register accesses. It is driven only when there is an AUDMUX access on the IP Bus.

16.4.3.2 AUDMUX Clock Diagram

The figure below shows the clocking used in the AUDMUX.

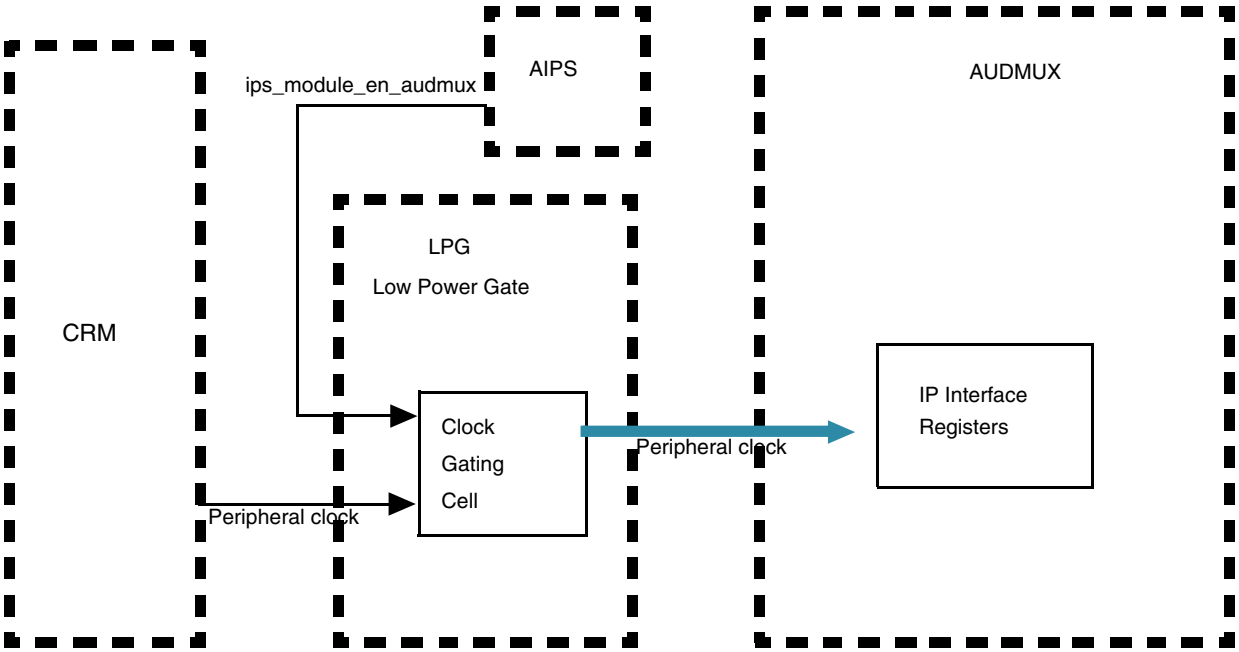


Figure 16-13. AUDMUX Clocking Scheme

16.4.3.3 Clocking Restrictions

16.5 Programmable Registers

This section includes the block memory map and detailed descriptions of all registers. For the base address of a specific sub-block instantiation, see the system memory map in this manual.

The AUDMUX memory map is shown in the following table.

AUDMUX memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FD_0000	Port Timing Control Register 1 (AUDMUX_PTCR1)	32	R/W	AD40_0800h	16.5.1/ 797

Table continues on the next page...

AUDMUX memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FD_0008	Port Timing Control Register 2 (AUDMUX_PTCR2)	32	R/W	A500_0800h	16.5.2/ 799
63FD_0010	Port Timing Control Register 3 (AUDMUX_PTCR3)	32	R/W	9CC0_0800h	16.5.3/ 801
63FD_0018	Port Timing Control Register n (AUDMUX_PTCR)	32	R/W	0000_0800h	16.5.4/ 803

16.5.1 Port Timing Control Register 1 (AUDMUX_PTCR1)

PTCR1 is the Port Timing Control Register for Port 1.

Address: AUDMUX_PTCR1 is 63FD_0000h base + 0h offset = 63FD_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TFS DIR	TFSEL[3:0]				TCLKDIR	TCSEL[3:0]				RFS DIR	RFSEL[3:0]				RCLKDIR
Reset	1	0	1	0	1	1	0	1	0	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RCSEL[3:0]				SYN	0										
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

AUDMUX_PTCR1 field descriptions

Field	Description
31 TFS DIR	Transmit Frame Sync Direction Control. This bit sets the direction of the TxFS pin of the interface as an output or input. When set as an input, the TFSEL settings are ignored. When set as an output, the TFSEL settings determine the source port of the frame sync. 0 TxFS is an input. 1 TxFS is an output.
30–27 TFSEL[3:0]	Transmit Frame Sync Select. Selects the source port from which TxFS is sourced. 0xxx Selects TxFS from port. 1xxx Selects RxFS from port. x000 Port 1 x110 Port 7 x101 Port 6 x111 Reserved x11x Reserved

Table continues on the next page...

AUDMUX_PTCR1 field descriptions (continued)

Field	Description
26 TCLKDIR	Transmit Clock Direction Control. This bit sets the direction of the TxClk pin of the interface as an output or input. When set as an input, the TCSEL settings are ignored. When set as an output, the TCSEL settings determine the source port of the clock. 0 TxClk is an input. 1 TxClk is an output.
25–22 TCSEL[3:0]	Transmit Clock Select. Selects the source port from which TxClk is sourced. 0xxx Selects TxClk from port. 1xxx Selects RxClk from port. x000 Port 1 x110 Port 7 x101 Port 6 x111 Reserved x11x Reserved
21 RFS DIR	Receive Frame Sync Direction Control. This bit sets the direction of the RxFS pin of the interface as an output or input. When set as an input, the RFSEL settings are ignored. When set as an output, the RFSEL settings determine the source port of the frame sync. 0 RxFS is an input. 1 RxFS is an output.
20–17 RFSEL[3:0]	Receive Frame Sync Select. Selects the source port from which RxFS is sourced. RxFS can be sourced from TxFS and RxFS from other ports. 0xxx Selects TxFS from port. 1xxx Selects RxFS from port. x000 Port 1 x110 Port 7 x101 Port 6 x111 Reserved x11x Reserved
16 RCLKDIR	Receive Clock Direction Control. This bit sets the direction of the RxClk pin of the interface as an output or input. When set as an input, the RCSEL settings are ignored. When set as an output, the RCSEL settings determine the source port of the clock. **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 RxClk is an input. 1 RxClk is an output.
15–12 RCSEL[3:0]	Receive Clock Select. Selects the source port from which RxClk is sourced. RxClk can be sourced from TxClk and RxClk from other ports. 0xxx Selects TxClk from port. 1xxx Selects RxClk from port. x000 Port 1 x110 Port 7 x101 Port 6 x111 Reserved x11x Reserved

Table continues on the next page...

AUDMUX_PTCR1 field descriptions (continued)

Field	Description
11 SYN	Synchronous/Asynchronous Select. When SYN is set, synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals (that is, the port is a 4-wire interface). When SYN is cleared, asynchronous mode is chosen and separate clock and frame sync signals are used for the transmit and receive sections (that is, the port is a 6-wire interface). **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 Asynchronous mode 1 Synchronous mode (default)
10–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

16.5.2 Port Timing Control Register 2 (AUDMUX_PTCR2)

PTCR2 is the Port Timing Control Register for Port 2.

Address: AUDMUX_PTCR2 is 63FD_0000h base + 8h offset = 63FD_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TFS DIR	TFSEL[3:0]				TCLKDIR	TCSEL[3:0]				RFS DIR	RFSEL[3:0]				RCLKDIR
Reset	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RCSEL[3:0]				SYN	0										
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

AUDMUX_PTCR2 field descriptions

Field	Description
31 TFS DIR	Transmit Frame Sync Direction Control. This bit sets the direction of the TxFS pin of the interface as an output or input. When set as an input, the TFSEL settings are ignored. When set as an output, the TFSEL settings determine the source port of the frame sync. 0 TxFS is an input. 1 TxFS is an output.
30–27 TFSEL[3:0]	Transmit Frame Sync Select. Selects the source port from which TxFS is sourced. 0xxx Selects TxFS from port. 1xxx Selects RxFS from port. x000 Port 1 x110 Port 7

Table continues on the next page...

AUDMUX_PTCR2 field descriptions (continued)

Field	Description
	x101 Port 6 x111 Reserved x11x Reserved
26 TCLKDIR	Transmit Clock Direction Control. This bit sets the direction of the TxClk pin of the interface as an output or input. When set as an input, the TCSEL settings are ignored. When set as an output, the TCSEL settings determine the source port of the clock. 0 TxClk is an input. 1 TxClk is an output.
25–22 TCSEL[3:0]	Transmit Clock Select. Selects the source port from which TxClk is sourced. 0xxx Selects TxClk from port. 1xxx Selects RxClk from port. x000 Port 1 x110 Port 7 x101 Port 6 x111 Reserved x11x Reserved
21 RFS DIR	Receive Frame Sync Direction Control. This bit sets the direction of the RxFS pin of the interface as an output or input. When set as an input, the RFSEL settings are ignored. When set as an output, the RFSEL settings determine the source port of the frame sync. 0 RxFS is an input. 1 RxFS is an output.
20–17 RFSEL[3:0]	Receive Frame Sync Select. Selects the source port from which RxFS is sourced. RxFS can be sourced from TxFS and RxFS from other ports. 0xxx Selects TxFS from port. 1xxx Selects RxFS from port. x000 Port 1 x110 Port 7 x101 Port 6 x111 Reserved x11x Reserved
16 RCLKDIR	Receive Clock Direction Control. This bit sets the direction of the RxClk pin of the interface as an output or input. When set as an input, the RCSEL settings are ignored. When set as an output, the RCSEL settings determine the source port of the clock. **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 RxClk is an input. 1 RxClk is an output.
15–12 RCSEL[3:0]	Receive Clock Select. Selects the source port from which RxClk is sourced. RxClk can be sourced from TxClk and RxClk from other ports. 0xxx Selects TxClk from port. 1xxx Selects RxClk from port. x000 Port 1

Table continues on the next page...

AUDMUX_PTCR2 field descriptions (continued)

Field	Description
	x110 Port 7 x101 Port 6 x111 Reserved x11x Reserved
11 SYN	Synchronous/Asynchronous Select. When SYN is set, synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals (that is, the port is a 4-wire interface). When SYN is cleared, asynchronous mode is chosen and separate clock and frame sync signals are used for the transmit and receive sections (that is, the port is a 6-wire interface). **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 Asynchronous mode 1 Synchronous mode (default)
10–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

16.5.3 Port Timing Control Register 3 (AUDMUX_PTCR3)

PTCR3 is the Port Timing Control Register for Port 3.

Address: AUDMUX_PTCR3 is 63FD_0000h base + 10h offset = 63FD_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TFS DIR	TFSEL[3:0]				TCLKDIR	TCSEL[3:0]				RFS DIR	RFSEL[3:0]				RCLKDIR
Reset	1	0	0	1	1	1	0	0	1	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RCSEL[3:0]				SYN	0										
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

AUDMUX_PTCR3 field descriptions

Field	Description
31 TFS DIR	Transmit Frame Sync Direction Control. This bit sets the direction of the TxFS pin of the interface as an output or input. When set as an input, the TFSEL settings are ignored. When set as an output, the TFSEL settings determine the source port of the frame sync. 0 TxFS is an input. 1 TxFS is an output.

Table continues on the next page...

AUDMUX_PTCR3 field descriptions (continued)

Field	Description
30–27 TFSEL[3:0]	Transmit Frame Sync Select. Selects the source port from which TxFS is sourced. 0xxx Selects TxFS from port. 1xxx Selects RxFS from port. x000 Port 1 x110 Port 7 x101 Port 6 x111 Reserved x11x Reserved
26 TCLKDIR	Transmit Clock Direction Control. This bit sets the direction of the TxClk pin of the interface as an output or input. When set as an input, the TCSEL settings are ignored. When set as an output, the TCSEL settings determine the source port of the clock. 0 TxClk is an input. 1 TxClk is an output.
25–22 TCSEL[3:0]	Transmit Clock Select. Selects the source port from which TxClk is sourced. 0xxx Selects TxClk from port. 1xxx Selects RxClk from port. x000 Port 1 x110 Port 7 x101 Port 6 x111 Reserved x11x Reserved
21 RFS DIR	Receive Frame Sync Direction Control. This bit sets the direction of the RxFS pin of the interface as an output or input. When set as an input, the RFSEL settings are ignored. When set as an output, the RFSEL settings determine the source port of the frame sync. 0 RxFS is an input. 1 RxFS is an output.
20–17 RFSEL[3:0]	Receive Frame Sync Select. Selects the source port from which RxFS is sourced. RxFS can be sourced from TxFS and RxFS from other ports. 0xxx Selects TxFS from port. 1xxx Selects RxFS from port. x000 Port 1 x110 Port 7 x101 Port 6 x111 Reserved x11x Reserved
16 RCLKDIR	Receive Clock Direction Control. This bit sets the direction of the RxClk pin of the interface as an output or input. When set as an input, the RCSEL settings are ignored. When set as an output, the RCSEL settings determine the source port of the clock. **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 RxClk is an input. 1 RxClk is an output.

Table continues on the next page...

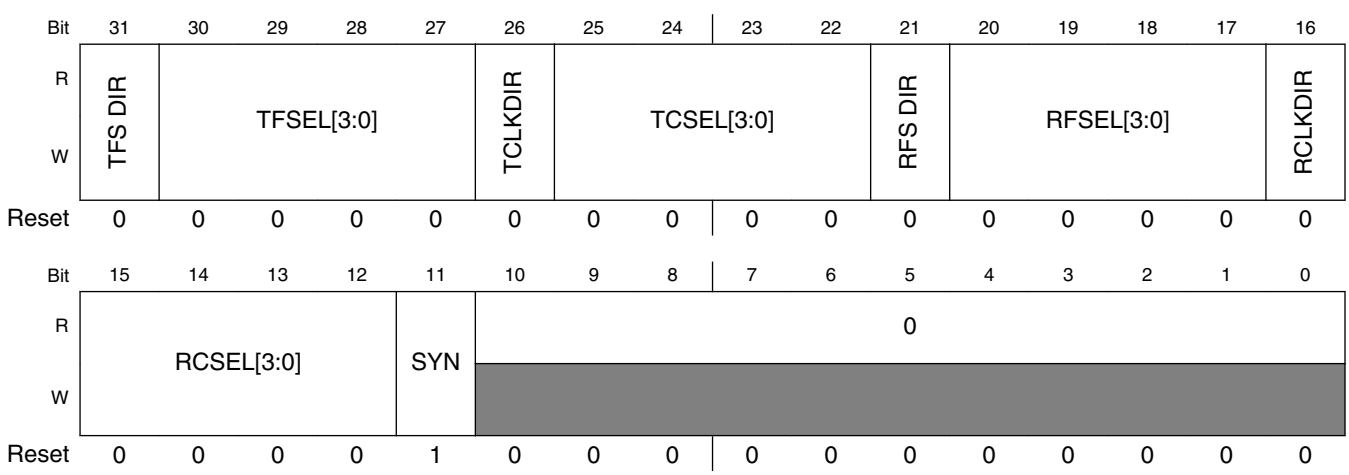
AUDMUX_PTCR3 field descriptions (continued)

Field	Description
15–12 RCSEL[3:0]	Receive Clock Select. Selects the source port from which RxClk is sourced. RxClk can be sourced from TxClk and RxClk from other ports. 0xxx Selects TxClk from port. 1xxx Selects RxClk from port. x000 Port 1 x110 Port 7 x101 Port 6 x111 Reserved x11x Reserved
11 SYN	Synchronous/Asynchronous Select. When SYN is set, synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals (that is, the port is a 4-wire interface). When SYN is cleared, asynchronous mode is chosen and separate clock and frame sync signals are used for the transmit and receive sections (that is, the port is a 6-wire interface). **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 Asynchronous mode 1 Synchronous mode (default)
10–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

16.5.4 Port Timing Control Register n (AUDMUX_PTCR)

PTCR*n* is the Port Timing Control Register for Port *n*, where *n* ranges from 4 through 7.

Address: AUDMUX_PTCR is 63FD_0000h base + 18h offset = 63FD_0018h



AUDMUX_PTCR*n* field descriptions

Field	Description
31 TFS DIR	Transmit Frame Sync Direction Control. This bit sets the direction of the TxFS pin of the interface as an output or input. When set as an input, the TFSEL settings are ignored. When set as an output, the TFSEL settings determine the source port of the frame sync. 0 TxFS is an input. 1 TxFS is an output.
30–27 TFSEL[3:0]	Transmit Frame Sync Select. Selects the source port from which TxFS is sourced. 0xxx Selects TxFS from port. 1xxx Selects RxFS from port. x000 Port 1 x110 Port 7 x101 Port 6 x111 Reserved x11x Reserved
26 TCLKDIR	Transmit Clock Direction Control. This bit sets the direction of the TxClk pin of the interface as an output or input. When set as an input, the TCSEL settings are ignored. When set as an output, the TCSEL settings determine the source port of the clock. 0 TxClk is an input. 1 TxClk is an output.
25–22 TCSEL[3:0]	Transmit Clock Select. Selects the source port from which TxClk is sourced. 0xxx Selects TxClk from port. 1xxx Selects RxClk from port. x000 Port 1 x110 Port 7 x101 Port 6 x111 Reserved x11x Reserved
21 RFS DIR	Receive Frame Sync Direction Control. This bit sets the direction of the RxFS pin of the interface as an output or input. When set as an input, the RFSEL settings are ignored. When set as an output, the RFSEL settings determine the source port of the frame sync. 0 RxFS is an input. 1 RxFS is an output.
20–17 RFSEL[3:0]	Receive Frame Sync Select. Selects the source port from which RxFS is sourced. RxFS can be sourced from TxFS and RxFS from other ports. 0xxx Selects TxFS from port. 1xxx Selects RxFS from port. x000 Port 1 x110 Port 7 x101 Port 6 x111 Reserved x11x Reserved

Table continues on the next page...

AUDMUX_PTCR*n* field descriptions (continued)

Field	Description
16 RCLKDIR	Receive Clock Direction Control. This bit sets the direction of the RxClk pin of the interface as an output or input. When set as an input, the RCSEL settings are ignored. When set as an output, the RCSEL settings determine the source port of the clock. **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 RxClk is an input. 1 RxClk is an output.
15–12 RCSEL[3:0]	Receive Clock Select. Selects the source port from which RxClk is sourced. RxClk can be sourced from TxClk and RxClk from other ports. 0xxx Selects TxClk from port. 1xxx Selects RxClk from port. x000 Port 1 x110 Port 7 x101 Port 6 x111 Reserved x11x Reserved
11 SYN	Synchronous/Asynchronous Select. When SYN is set, synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals (that is, the port is a 4-wire interface). When SYN is cleared, asynchronous mode is chosen and separate clock and frame sync signals are used for the transmit and receive sections (that is, the port is a 6-wire interface). **NOTE:** RCLKDIR and SYN should not be changed at the same time. 0 Asynchronous mode 1 Synchronous mode (default)
10–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

Chapter 17
Clock Amplifier (CAMP)

17.1 Introduction

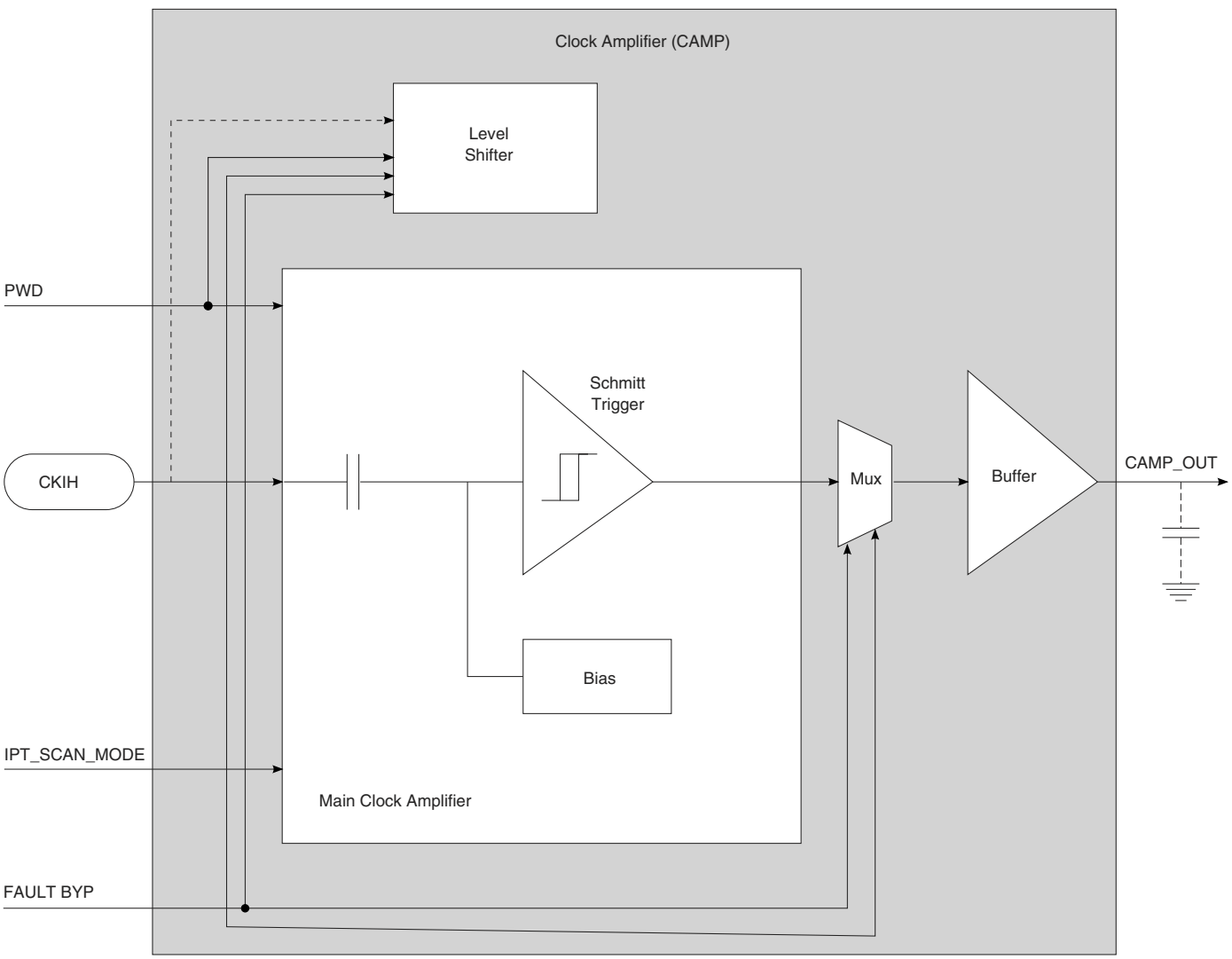


Figure 17-1. Clock Amplifier Block Diagram

17.1.1 Overview

The Clock Amplifier converts a square wave/sinusoidal input of frequency range 8-40 MHz, into a rail to rail square wave (CAMP supply voltage).

The input of the CAMP block is CKIH and the output is CAMP_OUT. The input to CAMP is internally AC coupled (in normal mode); no external coupling is required.

17.1.2 Features

The CAMP includes the following features:

- Converts sinusoidal input to square wave.
- Can accept a square wave of amplitude greater than the supply voltage of the block.

17.1.3 Modes of Operation

The CAMP supports three modes of operation.

17.1.3.1 Normal Mode

In this mode of operation, the Clock Amplifier accepts a sinusoidal/square wave as input and gives a rail to rail square wave output.

17.1.3.2 Power Down Mode

In this mode the CAMP is disabled and put in the low power state. CAMP_OUT remains at logic'0' level during power down.

17.1.3.3 Test Mode (Fault Bypass or Scan)

The Test mode is to be used for testing purpose or during scan. The MAIN CLOCK AMPLIFIER is bypassed in this mode.

17.2 External Signal Description

17.2.1 Signals Overview

Table 17-1. CAMP External Signal Properties

Name	Direction	Function
CKIH	Input	Input clock

Table 17-2. CAMP Interface Signal Properties

Name	Direction	Function
VDD	Input	Power supply
VSS	Input	Ground
PWD	Input	Power down signal
FAULT_BYP	Input	Test mode control signal
IPT_SCAN_MODE	Input	Scan mode signal
CAMP_OUT	Output	Output clock from CAMP

17.2.2 Detailed Signal Description

17.2.2.1 CKIH - External Clock Input

The signal CKIH is the clock input signal to the CAMP. It can be a sinusoidal input with a minimum swing of 400mV p-p or a square wave. It comes directly from the pad.

17.2.2.2 VDD - Power supply

The VDD is the power supply for the CAMP. It supplies the main clock amplifier, buffer and level shifter.

17.2.2.3 VSS - Ground

The VSS input is the ground for the CAMP.

17.2.2.4 IPT_SCAN_MODE - Scan Signal

This signal is active high. When this signal is asserted CAMP is forced to "Test Mode" irrespective of states of fault_byp or pwd signals.

17.2.2.5 PWD - Power Down Signal

This signal when asserted (and ipt_scan_mode = 0), puts the CAMP in the low power mode. This signal is active high.

17.2.2.6 FAULT_BYP - Control Signal for Test Mode

This signal when asserted puts CAMP in test mode. This signal is active high. When this signal is low (and ipt_scan_mode = 0) CAMP works in normal mode.

17.2.2.7 CAMP_OUT - Clock Output from CAMP

This signal is the output of the CAMP.

17.3 Memory Map/Register Definition

The CAMP gets its control signals PWD and FAULT_BYP from registers residing in the CRM. It gets its control signal IPT_SCAN_MODE from TSCM (Test and Scan Control sub-block).

17.4 Functional Description

This section provides a functional description of the Clock Amplifier. The block can be divided into three parts. First the Main Clock Amplifier, second the Output Buffer and third the Level Shifter.

17.4.1 CAMP Sub-Blocks

This section describes the Main Clock amplifier, the Output Buffer and Level Shifter of CAMP. (See Figure 17-1).

17.4.1.1 Main Clock Amplifier

The input clock is ac coupled to the input of a Schmitt trigger. A dc bias generation circuit is used to provide a fixed dc to the input of the Schmitt trigger. The Schmitt trigger converts a sinusoidal signal to a square wave.

17.4.1.2 Output Buffer

The output of the main clock amplifier is buffered by the output buffer.

17.4.1.3 Level Shifter

This is activated in the test mode. It level shifts the input signal to CAMP supply.

17.4.2 CAMP Modes of Operation

Table 17-3. Truth Table for CAMP

IPT_SCAN_MODE	PWD	FAULT_BYP	CAMP_OUT	Main Clock Amplifier Path	Level Shifter Path
0	0	0	Square Wave	Enabled	Disabled
0	0	1	Square Wave	Disabled	Enabled
0	1	X	Logic'0'	Disabled	Disabled
1	X	X	Square Wave	Disabled	Enabled

17.4.2.1 Normal Mode

In this mode of operation the IPT_SCAN_MODE, PWD and FAULT_BYP signals are de-asserted. The sinusoidal (or square wave) clock input (with peak to peak amplitude not exceeding 3 Volts) is converted to square wave with amplitude equal to the Camp supply.

17.4.2.2 Power Down Mode

In this mode of operation, the CAMP is powered down. The supply to Main Clock Amplifier and level shifter is turned off by power gating. The CAMP_OUT is a t logic '0' level.

17.4.2.3 Test Mode

In this mode the MAIN CLOCK AMPLIFIER is bypassed via a level shifter. It accepts only square wave input with voltage swing greater than or equal to CAMP supply voltage. There is no guarantee of output clock duty cycle in this mode. To obtain 45%-55% duty cycle output from CAMP in this mode, the input must be a square wave at CAMP's supply level with slews less than or equal to 2ns.

17.5 Initialization/Application Information

During initialization it has to be ensured that PWD and IPT_SCAN_MODE remain de-asserted.

Chapter 18
Clock Control Module (CCM)

18.1 Overview

The Clock Control Module (CCM) controls the clocks for i.MX53 blocks. This block uses the available clock sources to generate the clock roots. Figure 1-1 shows the CCM block diagram.

The Clock Control Module controls the following functions in i.MX53:

1. Uses the available clock sources to generate clock roots to various parts of the chip.
2. Uses program able bits to control frequencies of the clock roots.
3. Controls the low power mechanism.
4. Provides control signals to low power clock gating (LPCG) for gating clocks.
5. Provides handshake with system reset controller (SRC) for reset performance.
6. Provides handshake with general power controller (GPC) for support of dvfs, dptc and power gating operations.

18.1.1 Features

The CCM includes these distinctive features:

- Clock switcher block to generate four switcher source clocks based on four DPLL's.
- Root clocks generation - provides root clock to i.MX53's blocks based on four switcher source clocks.
- ARM core root clock generated from dedicated switcher source clock.
- Includes separate dividers to control generation of core and bus root clocks (axi's, ahb, ipg).
- Includes separate dividers and clock sources selectors for each serial root clock.
- Option for external clock to bypass DPLL's clocks.
- Selects clock signals to output on CLKO and CLKO2 on to the pads for observability.
- Controllable registers are accessible via IP Bus.

- Manages the Low Power Modes, namely RUN, WAIT, and STOP. The gating of the peripheral clocks is programmable in RUN and WAIT modes.
- Manages frequency scaling procedure for ARM core clock by shifting between DPLL sources, without loss of clocks.
- Manages frequency scaling procedure for peripheral clock roots by programable divider. The division is done on the fly without the loss of clocks.
- DFT feature support.
- Interface for the following block's
  - DPLLC - DPLLC interfaces for each DPLL on the IC.
  - LPCG - Low Power Clock Gating unit
  - SRC - System reset Controller
  - GPC - General Power Controller

18.1.2 CCM Block Diagram

CCM includes the following sub-blocks:

CCM_CLK_IGNITION - This sub-block manages the ignition process. This block functions once the CCM comes out of reset. It starts the Clock Amplifier (CAMP-1 and CAMP-2), the DPLL's and creates stable output root clocks after reset.

CCM_CLK_SWITCHER - This sub-block receives the clock outputs from the four DPLL's, together with the respective bypass clocks from the input and generates four switcher clock outputs (pll1_sw_clk, pll2_sw_clk, pll3_sw_clk, pll4_sw_clk) for the CCM_CLK_ROOT_GEN sub-block.

CCM_CLK_ROOT_GEN - This sub-block receives the four main clocks (pll1_sw_clk, pll2_sw_clk, pll3_sw_clk, pll4_sw_clk) and generates the output root clocks.

CCM_CLK_LOGIC - This sub-block generates the clock enables. It generates the clock enable signals based on information from CCM_LPM and CCM_IP. The clock enables are used by LPCG to turn on and off the clock branches it generates.

CCM_LPM: This sub-block manages the low power modes of the IC.

CCM_GSBI_TOP: This sub-block manages the memory map of CCM. It holds the programable registers and the connectivity to the IP bus. This block is connected to all the sub-blocks that need definition of programmable bits. Note that in the diagram below appears only connectivity to the ccm_clk_logic sub-blocks.

CCM_CLK_SRC_DIV - This sub-block multiplexes different clocks to the two output clocks, CLKO and CLKO2. These output clocks are connected to the pads and can support testing.

CCM_HND_SK - This sub-block manages the handshake when changing root clock dividers that require handshake, and manages the frequency change in case of dvfs scenario.

The following figure describes the CCM boundary.

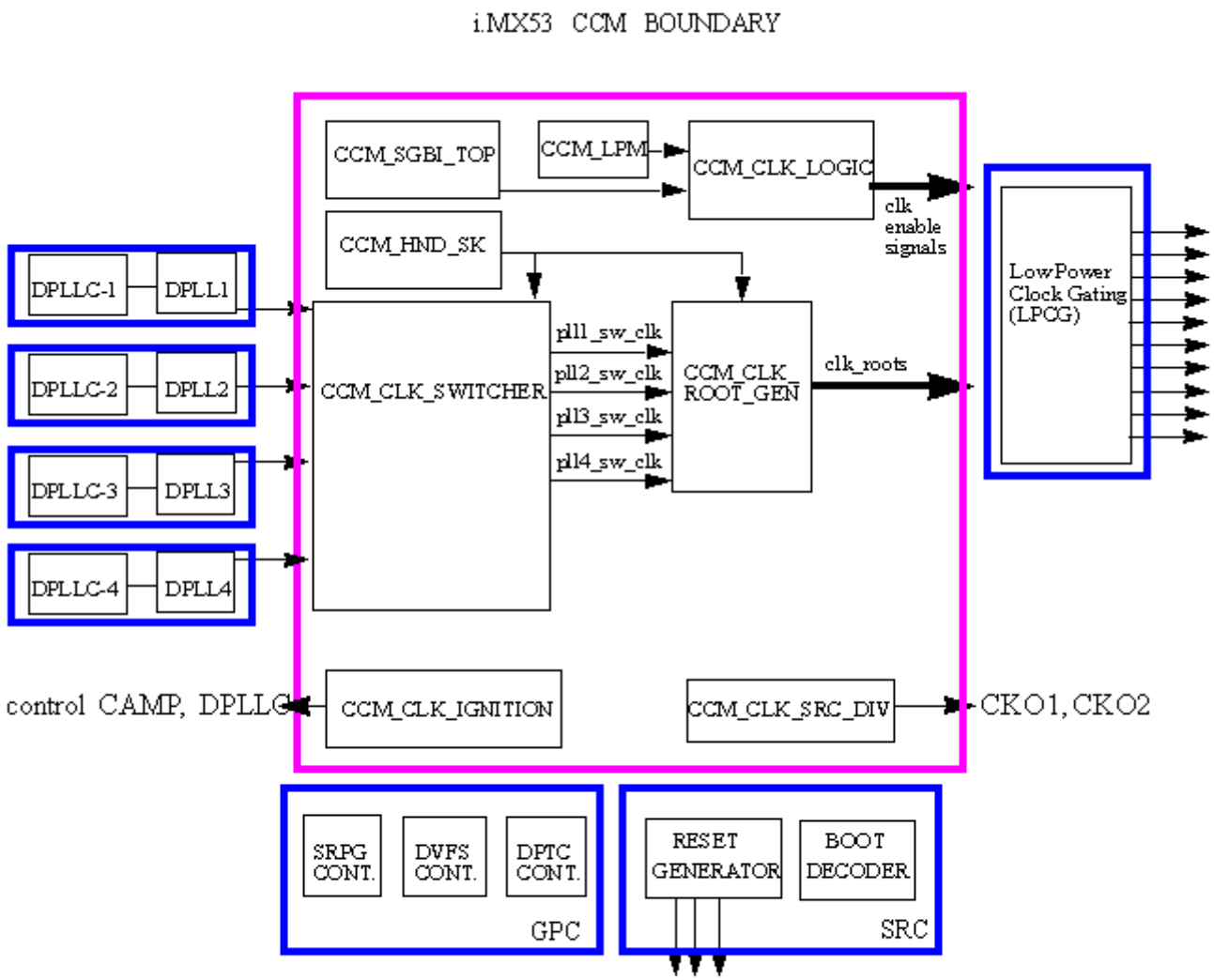


Figure 18-1. CCM Block Diagram

The following figure shows the entire clock tree for i.MX53

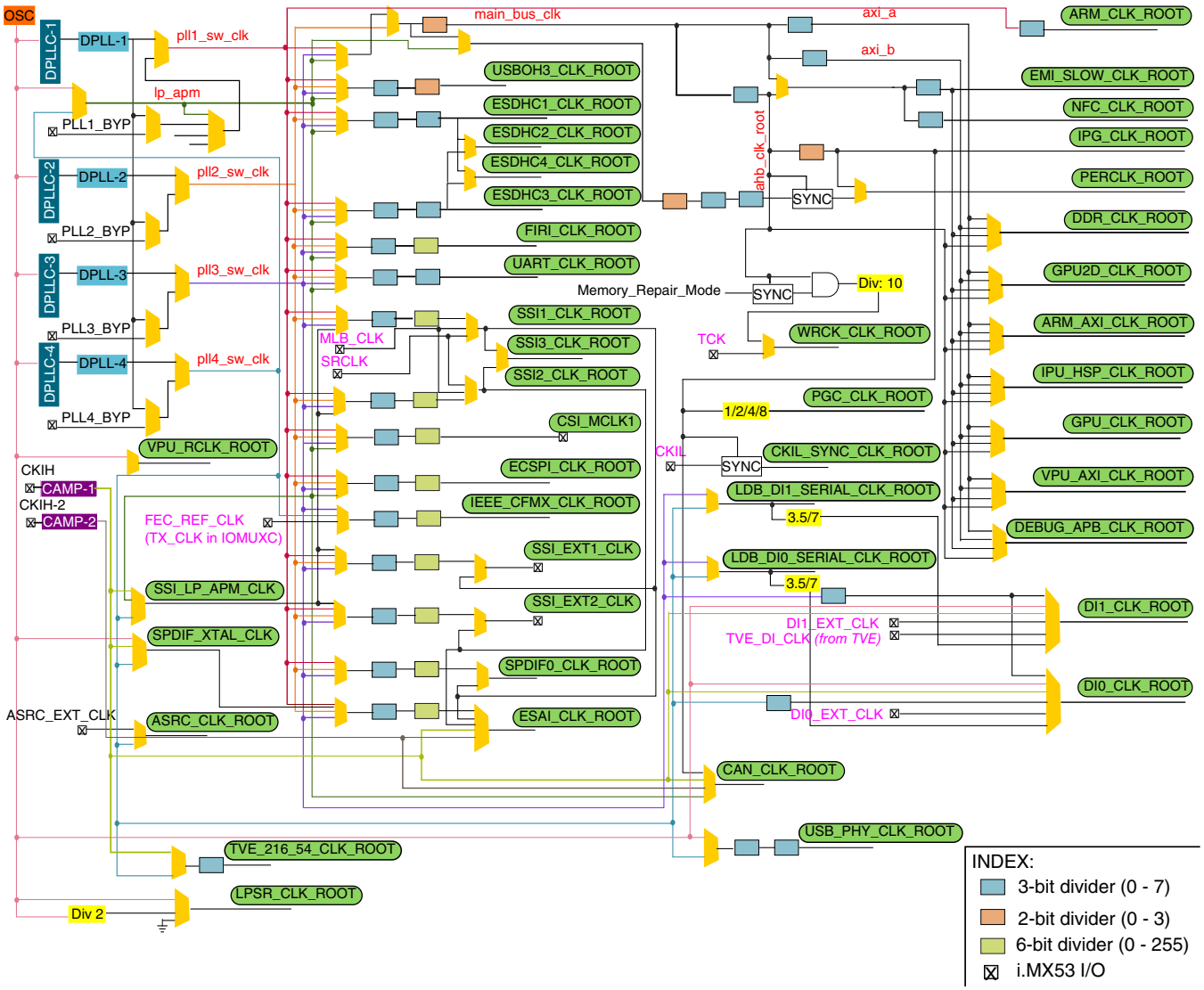


Figure 18-2. CCM Clock Tree

18.2 Functional Description

18.2.1 Clock Generation

18.2.1.1 External Low Frequency Clock - CKIL

i.MX53 can use either a 32 kHz, 32.768 kHz or a 38.4 kHz crystal as the external low frequency source. Throughout this chapter, the low frequency crystal is referred to as the 32 kHz crystal. CMOS input buffer with shmitt trigger feature is used as receiver of

32KHz clock. This clock source should be active all the time i.MX53 is powered on. The 32kHz entering the CCM is referred to as CKIL. It is synchronized to ipg_clk and supplied to blocks that need it.

18.2.1.2 External High Frequency Clock - CKIH and internal oscillator

i.MX53 uses internal oscillator to generate the reference clock. The internal oscillator is connected to an external crystal. The oscillator generates frequencies in the range of 22-27Mhz.

i.MX53 uses CKIH and CKIH2 input pins to generate special serial clock demands (like TVE, SPDIF, etc.) as a backup to the DPLL clocks.

18.2.1.3 DPLL reference clock

There are four DPLL's in i.MX53 project namely:

- DPLL-1 (typical functional frequency 800Mhz)
- DPLL-2 (typical functional frequency 400Mhz)
- DPLL-3 (typical functional frequency 216Mhz)
- DPLL-4 (typical functional frequency 595Mhz)

Each DPLL is controlled by a DPLLC-n interface block. Each DPLLC interface block uses the output of on chip oscillator (typical frequency is 24Mhz) as DPLL reference clock:

18.2.1.4 CCM Internal Clock Generation

The clock generation is comprised of three sub-blocks:

- CCM_CLK_SWITCHER
- CCM_CLK_IGNITION
- CCM_CLK_ROOT_GEN

CCM_CLK_SWITCHER sub-block receives the DPLL output clocks and the DPLL bypass clocks. It generates 4 main switcher clocks to be delivered to CCM_CLK_ROOT_GEN:

- pll1_sw_clk (typical functional frequency 800Mhz - used to supply ARM platform)
- pll2_sw_clk (typical functional frequency 400Mhz - used to supply axi/ahb/ip buses clocks)
- pll3_sw_clk (typical functional frequency 216Mhz - used to supply serial clocks like usb, ssi, etc.)

pll4_sw_clk (typical functional frequency 595Mhz - used to supply LDB clocks)

The figure below describes the generation of the four switcher clocks and a few root clocks.

The figure also includes Frequency Switch Control sub-block that is responsible of frequency change during DVFS operation.

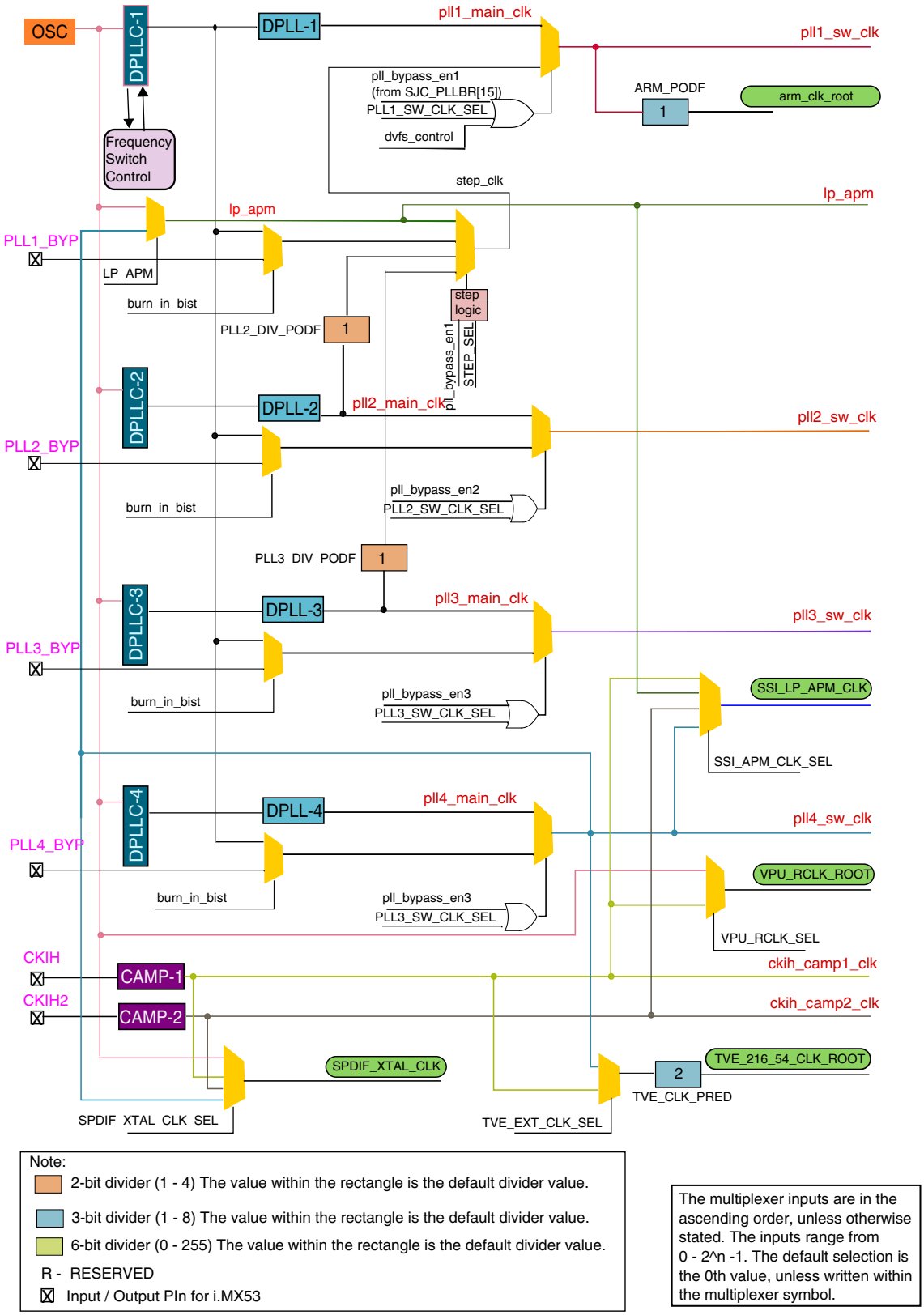


Figure 18-3. Detailed Clock Tree 1

18.2.1.4.1 DPLL Bypass Procedure

CCM_CLK_SWITCHER sub-block includes capability for each of the pll_main_clk's to be bypassed with an external bypass clk. In burn_in_bist mode the DPLL bypass clocks for all DPLLs are supplied from pll1_ref_clk (the output of DPLLC-1).

The decision to shift from pll_main_clk to pll_bypass_clk is done either by programable bits in CCM_CCSR or by pll_bypass_enn signals which are driven from the jtag block. As the switch between the clocks is done by a glitch less multiplexer, both pll_main_clk and pll_bypass_clk should be running to complete the shift glitchlessly. In case the selection is done during reset, then it is not necessary for both clocks to be available simultaneously.

As the bypass clock for pll1_sw_clk is generated from a multiplexer, the bypass clock needs to be set via CCM_CCSR[step_sel] before shifting the glitch less multiplexer via CCM_CCSR[PLL1_SW_CLK_SEL]. Similarly, when returning from pll_bypass, system should first change the glitch less multiplexer via PLL1_SW_CLK_SEL and only after that should it change the STEP_SEL bits.

18.2.1.4.2 Step Logic

The step logic allows the DPLL bypass enable signal to bypass the step_sel bits and to set the 4*1 multiplexer to the pll1_bypass_clk option.

18.2.1.4.3 DPLLC Reference Clock Connectivity

For all four DPLLC's:

The input clk2 is connected to on board oscillator clock output.

The input init_dp_ctl_dpllip [8] is connected to gnd.

The input init_dp_ctl_dpllip [9] is connected to vcc.

The above connections allows DPLLC to choose the on chip oscillator as the reference clock to the DPLL.

18.2.1.4.4 DPLL Clock Change

If the DPLL clock output of a specific DPLL needs to change, or if a specific DPLL needs to be updated, then first move all the clocks generated by the DPLL to another DPLL which is not changed. This should be done via the glitch less multiplexers for the clocks which can not be stopped (core and bus clocks). Then, reprogram the respective DPLLC: First configure DPLLC_CONFIG[1], so that it allows auto restart of the DPLL

on next update of the reference clock. Then change the DPLL values to the new aimed values. In this case the respective DPLL restarts and once its locked, it sends a signal to CCM. This signal can cause an interrupt from CCM (if configured in CCM_CIMR register). This interrupt, notifies that the DPLL is ready with the new reference clocks. Take care while reprograming the DPLL settings to match the new reference clock. At this point the multiplexer can be changed via the glitch less multiplexers to supply clock from the respective DPLL that was updated.

18.2.1.4.5 CCM_CLK_IGNITION

The responsibility of the CCM_CLK_IGNITION sub-block is to manage the wake up after reset of the on chip oscillator, CAMP-1, CAMP-2, DPLLC and DPLL blocks. Its task starts with de-assertion of early_reset from the reset controller and finishes with the assertion of signal that notifies the reset controller that the root clocks are ready.

As the on chip oscillator is the reference clock for the DPLLs, the first step in the ignition process is to assert CCM_CCR[COSC_EN] signal and to de-assert CCM_CLPCR[COSC_PWRDOWN] to start the on chip oscillator. The external oscillator and CAMPs are enabled during ignition proces, that is, CCM_CCR[CAMP1_EN] and CCM_CCR[CAMP2_EN] are asserted (= '1') and CCM_CSR[REF_EN_B] is de-asserted (= '0') so that ckih_camp1 and ckih2_camp2 clocks are started. Next CCM_CLK_IGNITION counts CKIL cycles defined in CCM_CCR[OSCNT]. Once the counter has elapsed, it is assumed that the clock output of the on chip oscillator and CAMP's is ready. In this stage CCM asserts the COSC_READY, CAMP1_READY and CAMP2_READY bits.

Both OSCNT counts for on board oscillator and external oscillator are performed in parallel, so that when the count elapses, it implies that both oscillators are ready. This is done to save time during the ignition process.

Next, CCM_CLK_IGNITION asserts a signal to enable the reference clocks to the DPLL's.

Next, CCM_CLK_IGNITION asserts enable signals for the DPLLCs so that dpllip_cpen is generated.

NOTE

For this to happen, initial value of UPEN should be '1' i.e. init_dp_ctl_dpllip[5] for all DPLL's should be tied to '1'.,

Once the DPLL's are locked, DPLLCs generate DPLL locked signal to CCM_CLK_IGNITION.

Upon assertion of all four DPLL locked signals, CCM_CLK_IGNITION generates a clock ready signal for the reset controller to indicate that DPLL clocks are ready

If DPLL bypass signal is asserted, then clock ready signal to the reset controller is asserted immediately after exiting reset. Once the clock ready signal arrives at the reset controller, the reset sequence continues.

18.2.1.4.6 Reset Values for DPLLC

Reset values that are hard coded as DPLLC initialization values are:

DPLL1 - initial value =192Mhz

DPLL2 - initial value = 192Mhz

DPLL3 - initial value = 168Mhz

DPLL4 - initial value = 168Mhz

These frequencies correspond to the reference clock source of on chip oscillator (24 MHz) as the source for DPLL. For different frequencies, a linear ratio should be applied.

NOTE

The frequencies above are the reset values. The BOOT ROM code can change the DPLL settings according to the configured BOOT mode configuration and FUSE settings.

18.2.1.4.7 CCM_CLK_ROOT_GEN

CCM_CLK_ROOT_GEN sub-block generates the root clocks to be delivered to LPCG.

The following is a list of the root clocks generated by this block. The clocks are supposed to be asynchronous unless mentioned differently:

ARM_CLK_ROOT - generated from pll1_sw_clk.

EMI_SLOW_CLK_ROOT

eNFC_CLK_ROOT - generated by division of EMI_SLOW_CLK_ROOT and balanced with it.

VPU_RCLK_ROOT

AHB_CLK_ROOT

IPG_CLK_ROOT - generated by division of AHB_CLK_ROOT and balanced with it.

PERCLK_ROOT - this clock is synchronized and balanced to AHB_CLK_ROOT. For the synchronization process, peripherals clock pre-divider and post-divider should generate clock with frequency 2.5 times lower than AHB clock. During DVFSP operation, when AHB is reduced, peripherals clock should be configured to be 2.5 times lower than the minimum value of AHB_CLK_ROOT (the DVFS value of ahb_clk).

DDR_CLK_ROOT

ARM_AXI_CLK_ROOT

IPU_HSP_CLK_ROOT

CKIL_SYNC_CLK_ROOT - ckil clock is synchronized to IPG_CLK_ROOT and balanced with it, when not in stop mode. Synchronizer is bypassed, when in stop mode.

USBOH3_CLK_ROOT

ESDHC1_CLK_ROOT

ESDHC2_CLK_ROOT

ESDHC3_CLK_ROOT

UART_CLK_ROOT

SSI1_CLK_ROOT

SSI2_CLK_ROOT

SSI_EXT1_CLK - connected to external pad

SSI_EXT2_CLK - connected to external pad

USB_PHY_CLK_ROOT

TVE_216_54_CLK_ROOT - The default clock frequency is 595MHz. For HDTV the user needs to change the frequency to 594MHz.

DI_CLK_ROOT

SPDIF0_CLK_ROOT

ECSPI_CLK_ROOT

WRCK_CLK_ROOT

LPSR_CLK_ROOT

PGC_CLK_ROOT - generated as division of ipg_clk and balanced to it.

Following figures describe the above clocks generation.

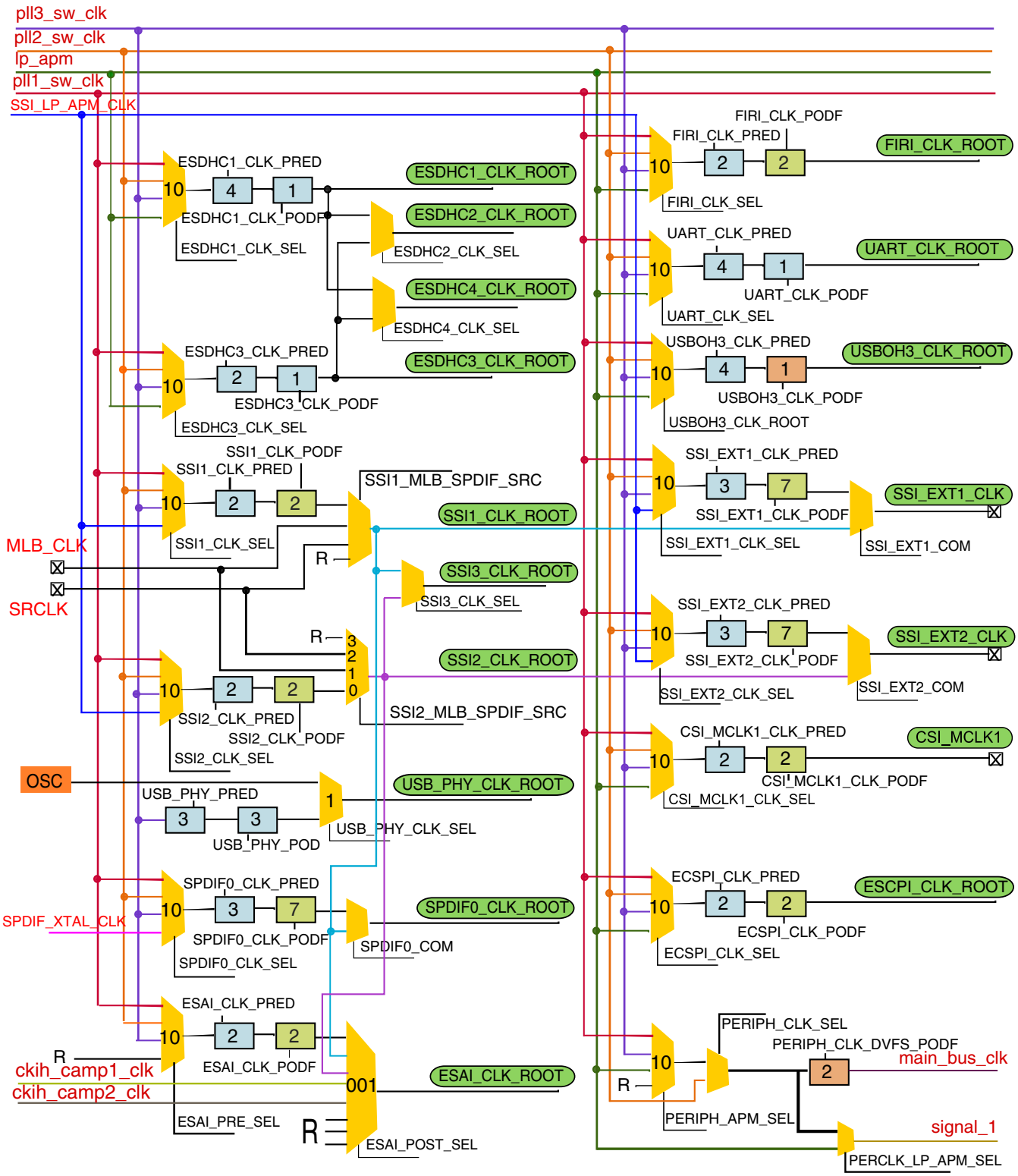


Figure 18-4. Root Clock Generation

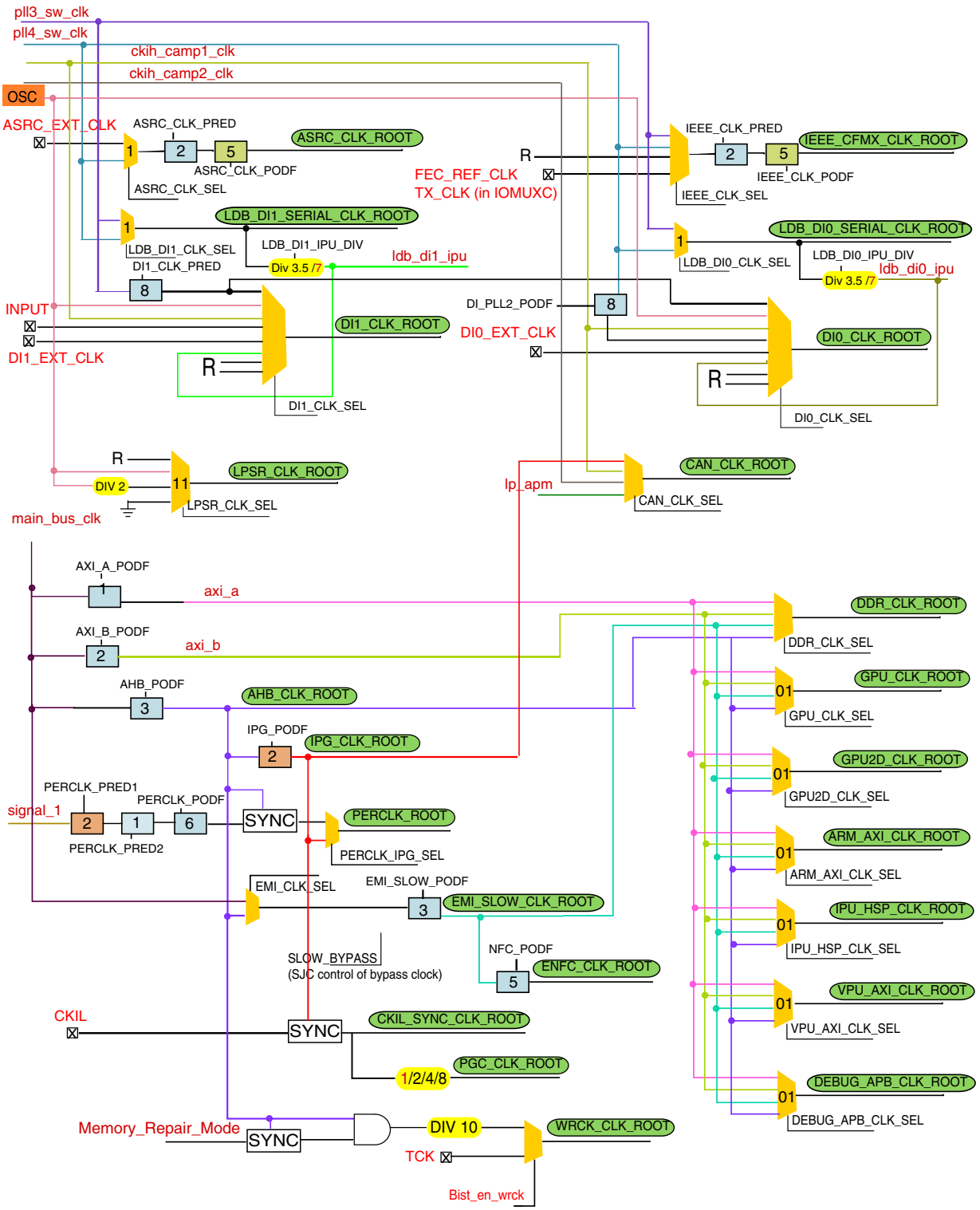
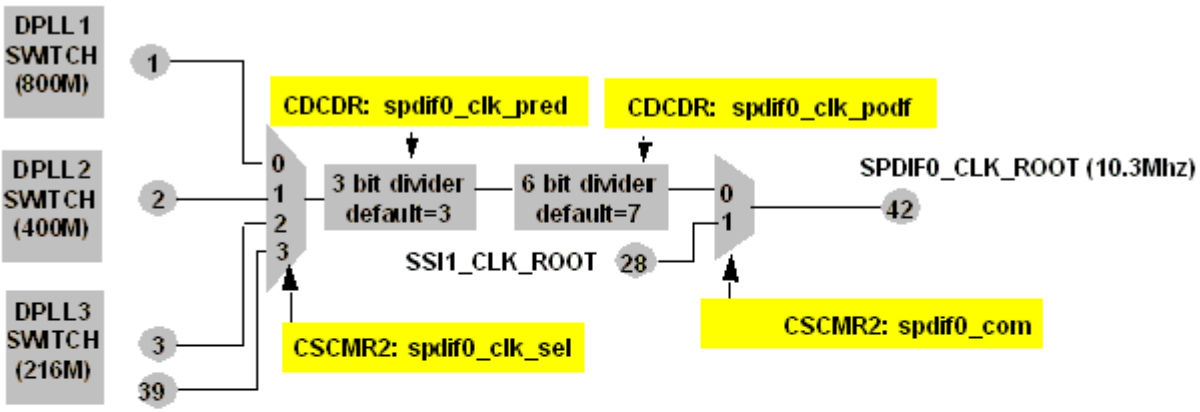
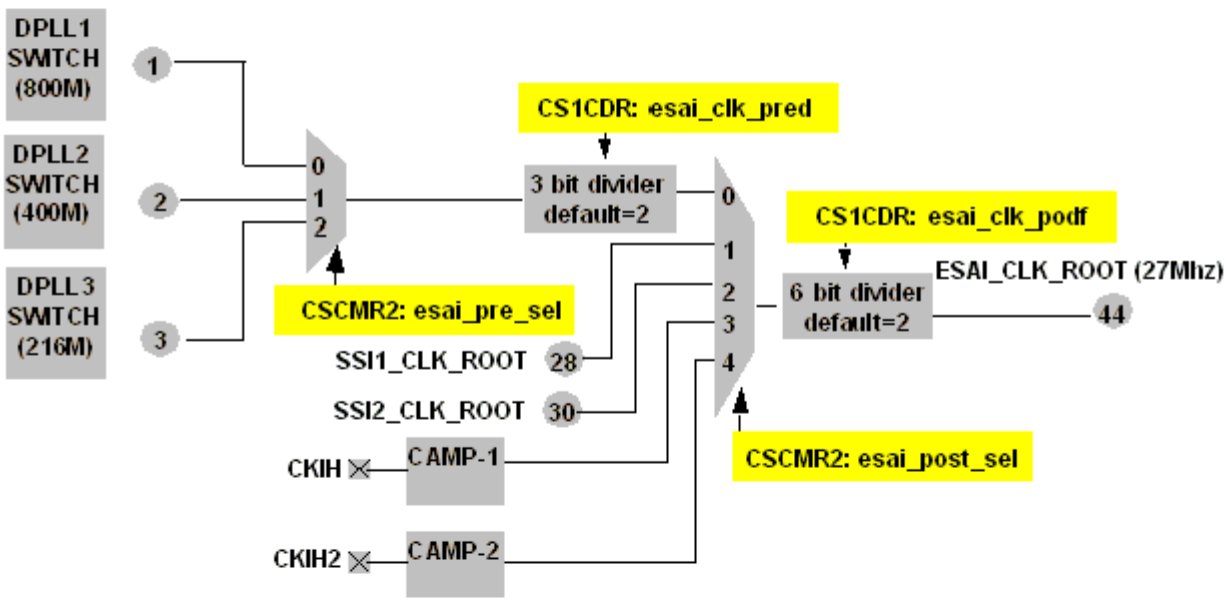
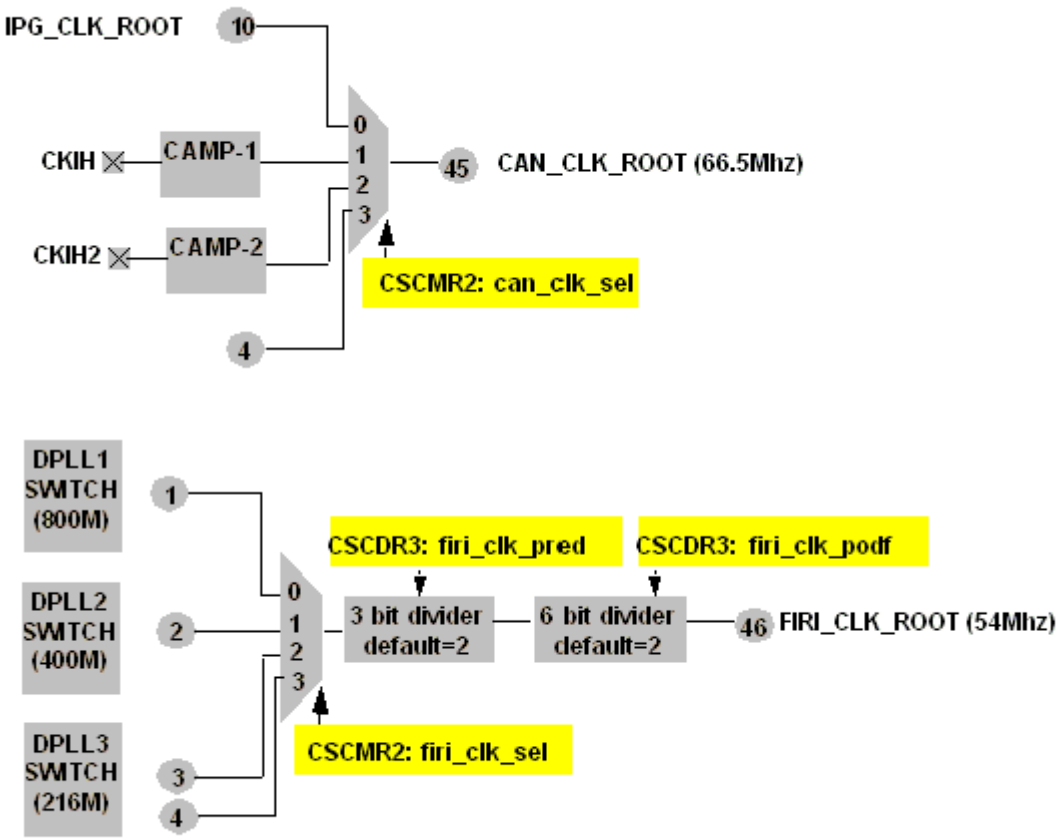
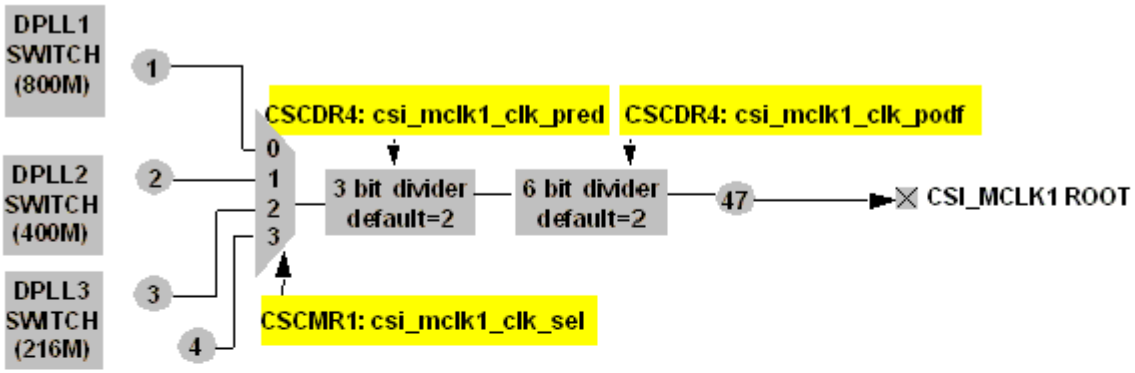


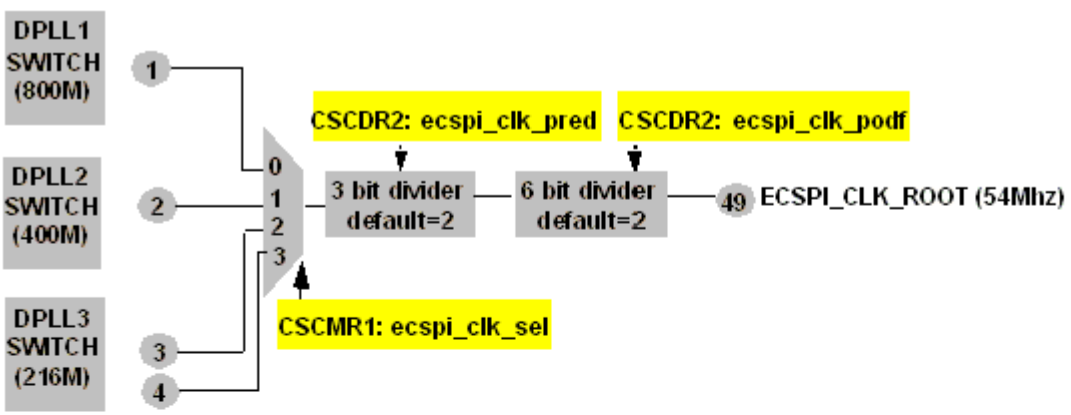
Figure 18-5. Root Clock Generation 2

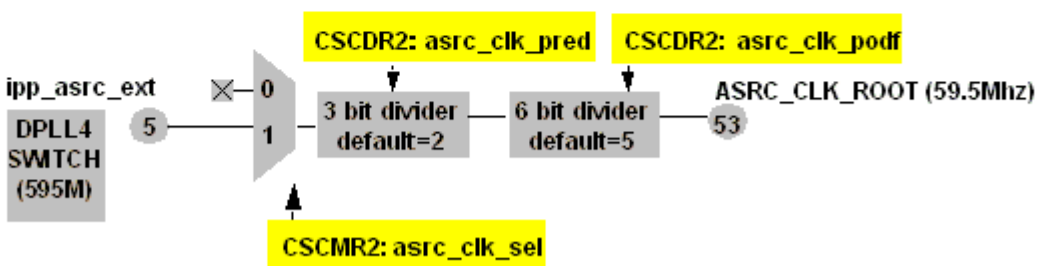

DPLL1 SWITCH (800M)
DPLL2 SWITCH (400M)
DPLL3 SWITCH (216M)
1
2
3
39
0
1
2
3
CDCDR: spdif0_clk_pred
3 bit divider default=3
CDCDR: spdif0_clk_podf
6 bit divider default=7
0
1
SPDIF0_CLK_ROOT (10.3Mhz)
42
SSI1_CLK_ROOT
28
CSCMR2: spdif0_clk_sel
CSCMR2: spdif0_com

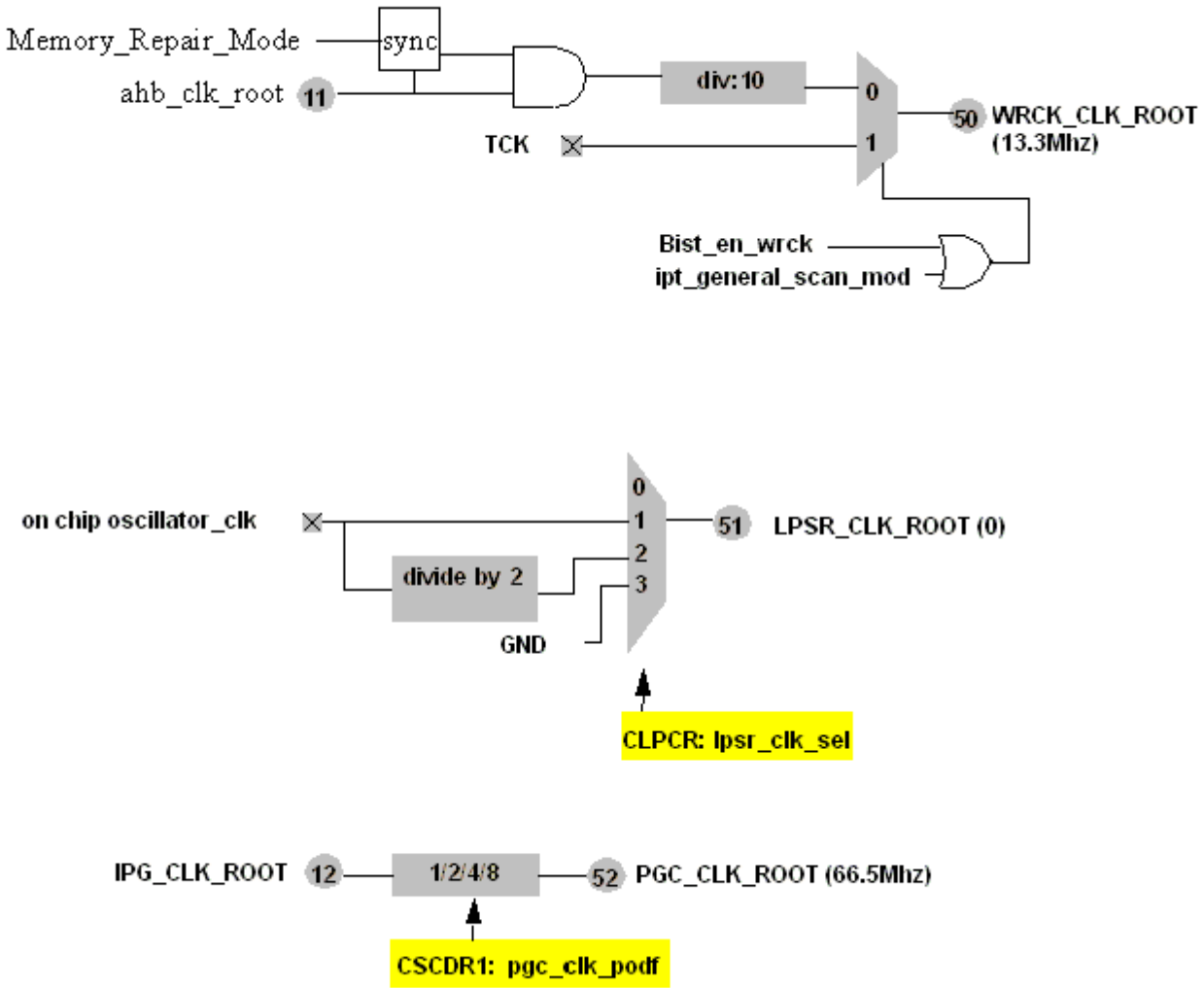

DPLL1 SWITCH (800M)
DPLL2 SWITCH (400M)
DPLL3 SWITCH (216M)
1
2
3
0
1
2
CS1CDR: esai_clk_pred
3 bit divider default=2
CSCMR2: esai_pre_sel
0
1
2
3
4
CS1CDR: esai_clk_podf
6 bit divider default=2
ESAI_CLK_ROOT (27Mhz)
44
SSI1_CLK_ROOT
28
SSI2_CLK_ROOT
30
CKIH
CAMP-1
CKIH2
CAMP-2
CSCMR2: esai_post_sel


IPG_CLK_ROOT
10
CKIH
CAMP-1
CKIH2
CAMP-2
0
1
2
3
4
45
CAN_CLK_ROOT (66.5Mhz)
CSCMR2: can_clk_sel
DPLL1 SWITCH (800M)
DPLL2 SWITCH (400M)
DPLL3 SWITCH (216M)
1
2
3
4
CSCDR3: firi_clk_pred
CSCDR3: firi_clk_podf
3 bit divider default=2
6 bit divider default=2
46
FIRI_CLK_ROOT (54Mhz)
CSCMR2: firi_clk_sel


DPLL1 SWITCH (800M)
DPLL2 SWITCH (400M)
DPLL3 SWITCH (216M)
1
2
3
4
0
1
2
3
CSCDR4: csi_mclk1_clk_pred
CSCDR4: csi_mclk1_clk_podf
3 bit divider default=2
6 bit divider default=2
47
CSI_MCLK1 ROOT
CSCMR1: csi_mclk1_clk_sel


DPLL1 SWITCH (800M)
DPLL2 SWITCH (400M)
DPLL3 SWITCH (216M)
1
2
3
4
0
1
2
3
CSCDR2: ecspi_clk_pred
CSCDR2: ecspi_clk_podf
3 bit divider default=2
6 bit divider default=2
49
ECSPI_CLK_ROOT (54Mhz)
CSCMR1: ecspi_clk_sel


ipp_asrc_ext
DPLL4 SWITCH (595M)
5
0
1
CSCDR2: asrc_clk_pred
CSCDR2: asrc_clk_podf
3 bit divider default=2
6 bit divider default=5
53
ASRC_CLK_ROOT (59.5Mhz)
CSCMR2: asrc_clk_sel



All 6 bit post-dividers in the diagrams above are able to work on frequencies lower than 400MHz. The 6 bit divider should not have an input clock with a frequency higher than this frequency.

The following pre-dividers should use divider values larger than '1'. The option '000' is not allowed for them: USBOH3_CLK_PRED, SSI1_CLK_PRED, SSI2_CLK_PRED, SSI_EXT1_CLK_PRED, SSI_EXT2_CLK_PRED, ECSPI_CLK_PRED.

18.2.1.4.8 Initial Values Controlled by SJC

The initial values of the following dividers and multiplexers can be controlled by SJC. In regular functional mode, the SJC drives the reset values stated in the CCM register memory map. If SJC is programmed to change these values, then the reset value for these

dividers/multiplexers are taken from the SJC programmability. These values can be changed after the reset sequence. The control signals and the dividers/muxes are listed below:

init_perclk_pred1[1:0] to control reset value of PERCLK_PRED1.

init_perclk_pred2[2:0] to control reset value of PERCLK_PRED2.

init_perclk_podf[2:0] to control reset value of PERCLK_PODF.

init_ipg_podf[1:0] to control reset value of IPG_PODF.

init_ahb_podf[2:0] to control reset value of AHB_PODF.

init_axi_a_podf[2:0] to control reset value of AXI_A_PODF.

init_axi_b_podf[2:0] to control reset value of AXI_B_PODF.

init_emi_slow_podf[2:0] to control reset value of EMI_SLOW_PODF.

init_nfc_podf[2:0] to control reset value of NFC_PODF.

init_periph_apm_sel[1:0] to control reset value of PERIPH_APM_SEL.

init_periph_clk_sel to control reset value of PERIPH_CLK_SEL.

init_ssi_apm_clk_sel[1:0] to control reset value of SSI_APM_CLK_SEL.

18.2.1.4.9 Divider Change Handshake

Below list describes the dividers that might involve block handshake on every divider update:

Divider	Handshake with blocks	Comment
emi_slow_podf	EXTMC IPU	Handshake with IPU exists if EMI_SLOW_PODF is chosen as source for ipu_hsp_clk. Handshake with EXTMC is through emi_dvfs_req_slow, emi_dvfs_req_int1,emi_dvfs_req_int2, emi_dvfs_ack_slow, emi_dvfs_ack_int1, emi_dvfs_ack_int2. If emi_slow_podf is chosen as source of gpu_clk,i.e. the change of emi_slow_podf is affect gpu_clk then the handshake of emi_dvfs_req_garb, emi_dvfs_ack_garb should also commence. If this divider is chosen as source of ddr_clk,then the handshake of emi_dvfs_req_fast, emi_dvfs_ack_fast should also commence.
nfc_podf	EXTMC	Handshake with EXTMC is through emi_dvfs_req_slow, emi_dvfs_ack_slow.

Table continues on the next page...

Divider	Handshake with blocks	Comment
axi_a_podf	EXTMC IPU	Handshake with EXTMC exists only if axi_a_podf is chosen as source to ddr_clk. In this case handshake with EXTMC is through emi_dvfs_req_fast, emi_dvfs_ack_fast. If axi_a_podf is chosen as source of gpu_clk,i.e. the change of axi_a_podf is affect gpu_clk then the handshake of emi_dvfs_req_garb, emi_dvfs_ack_garb should also commence. Handshake with IPU exists only if axi_a_podf is chosen as source to ipu_hsp_clk. If this divider is chosen as source of ddr_clk,then the handshake of emi_dvfs_req_fast, emi_dvfs_ack_fast should also commence.
axi_b_podf	EXTMC IPU	Handshake with EXTMC is exist only if axi_b_podf is chosen as source to ddr_clk. In this case handshake with EXTMC is through emi_dvfs_req_fast, emi_dvfs_ack_fast. If axi_b_podf is chosen as source of gpu_clk,i.e. the change of axi_b_podf is affect gpu_clk then the handshake of emi_dvfs_req_garb, emi_dvfs_ack_garb should also commence. Handshake with IPU is exist if axi_b_podf is chosen as source to ipu_hsp_clk. If this divider is chosen as source of ddr_clk,then the handshake of emi_dvfs_req_fast, emi_dvfs_ack_fast should also commence.
ahb_podf	EXTMC IPU	Handshake with EXTMC is exist only if ahb_podf is chosen as source to ddr_clk. In this case handshake with EXTMC is through emi_dvfs_req_fast, emi_dvfs_ack_fast. Handshake with IPU is exist only if ahb_podf is chosen as source to ipu_hsp_clk. If ahb_podf is chosen as source of emi_slow_clk generation, then similar handshake to the emi_slow_podf should take place. If ahb_podf is chosen as source of gpu_clk,i.e. the change of ahb_podf is affect gpu_clk then the handshake of emi_dvfs_req_garb, emi_dvfs_ack_garb should also commence. If this divider is chosen as source of ddr_clk,then the handshake of emi_dvfs_req_fast, emi_dvfs_ack_fast should also commence.

Any update of CCM_CBCDR might involve handshake with blocks based on the above table.

Once the CCM_CBCDR register is updated, CCM checks which of the above dividers were updated. For the updated dividers, CCM is check if a handshake with the respective blocks is needed. If the handshake is needed, CCM is request acknowledge from the respective block for frequency change. Once the acknowledge arrives, CCM is perform the actual frequency change, and is notify the block that the frequency has changed by de

asserting the request. IPU handshake also involves signal to notify it on the frequency change. This signal (periph_dvfs_sw_ack) is held for two clock cycles of ipg_clk on each change.

The handshake with the respective blocks is not be performed in case that the respective handshake is masked through CCDR[bits18-16]. In this case the write to the respective divider is commence immediately after the CBCDR register is updated.

Make sure that the respective blaocks are enabled and have the ablity to acknowledge CCM's request. If the respective block or its clocks are disabled, then it does not have the ability to generate the handshake process, and its handshake capability should be masked (through CCM_CCDR[bits18-16]).

Interrupts can be generated for change in each of the above dividers. See CCM Interrupt Status Register (CCM_CISR) for details on those interrupts.

CCM has divider handshake in process register (CCM_CDHIPR). This register has status bits for each of the above dividers that are involved in the handshake process. When the respective bit is asserted, it means that the divider is being updated, and no attempt to write to this divider should be made, until the bit is de-asserted. Any reads to the divider when it's status bit is asserted, reads the next value to be loaded into the divider, and not the actual value. To make sure that the actual dividers value is read, wait until it's CCM_CDHIPR bit is de-asserted.

For the serial multiplexers, make sure that the block is not using the respective clock before changing multiplexer settings that affect that clock. There is no hardware handshake for changing multiplexer settings.

The multiplexers for AXI buses that are controlled by CCM_CBCDR register, and are not glitchless multiplexers, and hence, should not be used on the fly. That is, make sure that the block is not using the respective clock before changing the multiplexer configuration.

Changing the periph_clk_sel bits in CCM_CBCDR involves handshake with external memory controller (EXTMC) and image processing unit (IPU) - a similar handshake to the one that is performed during load dividers for EXTMC and IPU. These handshakes can be masked by programming CCM_CCDR bits.

Changing the emi_clk_sel bits in CCM_CBCDR involves handshake with EXTMC and IPU - a similar handshake to the one that is performed during load dividers for EXTMC and IPU. The IPU handshake is performed only if IPU_HSP_CLK_ROOT uses EMI_SLOW_CLK_ROOT. These handshakes can be masked by programming CCM_CCDR bits.

NOTE

In case DVFS is enabled (through GPC), frequency changes are not allowed, that is, the frequency of the system should be set prior to DVFS enable, and once the DVFS operation is enabled, no divider from the above handshake group dividers can be changed. This might corrupt the handshake process.

18.2.1.4.10 CKIL Synchronizing to ipg_clk

CKIL is synchronized to ipg_clk when the system is in RUN mode. When system is in STOP mode, i.e. when there is no ipg_clk, the CKIL synchronizer is bypassed, and raw CKIL is supplied to the system.

18.2.1.4.11 Special Considerations for Configuring PERCLK

In addition to ensuring that PERCLK remains at least 2.5 times slower than the AHB clock, certain steps need to be followed to ensure robust operation of PERCLK when reconfiguring the PERCLK clock source.

To properly configure the PERCLK clock source, the following steps are required:

1. In the CCGR registers, gate the clocks to all PERCLK-dependent modules (see below for a list of PERCLK-dependent modules).
2. Select the desired input clock for the PERCLK root clock (to be either source from the peripherals main source clock or the lp_apm clock source). Refer to the CMCBR register, perclk_lp_apm_sel bit.
3. Configure the perclk_pred1, perclk_pred2, and perclk_podf dividers to the desired setting. Refer to the CBCDR register for details.
4. In the CCGR registers, enable the desired clocks for the PERCLK-dependent module clocks.

The following table lists the PERCLK-dependent module clocks and the associated CCGR register. Note that when configuring the PERCLK clock source, these clocks must be gated, however, other unused clock source may also be gated to minimize power consumption.

Table 18-2. PERCLK-dependent Module Clock Sources

PERCLK-dependent Module Clock Sources	Associated CCGR register
uart1_perclk (CG4)	CCGR1
uart2_perclk (CG6)	
uart3_perclk (CG8)	
epit1_highfreq (CG2)	CCGR2

Table continues on the next page...

Table 18-2. PERCLK-dependent Module Clock Sources (continued)

PERCLK-dependent Module Clock Sources	Associated CCGR register
epit2_highfreq (CG4)	
pwm1_highfreq (CG6)	
pwm2_highfreq (CG8)	
gpt_highfreq (CG10)	
esdhc1_perclk (CG1)	CCGR3
esdhc2_perclk (CG3)	
esdhc3_perclk (CG5)	
esdhc4_perclk (CG7)	
ecspi1_perclk (CG10)	CCGR4
ecspi2_perclk (CG12)	
uart4_perclk (CG5)	CCGR7
uart5_perclk (CG7)	

18.2.1.5 DPLL's Disabling / Enabling

DPLL enable and disable is done via the DPLLC block. First move all the clocks generated from a specific DPLL to another DPLL before disabling the DPLL through DPLLC. This move of clocks can be done via glitch less multiplexer, for clocks which are critical to the system, i.e. bus clocks. For serial clocks, first disable the block and the clock generated by it. Then move the multiplexer controlling the source of the clocks to another DPLL, and enable the block and its clocks. Only then it is safe to disable the DPLL. The multiplexer for the serial clocks is not glitchless hence the above procedure should be followed.

18.2.1.6 Low Power Clock Gating Module (LPCG)

The LPCG block receives the root clocks and splits them to clock branches for each block. The clock branches are gated clocks and their enables can come from five sources:

1. Clock enable signal from CCM - this signal is generated by configuration of the CGR bits in CCM. It is based on the low power mode.

2. Clock enable signal from the block - this signal is generated by the block based on its internal logic. Not every enable signal from the block is used. For clock enable signals used from the block, CCM generates override signals based on programable bits in CCM_CMEOR.

3. Clock enable signal from Reset controller (SRC) - this signal enables the clock during the reset procedure. Please refer to SRC spec for details on the clock enable signal during reset procedure.

4. Enable or disable clock in bist mode: This is based on bist_en signal. This is applicable for clocks that are not functional and needed only on memory repair or bist sequence (like sms_clk's and wrck).

The above possible enable signals are ANDed to generate the enable signal for the gating cell.

The enable signal for the gating cell is synchronized with the clock it needs to gate. This is done in order to prevent glitches on the gated clock.

Notifications are generated for the CCM, to indicate when to close and to open the clock roots. All notifications that correspond to the same clock root are ORed to generate one notification signal to CCM for clock root gating.

Figure 18-6 describes the implementation for each gating cell:

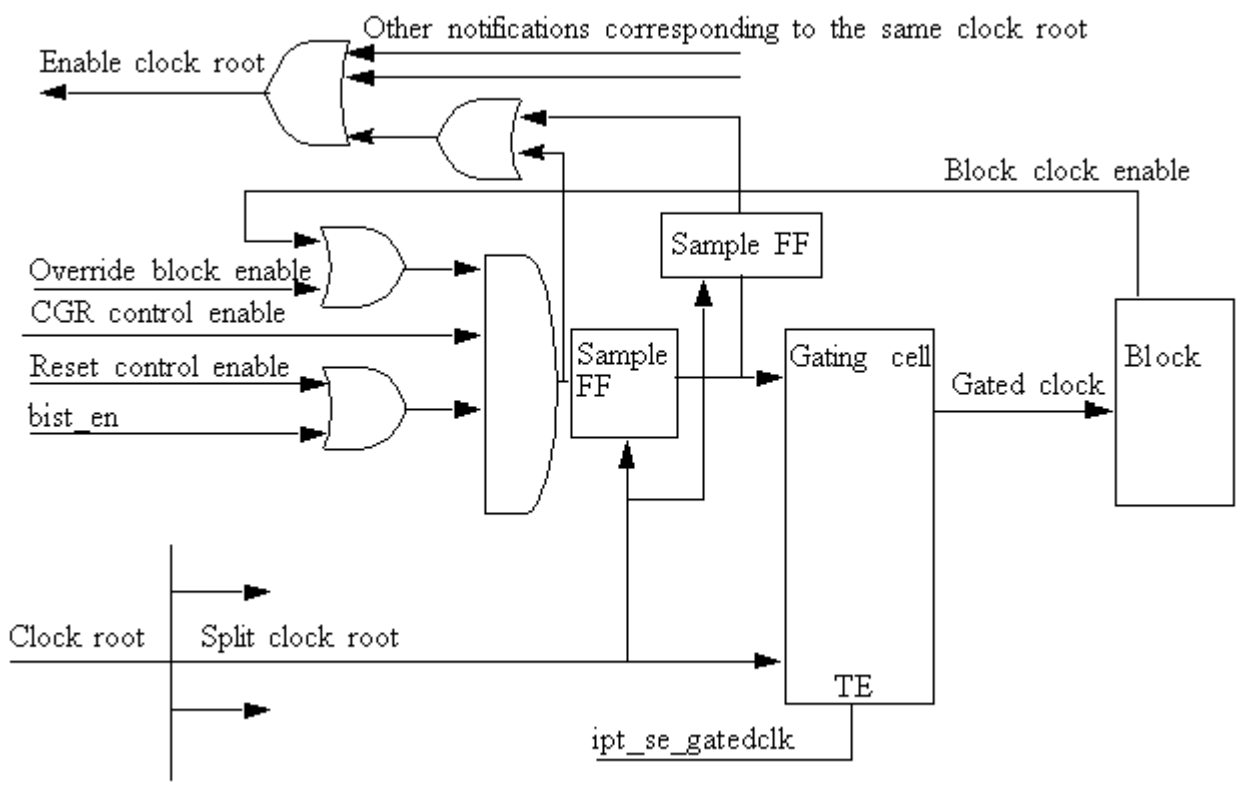


Figure 18-6 describes the clock split inside LPCG. It describes the case of 2 blocks, one block without enable signal and one with enable signal. (SRC enable signals and sync FF's are omitted from this figure).

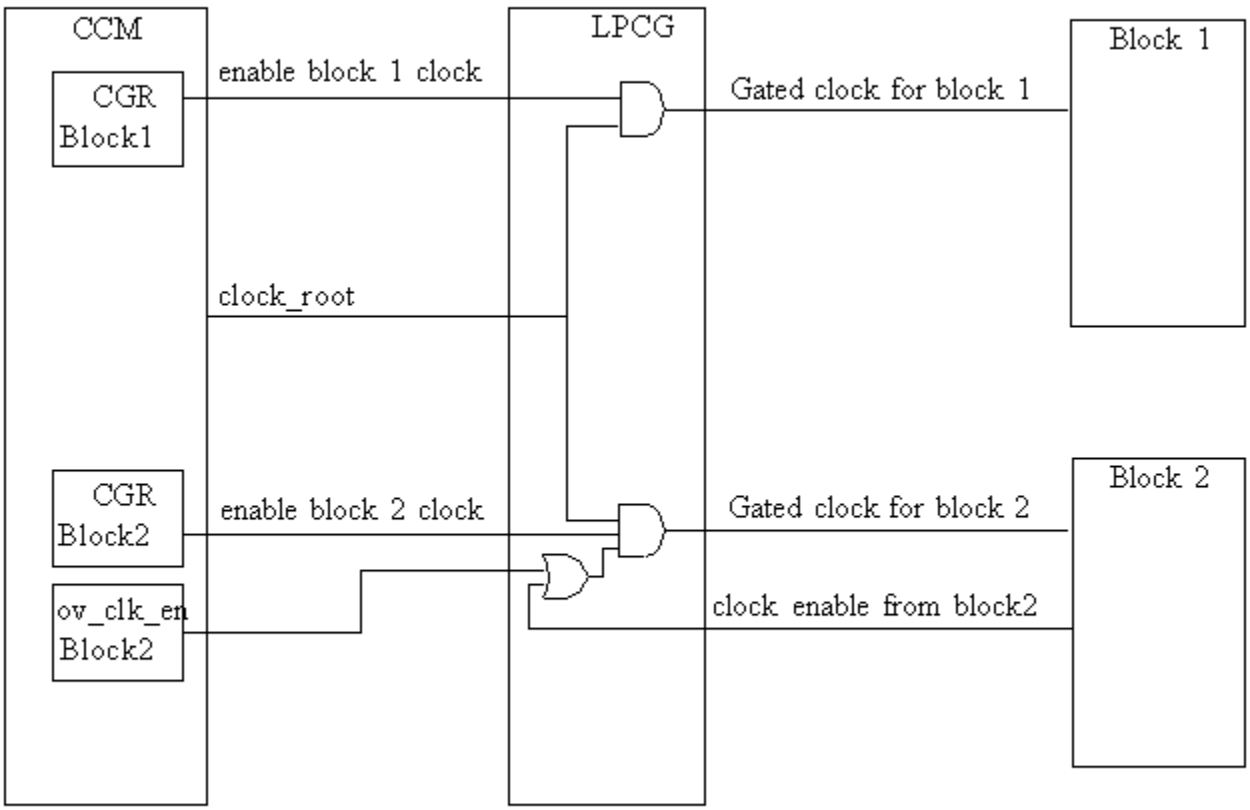


Figure 18-6. Clock split in LPCG

18.2.2 Creation of Sync Signal for IEEE_RTC Module

The IPTP block uses a ipg sync signal to be able to connect to the IP bus. CCM generates ipg_clk_sync_ieee_root signal to be used by LPCG to drive the ipg sync input of IPTP. LPCG gates the ipg sync signal with a same control as the ipg clock input to IPTP.

18.2.3 System clocks connectivity

The following tables show the CCM output clocks system level connectivity per block in i.MX53. The gating option in the table can be either CGR bit or clock enable from the block itself.

Table 18-3. Output clocks from CCM

Instance / Sub-Block	Clock	Root	Override by CCM_CMEOR bits	Gating by CCM_CCGR bits
ARM_Platform	ARM main clock (arm_clk)	arm_clk_root		Yes
	IPG clock (ipg_clk_lp)	ipg_clk_root		Yes
	ACLK (aclk_lp)	arm_axi_clk_root		Yes
	Debug PCLK (dbg_pclk_lp)	debug_apb_clk_root		Yes
	dbg_atclk_lp	debug_apb_clk_root		Yes
DAP	HCLK	ahb_clk_root	Yes	Yes
	HCLKEN	tie to '1'		
	PCLKDBG	debug_apb_clk_root	Yes	Yes
	PCLKENDBG	tie to '1'		
	PCLKSYS	ahb_clk_root	Yes	Yes
	PCLKENSYS	tie to '1'		
	DAPCLKao	debug_apb_clk_root		Yes
AIPSTZ-1	Main clock	ahb_clk_root		Yes
AIPSTZ-2	Main clock	ahb_clk_root		Yes
ASRC	Functional clock	ahb_clk_root		Yes
CSPI-3	Block interface clock	ipg_clk_root		Yes
	Low frequency clock	ckil_sync_clk_root		
eCSPI-1	Block interface clock	ipg_clk_root		Yes
	Low frequency clock	ckil_sync_clk_root		
	Reference clock	ecspi_clk_root		Yes
eCSPI-2	Block interface clock	ipg_clk_root		Yes
	Low frequency clock	ckil_sync_clk_root		
	Reference clock	ecspi_clk_root		Yes
EPIT-1	Peripheral clock (Block interface)	ipg_clk_root	Yes	Yes
	Low frequency clock	ckil_sync_clk_root		
	High frequency clock	perclk_root		Yes
EPIT-2	Peripheral clock (Block interface)	ipg_clk_root	Yes	Yes
	Low frequency clock	ckil_sync_clk_root		
	High frequency clock	perclk_root		Yes

Table continues on the next page...

Table 18-3. Output clocks from CCM (continued)

Instance / Sub-Block	Clock	Root	Override by CCM_CMEOR bits	Gating by CCM_CCGR bits
ESAI-1	ESAI logic clock	ahb_clk_root		Yes
	Block interface (registers accesses) clock	ipg_clk_root		Yes
	EXTAL clock	esai_clk_root		Yes
ESDHCv2-1	Block interface (register accesses) clock (ipg_clk)	ipg_clk_root	Yes	Yes
	AHB (DMA accesses) clock (hclk)	ahb_clk_root	Yes	Yes
	Card (SD/MMC) clock (ipg_clk_perclk)	esdhc1_clk_root	Yes	Yes
ESDHCv2-2	Block interface (register accesses) clock (ipg_clk)	ipg_clk_root	Yes	Yes
	AHB (DMA accesses) clock (hclk)	ahb_clk_root	Yes	Yes
	Card (SD/MMC) clock (ipg_clk_perclk)	esdhc2_clk_root	Yes	Yes
ESDHCv3-3	Block interface (register accesses) clock (ipg_clk)	ipg_clk_root	Yes	Yes
	AHB (DMA accesses) clock (hclk)	ahb_clk_root	Yes	Yes
	Card (SD/MMC) clock (ipg_clk_perclk)	esdhc3_clk_root	Yes	Yes
ESDHCv2-4	Block interface (register accesses) clock (ipg_clk)	ipg_clk_root	Yes	Yes
	AHB (DMA accesses) clock (hclk)	ahb_clk_root	Yes	Yes
	Card (SD/MMC) clock (ipg_clk_perclk)	esdhc4_clk_root	Yes	Yes

Table continues on the next page...

Table 18-3. Output clocks from CCM (continued)

Instance / Sub-Block	Clock	Root	Override by CCM_CMEOR bits	Gating by CCM_CCGR bits
EXTMC	AXI interface master 0 (IPU) clock	ipu_hsp_clk_root		Yes
	AXI interface master 1 (VPU) clock	vpu_axi_clk_root		Yes
	AXI interface master 5 (GPU3D) clock	gpu_clk_root		Yes
	AXI interface master 6 (GPU2D) clock	gpu2d_clk_root		Yes
	AXI interface master 7 (USB) clock	ahb_clk_root		Yes
	AXI interface master 4 (PL301_4x1) clock	ahb_clk_root		Yes
	AXI interface master 3 (ARM) clock	arm_axi_clk_root		Yes
	AXI interface master 2 (AHBMAX) clock	ahb_clk_root		Yes
	Fast channel clock	ddr_clk_root		Yes
	aclk_fast_ddr	ddr_clk_root		Yes
	aclk_fast_phy	ddr_clk_root		Yes
	NAND Flash clock	enfc_clk_root		Yes
	Slow channel clock	emi_slow_clk_root		Yes
	Internal 1 channel clock	ahb_clk_root		Yes
	Internal 2 channel clock	ahb_clk_root		Yes
FEC	System clock	ipg_clk_root		Yes
	AHB DMA clock	ahb_clk_root		Yes
FIRI	Block interface clock (ipg_clk)	ipg_clk_root		Yes
	FIRI baud clock (ipg_clk_firi_baud)	firi_clk_root		Yes
FLEXCAN-1	Bus clock	ipg_clk_root		Yes
	MBM clock	ipg_clk_root	Yes	Yes
	CPI clock	can_clk_root		Yes
FLEXCAN-2	Bus clock	ipg_clk_root		Yes
	MBM clock	ipg_clk_root	Yes	Yes
	CPI clock	can_clk_root		Yes

Table continues on the next page...

Table 18-3. Output clocks from CCM (continued)

Instance / Sub-Block	Clock	Root	Override by CCM_CMEOR bits	Gating by CCM_CCGR bits
GPT	Block interface cklock	ipg_clk_root	Yes	Yes
	Low reference clck	ckil_sync_clk_root		
	HIgh reference clock	perclk_root		Yes
GPU2D (OpenVG)	Main/Functional clock	gpu2d_clk_root	Yes	Yes
GPU3D	GMEM clock	gpu_clk_root		Yes
	GPU3D core clock	gpu_clk_root	Yes	Yes
I2C-1	Peripheral clock (IP Bus)	ipg_clk_root		Yes
	Block clock	perclk_root		Yes
I2C-2	Peripheral clock (IP Bus)	ipg_clk_root		Yes
	Block clock	perclk_root		Yes
I2C-3	Peripheral clock (IP Bus)	ipg_clk_root		Yes
	Block clock	perclk_root		Yes
IPTP / PTP	ipg_clk	ipg_clk_root		Yes
IPTP / RTC	ipg_ce_clk	ahb_clk_root		Yes
	ipg_clk	ipg_clk_root		Yes
	ipg_sync	ipg_clk_sync_ieee_root		Yes
	RTC source clock	ieee_cemx_clk_root		Yes
IPU	IPU main clock	ipu_hsp_clk_root		Yes
	Display interface 1 clock	di1_clk_root		Yes
	Display interface 0 clock	di0_clk_root		Yes
KPP	Block interface clock	ipg_clk_root		Yes
	Low frequency clock	ckil_sync_clk_root		
LDB	Channel 0 interface serializer clock	ldb_di0_serial_clk_root		Yes
	Display 0 interface pixel clock	di0_clk_root		Yes
	Channel 1 interface serializer clock	ldb_di1_serial_clk_root		Yes
	Display 1 interface pixel clock	di1_clk_root		Yes
MLB	Main clock	ahb_clk_root		Yes
	IPG clock	ipg_clk_root		Yes
	MLB memory clock	ahb_clk_root		Yes

Table continues on the next page...

Table 18-3. Output clocks from CCM (continued)

Instance / Sub-Block	Clock	Root	Override by CCM_CMEOR bits	Gating by CCM_CCGR bits
OCRAM	Main/Functional clock	ahb_clk_root		Yes
OWIRE	Block interface clock	ipg_clk_root	Yes	Yes
	Main clock	perclk_root		Yes
PATA	Block interface clock	ipg_clk_root		Yes
	AHB (DMA) clock	ahb_clk_root		Yes
PLARB-2	Main clock	ahb_clk_root		Yes
	Slave 0 clock (GPU3D)	gpu_clk_root		Yes
	Master 0 clock (VPU)	vpu_axi_clk_root		Yes
	Register access PCLK clock	ahb_clk_root		Yes
PWM-1	Block interface clock (ipg_clk)	ipg_clk_root		Yes
	Low frequency clock (ipg_clk_32k)	ckil_sync_clk_root		
	ipg_clk_highfreq	perclk_root		Yes
PWM-2	Block interface clock (ipg_clk)	ipg_clk_root		Yes
	Low frequency clock (ipg_clk_32k)	ckil_sync_clk_root		
	ipg_clk_highfreq	perclk_root		Yes
ROM64K	Functional/Main clock	ahb_clk_root		Yes
RTIC	AHB clock	ahb_clk_root		Yes
	Block interface clock	ipg_clk_root		Yes
	Low frequency clock	ckil_sync_clk_root		
SAHARA	Block interface clock	ipg_clk_root		Yes
	AHB (DMA) clock	ahb_clk_root	Yes	Yes
	Low frequency clock	ckil_sync_clk_root		
SATA	SATA DMA clock	ahb_clk_root		Yes
SCC	AHB (DMA) clock	ahb_clk_root		Yes
	Block interface clock	ipg_clk_root		Yes
SDMA	SDMA AP DMA (AHB) clock	ahb_clk_root		Yes
	SDMA core clock	ipg_clk_root		Yes
SPDIF	Tx clock	spdif0_clk_root		Yes
	Global/System clock	ipg_clk_root		Yes
	Block interface clock	ipg_clk_root		Yes

Table continues on the next page...

Table 18-3. Output clocks from CCM (continued)

Instance / Sub-Block	Clock	Root	Override by CCM_CMEOR bits	Gating by CCM_CCGR bits
SRC	Block interface clock	ipg_clk_root		Yes
	Low frequency clock	ckil_sync_clk_root		
SRTC	ipg_clk	ipg_clk_root		Yes
	ipg_clk_s	ipg_clk_root		Yes
	CKIL clock (32KHz)	ckil_sync_clk_root		
SSI-1	Block interface clock	ipg_clk_root		Yes
	Bit clock (ccm_ssi_clk)	ssi1_clk_root		Yes
SSI-2	Block interface clock	ipg_clk_root		Yes
	Bit clock (ccm_ssi_clk)	ssi2_clk_root		Yes
SSI-3	Block interface clock	ipg_clk_root		Yes
	Bit clock (ccm_ssi_clk)	ssi3_clk_root		Yes
TVE	TVE clock	tve_216_54_clk_root		Yes
	Block interface clock	ipg_clk_root		Yes
TZIC	Functional/Main clock	ahb_clk_root		Yes
UART-1	Block clock (ipg_clk and ipg_clk_s)	ipg_clk_root		Yes
	Peripheral clock (ipg_perclk)	uart_clk_root		Yes
UART-2	Block clock (ipg_clk and ipg_clk_s)	ipg_clk_root		Yes
	Peripheral clock (ipg_perclk)	uart_clk_root		Yes
UART-3	Block clock (ipg_clk and ipg_clk_s)	ipg_clk_root		Yes
	Peripheral clock (ipg_perclk)	uart_clk_root		Yes
UART-4	Block clock (ipg_clk and ipg_clk_s)	ipg_clk_root		Yes
	Peripheral clock (ipg_perclk)	uart_clk_root		Yes
UART-5	Block clock (ipg_clk and ipg_clk_s)	ipg_clk_root		Yes
	Peripheral clock (ipg_perclk)	uart_clk_root		Yes

Table continues on the next page...

Table 18-3. Output clocks from CCM (continued)

Instance / Sub-Block	Clock	Root	Override by CCM_CMEOR bits	Gating by CCM_CCGR bits
USB	USB interface clock	ipg_clk_root		Yes
	USB internal PL301 clock	ipg_clk_root		Yes
	AHB (DMA) clock	ahb_clk_root		Yes
	60MHz clock	usboh3_clk_root		Yes
	Low frequency clcok	ckil_sync_clk_root		
VPU	VPU AXI clock (aclk)	vpu_axi_clk_root	Yes	Yes
	VPU core clock (cclk)	vpu_axi_clk_root	Yes	Yes
	VPU reference clock (rclk)	vpu_rclk_root	Yes	Yes
	Block interface clock (ipg_clk_s)	ipg_clk_root		Yes
WDOG-1	Block interface (registers accesses) clock (ipg_clk_s)	ipg_clk_root		Yes
	32KHz (CKIL) clock (ipg_clk_32k)	ckil_sync_clk_root		
WDOG-2	Block interface (registers accesses) clock (ipg_clk_s)	ipg_clk_root		Yes
	32KHz (CKIL) clock (ipg_clk_32k)	ckil_sync_clk_root		

18.2.4

Clock roots that affect processing unit (VPU, GPU2D, GPU3D) include BRM capability on their clock root generation. The BRM allows finer granularity of the effective frequency generated on each one of those clock roots, as compared to using a divider. BRM changes the effective frequency by eliminating clock pulses, and hence reducing the average effective frequency over time. The value programed in the control register of the BRM affects the clock cycle that is eliminated. For instance, if the input clock to the BRM is 200Mhz, then the BRM output has the following options:

Table 18-4. BRM Effective Clock Output in case the Input Frequency is 200Mhz

BRM parameter setting	BRM Output frequency (Mhz)
0000	200 (BRM in bypass mode - all clock cycles are passed to the output)

Table continues on the next page...

Table 18-4. BRM Effective Clock Output in case the Input Frequency is 200Mhz (continued)

BRM parameter setting	BRM Output frequency (Mhz)
0001	100
0010	150
0011	166.6
0100	175
0101	180
0110	183.3
0111	185.7
1000	187.5
1001	188.9
1010	190
1011	190.9
1100	191.6
1101	192.3
1110	192.8
1111	193.3

18.2.5 DVFS Support

Frequency shift during DVFS procedure in i.MX53 can be done on two domains:

1. ARM clock domain frequency shift.
2. Peripherals clock domain frequency shift (including buses).

GPC(General Power Controller) initiates the frequency shift procedure. It initiates only one frequency shift at a time. There is no option to commence both frequency shifts simultaneously. GPC is aided by SDMA and CCM to execute the frequency shift.

The following paragraphs explain in detail the frequency shift options.

18.2.5.1 ARM Clock Domain Frequency Shift:

SDMA can request ARM frequency shift by requesting change in ARM_PODF divider, or by DPLL-1 relock.

The following steps are configured by SDMA:

1. SDMA can request ARM frequency shift by dividers, or by DPLL-1 relock. This request is written to CCM_CDCR[ARM_FREQ_SHIFT_DIVIDER] bit. If it is 0, DPLL-1 relock shift is requested.

   If it is 1, ARM_PODF divider shift is requested. In this case any new writes to ARM_PODF is held until change ARM frequency signal is asserted by GPC. CCM holds a status bit to notify that the ARM_PODF divider is being updated. Need to wait until the write finishes before writing any new value to ARM_PODF. See CCM Divider Handshake In-Process Register (CCM_CDHIPR). CCM can also generate an interrupt once the actual ARM_PODF has been written during DVFSC operation. See CCM Interrupt Status Register (CCM_CISR).

2. Configure new ARM domain dividers (if ARM_FREQ_SHIFT_DIVIDER = 1: ARM_PODF divider shift is requested)

   SDMA continues with next steps only if (ARM_FREQ_SHIFT_DIVIDER = 0: DPLL-1 relock shift is requested).

3. Check which frequency mode is active in DPLLC-1, HFS or LFS: DPLLC-1 bit HFSM of DPLLC_CTL register.
4. Configure the non active frequency mode (HFS or LFS) of DPLLC-1, with the new frequency needed for DPLL-1 to relock on: DPLLC-1 registers DPLLC_OP, DPLLC_MFD, DPLLC_MFN, DPLLC_HFS_OP, DPLLC_HFS_MFD, DPLLC_HFS_MFN.
5. Configure the step frequency divider: CCM_CCSR register, bit pll2_div_podf or pll3_div_podf (depending on the decision in the step mux in step 6).
6. Configure the step frequency mux: CCM bit step_sel of CCM_CNT register.

   Once those configurations are done, sdma is notify GPC to continue with the frequency shift procedure.

   The actual shift is start by assertion of arm_clk_switch_req signal from GPC to CCM.

   CCM is commence the following steps upon assertion of arm_clk_switch_req signal:

If arm_freq_shift_divider = 1 dividers shift is requested):

1. Load the ARM domain new dividers.
2. After reloading the dividers (all counters in 0), indicate to GPC that frequency has been updated.

If arm_freq_shift_divider = 0 (DPLL1 relock shift is requested):

1. Switch to the step frequency by a glitch less multiplexer.

2. After the glitchless multiplexer has changed to the step frequency, invert LVS select signal to DPLLC-1 so that DPLL-1 is locked to the new frequency.
3. On pll_lrf_sticky1 assertion move to the new reloaded frequency by the glitch less mux.
4. After actual shift of the mux assert clk_switch_ack for 2 ipg cycle. This is done so GPC is have indication that frequency has been updated.

Upon clk_switch_ack assertion, GPC is de assert arm_clk_switch_req signals.

On this point, ARM has changed frequency and system is ready for a new frequency switch procedure.

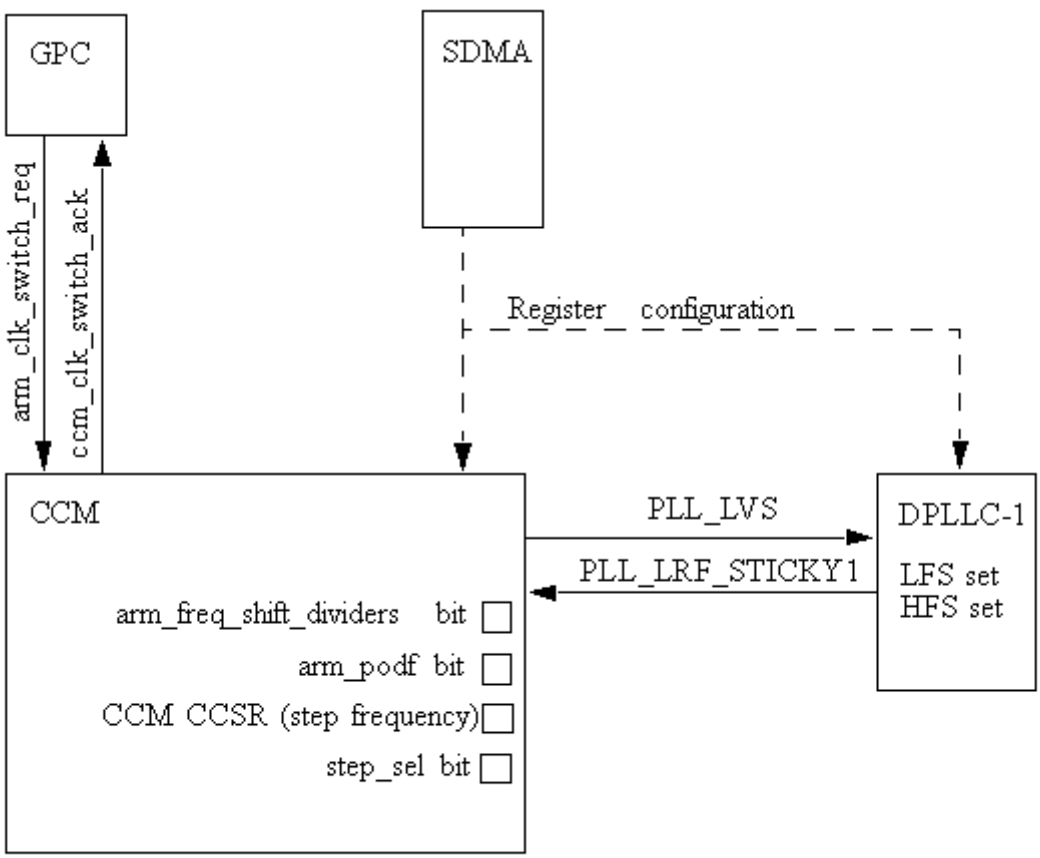


Figure 18-7. ARM dvfs connectivity

18.2.5.2 Peripheral Clock Domain Frequency Shift:

CCM includes capability to divide by 2/3/4 the root of bus clocks (ahb, axi, ipg, int_mem, ddr_clk, nfc_clk).

The actual division factor is defined by the bits in CCM_CDCR[1:0]. Set these bits prior to enabling DVFSP operation. The value written to the DVFSP register divider (CCM_CDCR[1:0]) is not loaded immediately to the actual divider. It waits until DVFSP procedure commences.

During DVFSP operation, the root clocks which are not affected by DVFS divider should be generated from a low frequency DPLL. This frequency should not be higher than 240Mhz. Can use DPLL-3 with frequency 216Mhz for this operation. The setting for the serial clocks is done through CCM_CSCMR1 register.

The blocks whose clock is about to change (IPU,EXTMC, etc.) should be configured with values that determine the frequency of the new clock, prior to the clock switch operation. Please refer to the respective blocks specification for clarification on the needed configuration.

Divide by 2/3/4 procedure (lowering frequency): Upon assertion of Peripherals clock division request signal from GPC, the actual division procedure takes place. The periph_clk_div_req signal from GPC remains asserted as long as GPC requests a division of the peripheral clocks. This signal is de-asserted only when GPC requests frequency multiplication and return to division by '1'.

NOTE

DVFS operation can be started by asserting SW_PERIPH_CLK_DIV_REQ bit in CCM_CDCR. If the bit is asserted, GPC control is ignored and the the acknowledge to GPC is not asserted.

There is no need to synchronize the change in the 3 dividers, because they work on asynchronous clock domains.

Prior to the actual division there is need for acknowledge from peripherals that are sensitive to the frequency change (that is, IPU and EXTMC). The handshakes for IPU and EXTMC are described as follows:

- CCM asserts change frequency request signal to request approval from IPU and EXTMC in order to commence the frequency shift.
- IPU and EXTMC asserts their respective acknowlege signals to notify the time window that its alright to change the frequency.

CCM commences the actual division once both the acknowledge signals (from IPU and EXTMC) are asserted.

After completing the division, CCM notifies GPC by assertion of clock switch acknowledge for 2 ipg cycles (rising on ipg_clk) and notifies IPU by assertion of IPU clock changed signal for at least 2 hsp clock cycles (rising on the edge of the new hsp clock). CCM also negates the frequency change request signals to IPU and EXTMC

Multiply by 2/3/4 procedure (increasing frequency): Upon negation of peripherals clock division request signal from GPC, the actual multiplication procedure takes place by removing the division by 2/3/4 and returning to division by '1'.

NOTE

DVFS operation can be stopped de-asserting SW_PERIPH_CLK_DIV_REQ bit in CCM_CDCR. The GPC control is ignored and the CCM clock switch acknowledge is not asserted.

There is no need to synchronize the 3 dividers change since they work on async clock domains.

Prior to the actual frequency change, CCM requires acknowledge from peripherals that are sensitive to the frequency change. These handshakes are described as follows:

- CCM asserts change frequency request signal to request approval from IPU and EXTMC in order to commence the frequency shift.
- IPU and EXTMC assert their respective acknowlege signals to notify the time window that its alright to change the frequency.

CCM commences the actual division once both the acknowledge signals (from IPU and EXTMC) are asserted.

After completing the frequency change, CCM notifies GPC by assertion of clock switch acknowledge for 2 ipg cycles and notifies IPU by assertion of IPU clock changed for at least 2 hsp clock cycles. CCM also negates the frequency change request signals to IPU and EXTMC.

Following figure describes the connectivity.

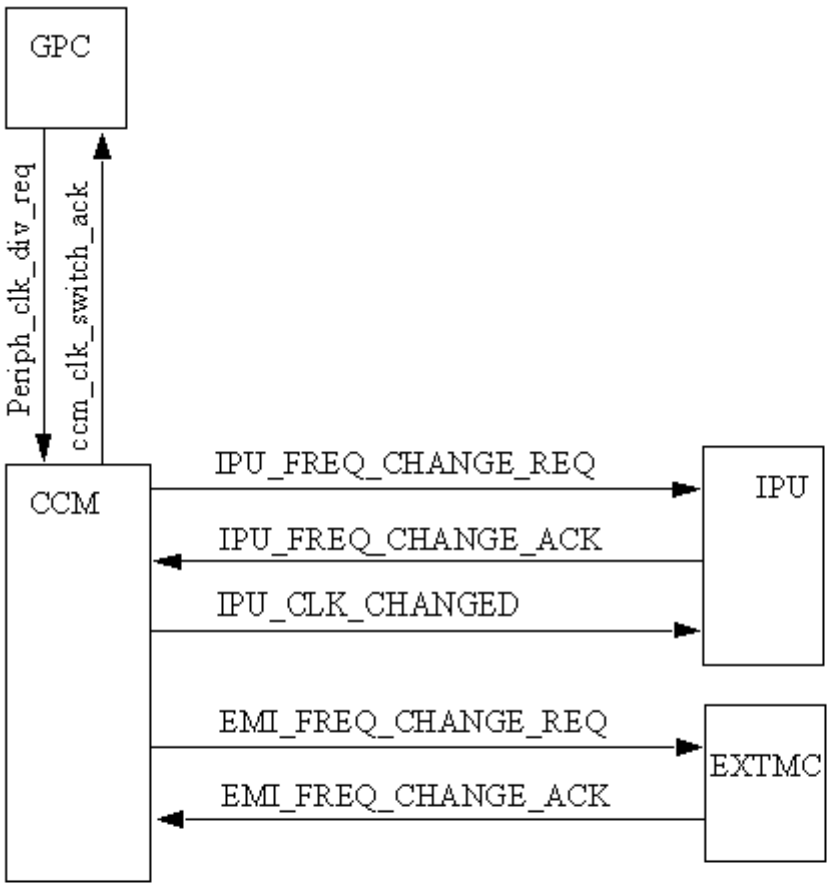


Figure 18-8. DVFSP connectivity

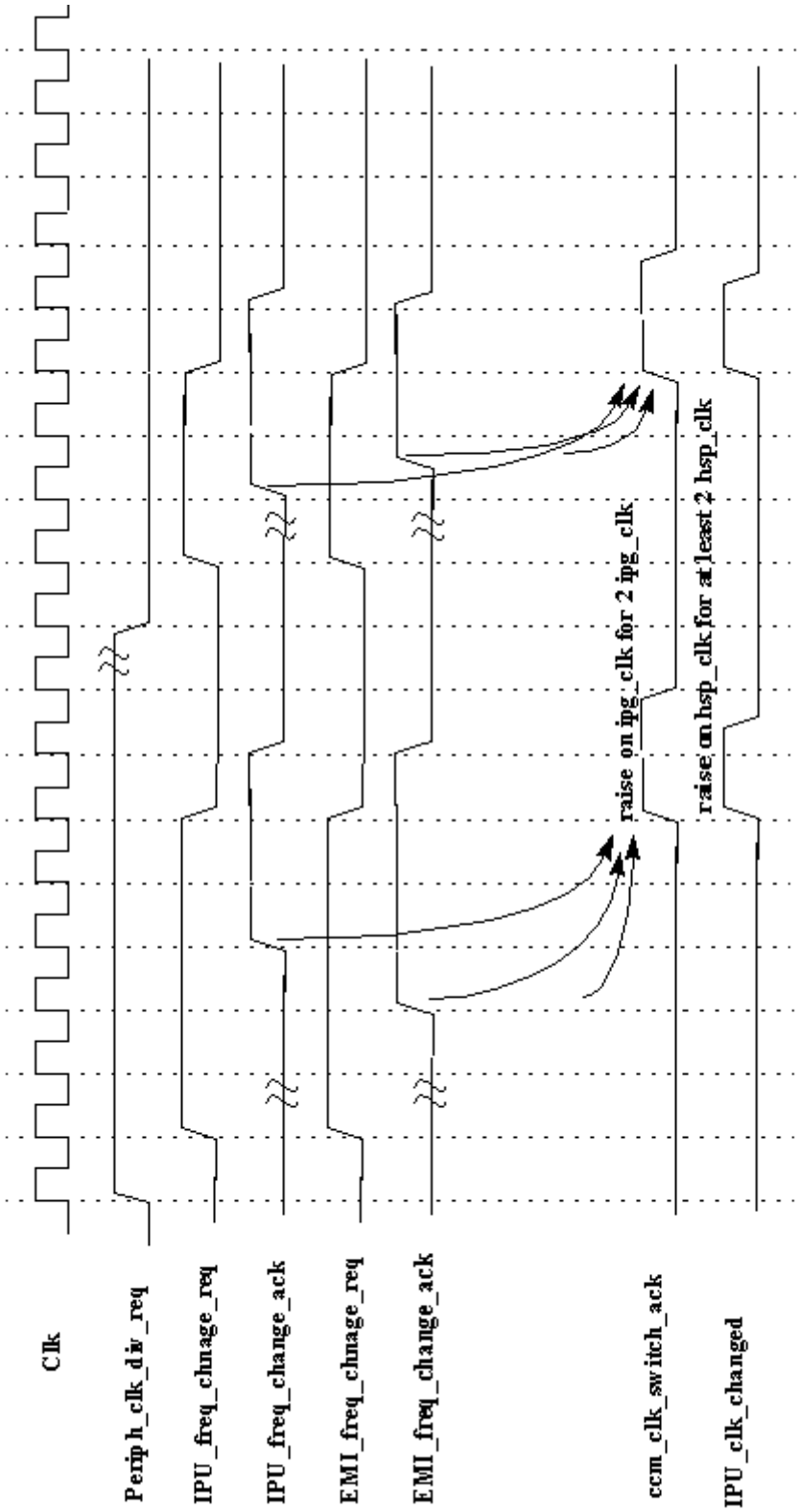


Figure 18-9. DVFSP signals

NOTE

There is an option to generate the EXTMC request only after the acknowledge signals form the EXTMC master (IPU) is received in CCM. This option allows the master to finish it's tasks that require EXTMC performance, and does not interrupt EXTMC performance before the master has acknowledged the frequency change. See CCM DVFS Control Register (CCM_CDCR) for details.

18.2.5.3 Peripherals Restrictions in DVFS Scenario

In DVFSP scenario, ipg and ahb clocks are divided from their nominal frequency.

The serial dividers in the low voltage case, should use a low frequency DPLL input (below 300Mhz) to work correctly.

Listed below are restrictions that need to be followed to support the DVFS frequency shift:

Table 18-5. DVFS restrictions

Block	Support	Restriction
R - blocks that is work with restriction		
UART	R	Need to configure uart rate to be low_frequency_ipg_clk/16. Hence rates of 1.875M and 4M cannot be reached in dvfs scenario.
ESDHC	R	Need to configure card rate to be low_frequency_ahb_clk/2. Hence rated of 52M for SD card cannot be reach in dvfs scenario.
MSHC	R	Need to configure card rate to be low_ipg_clk. Hence rated of 40M for card cannot be reach in dvfs scenario.
SSI	R	Master clock should be generated from ssi_ext1 and ssi_ext2 pads.
CSPI	R	Need to configure spi rate to be low_ipg_clk/4.
DDR memory	R	Under the low frequency limits of DDR memory (125Mhz for DDR2, and 300Mhz for DDR3). Without DLL, DDR3 is able to operate under 125Mhz.
N - blocks that is not work		

Table continues on the next page...

Table 18-5. DVFS restrictions (continued)

Block	Support	Restriction
USB, USB_PHY	N	Needs ipg_clk>60Mhz. Hence is not work under dvfs scenario
FEC	N	Needs ipg_clk>50Mhz. Hence is not work under dvfs scenario
TVE	N	Is not synthesized to twice the frequency
LDB	N	Is not synthesized to twice the frequency
SPDIF	N	Is not synthesized to twice the frequency
SATA	N	Is not synthesized to twice the frequency
P-ATA	N	Needs ipg_clk to be fixed. In case ARM core manages dvfs scenario, it can work only in times ipg_clk is fixed, i.e. before and after dvfs scenario.

18.2.5.4 CCM Handshake with the ESDCTL

When clock frequencies are changed to the DDR memory (either through DVFS or manually), the CCM will issue a handshake with the ESDCTL controller. In turn the ESDCTL will place the external DDR memory in self refresh during the frequency change.

In the case of a manual frequency change, the CCM will issue a handshake to the ESDCTL when changing the main_bus_clk clock MUX (via the PERIPH_CLK_SEL bits[24:22] in the CCM_CBCDR). The CCM will also issue a handshake when changing the post dividers (PODF) that source the DDR clock: AXI_A_PODF, AXI_B_PODF, EMI_SLOW_CLK_ROOT, and AHB_CLK_ROOT.

18.2.6 Power Modes

i.MX53 supports 3 low power modes: RUN mode, WAIT mode, and STOP mode.

18.2.6.1 Run Mode

This is the normal/functional operating mode. In this mode ARM runs in its normal operational mode. The frequency and voltage can be changed by DVFS operation as described in DVFS Support. Clocks to the blocks can be gated by configuring the corresponding CCM_CCGR*n* register bits.

18.2.6.2 Wait Mode

In this mode the ARM clock is gated. All other clocks are functional and can be gated by programing their CCGR bits. Power gating can be done for ARM platform.

18.2.6.2.1 Wait Mode is Entered by the Following Procedure:

ARM asserts the wait mode by writing to CCM_CLPCR[LPM] bits. It then writes to TZIC_DSMINT[0] (asserting DSM interrupt hold off).

DSM interrupt hold off assertion forces the interrupt controller to stop the synchronizer clock for incoming interrupts. Any interrupts that had previously been synchronized by the interrupt controller remain asserted.

If a wakeup event occurrs (an interrupt is serviced) or is pending, the ARM platform aborts the DSM sequence by exiting the main shutdown code sequence and returning to the desired security mode.

Once the L2 cache is inactive, the ARM platform asserts the DSM request signal to the CCM.

If the DSM wakeup signal from TZIC is negated and the DSM request signal from ARM is asserted, the CCM begins the WAIT sequence.

The ARM platform clock is stopped only if CCM_CLPCR[5] = 1 and SJC_GPUCR3[15] = 0. If either value is different, then ARM clocks are not gated off and the CCM continues to next step.

CCM shuts down the clocks to SAHARA,RTIC, IPU, SDMA, SCC, EXTMC and AHBMAX only if their corresponding bits in CCM_CCGR indicate that they should be closed in WAIT mode and if their clocks were functional in RUN mode. The request is issued if the handshake is not bypassed by programing the CLPCR[23:16] register bits. If the corresponding bits are set, then the request signal is not issued to the corresponding block and CCM does not wait for its acknowledge in the process of entering low power mode.

NOTE

Since EXTMC requires the DDR clock to be active for it to answer the handshake request, make sure to enable the clock by setting the appropriate CCM_CCGR bits.

NOTE

The SCC clocks are stopped, only if SAHARA clocks were programmed to be stopped in WAIT made. Hence, the SCC handshake is performed only if the SAHARA clocks were programmed to be closed as well.

Once CCM is in WAIT mode, it is check if TZIC has asserted the DSM wakeup signal, or if ARM DSM request signal has negated during the process of entering wait mode. If it has occurred, CCM exits the WAIT mode.

If neither of the signals have changed, CCM generates a request to GPC to power down the ARM platform. See Figure 18-10 diagram. The GPC sends a power down acknowledge signal at the end of the power down sequence.

NOTE

During WAIT mode, if SCC asserts either of its clock enable signals, CCM enables the SCC clocks. The WAIT mode is not exited due to this assertion.

18.2.6.2.2 Wait mode is Exited by the Following Procedure

As soon as the synchronized TZIC wakeup signal is seen asserted or the ARM DSM request signal is negated, CCM begins the process of exiting WAIT mode.

CCM request GPC to restore power to the ARM platform and then exits from WAIT mode by first resetting the blocks to which the clocks have been stopped. Then it enables all the block clocks (depending on the CCM_CCGR bits).

Once the interrupt propagates through all the synchronization logic, the ARM core recognizes and services it. This forces the negation of the ARM DSM request signal. Prior to this point, ARM DSM request signal and TZIC DSM wakeup signals were simultaneously asserted. Since TZIC DSM wakeup signal has top priority, the system is able to wake up.

Before the ARM exits the ISR, it clears the interrupt source. This is causes the TZIC DSM wakeup signal to be de-asserted. At this point, the two low power control signals (TZIC_DSM_WAKEUP and ARM_DSM_REQUEST) are negated, and the WAIT sequence can be repeated at any time.

18.2.6.3 Stop Mode

In this mode all system clocks and DPLL's are stopped. Power gating can be done on the ARM platform.

18.2.6.3.1 Stop Mode is Entered by the Following Procedure:

ARM asserts the stop mode by writing to CCM_CLPCR[LPM] bits and the TZIC_DSMINT[0] (DSM interrupt hold off) bit.

DSM interrupt hold off assertion forces the interrupt controller to stop the synchronizer clock for incoming interrupts. Any interrupts that had previously been synchronized by the interrupt controller remain asserted.

If a wakeup event occurrs or an interrupt is serviced or is pending, the ARM platform aborts the STOP mode sequence by exiting the main shutdown code sequence and returning to the desired security mode.

Once the L2 cache controller becomes inactive, the ARM platform asserts the ARM DSM request signal to the CCM.

The CCM continuously synchronizes the DSM requests from ARM and TZIC to the clock domain used by its internal state machine.The only time when those signals are not synchronized is when the DPLLs are closed. In this case, exit from STOP mode is done asynchronously. (See Stop Mode is Exited by the Following Procedure:.)

If DSM wakeup signal from TZIC is negated and DSM request from ARM is asserted, CCM begins the STOP (shutdown) sequence.

CCM stops the ARM clock and generates signals to indicate that it has started STOP procedure.

CCM closes SAHARA, RTIC, IPU, SDMA, SCC, EXTMC and AHBMAX clocks. (If they were active in RUN mode.) The request is issued if the handshake is not bypassed by programing the CCM_CLPCR[23:16] register bits. If the corresponding bits are set, then the request signal to the corresponding block is not issued. CCM does not wait for its acknowledge in the process of entering low power mode.

NOTE

Since EXTMC needs the DDR clock to be active for it to answer with handshake signal, make sure to turn on the DDR clock before entering low power mode.

Once the blocks have finished their operation and are ready to enter low power mode, they acknowledge CCM that it is safe to turn off their clocks.

The requests are generated by CCM in stages. CCM continues from stage to stage once all the acknowledges of a present stage are received. The stages are the following:

Stage 1: SAHARA, RTIC, IPU, SDMA, CAN1, CAN2

Stage 2: SCC

Stage 3: AHBMAX

Stage 4: EXTMC

Once CCM receives all the acknowledge signals needed, it enters stop mode and does the following tasks in the order they are written below:

- Closees the system block clocks, not including pgc_clk. The clocks closed include the synchronized clocks of Megamix (all blocks except ARM, IPU, VPU, GPUs, EXTMC and DPLLs) only if GPC indicates that Megamix is powered down.
- Asserts signal indicating that the system is in STOP mode.

Once CCM is in STOP mode, it checks both the DSM request and DSM wakeup signals for a change in value. If ARM DSM request is de-asserted or TZIC DSM wakeup signal is asserted, CCM exits stop mode. Otherwise, CCM generates a request to GPC to power down ARM platform. If in GPC, the ARM platform is programmed to be powered down, GPC powers down the system as follows:

- Closes all the DPLLs and DPLLCs.
- CCM closes the CAMPs.
- If CCM_CLPCR[SBYOS] bit is set, then CCM generates a signal to close the external oscillator and also puts the on board oscillator in power down mode. (If they were not already powered down. If one of them was already closed, then perform the SBYOS operation only on the one that is working. If both of them were closed in run mode, then do nothing.)
- If CCM_CLPCR[VSTBY] bit is set, then assert a signal that indicates to the power management IC, a shift in voltage to standby voltage.

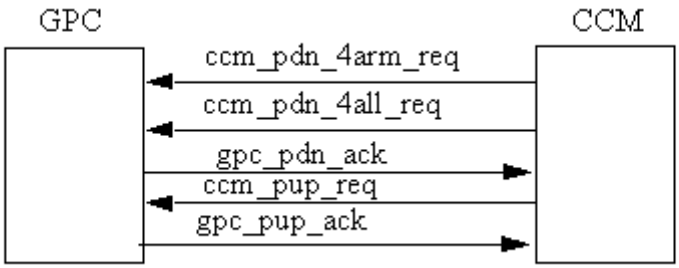


Figure 18-10. CCM-GPC connectivity

18.2.6.3.2 Stop Mode is Exited by the Following Procedure:

As soon as the unsynchronized TZIC DSM wakeup signal is seen asserted or ARM DSM request is seen de-asserted, the CCM start exit from STOP mode.

The following is take place:If CCM_CLPCR[VSTBY] bit is set, notify power management IC to change voltage from standby voltage to functional voltage.

If CCM_CLPCR[SBYOS] is set, and CCM closed either the external oscillator or the on board oscillator, then CCM starts the respective oscillator by asserting ref_en_b signal and deasserting cosc_pwrdown signal respectively.

After a pre-determined amount of CKIL cycles defined in STBY_COUNT, CCM waits until the Power Management IC asserts a signal indicating the voltage is at the "Functional Voltage" level. (Figure Figure 18-12 below describes the pmic signals connectivity.) Once the voltage is ready, CCM continues with the next steps:

- Starts CAMPs (based on definition of CCM_CCR[CAMP1_EN] and CCM_CCR[CAMP2_EN] bits)
- If any CAMP or the on board oscillator is started, waits until CCM_CCR[OSCNT] has finished counting to make sure that the external oscillator, CAMPs output and on board oscillator are ready.
- Start the DPLLs. Only the DPLLs that were configured to be on prior to the entrance to stop mode are started.
- Enable VPU, GPU2D, and GPU clocks.

CCM requests GPC to restore power to the ARM platform. GPC notifys CCM that power to ARM is back on, and it is safe to exit from STOP mode. Only then CCM carries out the following process:

- Once the powered down blocks are reset, CCM negates the low power request signals to all blocks and enables all block clocks including ARM clocks and synchronized CKIL of Megamix (if it was turned off), and returns to RUN mode.

(The clocks whose CCGR bits are set to 2b00, (that is, off in all modes) are not started in RUN mode and continue to be gated in RUN mode).

Once the interrupt propagates through all the synchronization logic, the ARM platform recognizes and services it. This forces the negation of the ARM DSM request. Prior to this point, the ARM DSM request and TZIC DSM wakeup were simultaneously asserted. Since TZIC DSM wakeup has top priority, the system is able to wake up.

Before the ARM exits the ISR, it clears the interrupt source. This causes the TZIC DSM wakeup signal to be negated. At this point, the two low power control signals are negated, and the STOP sequence can be repeated at any time.

Next figure describes connectivity of ARM, CCM and TZIC.

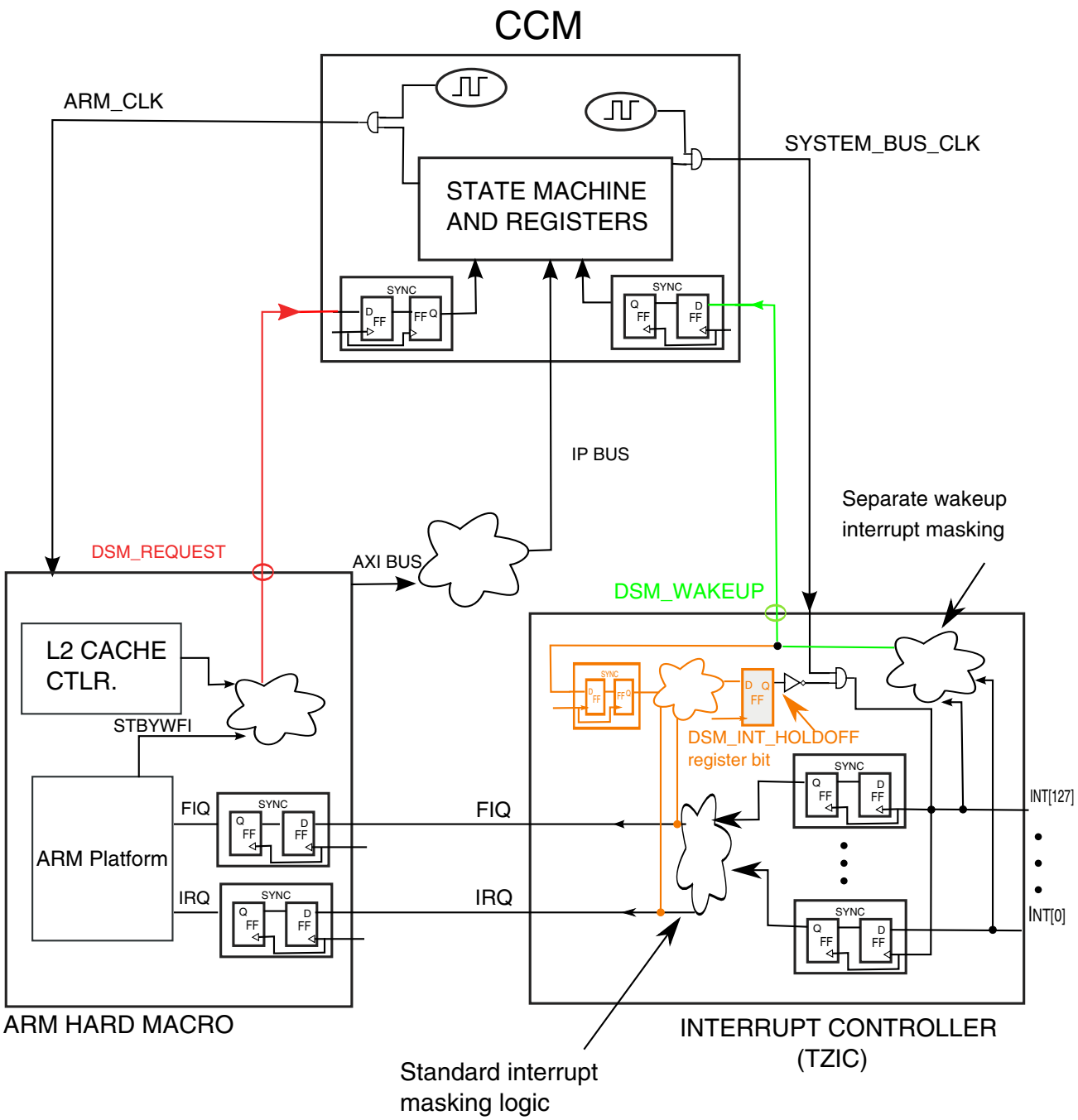


Figure 18-11. ARM Low Power Request

Next figure describes CCM's Low Power State Machine:

18.2.6.3.3 PMIC Signal Description:

stby_count value should be larger than t1 - larger than the amount of time that it takes between CCM's negation of pmic_vstby_req, and the negation of pmic_vfuncional_ready (the signal coming back from PMIC to indicate that the voltage started to change).

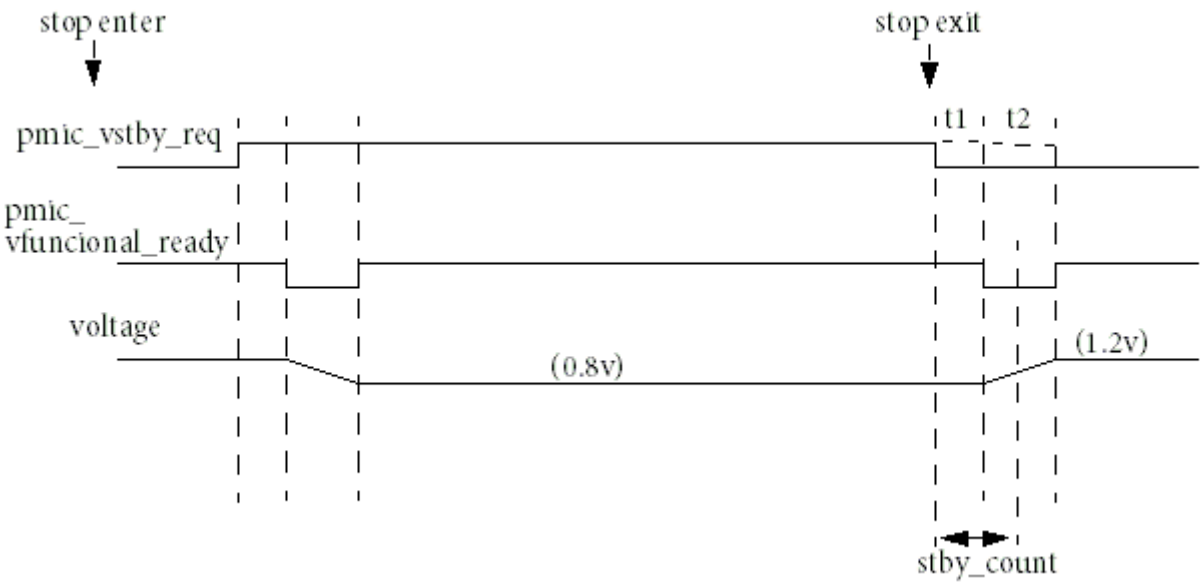


Figure 18-12. PMIC Signal Description

NOTE

Software should configure the IOMUXC so that pmic_vstby_req and pmic_vfuncional_ready is operating in the correct ALT mode, so that they are connected between CCM and the pads.

18.2.6.4 LPMD Request from SRC (Reset Controller)

SRC generates a signal named emi_dvfs_req. This signal is asserted in case of WARM reset to request EXTMC to enter sdram into SELFREFRESH mode. Since CCM is also a generator for the request to EXTMC for entering sdram to selfrefresh, then the SRC request is connected to CCM, and CCM is generate one common request to EXTMC for sdram selfrefresh. This common request is assert either if CCM is the originator of the request or if the SRC is the originator of the request. The SRC request (named emi_dvfs_req) is connected to CCM's input src_warm_dvfs_req. CCM generates the request to EXTMC via dvfs request . This signal generation is combined with the signal coming from SRC (src_warm_dvfs_req) to generate one dvfs request signal to EXTMC.

The acknowledge from EXTMC regarding the selfrefresh of EXTMC is connected to both CCM and SRC.

18.2.6.5 Low Power Audio Playback Mode (LP-APM)

Low Power Audio Playback mode (The term APM is used for this mode, across this document), defines special low power mode, dedicated for Audio only playback mode. It involves DPLL on/off and frequency settings, as well as voltage and clock gating aspects, to enable lowest possible power consumption.

This section covers APM, as well as assumptions, and Enter / Exit flows from the APM mode.

18.2.6.5.1 Low Power APM Mode Definition

1. Only one DPLL of i.MX53 is running - ARM DPLL. This DPLL is set to the maximal frequency that ARM can run at that lowest point, while still ensure ARM and bus system processing needs. Targeting.
2. The peripherals is also work on low voltage (1.0v). The peripherals is work on frequency suitable for this low voltage (about third of the nominal frequency).
3. For i.MX53 used as master on SSI clocks, the ARM platform DPLL frequency has to be set to an integer multiplication of SSI clock

18.2.6.5.2 LP-APM Mode Restrictions

- Entrance and exit to/from Low Power APM mode has effect on Display and SSI clocks. If i.MX53 is master on SSI clocks, audio playback is interrupted during the switch. It is not recommended to switch modes during audio playback.
- Due to impact to Display controller clock, switch to/from LP-APM, SW should take that into consideration. This is not applicable when using Smart Display.

Following diagram describes the possible sequence to be performed to switch from a high power settings (3 DPLL's, high voltage) to the low power APM.

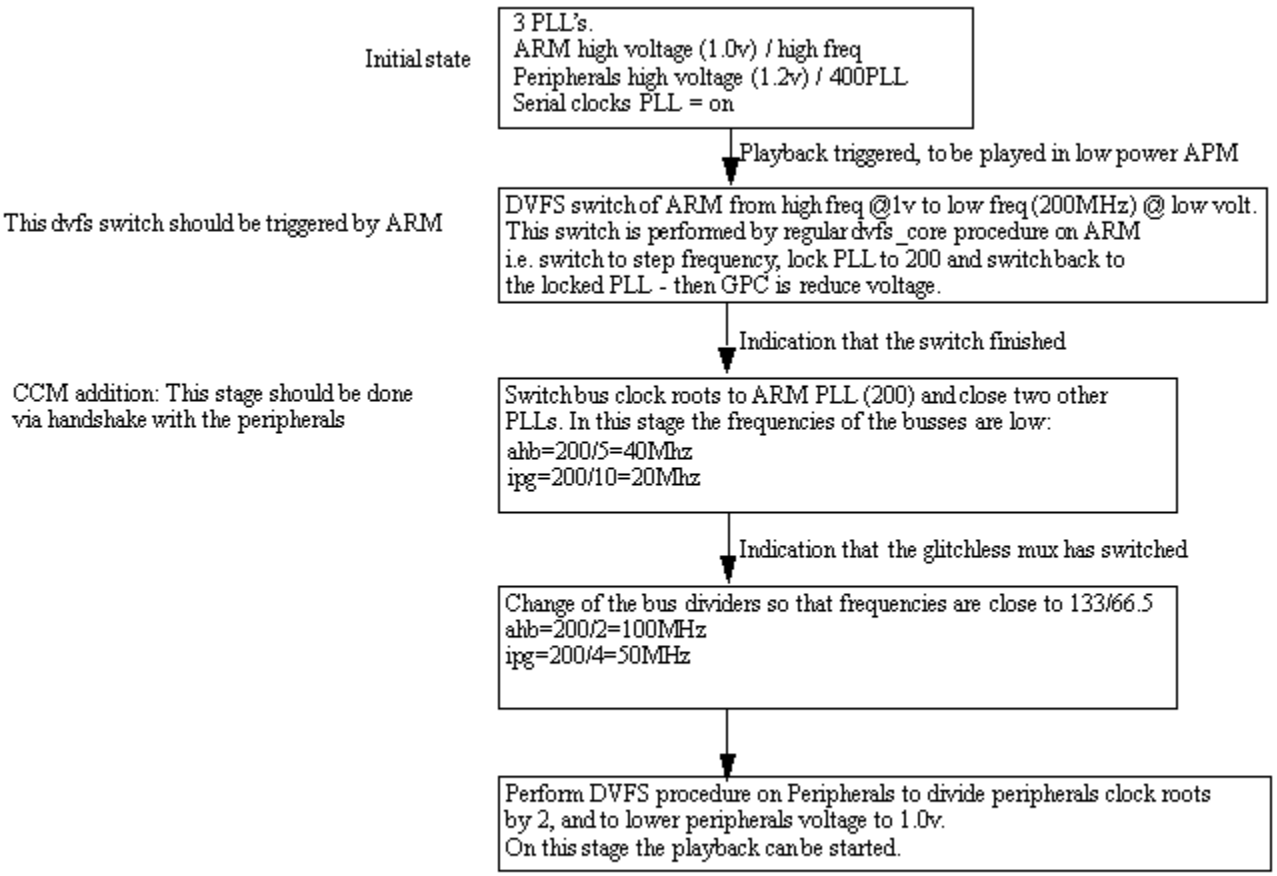


Figure 18-13. Flow of entering low power APM on i.MX53

Table 18-6. LP-APM flow frequencies

Stage	Farm [MHz]	Varm [v]	Fahb [MHz]	Fipg [MHz[	Vper [v]	comments
initial state	800*	1.0*	133	66.5	1.2	run system from 3 PLL's: ARM PLL = 800MHz supplying ARM PERIPH PLL = 665MHz supplying busses (ahb, ipg). Serial PLL = 216MHz supplying USB, TVE etc.
DVFS on ARM	200*	0.8*	133	66.5	1.2	GPC operation triggered by software

Table continues on the next page...

Table 18-6. LP-APM flow frequencies (continued)

Stage	Farm [MHz]	Varm [v]	Fahb [MHz]	Fipg [MHz[	Vper [v]	comments
Switch bus clocks to ARM PLL	200*	0.8*	40	20	1.2	Peripheral dividers are not changed during the shift. The shift should involve EXTMC handshake
change bus dividers	200*	0.8*	100	50	1.2	
DVFS on peripherals	200*	0.8*	50	25	1.0	DVFS on peripherals with div=2

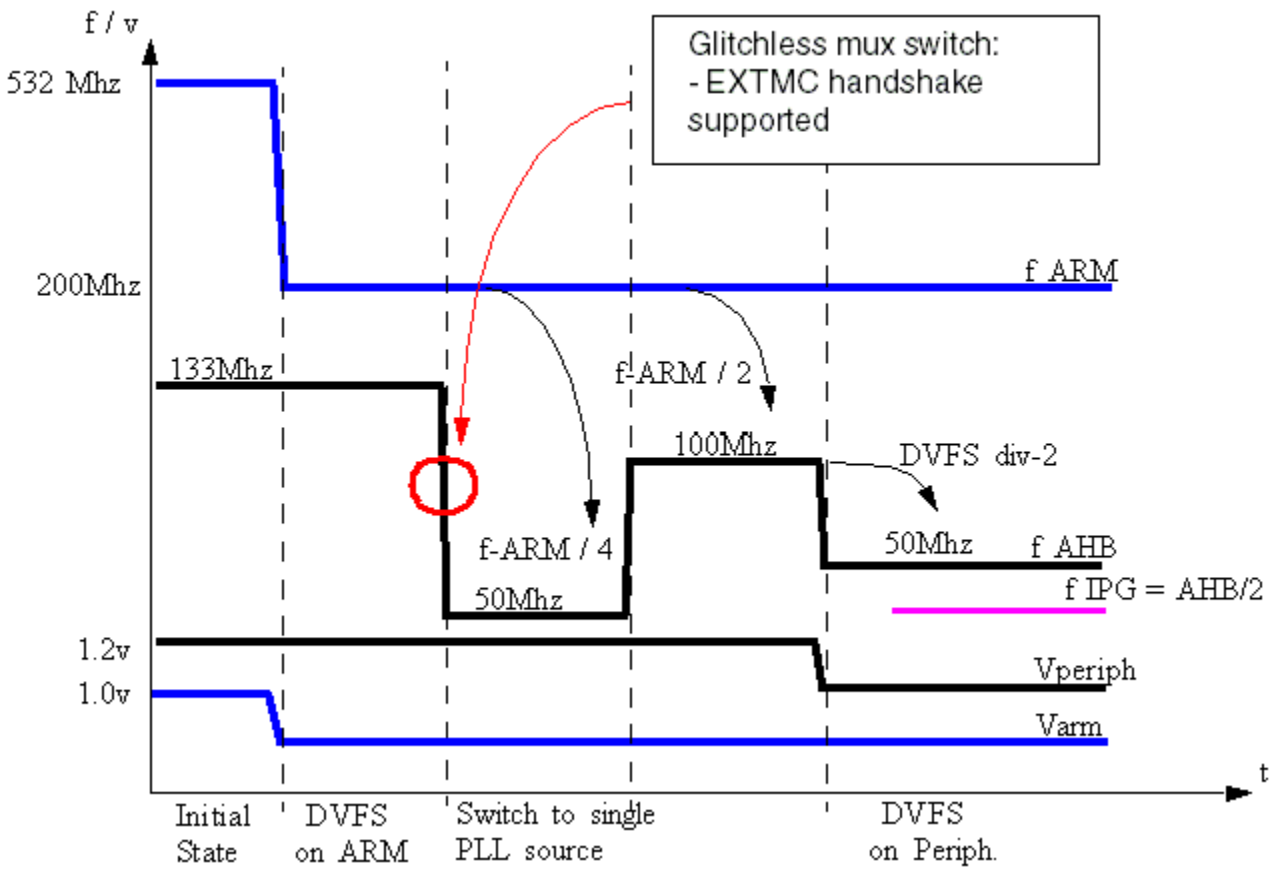


Figure 18-14. Frequency and Voltage flow

Alternative LP-APM clock settings can be applied:

- Entering LPAPM:
  - EXTMC clocks should be enabled.
  - IPU handshake should be masked, since IPU is off during LPAPM mode.
  - Disable DVFSP operation.

- Perform DVFSC procedure to lower the DPLL-1 clock (to ~200 MHz).
- Mask DVFSC operation.
- Set DPLL-1 as source of main_bus_clk.
- Set the AHB and EXTMC dividers to divide by 8, equal to 25MHz. Poll the load dividers bit after EMI_PODF is set.
- EXTMC root clock can be sourced from AHB clock.

- Exiting LP-APM mode
  - Set the AHB and EXTMC dividers to divide by 3.
  - Set DPLL-2 as the source of the main_bus_clk.
  - Perform DVFSC procedure to restore (increase) the DPLL-1 clock.
  - Enable (if required) DVFSP operation.

18.2.6.6 Recommendations for using low power consumption from CCM

In addition to using APM mode for low power, here are recommendations for configuring CCM so that it is consume less power. Those recommendations should be followed when applicable:

1. Using AHB clock source as the source of all AXI buses and VPU core clk: here, the power from the AXI dividers is preserved.
2. Working from one DPLL and disabling the other DPLL's.
3. Generate SSI external clocks from SSI clock roots: here, power consumption by SSI clock dividers is preserved.
4. Generate SPDIF clocks from SSI clock roots: here, the power consumption by SPDIF clock dividers is prevented.

18.3 Programmable Registers

CCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FD_4000	CCM Control Register (CCM_CCR)	32	R/W	0000_16FFh	18.3.1/ 866
53FD_4004	CCM Control Divider Register (CCM_CCDR)	32	R/W	0000_0000h	18.3.2/ 868

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FD_4008	CCM Status REgister (CCM_CSR)	32	R	0000_0010h	18.3.3/ 869
53FD_400C	CCM Clock Swither Register (CCM_CCSR)	32	R/W	0000_0000h	18.3.4/ 870
53FD_4010	CCM Arm Clock Root Register (CCM_CACRR)	32	R/W	0000_0000h	18.3.5/ 872
53FD_4014	CCM Bus Clock Divider Register (CCM_CBCDR)	32	R/W	0088_8945h	18.3.6/ 873
53FD_4018	CCM Bus Clock Multiplexer Register (CCM_CBCMR)	32	R/W	0001_6154h	18.3.7/ 876
53FD_401C	CCM Serial Clock Multiplexer Register 1 (CCM_CSCMR1)	32	R/W	A6A2_A020h	18.3.8/ 878
53FD_4020	CCM Serial Clock Multiplexer Register 2 (CCM_CSCMR2)	32	R/W	00B1_2F02h	18.3.9/ 881
53FD_4024	CCM Serial Clock Divider Register 1 (CCM_CSCDR1)	32	R/W	0043_0318h	18.3.10/ 884
53FD_4028	CCM SSI1 Clock Divider Register (CCM_CS1CDR)	32	R/W	0286_0241h	18.3.11/ 887
53FD_402C	CCM SSI2 Clock Divider Register (CCM_CS2CDR)	32	R/W	0086_0041h	18.3.12/ 888
53FD_4030	CCM D1 Clock Divider Register (CCM_CDCDR)	32	R/W	1437_01D2h	18.3.13/ 890
53FD_4034	CCM HSC Clock Divider Register (CCM_CHSCCDR)	32	R/W	0000_00A0h	18.3.14/ 892
53FD_4038	CCM Serial Clock Divider Register 2 (CCM_CSCDR2)	32	R/W	1208_0844h	18.3.15/ 893
53FD_403C	CCM Serial Clock Divider Register 3 (CCM_CSCDR3)	32	R/W	0000_0041h	18.3.16/ 895
53FD_4040	CCM Serial Clock Divider Register 4 (CCM_CSCDR4)	32	R/W	0000_0241h	18.3.17/ 895
53FD_4048	CCM Divider Handshake In-Process Register (CCM_CDHIPR)	32	R	0000_0000h	18.3.18/ 897
53FD_404C	CCM DVFS Control Register (CCM_CDCR)	32	R/W	0000_0000h	18.3.19/ 899
53FD_4054	CCM Low Power Control Register (CCM_CLPCR)	32	R/W	0000_0079h	18.3.20/ 901
53FD_4058	CCM Interrupt Status Register (CCM_CISR)	32	R/W	0000_0000h	18.3.21/ 905
53FD_405C	CCM Interrupt Mask Register (CCM_CIMR)	32	R/W	FFFF_ FFFFh	18.3.22/ 907

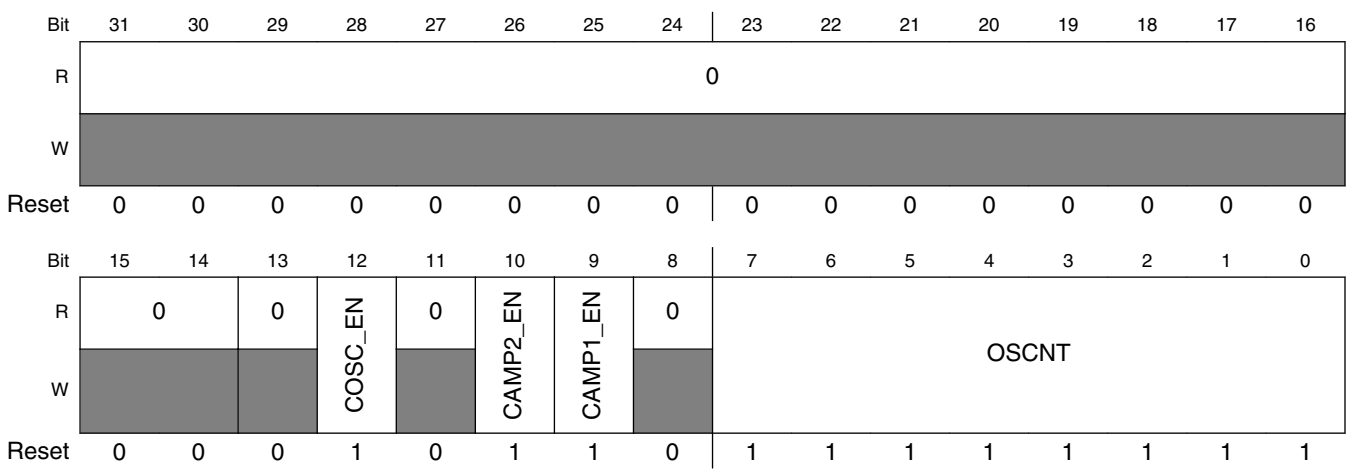
Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FD_4060	CCM Clock Output Source Register (CCM_CCOSR)	32	R/W	000A_0001h	18.3.23/ 909
53FD_4064	CCM General Purpose Register (CCM_CGPR)	32	R/W	0000_0000h	18.3.24/ 912
53FD_4068	CCM Clock Gating Register 0 (CCM_CCGR0)	32	R/W	FFFF_ FFFFh	18.3.25/ 913
53FD_406C	CCM Clock Gating Register 1 (CCM_CCGR1)	32	R/W	FFFF_ FFFFh	18.3.26/ 914
53FD_4070	CCM Clock Gating Register 2 (CCM_CCGR2)	32	R/W	FFFF_ FFFFh	18.3.27/ 915
53FD_4074	CCM Clock Gating Register 3 (CCM_CCGR3)	32	R/W	FFFF_ FFFFh	18.3.28/ 916
53FD_4078	CCM Clock Gating Register 4 (CCM_CCGR4)	32	R/W	FFFF_ FFFFh	18.3.29/ 917
53FD_407C	CCM Clock Gating Register 5 (CCM_CCGR5)	32	R/W	FFFF_ FFFFh	18.3.30/ 919
53FD_4080	CCM Clock Gating Register 6 (CCM_CCGR6)	32	R/W	FFFF_ FFFFh	18.3.31/ 920
53FD_4084	CCM Clock Gating Register 7 (CCM_CCGR7)	32	R/W	FFFF_ FFFFh	18.3.32/ 921
53FD_4088	CCM Module Enable Overide Register (CCM_CMEOR)	32	R/W	FFFF_ FFFFh	18.3.33/ 923

18.3.1 CCM Control Register (CCM_CCR)

Address: CCM_CCR is 53FD_4000h base + 0h offset = 53FD_4000h



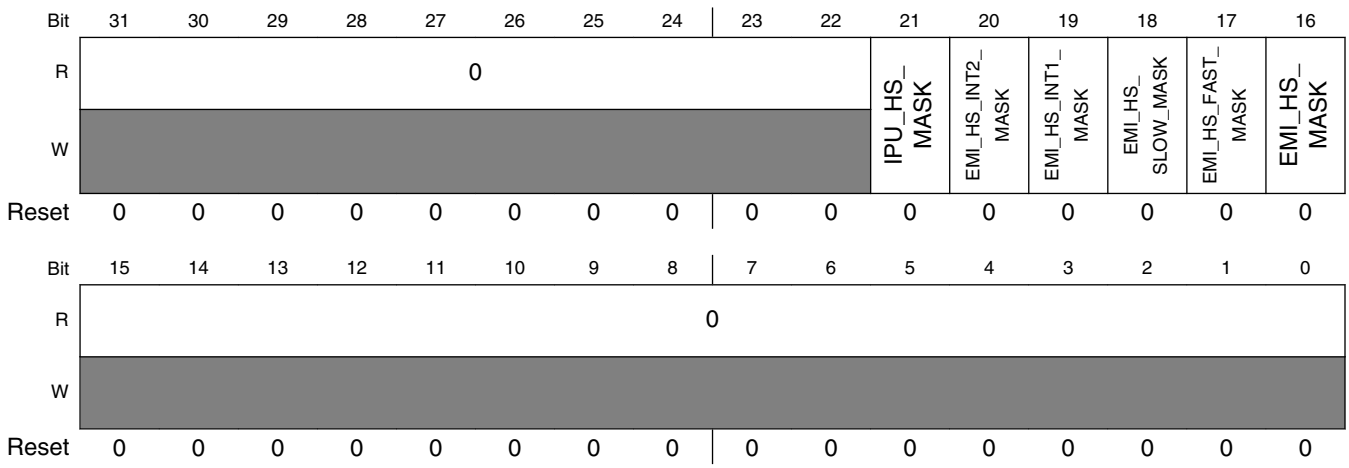
CCM_CCR field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 COSC_EN	On chip oscillator enable bit. This bit value is reflected on the output COSC_EN. The system starts with the on chip oscillator enabled to supply source for the DPLL's. This bit can be changed if a transition to the bypass DPLL clocks is desired for all the DPLLs. In case this bit is changed from '0' to '1', CCM enables the on chip oscillator and after counting OSCNT CKIL clock cycles it notifies that the on chip oscillator is ready by an interrupt cosc_ready and by status bit COSC_READY. The COSC_EN bit can be changed only when the on chip oscillator is not chosen as the clock source. 0 disable on chip oscillator 1 enable on chip oscillator
11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10 CAMP2_EN	CAMP-2 Enable bit. The ignition process always sets this bit. In RUN mode, this bit controls the enable/disable of CAMP-2. If this bit is changed from '0' to '1' then CCM enables CAMP-2 and after counting OSCNT CKILs, it notifies that CAMP-2 is ready by an interrupt camp2_ready and by status bit CAMP2_READY. The CAMP2_EN bit should be changed only when CAMP-2 is not chosen as a clock source. **NOTE:** CCM has an output ccm_camp2_dis. This signal is inverted from the CAMP2_EN bit, i.e. when CAMP2_EN bit is '1' the ccm_camp2_dis signal is '0'. 0 disable CAMP-2 1 enable CAMP-2
9 CAMP1_EN	CAMP-1 Enable bit. The ignition process always sets this bit. In RUN mode, this bit controls the enable/disable of CAMP-1. If this bit is changed from '0' to '1' then CCM enables CAMP-1 and after counting OSCNT CKILs, it notifies that CAMP-1 is ready by an interrupt camp1_ready and by status bit CAMP1_READY. The CAMP1_EN bit should be changed only when CAMP-1 is not chosen as a clock source. **NOTE:** CCM has an output ccm_camp1_dis. This signal is inverted from the CAMP1_EN bit, i.e. when CAMP1_EN bit is '1' the ccm_camp1_dis signal is '0'. 0 disable CAMP-1 1 enable CAMP-1
8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–0 OSCNT	Oscillator ready counter value. These bits define the value of 32KHz counter, that serves as a counter for oscillator lock time. This is used for both the on chip oscillator lock time and the time that CAMP-1 and CAMP-2 require to be ready, since the CAMPs receives the external oscillator clock. Current estimation is ~5ms. This counter is used in ignition sequence and in wake from STOP sequence if SBYOS bit was defined. It notifies that the on chip oscillator output is ready for DPLLC to use.Only then can the gate in DPLLC be opened. 00000000 count 1 ckil 11111111 count 256 ckil's (Default)

18.3.2 CCM Control Divider Register (CCM_CCDR)

The figure below represents the CCM Control Divider Register (CCM_CCDR), which contains bits that control the loading of the dividers that need handshake with the blocks they affect.

Address: CCM_CCDR is 53FD_4000h base + 4h offset = 53FD_4004h



CCM_CCDR field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value zero. Reserved
21 IPU_HS_MASK	During the divider ratio change procedure, mask handshake with IPU. 0 allow handshake with IPU. 1 mask handshake with IPU. Request signal is not be generated.
20 EMI_HS_INT2_MASK	During the divider ratio change procedure, mask handshake with EXTMC int2 part only. 0 allow handshake with extermal memory controller (EXTMC). 1 mask handshake with EXTMC. Request signal is not be generated.
19 EMI_HS_INT1_MASK	During the divider ratio change procedure, mask handshake with emi int1 part only. 0 allow handshake with EXTMC 1 mask handshake with EXTMC. Request signal is not be generated.
18 EMI_HS_SLOW_MASK	During the divider ratio change procedure, mask handshake with emi slow part only. 0 allow handshake with EXTMC 1 mask handshake with EXTMC. Request signal is not be generated.
17 EMI_HS_FAST_MASK	During the divider ratio change procedure, mask handshake with emi fast part only. 0 allow handshake with EXTMC 1 mask handshake with EXTMC. Request signal is not be generated.

Table continues on the next page...

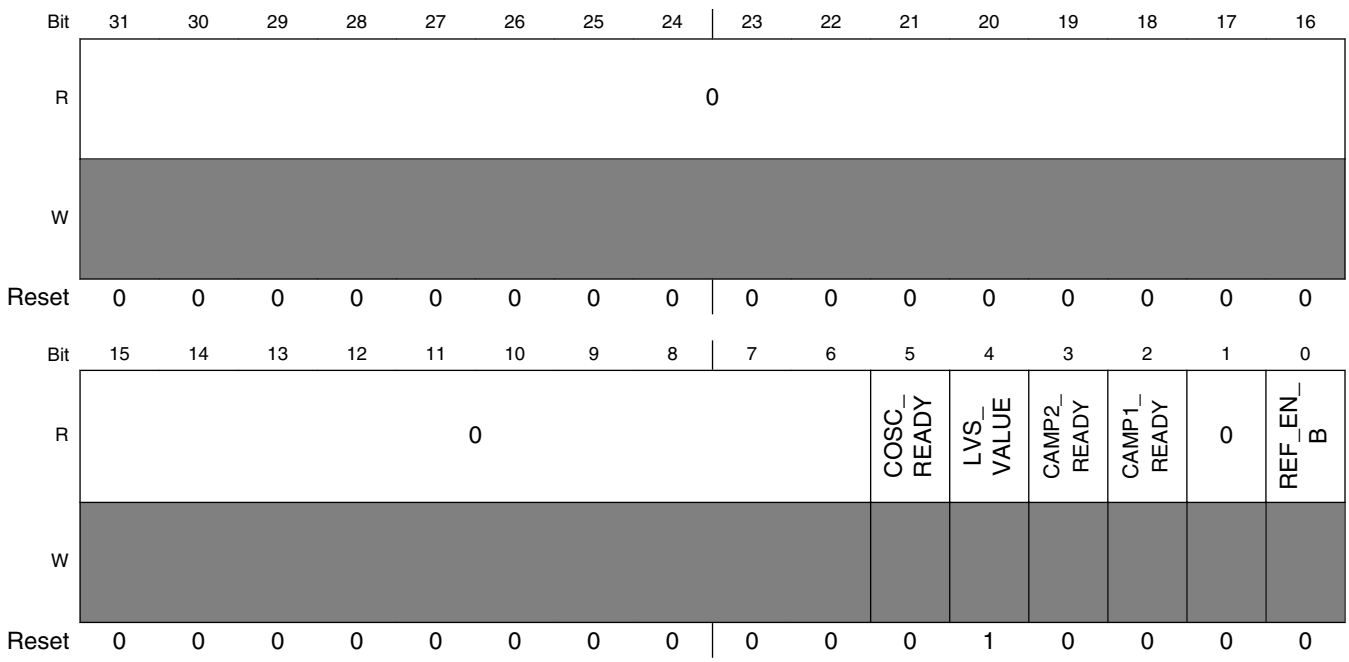
CCM_CCDR field descriptions (continued)

Field	Description
16 EMI_HS_MASK	During the divider ratio change procedure, mask handshake with EXTMC. This is used to mask all the EXTMC frequency change handshakes together. 0 allow handshake with EXTMC 1 mask handshake with EXTMC. Request signal is not be generated.
15–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

18.3.3 CCM Status REgister (CCM_CSR)

The figure below represents the CCM status Register (CCM_CSR). The status bits are read only bits.

Address: CCM_CSR is 53FD_4000h base + 8h offset = 53FD_4008h



CCM_CSR field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5 COSC_READY	Status indicator of on board oscillator. This bit is asserted if on chip oscillator is enabled and on chip oscillator is not powered down, and if OSCNT counter has finished counting. 0 on board oscillator is not ready. 1 on board oscillator is ready.

Table continues on the next page...

CCM_CSR field descriptions (continued)

Field	Description
4 LVS_VALUE	Status of the LVS value for DPLL (output from CCM to select between the LVS / HVS frequency configuration) 0 value of LVS select is '0' 1 value of LVS select is '1'
3 CAMP2_READY	Status indication of CAMP-2. This is asserted if CAMP-2 is enabled, and if OSCNT counter has finished counting. 0 CAMP-2 is not ready. 1 CAMP-2 is ready.
2 CAMP1_READY	Status indication of CAMP-1. This is asserted if CAMP-1 is enabled, and if OSCNT counter has finished counting. 0 CAMP-1 is not ready. 1 CAMP-1 is ready.
1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 REF_EN_B	Status of the value of ref_en_b output of CCM 0 value of ref_en_b is '0' 1 value of ref_en_b is '1'

18.3.4 CCM Clock Swither Register (CCM_CCSR)

The figure below represents the CCM Clock Switcher register (CCM_CCSR). The CCM_CCSR register contains bits to control the switcher sub-block dividers and multiplexers.

Address: CCM_CCSR is 53FD_4000h base + Ch offset = 53FD_400Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					LP_APM	PLL4_SW_CLK_SEL	STEP_SEL [1:0]		PLL2_DIV_PODF [1:0]		PLL3_DIV_PODF [1:0]		PLL1_SW_CLK_SEL	PLL2_SW_CLK_SEL	PLL3_SW_CLK_SEL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_CCSR field descriptions

Field	Description
31–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10 LP_APM	Selects the option to be chosen for the Low Power Audio Playback source clock See Low Power Audio Playback Mode (LP-APM). 0 Oscillator clock output 1 PLL4 clock source
9 PLL4_SW_CLK_SEL	Selects source to generate pll4_sw_clk. This bit should only be used for testing purposes. **NOTE:** If (pll_bypass_en4 = '1' OR PLL4_SW_CLK_SEL = '1'), pll4_main_clk = DPLL-4 bypass clock. Only if both are '0', pll4_main_clk = pll4_sw_clk. 0 pll4_main_clk(Default) 1 pll4 bypass clock
8–7 STEP_SEL [1:0]	Selects the option to be chosen for the step frequency when shifting ARM frequency. this is control the step_clk. **NOTE:** This multiplexer should be changed only if its output is not used, i.e. ARM uses the output of DPLL-1, and step_clk is not used. **NOTE:** To conserve power, it is preferred to not set this multiplexer to the DPLL-2 and DPLL-3 options when they are not needed. If they are not chosen, then the path from DPLL-2 and DPLL-3 to this multiplexer is closed and its power is conserved. 00 clock source 4 - source for lp_apm. (Default) 01 pll1 bypass clock 10 divided pll2 clock 11 divided pll3 clock
6–5 PLL2_DIV_PODF [1:0]	divider for DPLL-2 clock. **NOTE:** This podf is allowed to be changed only during the period when its output is not used. **NOTE:** This podf is allowed to be changed only during the period when its output is not used. 00 divide by 1 01 divide by 2 10 divide by 3 11 divide by 4
4–3 PLL3_DIV_PODF [1:0]	divider for DPLL-3 clock. 00 divide by 1 01 divide by 2 10 divide by 3 11 divide by 4
2 PLL1_SW_CLK_SEL	Selects source to generate pll1_sw_clk. **NOTE:** If (pll_bypass_en1 = 1 OR PLL1_SW_CLK_SEL = 1 OR dvfs_control =1), pll1_sw_clk = step_clock. Only if all 3 are 0, pll1_sw_clk = pll1_main_clk. 0 pll1_main_clk(Default) 1 step_clk

Table continues on the next page...

CCM_CCSR field descriptions (continued)

Field	Description
1 PLL2_SW_CLK_SEL	Selects source to generate pll2_sw_clk. This bit should only be used for testing purposes. **NOTE:** If (pll_bypass_en2 = 1 OR PLL2_SW_CLK_SEL = 1), pll2_main_clk = DPLL-2 bypass clock. Only if both are 0, pll2_main_clk = pll2_sw_clk. 0 pll2_main_clk(Default) 1 pll2 bypass clock
0 PLL3_SW_CLK_SEL	Selects source to generate pll3_sw_clk. This bit should only be used for testing purposes. **NOTE:** If (pll_bypass_en3 = 1 OR PLL3_SW_CLK_SEL = 1), pll3_main_clk = DPLL-3 bypass clock. Only if both are 0, pll3_main_clk = pll3_sw_clk. 0 pll3_main_clk(Default) 1 pll3 bypass clock

18.3.5 CCM Arm Clock Root Register (CCM_CACRR)

The figure below represents the CCM Arm Clock Root register (CCM_CACRR). The CCM_CACRR register contains bits to control the ARM clock root generation.

Address: CCM_CACRR is 53FD_4000h base + 10h offset = 53FD_4010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																													arm_podf [2:0]		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_CACRR field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 arm_podf [2:0]	Divider for ARM clock root. **NOTE:** If CCM_CDCR[ARM_FREQ_SHIFT_DIVIDER] is set to '1' then any new write to ARM_PODF is held until change ARM frequency request signal is asserted by GPC. 000 divide by 1(Default) 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8

18.3.6 CCM Bus Clock Divider Register (CCM_CBCDR)

The figure below represents the CCM Bus Clock Divider Register (CCM_CBCDR). The register contains bits to control the clock generation sub-block dividers.

Address: CCM_CBCDR is 53FD_4000h base + 14h offset = 53FD_4014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					EMI_CLK_SEL	PERIPH_CLK_SEL	EMI_SLOW_PODF [2:0]			AXI_B_PODF [2:0]			AXI_A_PODF [2:0]		
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NFC_PODF [2:0]			AHB_PODF [2:0]			IPG_PODF [1:0]		PERCLK_PRED1 [1:0]		PERCLK_PRED2 [2:0]			PERCLK_PODF [2:0]		
W																
Reset	1	0	0	0	1	0	0	1	0	1	0	0	0	1	0	1

CCM_CBCDR field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26 EMI_CLK_SEL	Selector for EXTMC clock group **NOTE:** Any change to this multiplexer might involve handshake with EXTMC and IPU. See CCM_CDHIPR register for the handshake busy bits. 0 derive clock from dvfs divider 1 derive clock from ahb clock root
25 PERIPH_CLK_SEL	Selector for peripheral main clock. **NOTE:** Any change to this multiplexer involves handshake with EXTMC and IPU - a similar handshake to the one that is performed on DVFSP. 0 derive clock from pll2_sw_clk clock source. 1 derive clock from periph_apm_clk clock source.
24–22 EMI_SLOW_PODF [2:0]	Selector for EXTMC slow post divider. **NOTE:** Any change of this divider might involve handshake with EXTMC and IPU. See CDHIPR register for the handshake busy bits. 000 divide by 1 001 divide by 2 010 divide by 3 (default) 011 divide by 4 100 divide by 5 101 divide by 6

Table continues on the next page...

CCM_CBCDR field descriptions (continued)

Field	Description
	110 divide by 7 111 divide by 8
21–19 AXI_B_PODF [2:0]	Selector for AXI B post-divider. **NOTE:** Any change of this divider might involve handshake with EXTMC and IPU. See CDHIPR register for the handshake busy bits. 000 divide by 1 001 divide by 2 (default) 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
18–16 AXI_A_PODF [2:0]	Selector for AXI A post-divider. **NOTE:** Any change of this divider might involve handshake with EXTMC and IPU. See CDHIPR register for the handshake busy bits. 000 divide by 1 (default) 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
15–13 NFC_PODF [2:0]	Selector for NFC post-divider. **NOTE:** Any change of this divider might involve handshake with EXTMC. See CDHIPR register for the handshake busy bits. 000 restricted 001 divide by 2(Default) 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
12–10 AHB_PODF [2:0]	Selector for AHB post-divider. **NOTE:** Any change of this divider might involve handshake with EXTMC and IPU. See CDHIPR register for the handshake busy bits. 000 divide by 1 001 divide by 2 010 divide by 3(Default) 011 divide by 4

Table continues on the next page...

CCM_CBCDR field descriptions (continued)

Field	Description
	100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
9–8 IPG_PODF [1:0]	Selector for ipg clock post-divider. **NOTE:** IPTP block does not support ratio of 1:3 for ahb_clk:ipg_clk. In case IEEE_RTC is used, then the ratio should not be used. **NOTE:** SDMA does not support ratio of 1:3 and 1:4 for ahb_clk:ipg_clk. In case SDMA is used, then these ratios should not be used. 00 divide by 1 01 divide by 2 (Default) 10 divide by 3 11 divide by 4
7–6 PERCLK_ PRED1 [1:0]	Selector for peripherals clock pre-divider1 **NOTE:** Divider should be updated when output clock is gated. 00 divide by 1 01 divide by 2 (default) 10 divide by 3 11 divide by 4
5–3 PERCLK_ PRED2 [2:0]	Selector for peripherals clock pre-divider2. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1(default) 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
2–0 PERCLK_PODF [2:0]	Selector for peripherals clock post-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6(default) 110 divide by 7 111 divide by 8

18.3.7 CCM Bus Clock Multiplexer Register (CCM_CBCMR)

The figure below represents the CCM Bus Clock Multiplexer Register (CCM_CBCMR). The CCM_CBCMR register contains bits to control the multiplexers that generate the bus clocks.

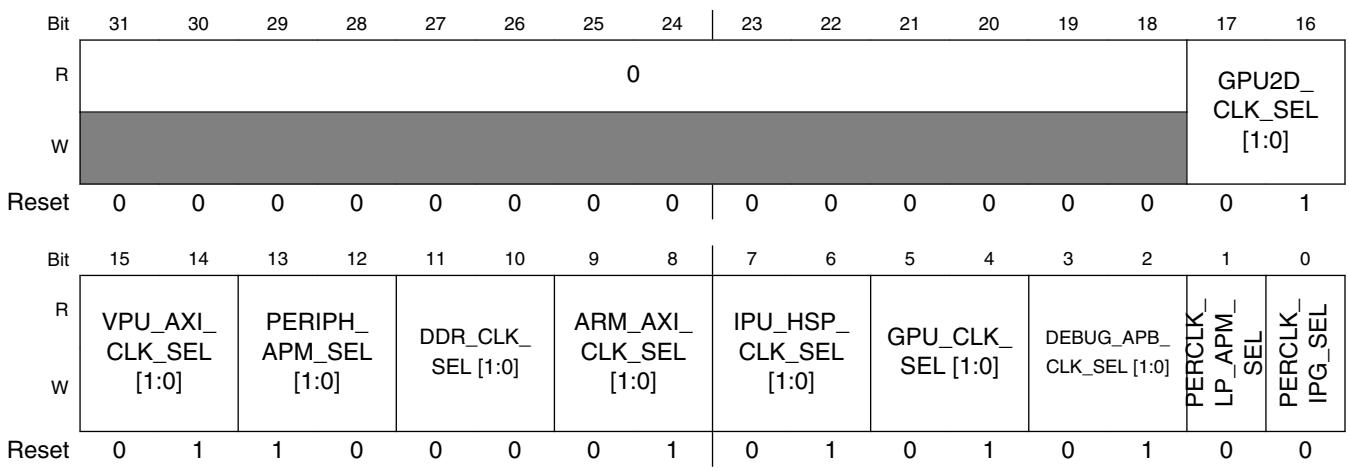
NOTE

Any change on the above multiplexer has to be done while the affected block's clock is not functional and the respective clock is gated in LPCG. If the change is done during operation of the block, it is not guaranteed that the block's operation cannot be harmed.

NOTE

The change for arm_axi_clk_sel should be done through SDMA so that ARM does not use this clock during the change, and the clock is gated in LPCG.

Address: CCM_CBCMR is 53FD_4000h base + 18h offset = 53FD_4018h



CCM_CBCMR field descriptions

Field	Description
31–18 Reserved	This read-only field is reserved and always has the value zero. Reserved
17–16 GPU2D_CLK_ SEL [1:0]	Selector for open vg clock multiplexer 00 derive clock from axi a 01 derive clock from axi b (Default) 10 derive clock from emi_slow_clk_root 11 derive clock from ahb clock root

Table continues on the next page...

CCM_CBCMR field descriptions (continued)

Field	Description
15–14 VPU_AXI_CLK_SEL [1:0]	Selector for VPU axi clock multiplexer 00 derive clock from axi a 01 derive clock from axi b (Default) 10 derive clock from emi_slow_clk_root 11 derive clock from ahb clock root
13–12 PERIPH_APM_SEL [1:0]	Selector for peripheral clock multiplexer 00 derive clock from pll1_sw_clk 01 derive clock from pll3_sw_clk 10 lp_apm clock (default) 11 reserved
11–10 DDR_CLK_SEL [1:0]	Selector for DDR clock multiplexer 00 derive clock from axi a (Default) 01 derive clock from axi b 10 derive clock from emi_slow_clk_root 11 derive clock from ahb clock root
9–8 ARM_AXI_CLK_SEL [1:0]	Selector for ARM axi clock multiplexer 00 derive clock from axi a 01 derive clock from axi b (Default) 10 derive clock from emi_slow_clk_root 11 derive clock from ahb clock root
7–6 IPU_HSP_CLK_SEL [1:0]	Selector for ipu hsp clock multiplexer 00 derive clock from axi a 01 derive clock from axi b (Default) 10 derive clock from emi_slow_clk_root 11 derive clock from ahb clock root
5–4 GPU_CLK_SEL [1:0]	Selector for gpu clock multiplexer 00 derive clock from axi a 01 derive clock from axi b (Default) 10 derive clock from emi_slow_clk_root 11 derive clock from ahb clock root
3–2 DEBUG_APB_CLK_SEL [1:0]	Selector for debug apb clock multiplexer 00 derive clock from axi a 01 derive clock from axi b (Default) 10 derive clock from emi_slow_clk_root 11 derive clock from ahb clock root
1 PERCLK_LP_APM_SEL	Controls if the root clock generation of perclk is from peripherals main clock or lp_apm source. 0 generate perclk_root from peripherals main clock source. 1 generate perclk_root from lp_apm source.

Table continues on the next page...

CCM_CBCMR field descriptions (continued)

Field	Description
0 PERCLK_IPG_SEL	Selector for perclk multiplexer - allows selection between ipg_clk_root or the division of chosen DPLL. 0 select perclk generation from division of DPLL frequency 1 select perclk generation from ipg_clk_root

18.3.8 CCM Serial Clock Multiplexer Register 1 (CCM_CSCMR1)

The figure below represents the CCM Serial Clock Multiplexer Register 1 (CCM_CSCMR1). The register contains bits to control the multiplexers that generate the serial clocks.

NOTE

Any change to the above multiplexer has to be done while the affected block's clock is not functional and the clock is gated. If the change is done during operation of the block, it is not guaranteed that the block's operation cannot be harmed.

Address: CCM_CSCMR1 is 53FD_4000h base + 1Ch offset = 53FD_401Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SSI_EXT2_CLK_SEL [1:0]		SSI_EXT1_CLK_SEL [1:0]		0	USB_PHY_CLK_SEL	UART_CLK_SEL[1:0]		USBOH3_CLK_SEL [1:0]		ESDHC1_CLK_SEL [1:0]		ESDHC2_CLK_SEL	ESDHC4_CLK_SEL	ESDHC3_CLK_SEL [1:0]	
W																
Reset	1	0	1	0	0	1	1	0	1	0	1	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SSI1_CLK_SEL [1:0]		SSI2_CLK_SEL [1:0]		SSI3_CLK_SEL	VPU_RCLK_SEL	SSI_APM_CLK_SEL		0	TVE_EXT_CLK_SEL	ECSPI_CLK_SEL [1:0]		SPDIF_XTAL_CLK_SEL		SSI_EXT2_COM	SSI_EXT1_COM
W																
Reset	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

CCM_CSCMR1 field descriptions

Field	Description
31–30 SSI_EXT2_CLK_SEL [1:0]	Selector for SSI EXT2_CLK clock multiplexer 00 derive clock from pll1_sw_clk 01 derive clock from pll2_sw_clk 10 derive clock from pll3_sw_clk(Default) 11 derive clock from ssi_lp_apm_clk

Table continues on the next page...

CCM_CSCMR1 field descriptions (continued)

Field	Description
29–28 SSI_EXT1_CLK_SEL [1:0]	Selector for SSI_EXT1_CLK multiplexer 00 derive clock from pll1_sw_clk 01 derive clock from pll2_sw_clk 10 derive clock from pll3_sw_clk(Default) 11 derive clock from ssi_lp_apm_clk
27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26 USB_PHY_CLK_SEL	Selector for USB PHY clock multiplexer 0 derive clock from oscillator 1 derive clock from divided output of PLL3 (Default)
25–24 UART_CLK_SEL[1:0]	Selector for UART clock multiplexer 00 derive clock from pll1_sw_clk 01 derive clock from pll2_sw_clk 10 derive clock from pll3_sw_clk(Default) 11 lp_apm clock
23–22 USBOH3_CLK_SEL [1:0]	Selector for USBOH3 clock multiplexer 00 derive clock from pll1_sw_clk 01 derive clock from pll2_sw_clk 10 derive clock from pll3_sw_clk(Default) 11 lp_apm clock
21–20 ESDHC1_CLK_SEL [1:0]	Selector for ESDHCV2-1 clock multiplexer 00 derive clock from pll1_sw_clk 01 derive clock from pll2_sw_clk 10 derive clock from pll3_sw_clk(Default) 11 lp_apm clock
19 ESDHC2_CLK_SEL	Selector for ESDHCV2-2 clock multiplexer 0 derive clock from esdhc1_clk_sel(Default) 1 derive clock from esdhc3_clk_sel
18 ESDHC4_CLK_SEL	Selector for ESDHCV2-4 clock multiplexer 0 derive clock from esdhc1_clk_sel(Default) 1 derive clock from esdhc3_clk_sel
17–16 ESDHC3_CLK_SEL [1:0]	Selector for ESDHCV3-3 clock multiplexer 00 derive clock from pll1_sw_clk 01 derive clock from pll2_sw_clk 10 derive clock from pll3_sw_clk(Default) 11 lp_apm clock
15–14 SSI1_CLK_SEL [1:0]	Selector for SSI-1 clock multiplexer 00 derive clock from pll1_sw_clk

Table continues on the next page...

CCM_CSCMR1 field descriptions (continued)

Field	Description
	01 derive clock from pll2_sw_clk 10 derive clock from pll3_sw_clk(Default) 11 derive clock from ssi_lp_apm_clk
13–12 SSI2_CLK_SEL [1:0]	Selector for SSI-2 clock multiplexer 00 derive clock from pll1_sw_clk 01 derive clock from pll2_sw_clk 10 derive clock from pll3_sw_clk(Default) 11 derive clock from ssi_lp_apm_clk
11 SSI3_CLK_SEL	Selector for SSI-3 clock multiplexer 0 derive clock from ssi1_clk_root(Default) 1 derive clock from ssi2_clk_root
10 VPU_RCLK_SEL	Selector for vpu_rclk clock multiplexer 0 derive clock from osc_clk(Default) 1 derive clock from ckih_camp1_clk
9–8 SSI_APM_CLK_SEL	Selector for the ssi_lp_apm_clk clock generation. 00 CAMP-1 of CKIH (default) 01 LP-APM clock selector output 10 CAMP-2 of CKIH2 11 PLL4
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6 TVE_EXT_CLK_SEL	Selector for the TVE external clock input. 0 PLL4 1 CKIh through CAMP-1
5–4 ECSPI_CLK_SEL [1:0]	Selector for ECSPI clock multiplexer 00 derive clock from pll1_sw_clk 01 derive clock from pll2_sw_clk 10 derive clock from pll3_sw_clk(Default) 11 lp_apm clock
3–2 SPDIF_XTAL_CLK_SEL	Selector for the spdif_xtal_clk clock generation. 00 Oscillator output (default) 01 CAMP-1 of CKIH 10 CAMP-2 of CKIH2 11 PLL4
1 SSI_EXT2_COM	Selector for SSI_EXT2_CLK clock generation based on SSI-2 root clock. 0 generate ssi_ext2_clk_root from dividers 1 generate ssi_ext2_clk_root from ssi2_clk_root.
0 SSI_EXT1_COM	Selector for SSI_EXT1_CLK clock generation based on SSI-1 root clock.

Table continues on the next page...

CCM_CSCMR1 field descriptions (continued)

Field	Description
	0 generate ssi_ext1_clk_root from dividers 1 generate ssi_ext1_clk_root from ssi1_clk_root.

18.3.9 CCM Serial Clock Multiplexer Register 2 (CCM_CSCMR2)

The figure below represents the CCM Serial Clock Multiplexer Register 2 (CCM_CSCMR2). The register contains bits to control the multiplexers that generate the serial clocks.

NOTE

Any change to the above multiplexer has to be done while the affected block's clock is not functional and the clock is gated. If the change is done during operation of the block, it is not guaranteed that the block's operation cannot be harmed.

Address: CCM_CSCMR2 is 53FD_4000h base + 20h offset = 53FD_4020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DI1_CLK_SEL [2:0]			DI0_CLK_SEL [2:0]			0		CSI_MCLK1_ CLK_SEL [1:0]		ASRC_CLK_ SEL	ESAI_PRE_ SEL		ESAI_POST_ SEL[2:0]		
Reset	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IEEE_CLK_ SEL [1:0]		FIRI_CLK_ SEL [1:0]		LDB_DI1_ IPU_DIV	LDB_DI0_ IPU_DIV	LDB_DI1_ CLK_SEL	LDB_DI0_ CLK_SEL	CAN_CLK_ SEL[1:0]		0	SPDIF0_ COM	0		SPDIF0_ CLK_SEL [1:0]	
Reset	0	0	1	0	1	1	1	1	0	0	0	0	0	0	1	0

CCM_CSCMR2 field descriptions

Field	Description
31–29 DI1_CLK_SEL [2:0]	Selector for Display Interface 1 (DI1) clock multiplexer 000 derive clock from divided DPLL-3. (Default) 001 derive clock from oscillator 010 derive clock from CKIH CAMP-1 clock. 011 derive clock from tve_di_clock - in this case TVE supplies the DI clock. 100 derive clock from iDI1_EXT_CLK - clock source is generated from the IOMUX. 101 derive clock from ldb_di1_clk

Table continues on the next page...

CCM_CSCMR2 field descriptions (continued)

Field	Description
	110 reserved. 111 reserved.
28–26 DI0_CLK_SEL [2:0]	Selector for Display Interface 0 (DI0) clock multiplexer 000 derive clock from divided DPLL-3. (Default) 001 derive clock from oscillator 010 derive clock from ckih CAMP-1 clock. 011 derive clock from divided DPLL-4. 100 derive clock from DI0_EXT_CLK - clock source is generated from the IOMUX. 101 derive clock from ldb_di0_clk 111 reserved. 111 reserved.
25–24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23–22 CSI_MCLK1_ CLK_SEL [1:0]	Selector for CSI-1 clock multiplexer 00 derive clock from pll1_sw_clk 01 derive clock from pll2_sw_clk 10 derive clock from pll3_sw_clk(Default) 11 lp_apm clock
21 ASRC_CLK_ SEL	Selector for ASRC clock multiplexer 0 **ASRC_EXT_CLK** 1 pll4_sw_clk(Default)
20–19 ESAI_PRE_SEL	Selector for ESAI pre-divider clock multiplexer 00 derive clock from pll1_sw_clk 01 derive clock from pll2_sw_clk 10 derive clock from pll3_sw_clk(Default) 11 Reserved
18–16 ESAI_POST_ SEL[2:0]	Selector for ESAI clock multiplexer 000 derive clock from divided DPLL. 001 derive clock from ssi1 clock root (Default) 010 derive clock from ssi2 clock root 011 derive clock from ckih_camp1_clk. 100 derive clock from ckih_camp2_clk. 101 gnd 110 reserved 111 reserved
15–14 IEEE_CLK_SEL [1:0]	Selector for IPTP clock multiplexer 00 derive clock from pll3_sw_clk (Default) 01 derive clock from pll4_sw_clk 10 derive clock from usbphy2_clkout - from USB 11 fec_phy_clock

Table continues on the next page...

CCM_CSCMR2 field descriptions (continued)

Field	Description
13–12 FIRI_CLK_SEL [1:0]	Selector for FIRI clock multiplexer 00 derive clock from pll1_sw_clk 01 derive clock from pll2_sw_clk 10 derive clock from pll3_sw_clk(Default) 11 lp_apm clock
11 LDB_DI1_IPU_ DIV	Selector for division of LDB clock for IPU DI1 0 divide by 3.5 1 divide by 7(Default)
10 LDB_DI0_IPU_ DIV	Selector for division of LDB clock for IPU DI0 0 divide by 3.5 1 divide by 7(Default)
9 LDB_DI1_CLK_ SEL	Selector for LDB DI1 clock multiplexer 0 pll3_sw_clk 1 pll4_sw_clk(Default)
8 LDB_DI0_CLK_ SEL	Selector for LDB DI0 clock multiplexer 0 pll3_sw_clk 1 pll4_sw_clk(Default)
7–6 CAN_CLK_ SEL[1:0]	Selector for FLEXCAN clock multiplexer 00 derive clock from ipg_clk_root (default) 01 derive clock from CKIH pad input 10 derive clock from CKIH2 pad input 11 derive clock from lp_apm clock source
5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SPDIF0_COM	Controlls the multiplexer to communize spdif0 clock generation based on ssi1 clock root. 0 generate spdif0_clk_root from dividers 1 generate spdif0_clk_root from ssi1_clk_root.
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 SPDIF0_CLK_ SEL [1:0]	Selector for spdif0 clock multiplexer 00 derive clock from pll1_sw_clk 01 derive clock from pll2_sw_clk 10 derive clock from pll3_sw_clk(Default) 11 SPDIF_XTAL_CLK clock

18.3.10 CCM Serial Clock Divider Register 1 (CCM_CSCDR1)

The figure below represents the CCM Serial Clock Divider Register 1 (CCM_CSCDR1). The register contains bits to control the clock generation sub-block dividers.

NOTE

Any change to the above dividers has to be done while the affected block's clock is not functional and the affected clock is gated. If the change is done during operation of the block, it is not guaranteed that the blocks operation will not be harmed.

Address: CCM_CSCDR1 is 53FD_4000h base + 24h offset = 53FD_4024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								ESDHC3_CLK_PRED [2:0]		ESDHC3_CLK_PODF[2:0]			ESDHC1_CLK_PRED [2:0]		
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PGC_CLK_PODF[1:0]		ESDHC1_CLK_PODF[2:0]			USBOH3_CLK_PRED[2:0]			USBOH3_CLK_PODF[1:0]		UART_CLK_PRED[2:0]			UART_CLK_PODF[2:0]		
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0

CCM_CSCDR1 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved
24–22 ESDHC3_CLK_PRED [2:0]	Selector for ESDHCV3-3 clock pre-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 (Default) 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
21–19 ESDHC3_CLK_PODF[2:0]	Selector for ESDHCV3-3 clock post-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1(Default) 001 divide by 2 010 divide by 3

Table continues on the next page...

CCM_CSCDR1 field descriptions (continued)

Field	Description
	011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
18–16 ESDHC1_CLK_ PRED [2:0]	Selector for ESDHCV2-1 clock pre-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4(Default) 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
15–14 PGC_CLK_ PODF[1:0]	Selector for pgc (power gating controller) clock post-divider. **NOTE:** Divider should be updated when output clock is gated. 00 divide by 1(Default) 01 divide by 2 10 divide by 4 11 divide by 8
13–11 ESDHC1_CLK_ PODF[2:0]	Selector for ESDHCV2-1 clock post-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1(Default) 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
10–8 USBOH3_CLK_ PRED[2:0]	Selector for USB clock pre-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 (do not use with high input frequencies) 001 divide by 2 010 divide by 3 011 divide by 4(Default) 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8

Table continues on the next page...

CCM_CSCDR1 field descriptions (continued)

Field	Description
7–6 USBOH3_CLK_ PODF[1:0]	Selector for USB clock post-divider. **NOTE:** Divider should be updated when output clock is gated. 00 divide by 1(Default) 01 divide by 2 10 divide by 3 11 divide by 4
5–3 UART_CLK_ PRED[2:0]	Selector for UART clock pre-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4(Default) 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
2–0 UART_CLK_ PODF[2:0]	Selector for UART clock post-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1(Default) 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8

18.3.11 CCM SSI1 Clock Divider Register (CCM_CS1CDR)

The figure below represents the CCM SSI-1 Clock Divider Register (CCM_CS1CDR). The register contains bits to control the ssi1 clock generation dividers.

Address: CCM_CS1CDR is 53FD_4000h base + 28h offset = 53FD_4028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	ESAI_CLK_PODF [5:0]							SSI_EXT1_CLK_PRED[2:0]		SSI_EXT1_CLK_PODF [5:0]					
W																
Reset	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				ESAI_CLK_PRED[2:0]			SSI1_CLK_PRED[2:0]			SSI1_CLK_PODF [5:0]					
W																
Reset	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1

CCM_CS1CDR field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–25 ESAI_CLK_PODF [5:0]	Selector for ESAI clock post-divider. The input clock to this divider should be lower than 300Mhz, the pre-divider can be used to achieve this. 000000 divide by 1 000001 divide by 2(default) 111111 divide by 2^6
24–22 SSI_EXT1_CLK_PRED[2:0]	Selector for ssi_ext1 clock pre-divider. 000 divide by 1 (do not use with high input frequencies) 001 divide by 2 010 divide by 3(Default) 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
21–16 SSI_EXT1_CLK_PODF [5:0]	Selector for ssi_ext1 clock post-divider. The input clock to this divider should be lower than 300Mhz, the pre-divider can be used to achieve this. 000000 divide by 1 111111 divide by 2^6
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

CCM_CS1CDR field descriptions (continued)

Field	Description
11–9 ESAI_CLK_ PRED[2:0]	Selector for ESAI clock pre-divider. 000 divide by 1 (do not use with high input frequencies) 001 divide by 2(Default) 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
8–6 SSI1_CLK_ PRED[2:0]	Selector for SSI-1 clock pre-divider. 000 divide by 1 (do not use with high input frequencies) 001 divide by 2(Default) 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
5–0 SSI1_CLK_ PODF [5:0]	Selector for SSI-1 clock post-divider. The input clock to this divider should be lower than 300Mhz, the pre-divider can be used to achieve this. 000000 divide by 1 111111 divide by 2^6

18.3.12 CCM SSI2 Clock Divider Register (CCM_CS2CDR)

The figure below represents the CCM SSI2 Clock Divider Register (CCM_CS2CDR). The register contains bits to control the ssi2 clock generation dividers.

Address: CCM_CS2CDR is 53FD_4000h base + 2Ch offset = 53FD_402Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							SSI_EXT2_CLK_PRED[2:0]			SSI_EXT2_CLK_PODF [5:0]					
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							SSI2_CLK_PRED[2:0]			SSI2_CLK_PODF [5:0]					
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

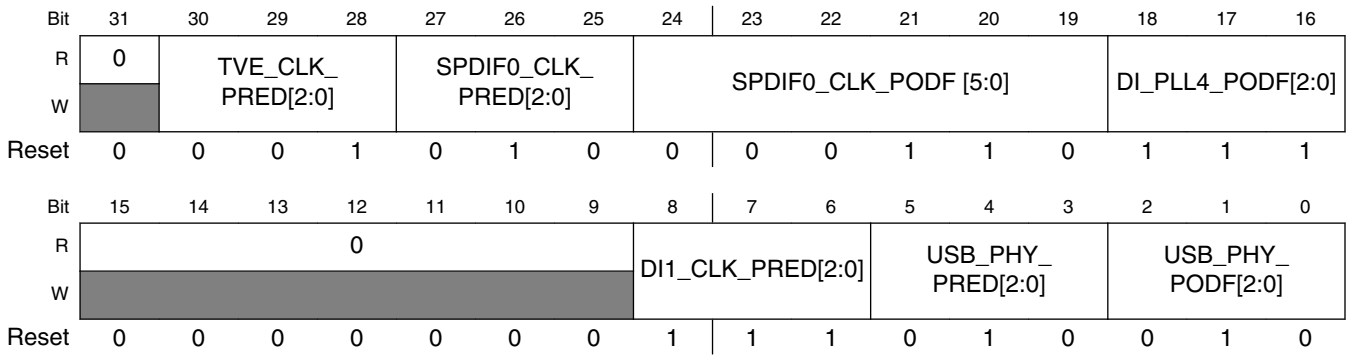
CCM_CS2CDR field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved
24–22 SSI_EXT2_ CLK_PRED[2:0]	Selector for ssi_ext2 clock pre-divider. 000 divide by 1 (do not use with high input frequencies) 001 divide by 2 010 divide by 3(Default) 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
21–16 SSI_EXT2_ CLK_PODF [5:0]	Selector for ssi_ext2 clock post-divider. **NOTE:** The input clock to this divider should be lower than 300Mhz, the pre-divider can be used to achieve this. 000000 divide by 1 111111 divide by 2^6
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8–6 SSI2_CLK_ PRED[2:0]	Selector for SSI-2 clock pre-divider. 000 divide by 1 (do not use with high input frequencies) 001 divide by 2(Default) 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
5–0 SSI2_CLK_ PODF [5:0]	Selector for SSI-2 clock post-divider. **NOTE:** The input clock to this divider should be lower than 300Mhz, the pre-divider can be used to achieve this. 000000 divide by 1 111111 divide by 2^6

18.3.13 CCM D1 Clock Divider Register (CCM_CDCDR)

The figure below represents the CCM DI Clock Divider Register (CCM_CDCDR). The register contains bits to control the DI clock, tve clock and usb_phy generation dividers.

Address: CCM_CDCDR is 53FD_4000h base + 30h offset = 53FD_4030h



CCM_CDCDR field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–28 TVE_CLK_ PRED[2:0]	Selector for TVE clock pre-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 (Default) 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
27–25 SPDIF0_CLK_ PRED[2:0]	Selector for spdif0 clock pre-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 (do not use with high input frequencies) 001 divide by 2 010 divide by 3(Default) 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
24–19 SPDIF0_CLK_ PODF [5:0]	Selector for spdif0 clock post-divider. **NOTE:** Divider should be updated when output clock is gated.

Table continues on the next page...

CCM_CDCDR field descriptions (continued)

Field	Description
	NOTE: The input clock to this divider should be lower than 300Mhz, the pre-divider can be used to achieve this. 000000 divide by 1 111111 divide by 2^6
18–16 DI_PLL4_ PODF[2:0]	Selector for di_pll4 clock pre-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8 (Default)
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8–6 DI1_CLK_ PRED[2:0]	Selector for DI clock pre-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8 (Default)
5–3 USB_PHY_ PRED[2:0]	Selector for usb_phy clock pre-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 (do not use with high input frequencies) 001 divide by 2 010 divide by 3(Default) 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
2–0 USB_PHY_ PODF[2:0]	Selector for usb_phy clock post-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 001 divide by 2

Table continues on the next page...

CCM_CDCDR field descriptions (continued)

Field	Description
	010 divide by 3(Default) 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8

18.3.14 CCM HSC Clock Divider Register (CCM_CHSCCDR)

The figure below represents the CCM HSC Clock Divider Register (CCM_CHSCCDR). The register contains bits to control the hsc clock generation dividers.

Address: CCM_CHSCCDR is 53FD_4000h base + 34h offset = 53FD_4034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												SSI2_MLB_SPDIF_SRC[1:0]		SSI1_MLB_SPDIF_SRC[1:0]	
W																
Reset	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0

CCM_CHSCCDR field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value zero. Reserved
3–2 SSI2_MLB_SPDIF_SRC[1:0]	Selector for SSI-2 root clock multiplexer 00 derive clock from SSI-2 clock divider (Default) 01 derive clock from mlb_mlbclk_in 10 derive clock from spdif_sclk 11 Restricted
1–0 SSI1_MLB_SPDIF_SRC[1:0]	Selector for SSI-1 root clock multiplexer 00 derive clock from SSI-1 clock divider (Default) 01 derive clock from mlb_mlbclk_in 10 derive clock from spdif_sclk 11 Restricted

18.3.15 CCM Serial Clock Divider Register 2 (CCM_CSCDR2)

The figure below represents the CCM Serial Clock Divider Register 2 (CCM_CSCDR2). The register contains bits to control the clock generation sub-block dividers.

Address: CCM_CSCDR2 is 53FD_4000h base + 38h offset = 53FD_4038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	ASRC_CLK_PRED[2:0]			ECSPI_CLK_PRED[2:0]			ECSPI_CLK_PODF [5:0]						0		
W																
Reset	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	ASRC_CLK_PODF[5:0]						IEEE_CLK_PRED[2:0]	IEEE_CLK_PODF[5:0]							
W																
Reset	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0

CCM_CSCDR2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–28 ASRC_CLK_PRED[2:0]	Selector for ASRC clock pre-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 (do not use with high input frequencies) 001 divide by 2(Default) 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
27–25 ECSPI_CLK_PRED[2:0]	Selector for ECSPI clock pre-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 (do not use with high input frequencies) 001 divide by 2(Default) 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
24–19 ECSPI_CLK_PODF [5:0]	Selector for ECSPI clock post-divider. **NOTE:** Divider should be updated when output clock is gated.

Table continues on the next page...

CCM_CSCDR2 field descriptions (continued)

Field	Description
	NOTE: The input clock to this divider should be lower than 300Mhz, the pre-divider can be used to achieve this. 000000 divide by 1 111111 divide by 2^6
18–15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–9 ASRC_CLK_ PODF[5:0]	Selector for ASRC clock post-divider. **NOTE:** Divider should be updated when output clock is gated. **NOTE:** The input clock to this divider should be lower than 300Mhz, the pre-divider can be used to achieve this. 000000 divide by 1 000001 divide by 2 000100 divide by 5 (Default) 111111 divide by 2^6
8–6 IEEE_CLK_ PRED[2:0]	Selector for IPTP clock pre-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 (do not use with high input frequencies) 001 divide by 2(Default) 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
5–0 IEEE_CLK_ PODF[5:0]	Selector for iptp clock post-divider. **NOTE:** Divider should be updated when output clock is gated. **NOTE:** The input clock to this divider should be lower than 300Mhz, the pre-divider can be used to achieve this. 000000 divide by 1 000001 divide by 2 000100 divide by 5 (Default) 111111 divide by 2^6

18.3.16 CCM Serial Clock Divider Register 3 (CCM_CSCDR3)

The figure below represents the CCM Serial Clock Divider Register 3 (CCM_CSCDR3). The register contains bits to control the clock generation sub-block dividers.

Address: CCM_CSCDR3 is 53FD_4000h base + 3Ch offset = 53FD_403Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								FIRI_CLK_PRED[2:0]		FIRI_CLK_PODF [5:0]					
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

CCM_CSCDR3 field descriptions

Field	Description
31–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8–6 FIRI_CLK_PRED[2:0]	Selector for FIRI clock pre-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 (do not use with high input frequencies) 001 divide by 2(Default) 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
5–0 FIRI_CLK_PODF [5:0]	Selector for FIRI clock post-divider. **NOTE:** Divider should be updated when output clock is gated. **NOTE:** The input clock to this divider should be lower than 300Mhz, the pre-divider can be used to achieve this. 000000 divide by 1 111111 divide by 2^6

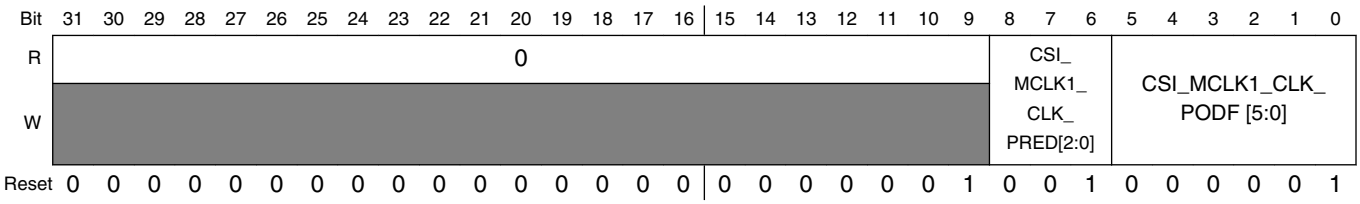
18.3.17 CCM Serial Clock Divider Register 4 (CCM_CSCDR4)

The figure below represents the CCM Serial Clock Divider Register 4 (CCM_CSCDR4). The register contains bits to control the clock generation sub-block dividers

NOTE

Any change to the serial clock dividers (all dividers in the registers CCM_CSCDR1, CCM_CECDR, CCM_CDCDR, CCM_CSCDR2) has to be done while the affected block's clock is not functional. If the change is done during operation of the block, it is not guaranteed that the block's operation will not be harmed.

Address: CCM_CSCDR4 is 53FD_4000h base + 40h offset = 53FD_4040h



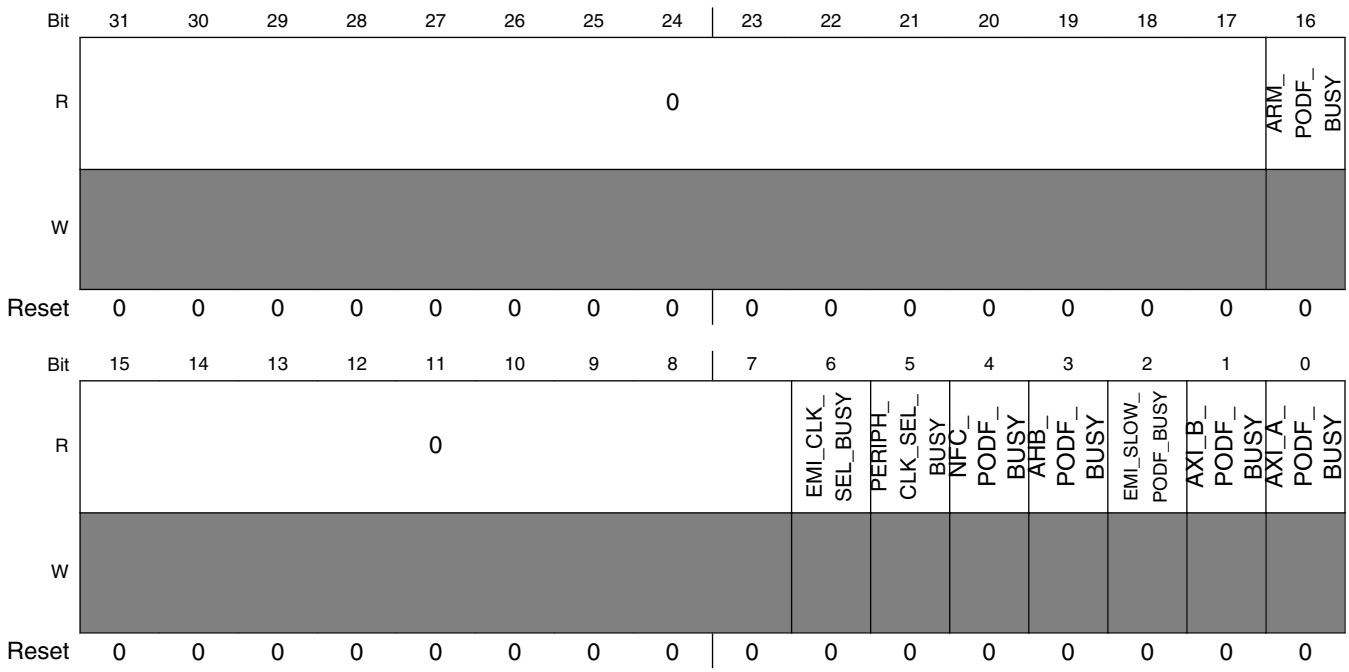
CCM_CSCDR4 field descriptions

Field	Description
31–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8–6 CSI_MCLK1_ CLK_PRED[2:0]	Selector for CSI mclk1 clock pre-divider. **NOTE:** Divider should be updated when output clock is gated. 000 divide by 1 (do not use with high input frequencies) 001 divide by 2 (Default) 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
5–0 CSI_MCLK1_ CLK_PODF [5:0]	Selector for CSI mclk1 clock post-divider. **NOTE:** Divider should be updated when output clock is gated. **NOTE:** The input clock to this divider should be lower than 300Mhz, the pre-divider can be used to achieve this. 000000 divide by 1 111111 divide by 2^6

18.3.18 CCM Divider Handshake In-Process Register (CCM_CDHIPR)

The figure below represents the CCM Divider Handshake In-Process Register (CCM_CDHIPR). The register contains read only bits that indicate if the CCM is in the process of updating dividers or multiplexers that might need handshake with blocks.

Address: CCM_CDHIPR is 53FD_4000h base + 48h offset = 53FD_4048h



CCM_CDHIPR field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value zero. Reserved
16 ARM_PODF_BUSY	Busy indicator for CCM_CACRR[ARM_PODF]. This bit corresponds to the ARM_PODF divider only if DVFSC operation is done through ARM_PODF change, that is, if CCM_CDCR[ARM_FREQ_SHIFT_DIVIDER] = '1'. If it is not a DVFSC operation, writes to ARM_PODF commence once the divider is written with the new value. **NOTE:** When busy, do not write to this bit. Any reads will return the next value that the divider holds and not the actual value. After the busy signal is de-asserted, CCM can generate an interrupt if it is not masked (see CCM Interrupt Mask Register (CCM_CIMR)) 0 divider is not busy and its value represents the actual division. 1 divider is busy with handshake process.
15–7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6 EMI_CLK_SEL_BUSY	Busy indicator for emi_clk_sel multiplexer control.

Table continues on the next page...

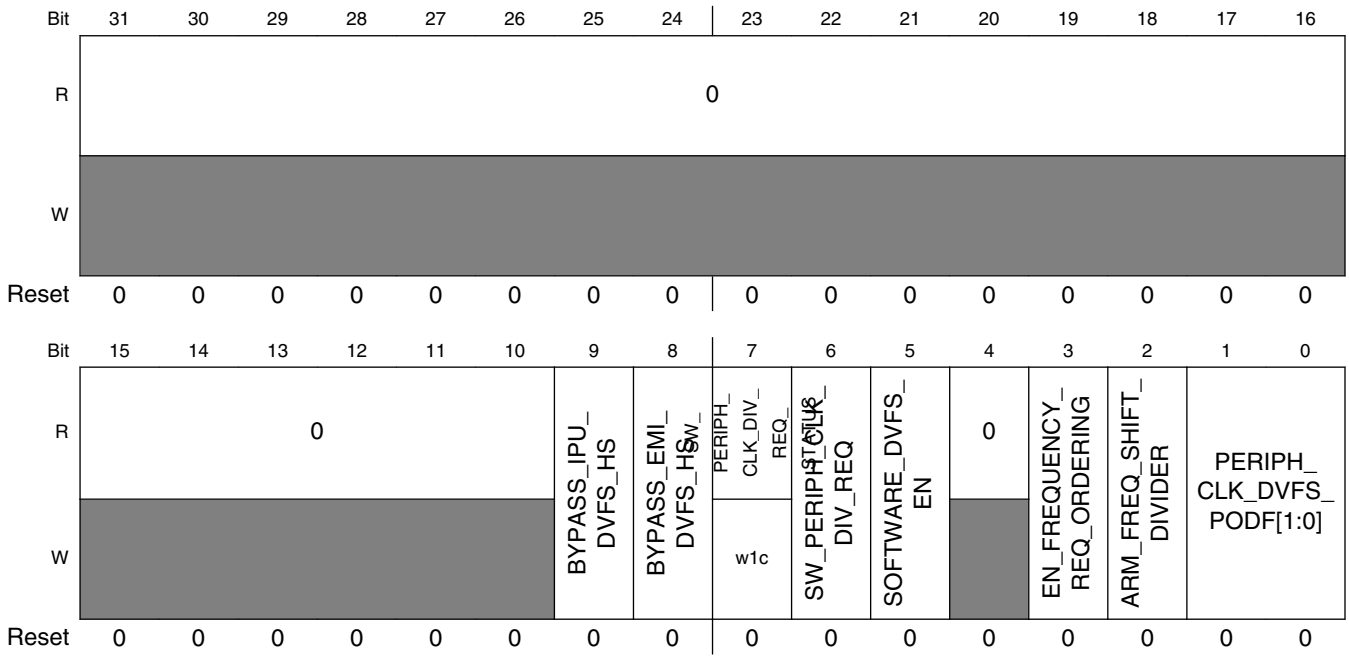
CCM_CDHIPR field descriptions (continued)

Field	Description
	NOTE: When busy, read to the multiplexer will return the next value it holds and not the actual value. After the busy signal is de-asserted, CCM can generate an interrupt if it is not masked (see CCM Module Enable Overide Register CCM Interrupt Mask Register) 0 Multiplexer is not busy and its value represents the actual division. 1 Multiplexer is busy in handshake process with EXTMC. The value read in the divider represents the next value that the divider holds after the handshake.
5 PERIPH_CLK_SEL_BUSY	Busy indicator for periph_clk_sel multiplexer control. **NOTE:** When busy, read to the multiplexer will return the next value it holds and not the actual value. After the busy signal is de-asserted, CCM can also generate an interrupt if it is not masked (see CCM Module Enable Overide Register CCM Interrupt Mask Register) 0 mux is not busy and its value represents the actual division. 1 mux is busy with handshake process with block. The value read in the divider represents the next value that the divider will hold after the handshake.
4 NFC_PODF_BUSY	Busy indicator for nfc_podf. 0 divider is not busy and its value represents the actual division. 1 divider is busy with handshake process with block. The value read in the divider represents the next value that the divider will hold after the handshake.
3 AHB_PODF_BUSY	Busy indicator for ahb_podf. 0 divider is not busy and its value represents the actual division. 1 divider is busy with handshake process with block. The value read in the divider represents the next value that the divider will hold after the handshake.
2 EMI_SLOW_PODF_BUSY	Busy indicator for emi_slow_podf. 0 divider is not busy and its value represents the actual division. 1 divider is busy with handshake process with block. The value read in the divider represents the next value that the divider will hold after the handshake.
1 AXI_B_PODF_BUSY	Busy indicator for axi_b_podf. 0 divider is not busy and its value represents the actual division. 1 divider is busy with handshake process with block. The value read in the divider represents the next value that the divider will hold after the handshake.
0 AXI_A_PODF_BUSY	Busy indicator for axi_a_podf. 0 divider is not busy and its value represents the actual division. 1 divider is busy with handshake process with block. The value read in the divider represents the next value that the divider will hold after the handshake.

18.3.19 CCM DVFS Control Register (CCM_CDCR)

The figure below represents the CCM DVFS Control Register (CCM_CDCR). The register contains bits to control the dvfs operation.

Address: CCM_CDCR is 53FD_4000h base + 4Ch offset = 53FD_404Ch



CCM_CDCR field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 BYPASS_IPU_DVFS_HS	Bypass handshake performed with IPU in case of DVFS frequency division. 1 In case of DVFS. bypass IPU handshake. CCM does not generate the request in this case, and does not wait for the acknowledge from IPU. 0 In case of DVFS, CCM performs handshake with IPU prior to the frequency division.
8 BYPASS_EMI_DVFS_HS	Bypass handshake performed with EXTMC in case of DVFS frequency division. 1 In case of DVFS. bypass EXTMC handshake. CCM does not generate the request in this case, and does not wait for the acknowledge from EXTMC. 0 In case of DVFS, CCM performs handshake with EXTMC prior to the frequency division.
7 SW_PERIPH_CLK_DIV_REQ_STATUS	When software control of DVFS is enabled, that is, SOFTWARE_DVFS_EN = '1' this bit indicates the status of the DVFS frequency switch operation. 1 DVFS frequency switch operation is completed. 0 DVFS frequency switch has not finished, or there is no frequency division request pending.

Table continues on the next page...

CCM_CDCR field descriptions (continued)

Field	Description
6 SW_PERIPH_ CLK_DIV_REQ	Start DVFS frequency division operation. This bit is operational only if SOFTWARE_DVFS_EN bit is set to '1'. 0 Disable DVFS divider operation - DVFS divider divides by '1'. 1 Enable DVFS divider operation. DVFS divider divides by the PERIPH_CLK_DVFS_PODF divider settings.
5 SOFTWARE_ DVFS_EN	Defines if the DVFS frequency division operation commences by software bit SOFTWARE_DVFS_EN or by GPC change frequency request signal. 0 DVFS frequency division operation is controlled by GPC change frequency request signal. 1 DVFS frequency division operation is controlled by software bit SW_PERIPHE_CLK_DIV_REQ.
4 Reserved	This read-only field is reserved and always has the value zero. Reserved
3 EN_ FREQUENCY_ REQ_ ORDERING	Defines if the frequency change request is sent to the EXTMC and to its master (IPU) together, or if CCM sends the request to EXTMC only after an acknowledge from the master (IPU) is received. This programing bit affects every clock change operation (DVFS operation, divider change operation, glitchless multiplexers change operation, etc.). 0 EXTMC request for frequency change is sent together with the request to EXTMC master (IPU). 1 EXTMC request for frequency change is sent after the acknowledge from master (IPU) is received.
2 ARM_FREQ_ SHIFT_DIVIDER	Defines if next DVFSC operation is through CCM_CACRR[ARM_PODF] change or DPLL relock. 0 Next DVFSC operation is done through DPLL relock. The pll relock process starts once GPC request an ARM frequenct change. (in this case any new writes to ARM_PODF commences immediately, without waiting for frequency change request from GPC). 1 Next DVFSC operation is done through ARM_PODF change. CCM holds updates to ARM_PODF until GPC requests for ARM frequency change.
1–0 PERIPH_CLK_ DVFS_ PODF[1:0]	Divider value for next DVFSP operation. This defines the value of the dividers affecting main_bus_clk. This divider takes place only during dvfs operation. Please refer to Peripheral Clock Domain Frequency Shift: for details. 00 divide by 1 (no frequency reduction during dvfs_per engagement, should be used for debug only) 01 divide by 2 (Default) 10 divide by 3 11 divide by 4

18.3.20 CCM Low Power Control Register (CCM_CLPCR)

The figure below represents the CCM Low Power Control Register (CCM_CLPCR). The register contains bits to control the low power modes operation.

Address: CCM_CLPCR is 53FD_4000h base + 54h offset = 53FD_4054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			BYPASS_CAN2_LPM_HS	BYPASS_CAN1_LPM_HS	BYPASS_SCC_LPM_HS	BYPASS_MAX_LPM_HS	BYPASS_SDMA_LPM_HS	BYPASS_EMI_INT2_LPM_HS	BYPASS_EMI_INT1_LPM_HS	BYPASS_EMI_SLOW_LPM_HS	BYPASS_EMI_FAST_LPM_HS	BYPASS_EMI_LPM_HS	BYPASS_IPU_LPM_HS	BYPASS_RTIC_LPM_HS	BYPASS_SAHARA_LPM_HS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				COSC_PWRDOWN	STBY_COUNT		VSTBY	DIS_REF_OSC	SBYOS	ARM_CLK_DIS_ON_LPM	LPSR_CLK_SEL		BYPASS_PMIC_VFUNCTIONAL_READY	LPM[1:0]	
W																
Reset	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1

CCM_CLPCR field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28 BYPASS_CAN2_LPM_HS	Bypass handshake with FLEXCAN-2 on next entrance to stop mode. **NOTE:** If bypass is enabled, CCM does not wait for an acknowledge from the block. 0 handshake with FLEXCAN-2 on next entrance to stop mode is performed. (default). 1 handshake with FLEXCAN-2 on next entrance to stop mode is bypassed.
27 BYPASS_CAN1_LPM_HS	Bypass handshake with FLEXCAN-1 on next entrance to stop mode. **NOTE:** If bypass is enabled, CCM does not wait for an acknowledge from the block. 0 handshake with FLEXCAN-1 on next entrance to stop mode is performed. (default). 1 handshake with FLEXCAN-1 on next entrance to stop mode is bypassed.
26 BYPASS_SCC_LPM_HS	Bypass handshake with scc on next entrance to stop mode. **NOTE:** If bypass is enabled, CCM does not wait for an acknowledge from the block. 0 handshake with SCC on next entrance to stop mode is performed. (default). 1 handshake with SCC on next entrance to stop mode is bypassed.
25 BYPASS_MAX_LPM_HS	Bypass handshake with max on next entrance to low power mode (wait or stop mode). **NOTE:** If bypass is enabled, CCM does not wait for an acknowledge from the block.

Table continues on the next page...

CCM_CLPCR field descriptions (continued)

Field	Description
	0 handshake with max on next entrance to low power mode is performed. (default). 1 handshake with max on next entrance to low power mode is bypassed.
24 BYPASS_ SDMA_LPM_HS	Bypass handshake with sdma on next entrance to low power mode (wait or stop mode). **NOTE:** If bypass is enabled, CCM does not wait for an acknowledge from the block. 0 handshake with sdma on next entrance to low power mode is performed. (default). 1 handshake with sdma on next entrance to low power mode is bypassed.
23 BYPASS_EMI_ INT2_LPM_HS	Bypass handshake with EXTMC int2 controller only on next entrance to low power mode (wait or stop mode). **NOTE:** If bypass is enabled, CCM does not wait for an acknowledge from the block. 0 handshake with EXTMC on next entrance to low power mode is performed. (default). 1 handshake with EXTMC on next entrance to low power mode is bypassed.
22 BYPASS_EMI_ INT1_LPM_HS	Bypass handshake with EXTMC int1 controller only on next entrance to low power mode (wait or stop mode). **NOTE:** If bypass is enabled, CCM does not wait for an acknowledge from the block. 0 handshake with EXTMC on next entrance to low power mode is performed. (default). 1 handshake with EXTMC on next entrance to low power mode is bypassed.
21 BYPASS_EMI_ SLOW_LPM_HS	Bypass handshake with EXTMC slow controller only on next entrance to low power mode (wait or stop mode). **NOTE:** If bypass is enabled, CCM does not wait for an acknowledge from the block. 0 handshake with EXTMC on next entrance to low power mode is performed. (default). 1 handshake with EXTMC on next entrance to low power mode is bypassed.
20 BYPASS_EMI_ FAST_LPM_HS	Bypass handshake with EXTMC fast controller only on next entrance to low power mode (wait or stop mode). **NOTE:** If bypass is enabled, CCM does not wait for an acknowledge from the block. 0 handshake with EXTMC on next entrance to low power mode is performed. (default). 1 handshake with EXTMC on next entrance to low power mode is bypassed.
19 BYPASS_EMI_ LPM_HS	Bypass handshake with EXTMC on next entrance to low power mode (wait or stop mode). This bit is used to bypass all the EXTMC handshakes or low power mode. **NOTE:** If bypass is enabled, CCM does not wait for an acknowledge from the block. 0 handshake with EXTMC on next entrance to low power mode is performed. (default). 1 handshake with EXTMC on next entrance to low power mode is bypassed.
18 BYPASS_IPU_ LPM_HS	Bypass handshake with IPU on next entrance to low power mode (wait or stop mode). **NOTE:** If bypass is enabled, CCM does not wait for an acknowledge from the block. 0 handshake with IPU on next entrance to low power mode is performed. (default). 1 handshake with IPU on next entrance to low power mode is bypassed.
17 BYPASS_RTIC_ LPM_HS	Bypass handshake with RTIC on next entrance to low power mode (wait or stop mode). **NOTE:** If bypass is enabled, CCM does not wait for an acknowledge from the block.

Table continues on the next page...

CCM_CLPCR field descriptions (continued)

Field	Description
	0 handshake with RTIC on next entrance to low power mode is performed. (default). 1 handshake with RTIC on next entrance to low power mode is bypassed.
16 BYPASS_ SAHARA_LPM_ HS	Bypass handshake with SAHARA on next entrance to low power mode (wait or stop mode). **NOTE:** If bypass is enabled, CCM does not wait for an acknowledge from the block. 0 handshake with SAHARA on next entrance to low power mode is performed. (default). 1 handshake with SAHARA on next entrance to low power mode is bypassed.
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11 COSC_ PWRDOWN	In run mode, on chip oscillator can be powered down by programming this bit to '1'. If the on chip oscillator is powered down, then stand-by clock oscillator (SBYOS) functionality for on chip oscillator is bypassed. The onchip oscillator should be powered down only in case the reference oscilator is not the source of all the clocks generation. 0 On chip oscillator is not be powered down.(default) 1 On chip oscillator is powered down.
10–9 STBY_COUNT	Standby counter. In the case of exit from STOP mode, (if VSTBY bit is set), STBY_COUNT defines the amount of time CCM waits between negation of PMIC_VSTBY_REQ and the check for assertion of PMIC_VFUNCIONAL_READY (input to i.MX53). 00 CCM waits 2 ckil clock cycles 01 CCM waits 4 ckil clock cycles 10 CCM waits 8 ckil clock cycles 11 CCM waits 16 ckil clock cycles
8 VSTBY	Voltage standby request bit. This bit defines if PMIC_VSTBY_REQ pin, which notifies external power management IC to change from functional voltage to standby voltage, is asserted in STOP mode. **NOTE:** When returning from STOP mode, the PMIC_VSTBY_REQ is deasserted (if it was asserted when entering STOP mode), and CCM waits for indication that functional voltage is ready (by sampling the assertion of PMIC_VFUNCIONAL_READY) before continuing the process of exiting from STOP mode. Please refer to STBY_COUNT bits (Bits 10-9). 0 voltage is not changed to standby voltage after next entrance to STOP mode. (PMIC_VSTBY_REQ remains negated - '0') 1 voltage is requested to change to standby voltage after next entrance to STOP mode. (PMIC_VSTBY_REQ is asserted - '1').
7 DIS_REF_OSC	In run mode, external reference oscillator clock can be disabled by generating '1' on CCM_CSR[REF_EN_B]. If the external reference clock is disabled by setting REF_EN_B to 1, then SBYOS functionality is bypassed. The external reference oscilator should be closed only in case the reference oscilator is not the source of any clock generation. 0 external high frequency oscillator is enabled, i.e. REF_EN_B = '0'.(default) 1 external high frequency oscillator is disabled, i.e. REF_EN_B = '1'
6 SBYOS	Standby clock oscillator. This bit defines if REF_EN_B pin, which disables external high frequency crystal, and COSC_PWRDOWN, which powers down the on chip oscillator, is asserted in STOP mode. This bit is discarded if DIS_REF_OSC = '1' for external oscillator, and if COSC_PWRDOWN='1' for the on chip oscillator.

Table continues on the next page...

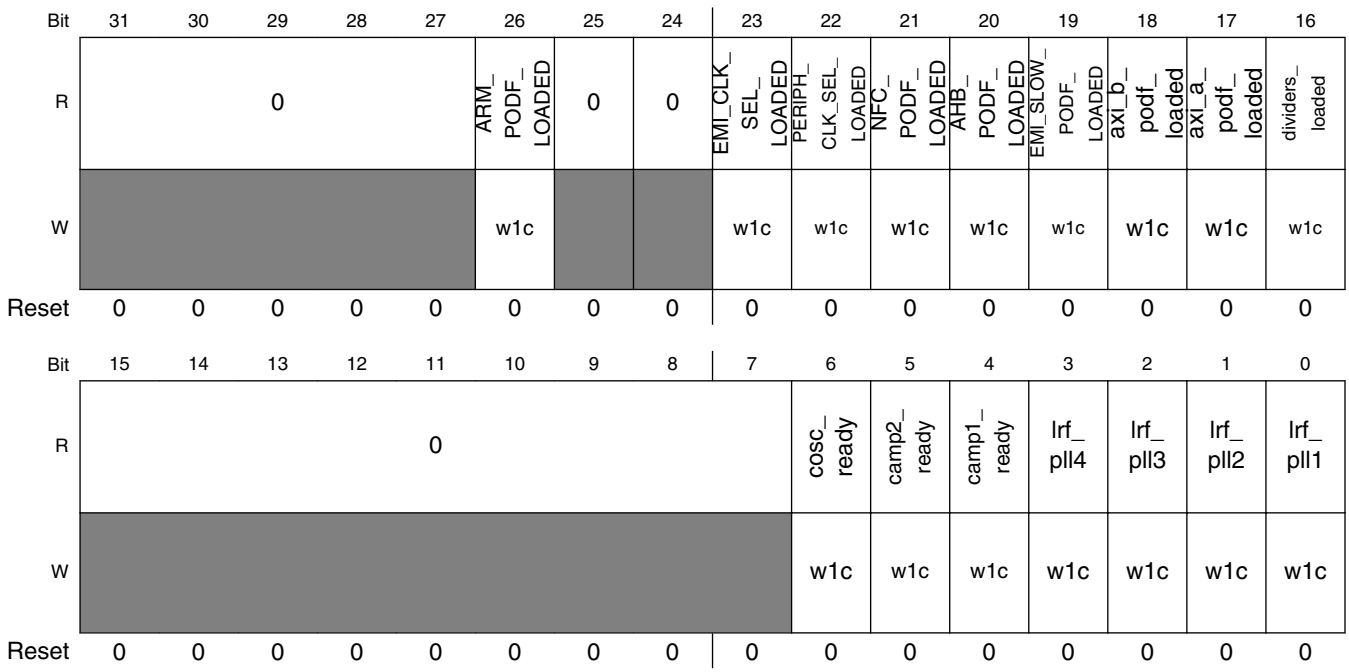
CCM_CLPCR field descriptions (continued)

Field	Description
	0 External high frequency oscillator is not disabled and on chip oscillator is not powered down, after next entrance to STOP mode. (REF_EN_B remains asserted - '0' and COSC_PWRDOWN remains de-asserted - '0') 1 External high frequency oscillator is disabled and on chip oscillator is powered down, after next entrance to STOP mode. (REF_EN_B is deasserted - '1' and COSC_PWRDOWN is asserted - '1'). (default). When returning from STOP mode, external oscillator is enabled again, on chip oscillator is returned to oscillator mode, and after OSCNT counts CCM continues with the exit from STOP mode process.
5 ARM_CLK_DIS_ON_LPM	Defines if ARM clocks are disabled in WAIT mode. This is useful in debug mode, when the user still wants to simulate entering WAIT mode and still keep the ARM clock functioning. **NOTE:** Software should not enable ARM power gating in wait mode if this bit is cleared. 0 ARM clock enabled in WAIT mode. 1 ARM clock disabled in WAIT mode. (default).
4–3 LPSR_CLK_SEL	Controlls the mux to select the lpsr_clk_root generation. Refer to CCM_CLK_ROOT_GEN and to PMIC Signal Description:, for details. 00 Reserved 01 generate clock from on chip oscillator clock. 10 generate clock from on chip oscillator clock/2 11 logical low ('0') - to be set when lpsr is not used, (for power consumption considerations) (default).
2 BYPASS_PMIC_VFUNCTIONAL_READY	By asserting this bit CCM bypasses waiting for pmic_vfunctional_ready signal when coming out of STOP mode. This should be used for PMIC's that don't support the pmic_vfunctional_ready signal. 0 Don't bypass the pmic_vfunctional_ready signal - CCM waits for it's assertion during exit from low power mode if standby voltage is enabled. 1 Bypass the pmic_vfunctional_ready signal - CCM does not wait for it's assertion during exit from low power mode if standby voltage is enabled.
1–0 LPM[1:0]	Setting the low power mode that system enters on next assertion of dsm_request signal. 00 Remain in RUN mode 01 Transfer to WAIT mode 10 Transfer to STOP mode 11 Reserved

18.3.21 CCM Interrupt Status Register (CCM_CISR)

The figure below represents the CCM Interrupt Status Register (CCM_CISR). This is a write one to clear register. Once a interrupt is generated, software should write one to clear it.

Address: CCM_CISR is 53FD_4000h base + 58h offset = 53FD_4058h



CCM_CISR field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26 ARM_PODF_LOADED	Interrupt ipi_int_1 (Interrupt #71) generated due to frequency change of ARM_PODF. The interrupt commences only if ARM_PODF is loaded during a DVFSC operation. 0 interrupt is not genrerated due to frequency change of ARM_PODF 1 interrupt is genrerated due to frequency change of ARM_PODF
25 Reserved	This read-only field is reserved and always has the value zero. Reserved
24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23 EMI_CLK_SEL_LOADED	Interrupt ipi_int_1 (Interrupt #71) generated due to update of EMI_CLK_SEL. 0 interrupt is not genrerated due to update of EMI_CLK_SEL. 1 interrupt is genrerated due to update of EMI_CLK_SEL.

Table continues on the next page...

CCM_CISR field descriptions (continued)

Field	Description
22 PERIPH_CLK_SEL_LOADED	Interrupt ipi_int_1 (Interrupt #71) generated due to update of PERIPH_CLK_SEL. 0 interrupt is not genrerated due to update of PERIPH_CLK_SEL. 1 interrupt is genrerated due to update of PERIPH_CLK_SEL.
21 NFC_PODF_LOADED	Interrupt ipi_int_1 (Interrupt #71) generated due to frequency change of NFC_PODF 0 interrupt is not genrerated due to frequency change of NFC_PODF 1 interrupt is genrerated due to frequency change of NFC_PODF
20 AHB_PODF_LOADED	Interrupt ipi_int_1 (Interrupt #71) generated due to frequency change of AHB_PODF 0 interrupt is not genrerated due to frequency change of AHB_PODF 1 interrupt is genrerated due to frequency change of AHB_PODF
19 EMI_SLOW_PODF_LOADED	Interrupt ipi_int_1 (Interrupt #71) generated due to frequency change of EMI_SLOW_PODF 0 interrupt is not genrerated due to frequency change of EMI_SLOW_PODF 1 interrupt is genrerated due to frequency change of EMI_SLOW_PODF
18 axi_b_podf_loaded	Interrupt ipi_int_1 (Interrupt #71) generated due to frequency change of AXI_B_PODF 0 interrupt is not genrerated due to frequency change of AXI_B_PODF 1 interrupt genrerated due to frequency change of AXI_B_PODF
17 axi_a_podf_loaded	Interrupt ipi_int_1 (Interrupt #71) generated due to frequency change of AXI_A_PODF 0 interrupt is not genrerated due to frequency change of AXI_A_PODF 1 interrupt genrerated due to frequency change of AXI_A_PODF
16 dividers_loaded	Interrupt ipi_int_1 (Interrupt #71) generated due to updated of AXI_A_PODF, AXI_B_PODF, EMI_SLOW_PODF, AHB_PODF, NFC_PODF, PERIPH_CLK_SEL, EMI_CLK_SEL. 0 interrupt is not genrerated due to updated of AXI_A_PODF, AXI_B_PODF, EMI_SLOW_PODF, AHB_PODF, NFC_PODF, PERIPH_CLK_SEL, EMI_CLK_SEL. 1 interrupt genrerated due to updated of AXI_A_PODF, AXI_B_PODF, EMI_SLOW_PODF, AHB_PODF, NFC_PODF, PERIPH_CLK_SEL, EMI_CLK_SEL.
15–7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6 cosc_ready	Interrupt ipi_int_2 (Interrupt #72) generated when on board oscillator is ready, i.e. OSCNT has finished counting. 0 interrupt is not genrerated due to on board oscillator ready 1 interrupt genrerated due to on board oscillator ready
5 camp2_ready	Interrupt ipi_int_2 (Interrupt #72) generated when CAMP-2 is ready, i.e. OSCNT has finished counting. 0 interrupt is not genrerated due to CAMP-2 ready 1 interrupt is genrerated due to CAMP-2 ready
4 camp1_ready	Interrupt ipi_int_2 (Interrupt #72) generated when CAMP-1 is ready, i.e. OSCNT has finished counting. 0 interrupt is not genrerated due to CAMP-1 ready 1 interrupt is genrerated due to CAMP-1 ready
3 lrf_pll4	Interrupt ipi_int_2 (Interrupt #72) is generated when DPLL-4 is locked

Table continues on the next page...

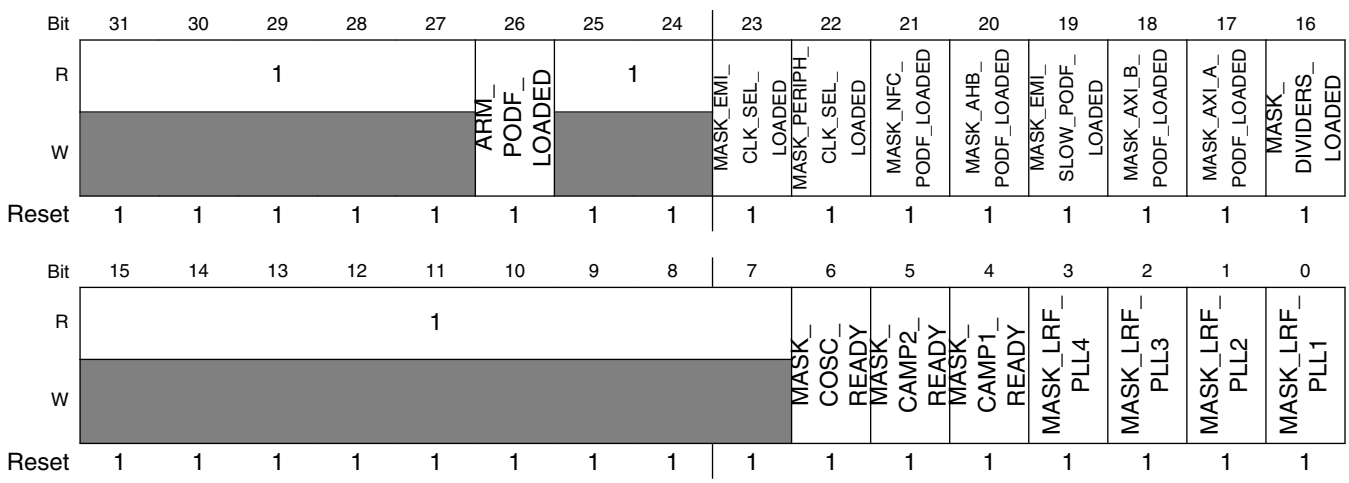
CCM_CISR field descriptions (continued)

Field	Description
	0 interrupt is not genrerated due to lock ready of DPLL-4 1 interrupt is genrerated due to lock ready of DPLL-4
2 lrf_pll3	Interrupt ipi_int_2 (Interrupt #72) generated when DPLL-3 is locked 0 interrupt is not genrerated due to lock ready of DPLL-3 1 interrupt is genrerated due to lock ready of DPLL-3
1 lrf_pll2	Interrupt ipi_int_2 (Interrupt #72) generated when DPLL-2 is locked 0 interrupt is not genrerated due to lock ready of DPLL-2 1 interrupt is genrerated due to lock ready of DPLL-2
0 lrf_pll1	Interrupt ipi_int_2 (Interrupt #72) generated when DPLL-1 is locked 0 interrupt is not genrerated due to lock ready of DPLL-1 1 interrupt is genrerated due to lock ready of DPLL-1

18.3.22 CCM Interrupt Mask Register (CCM_CIMR)

The figure below represents the CCM Interrupt Mask Register (CCM_CIMR).

Address: CCM_CIMR is 53FD_4000h base + 5Ch offset = 53FD_405Ch



CCM_CIMR field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value one. Reserved
26 ARM_PODF_LOADED	Mask interrupt generation due to frequency change of ARM_PODF 0 don't mask interrupt due to frequency change of ARM_PODF - interrupt is created 1 mask interrupt due to frequency change of ARM_PODF

Table continues on the next page...

CCM_CIMR field descriptions (continued)

Field	Description
25–24 Reserved	This read-only field is reserved and always has the value one. Reserved
23 MASK_EMI_CLK_SEL_LOADED	Mask interrupt generation due to update of EMI_CLK_SEL. 0 don't mask interrupt due to update of EMI_CLK_SEL - interrupt is created 1 mask interrupt due to update of EMI_CLK_SEL
22 MASK_PERIPH_CLK_SEL_LOADED	Mask interrupt generation due to update of PERIPH_CLK_SEL. 0 don't mask interrupt due to update of PERIPH_CLK_SEL - interrupt is created 1 mask interrupt due to update of PERIPH_CLK_SEL
21 MASK_NFC_PODF_LOADED	Mask interrupt generation due to frequency change of NFC_PODF 0 don't mask interrupt due to frequency change of NFC_PODF - interrupt is created 1 mask interrupt due to frequency change of NFC_PODF
20 MASK_AHB_PODF_LOADED	Mask interrupt generation due to frequency change of AHB_PODF 0 don't mask interrupt due to frequency change of AHB_PODF - interrupt is created 1 mask interrupt due to frequency change of AHB_PODF
19 MASK_EMI_SLOW_PODF_LOADED	Mask interrupt generation due to frequency change of EMI_SLOW_PODF 0 don't mask interrupt due to frequency change of EMI_SLOW_PODF - interrupt is created 1 mask interrupt due to frequency change of EMI_SLOW_PODF
18 MASK_AXI_B_PODF_LOADED	Mask interrupt generation due to frequency change of AXI_B_PODF 0 don't mask interrupt due to frequency change of AXI_B_PODF - interrupt is created 1 mask interrupt due to frequency change of AXI_B_PODF
17 MASK_AXI_A_PODF_LOADED	Mask interrupt generation due to frequency change of AXI_A_PODF 0 don't mask interrupt due to frequency change of AXI_A_PODF - interrupt is created 1 mask interrupt due to frequency change of AXI_A_PODF
16 MASK_DIVIDERS_LOADED	Mask interrupt generation due to updated in any one of: AXI_A_PODF, AXI_B_PODF, EMI_SLOW_PODF, AHB_PODF, NFC_PODF, PERIPH_CLK_SEL, EMI_CLK_SEL. 0 don't mask interrupt due to updated in any one of: AXI_A_PODF, AXI_B_PODF, EMI_SLOW_PODF, AHB_PODF, NFC_PODF, PERIPH_CLK_SEL, EMI_CLK_SEL. 1 mask interrupt due to updated in any one of: AXI_A_PODF, AXI_B_PODF, EMI_SLOW_PODF, AHB_PODF, NFC_PODF, PERIPH_CLK_SEL, EMI_CLK_SEL.
15–7 Reserved	This read-only field is reserved and always has the value one. Reserved
6 MASK_COSC_READY	Mask interrupt generation when on board oscillator is ready 0 don't mask interrupt due to on board oscillator ready - interrupt is created 1 mask interrupt due to on board oscillator ready
5 MASK_CAMP2_READY	Mask interrupt generation when CAMP-2 is ready 0 don't mask interrupt when CAMP-2 is ready - interrupt is created 1 mask interrupt when CAMP-2 is ready

Table continues on the next page...

CCM_CIMR field descriptions (continued)

Field	Description
4 MASK_CAMP1_READY	Mask interrupt generation when CAMP-1 is ready 0 don't mask interrupt when CAMP-1 is ready - interrupt is created 1 mask interrupt when CAMP-1 is ready
3 MASK_LRF_PLL4	Mask interrupt generation when DPLL-4 is locked 0 don't mask interrupt due to lock ready of DPLL-4 - interrupt is created 1 mask interrupt due to lock ready of DPLL-4
2 MASK_LRF_PLL3	Mask interrupt generation when DPLL-3 is locked 0 don't mask interrupt due to lock ready of DPLL-3 - interrupt is created 1 mask interrupt due to lock ready of DPLL-3
1 MASK_LRF_PLL2	Mask interrupt generation when DPLL-2 is locked 0 don't mask interrupt due to lock ready of DPLL-2 - interrupt is created 1 mask interrupt due to lock ready of DPLL-2
0 MASK_LRF_PLL1	Mask interrupt generation when DPLL-1 is locked 0 don't mask interrupt due to lock ready of DPLL-1 - interrupt is created 1 mask interrupt due to lock ready of DPLL-1

18.3.23 CCM Clock Output Source Register (CCM_CCOSR)

The figure below represents the CCM Clock Output Source Register (CCM_CCOSR). The register contains bits to control the clocks that are generated on the output CLKO and CLKO2.

Address: CCM_CCOSR is 53FD_4000h base + 60h offset = 53FD_4060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							CKO2_EN	CKO2_DIV[2:0]			CKO2_SEL[4:0]				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								cko1_en	cko1_div[2:0]			cko1_sel[3:0]			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

CCM_CCOSR field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved
24 CKO2_EN	Enable for CLKO2 clock 0 CLKO2 disabled. 1 CLKO2 enabled.
23–21 CKO2_DIV[2:0]	Setting the divider of CLKO2 000 divide by 1(Default) 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
20–16 CKO2_SEL[4:0]	Selection of the clock to be generated on cko2 00000 dptc_core (this is the ccm input dptc_core1_clk_clko generated to the cko2 pad). 00001 dptc_periph (this is the ccm input dptc_peripheral1_clk_clko generated to the cko2 pad). 00010 Reserved 00011 ESDHC1_CLK_ROOT 00100 USBOH3_CLK_ROOT 00101 WRCK_CLK_ROOT 00110 ECSPI_CLK_ROOT 00111 pll1_ref_clk 01000 ESDHC3_CLK_ROOT 01001 DDR_CLK_ROOT 01010 ARM_AXI_CLK_ROOT (Default) 01011 Output clock from USB PHY 01100 VPU_RCLK_ROOT 01101 IPU_HSP_CLK_ROOT 01110 osc_clk 01111 ckih_camp1_clk 10000 Reserved 10001 ESDHC2_CLK_ROOT 10010 SSI1_CLK_ROOT 10011 SSI2_CLK_ROOT 10100 Reserved 10101 Reserved 10110 LPSR_CLK_ROOT 10111 PGC_CLK_ROOT 11000 output clock generated by TVE 11001 USB_PHY_CLK_ROOT 11010 TVE_216_54_CLK_ROOT 11011 lp_apm clock 11100 UART_CLK_ROOT

Table continues on the next page...

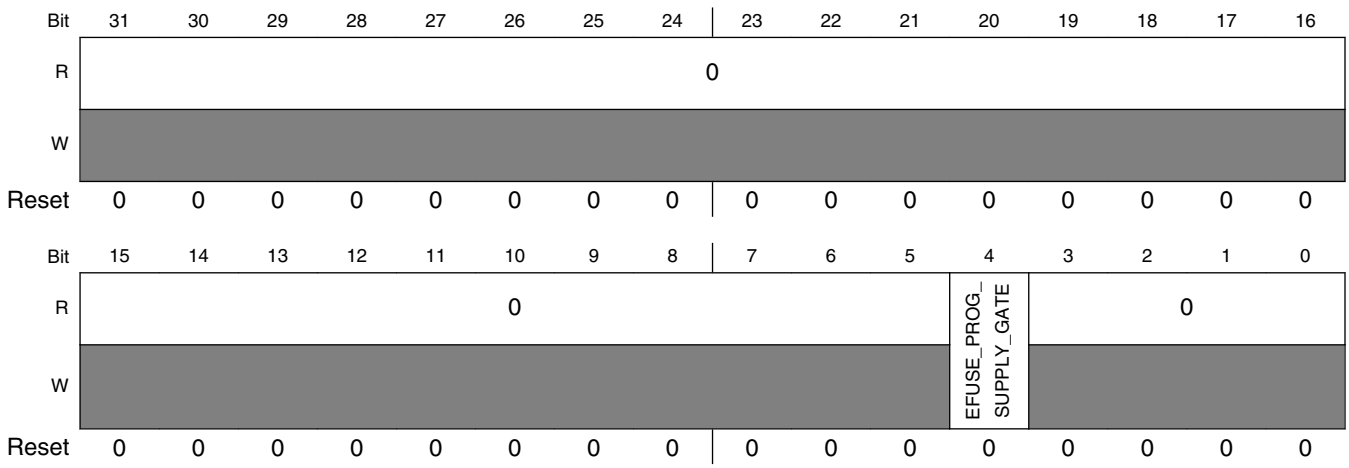
CCM_CCOSR field descriptions (continued)

Field	Description
	11101 SPDIF0_CLK_ROOT 11110 Reserved 11111 Reserved.
15–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7 cko1_en	Enable of CLKO1 clock 0 CLKO1 disabled. 1 CLKO1 enabled.
6–4 cko1_div[2:0]	Setting the divider of CLKO1 000 divide by 1(Default) 001 divide by 2 010 divide by 3 011 divide by 4 100 divide by 5 101 divide by 6 110 divide by 7 111 divide by 8
3–0 cko1_sel[3:0]	Selection of the clock to be generated on cko1 0000 ARM_CLK_ROOT 0001 pll1_sw_clk (Default) 0010 pll2_sw_clk 0011 pll3_sw_clk 0100 EMI_SLOW_CLK_ROOT 0101 pll4_sw_clk 0110 ENFC_CLK_ROOT 0111 Reserved 1000 DI_CLK_ROOT 1001 Reserved 1010 Reserved 1011 AHB_CLK_ROOT 1100 IPG_CLK_ROOT 1101 PERCLK_ROOT 1110 CKIL_SYNC_CLK_ROOT 1111 reserved

18.3.24 CCM General Purpose Register (CCM_CGPR)

The figure below represents the CCM General Purpose Register (CCM_CGPR). This is a regular read/write implemented register, which can serve for future possible usage. Few bits are in use, the other bits are free to be used in future.

Address: CCM_CGPR is 53FD_4000h base + 64h offset = 53FD_4064h



CCM_CGPR field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved.
4 EFUSE_PROG_SUPPLY_GATE	Gating of supply control for efuse programing 0 fuse programing supply voltage is gated off to the FUSEBOX 1 allow fuse programing.
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

18.3.25 CCM Clock Gating Register 0 (CCM_CCGR0)

The figure below represents the CCM Clock Gating Register 0 (CCM_CCGR0). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is according to the number of peripherals in the system.

Address: CCM_CCGR0 is 53FD_4000h base + 68h offset = 53FD_4068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8		CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR0 field descriptions

Field	Description
31–30 CG15	iim clocks: Also affects fusebox clocks (iim_clk_en)
29–28 CG14	ahb_max clocks (ahb_max_clk_enable)
27–26 CG13	aips_tz2 clocks (aips_tz2_clk_enable)
25–24 CG12	aips_tz1 clocks (aips_tz1_clk_enable)
23–22 CG11	rom clocks (rom_clk_enable)
21–20 CG10	romcp clocks (romcp_clk_enable)
19–18 CG9	ahbmux2 clocks (ahbmux2_clk_enable)
17–16 CG8	ahbmux1 clocks (ahbmux1_clk_enable)
15–14 CG7	cti3 clocks (cti3_clk_enable)
13–12 CG6	cti2 clocks (cti2_clk_enable)
11–10 CG5	tpiu clocks (tpiu_clk_enable)
9–8 CG4	dap clocks (dap_clk_enable)
7–6 CG3	tzic: to control TZIC clocks (tzic_clk_enable)
5–4 CG2	arm_debug: to control pclk and atclk busses of ARM (arm_debug_clk_enable)

Table continues on the next page...

CCM_CCGR0 field descriptions (continued)

Field	Description
3–2 CG1	arm_axi: to control arm_axi input clock (arm_axi_clk_enable)
1–0 CG0	arm_bus: to control ipg bus of ARM (arm_bus_clk_enable)

18.3.26 CCM Clock Gating Register 1 (CCM_CCGR1)

The figure below represents the CCM Clock Gating Register 1 (CCM_CCGR1). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is according to the number of peripherals in the system.

Address: CCM_CCGR1 is 53FD_4000h base + 6Ch offset = 53FD_406Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8		CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR1 field descriptions

Field	Description
31–30 CG15	SCC clocks: affects sccv2_clk_enable. it is not possible to close SCC clocks during RUN mode. Writing '00' to this CCM_CGR register does not affect the clocks of SCC. Closing the clocks of SCC eventually blocks access to the internal memory. In WAIT mode the SCC clocks can be closed only if it is guaranteed that no master is accessing the internal memory during the time spend in WAIT mode. The SCC clocks are closed in WAIT mode only if SAHARA clocks were closed as well. During WAIT mode, if SCC asserts either of it's clock enable signals, CCM enables the SCC clocks. The WAIT mode is not exited due to this assertion.
29–28 CG14	firi serial clock (firi_serial_clk_enable)
27–26 CG13	firi ipg clock (firi_clk_enable)
25–24 CG12	Reserved
23–22 CG11	i2c3 clocks (i2c3_serial_clk_enable)
21–20 CG10	i2c2 clocks (i2c2_serial_clk_enable)

Table continues on the next page...

CCM_CCGR1 field descriptions (continued)

Field	Description
19–18 CG9	i2c1 clocks (i2c1_serial_clk_enable)
17–16 CG8	uart3_perclk: affects ipg_perclk input to UART-3 (uart3_serial_clk_enable)
15–14 CG7	uart3_ipg_clk: affects ipg_clk input to UART-3 (uart3_clk_enable)
13–12 CG6	uart2_perclk: affects ipg_perclk input to UART-2 (uart2_serial_clk_enable)
11–10 CG5	uart2_ipg_clk: affects ipg_clk input to UART-2 (uart2_clk_enable)
9–8 CG4	uart1_perclk:affects ipg_perclk input to UART-1 (uart1_serial_clk_enable)
7–6 CG3	uart1_ipg_clk: affects ipg_clk input to UART-1 (uart1_clk_enable)
5–4 CG2	tmax3 clocks (tmax3_clk_enable)
3–2 CG1	tmax3 clocks (tmax2_clk_enable)
1–0 CG0	tmax1 clocks (tmax1_clk_enable)

18.3.27 CCM Clock Gating Register 2 (CCM_CCGR2)

The figure below represents the CCM Clock Gating Register 2 (CCM_CCGR2). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is according to the number of peripherals in the system.

Address: CCM_CCGR2 is 53FD_4000h base + 70h offset = 53FD_4070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8		CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR2 field descriptions

Field	Description
31–30 CG15	tve clock (tve_clk_enable)
29–28 CG14	usboh3_60M: affects ipg_clk_60Mhz input of USB (usboh3_serial_clk_enable)

Table continues on the next page...

CCM_CCGR2 field descriptions (continued)

Field	Description
27–26 CG13	usboh3_ipg_ahb: affects ipg_clk and ipg_ahb_clk inputs of USB (usboh3_clk_enable)
25–24 CG12	fec clocks: affects also csync_fec (fec_clk_enable)
23–22 CG11	owire clocks (owire_serial_clk_enable)
21–20 CG10	gpt_highfreq: affects ipg_clk_highfreq input of GPT (gpt_serial_clk_enable)
19–18 CG9	gpt_ipg_clk: affects ipg_clk input of GPT (gpt_clk_enable)
17–16 CG8	pwm2_highfreq: affects ipg_clk_highfreq input of PWM-2 (pwm2_serial_clk_enable)
15–14 CG7	pwm2_ipg_clk: affects ipg_clk input of PWM-2 (pwm2_clk_enable)
13–12 CG6	pwm1_highfreq: affects ipg_clk_highfreq input of PWM-1 (pwm1_serial_clk_enable)
11–10 CG5	pwm1_ipg_clk: affects ipg_clk input of PWM-1 (pwm1_clk_enable)
9–8 CG4	epit2_highfreq: affects ipg_clk_highfreq input of EPIT-2 (epit2_serial_clk_enable)
7–6 CG3	epit2_ipg_clk: affects ipg_clk input of EPIT-2 (epit2_clk_enable)
5–4 CG2	epit1_highfreq: affects ipg_clk_highfreq input of EPIT-1 (epit1_serial_clk_enable)
3–2 CG1	epit1_ipg_clk: affects ipg_clk input of EPIT-1 (epit1_clk_enable)
1–0 CG0	reserved

18.3.28 CCM Clock Gating Register 3 (CCM_CCGR3)

The figure below represents the CCM Clock Gating Register 3 (CCM_CCGR3). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is according to the number of peripherals in the system.

Address: CCM_CCGR3 is 53FD_4000h base + 74h offset = 53FD_4074h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8		CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR3 field descriptions

Field	Description
31–30 CG15	ssi_ext2 (ssi_ext2_clk_enable)
29–28 CG14	ssi_ext1 (ssi_ext1_clk_enable)
27–26 CG13	ssi3_ssi_clk: affects ccm_ssi_clk input of SSI-3 (ssi3_serial_clk_enable)
25–24 CG12	ssi3_ipg: affects ipg_clk input of SSI-3 (ssi3_clk_enable)
23–22 CG11	ssi2_ssi_clk: affects ccm_ssi_clk input of SSI-2 (ssi2_serial_clk_enable)
21–20 CG10	ssi2_ipg: affects ipg_clk input of SSI-2 (ssi2_clk_enable)
19–18 CG9	ssi1_ssi_clk: affects ccm_ssi_clk input of SSI-1 (ssi1_serial_clk_enable)
17–16 CG8	ssi1_ipg: affects ipg_clk input of SSI-1 (ssi1_clk_enable)
15–14 CG7	esdhc4_perclk: affects ipg_clk_perclk input of ESDEHC-4 (esdhc4_serial_clk_enable)
13–12 CG6	esdhc4_ipg_hclk: affects ipg_clk and hclk inputs of ESDEHC-4 (esdhc4_clk_enable)
11–10 CG5	esdhc3_perclk: affects ipg_clk_perclk input of ESDEHC-3 (esdhc3_serial_clk_enable)
9–8 CG4	esdhc3_ipg_hclk: affects ipg_clk and hclk inputs of ESDEHC-3 (esdhc3_clk_enable)
7–6 CG3	esdhc2_perclk: affects ipg_clk_perclk input of ESDEHC-2 (esdhc2_serial_clk_enable)
5–4 CG2	esdhc2_ipg_hclk: affects ipg_clk and hclk inputs of ESDEHC-2 (esdhc2_clk_enable)
3–2 CG1	esdhc1_perclk:affects ipg_clk_perclk input of ESDEHC-1 (esdhc1_serial_clk_enable)
1–0 CG0	esdhc1_ipg_hclk: affects ipg_clk and hclk inputs of ESDEHC-1 (esdhc1_clk_enable)

18.3.29 CCM Clock Gating Register 4 (CCM_CCGR4)

The figure below represents the CCM Clock Gating Register 4 (CCM_CCGR4). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is according to the number of peripherals in the system.

Address: CCM_CCGR4 is 53FD_4000h base + 78h offset = 53FD_4078h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8		CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR4 field descriptions

Field	Description
31–30 CG15	sdma clocks (sdma_clk_enable)
29–28 CG14	srtc clocks (srtc_clk_enable)
27–26 CG13	cspi_ipg: affects ipg_clk input of CSPI (cspi_clk_enable)
25–24 CG12	ecspi2_perclk: affects ipg_clk_per input of ECSPI-2 (ecspi2_serial_clk_enable)
23–22 CG11	ecspi2_ipg: affects ipg_clk input of ECSPI-2 (ecspi2_clk_enable)
21–20 CG10	ecspi1_perclk: affects ipg_clk_per input of ECSPI-1 (ecspi1_serial_clk_enable)
19–18 CG9	ecspi1_ipg: affects ipg_clk input of ECSPI-1 (ecspi1_clk_enable)
17–16 CG8	rtic clocks: it is not possible to close RTIC clocks during RUN mode. Writing '00' to this CCGR register does not affect the clocks of RTIC. (rtic_clk_enable)
15–14 CG7	sahara (sahara_clk_enable)
13–12 CG6	usb ph2 clock (usb_phy2_clk_enable)
11–10 CG5	usb phy1 clock (usb_phy1_clk_enable)
9–8 CG4	can2_serial: affects ipg clock cpi domain of FLEXCAN-2 (can2_serial_clk_enable)
7–6 CG3	can2_ipg: affects ipg_clk and ipg_clk_mbm inputs to FLEXCAN-2 (can2_clk_enable)
5–4 CG2	Reserved
3–2 CG1	sata (sata_clk_enable)
1–0 CG0	pata (pata_clk_enable)

18.3.30 CCM Clock Gating Register 5 (CCM_CCGR5)

The figure below represents the CCM Clock Gating Register 5 (CCM_CCGR5). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is according to the number of peripherals in the system.

Address: CCM_CCGR5 is 53FD_4000h base + 7Ch offset = 53FD_407Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8		CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR5 field descriptions

Field	Description
31–30 CG15	spdif ipg clock (spdif_clk_enable)
29–28 CG14	Reserved
27–26 CG13	spdif0 clock (spdif0_clk_enable)
25–24 CG12	gpc ipg clock (gpc_clk_enable)
23–22 CG11	emi wrck clock: affects the WRCK clock of EXTMC (emi_clk_enable)
21–20 CG10	emi_enfc: affects only NFC clock of EXTMC (emi_enfc_clk_enable)
19–18 CG9	emi_int1: affects only int1 clock of EXTMC. As accesses to any configuration register in the system (for any given block) passes through the EXTMC internal channel logic, gating the emi_int1 clock will cause the system to hang. Clearing bits [19:18] is not recommended. (emi_int1_clk_enable)
17–16 CG8	emi_slow: affects only slow clock of EXTMC (emi_slow_clk_enable)
15–14 CG7	emi_fast: affects only fast clock of EXTMC (emi_fast_clk_enable)
13–12 CG6	ipmux1 (ipmux1_clk_enable)
11–10 CG5	ipu clocks (ipu_clk_enable)
9–8 CG4	vpu reference clock (vpu_serial_clk_enable)

Table continues on the next page...

CCM_CCGR5 field descriptions (continued)

Field	Description
7–6 CG3	vpu clocks (vpu_clk_enable)
5–4 CG2	garb clocks (garb_clk_enable)
3–2 CG1	gpu clocks (gpu_clk_enable)
1–0 CG0	spba clocks (spba_clk_enable)

18.3.31 CCM Clock Gating Register 6 (CCM_CCGR6)

The figure below represents the CCM Clock Gating Register 6 (CCM_CCGR6). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is according to the number of peripherals in the system.

Address: CCM_CCGR6 is 53FD_4000h base + 80h offset = 53FD_4080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8		CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR6 field descriptions

Field	Description
31–30 CG15	ldb_di1 (ldb_di1_clk_enable)
29–28 CG14	ldb_di0 (ldb_di0_clk_enable)
27–26 CG13	pl301_2x2
25–24 CG12	pl301_4x1
23–22 CG11	can1_serial: affects ipg clock cpi domain of FLEXCAN-1 (can1_serial_clk_enable)
21–20 CG10	can1_ipg: affects ipg_clk and ipg_clk_mbm inputs to FLEXCAN-1 (can1_clk_enable)
19–18 CG9	easi_root (easi_root_clk_enable)
17–16 CG8	esai_ipg (esai_clk_enable)

Table continues on the next page...

CCM_CCGR6 field descriptions (continued)

Field	Description
15–14 CG7	gpu2d clock (gpu2d_clk_enable)
13–12 CG6	ipu di1 clock (ipu_di1_clk_enable)
11–10 CG5	ipu di0 clock (ipu_di0_clk_enable)
9–8 CG4	emi_int2: affects only int2 clock to EXTMC (emi_int2_clk_enable)
7–6 CG3	Reserved
5–4 CG2	CSI mclk1
3–2 CG1	ocram (ocram_clk_enable)
1–0 CG0	ipmux2 (ipmux2_clk_enable)

18.3.32 CCM Clock Gating Register 7 (CCM_CCGR7)

The figure below represents the CCM Clock Gating Register 7 (CCM_CCGR7). The clock gating Registers define the clock gating for power reduction of each clock (CG(i) bits). There are 8 CGR registers. The number of registers required is according to the number of peripherals in the system.

CG(i) bits CCGR0-7

NOTE

This bits are used to turn on/off the clock to each block independently.The following table details the possible clock activity conditions for each block

Table 18-40. CG Bit Description

CGR value	Clock Activity Description
00	clock is off during all modes. stop enter hardware handshake is disabled.
01	clock is on in run mode, but off in wait and stop modes
10	Not applicable (Reserved).
11	clock is on during all modes, except stop mode.

NOTE

Blocks should be stopped before it's bits are set to "0", as clocks to the block are stopped immediately.

NOTE

The tables above show the register mappings for the different Clock Gating Registers. The clock connectivity table should be used to match the signal mentioned to the actual clocks going into the blocks.

Address: CCM_CCGR7 is 53FD_4000h base + 84h offset = 53FD_4084h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8		CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CCGR7 field descriptions

Field	Description
31–30 CG15	Reserved
29–28 CG14	Reserved
27–26 CG13	Reserved
25–24 CG12	Reserved
23–22 CG11	Reserved
21–20 CG10	Reserved
19–18 CG9	Reserved
17–16 CG8	Reserved
15–14 CG7	uart5_perclk: affects ipg_perclk input to UART-5 (uart5_serial_clk_enable)
13–12 CG6	uart5_ipg_clk: affects ipg_clk input to UART-5 (uart5_clk_enable)
11–10 CG5	uart4_perclk: affects ipg_perclk input to UART-4 (uart4_serial_clk_enable)
9–8 CG4	uart4_ipg_clk: affects ipg_clk input to UART-4 (uart4_clk_enable)
7–6 CG3	ieee1588 (ieee1588_clk_enable)

Table continues on the next page...

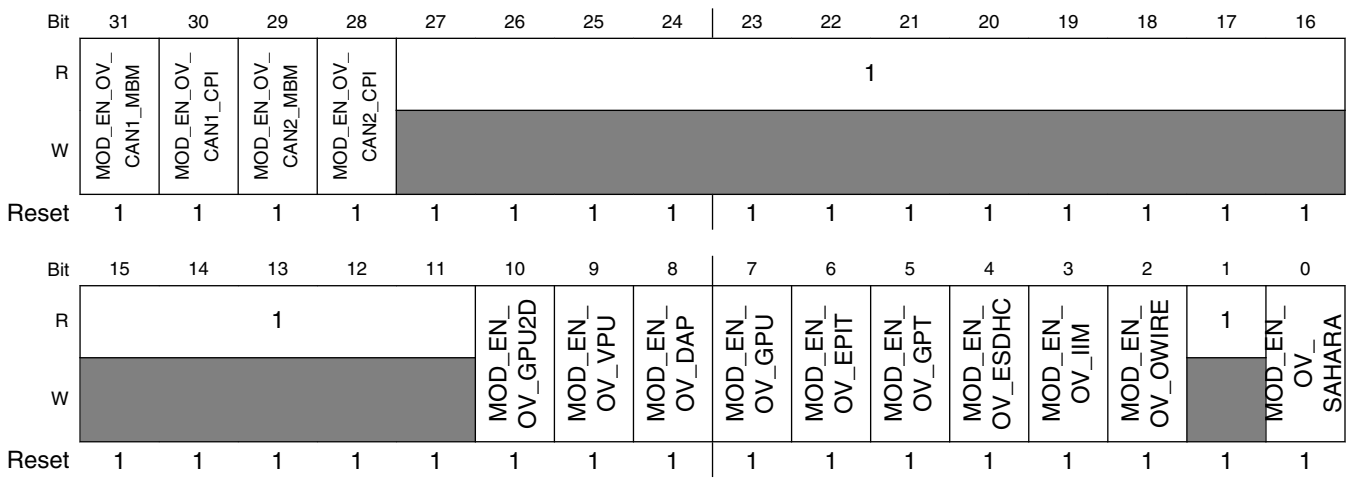
CCM_CCGR7 field descriptions (continued)

Field	Description
5–4 CG2	mlb_clk (mlb_clk_enable)
3–2 CG1	asrc_asrck_clock_d: affects asrck_clock_d input of ASRC (asrc_serial_clk_enable)
1–0 CG0	asrc_ipg_clk: affects ipg_clk input of ASRC (asrc_clk_enable)

18.3.33 CCM Module Enable Overide Register (CCM_CMEOR)

The figure below represents the CCM Module Enable Override Register (CCM_CMEOR). The CMEOR register contains bits to override the clock enable signal from the block. This should be used in case that it is decided to bypass the clock enable signals from the blocks. This bit is applicable only for block that their clock enable signal is used.

Address: CCM_CMEOR is 53FD_4000h base + 88h offset = 53FD_4088h



CCM_CMEOR field descriptions

Field	Description
31 MOD_EN_OV_CAN1_MBM	Override clock enable signal from FLEXCAN-1 - clock is not be gated based on can's signal 'enable_clk_mbm'. 0 dont override block enable signal 1 override block enable signal
30 MOD_EN_OV_CAN1_CPI	Override clock enable signal from FLEXCAN-1 - clock is not be gated based on can's signal 'enable_clk_cpi'. 0 dont override block enable signal 1 override block enable signal

Table continues on the next page...

CCM_CMEOR field descriptions (continued)

Field	Description
29 MOD_EN_OV_ CAN2_MBM	Override clock enable signal from FLEXCAN-2 - clock is not be gated based on can's signal 'enable_clk_mbm'. 0 dont override block enable signal 1 override block enable signal
28 MOD_EN_OV_ CAN2_CPI	Override clock enable signal from FLEXCAN-2 - clock is not gated based on FLEXCAN-2's enable signal 0 dont override block enable signal 1 override block enable signal
27–11 Reserved	This read-only field is reserved and always has the value one. Reserved
10 MOD_EN_OV_ GPU2D	override clock enable signal from GPU2D - clock is not gated based on GPU2D's busy signal. 0 dont override block enable signal 1 override block enable signal
9 MOD_EN_OV_ VPU	override clock enable signal from VPU- clock is not gated based on vpu's idle signal. 0 dont overrid eblock enable signal 1 override block enable signal
8 MOD_EN_OV_ DAP	override clock enable signal from (debug access port) DAP- clock is not gated based on dap's signal 'dap_dbgen'. 0 dont averred block enable signal 1 override block enable signal
7 MOD_EN_OV_ GPU	override clock enable signal from GPU. 0 dont override block enable signal 1 override block enable signal
6 MOD_EN_OV_ EPIT	override clock enable signal from EPIT - clock is not gated based on EPIT's enable signal. 0 dont averred block enable signal 1 override block enable signal
5 MOD_EN_OV_ GPT	override clock enable signal from GPT - clock is not gated based on GPT's enable signal. 0 dont averred block enable signal 1 override block enable signal
4 MOD_EN_OV_ ESDHC	override clock enable signal from ESDHC - clock is not gated based on ESDHC's enable signals. 0 Do not override block enable signal 1 Override block enable signal
3 MOD_EN_OV_ IIM	override clock enable signal from IIM- clock is not gated based on IIM's signal 'iim_clk_en'. 0 dont averred block enable signal 1 override block enable signal
2 MOD_EN_OV_ OWIRE	override clock enable signal from OWIRE - clock is not gated based on OWIRE's signal 'owire_clk_en'.

Table continues on the next page...

CCM_CMEOR field descriptions (continued)

Field	Description
	0 dont override block enable signal 1 override block enable signal
1 Reserved	This read-only field is reserved and always has the value one. Reserved
0 MOD_EN_OV_SAHARA	override clock enable signal from SAHARA. 0 dont override block enable signal 1 override block enable signal

Chapter 19
Configurable Serial Peripheral Interface (CSPI)

19.1 Overview

This chapter describes a block integrated into an SoC. The chapter is intended for a block driver software developer. It describes block-level operation and programming. To understand how the block is integrated at the SoC level, a system software developer should see discussions of the block in the appropriate SoC-level chapter(s).

The Configurable Serial Peripheral Interface (CSPI) block is a full-duplex, synchronous, four-wire serial communication block. The CSPI block contains an 8 x 32 receive buffer (RXFIFO) and an 8 x 32 transmit buffer (TXFIFO). With data FIFOs, the CSPI allows rapid data communication with fewer software interrupts. Figure 19-1 shows a block diagram of the CSPI.

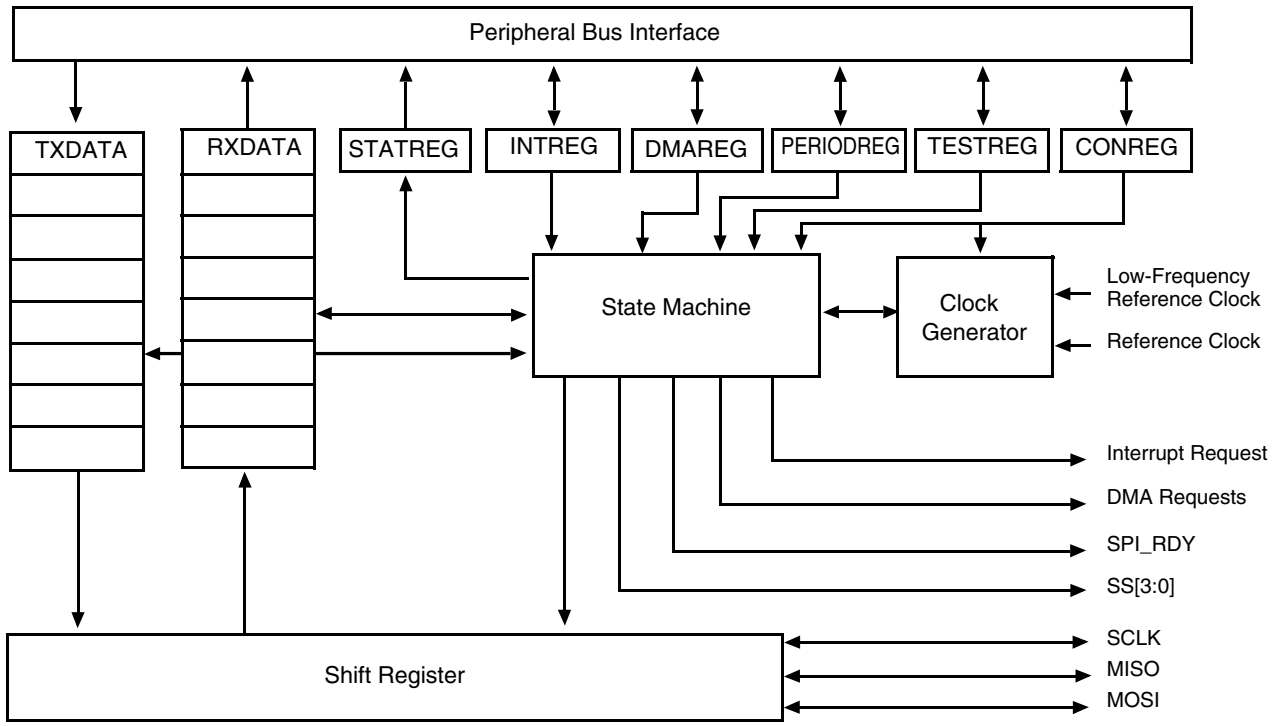


Figure 19-1. CSPI Block Diagram

19.1.1 Features

Key features of the CSPI include:

- Full-duplex synchronous serial interface
- Master/Slave configurable
- Four Chip Select (SS) signals to support multiple peripherals
- Transfer continuation function allows unlimited length data transfers
- 32-bit wide by 8 -entry FIFO for both transmit and receive data
- Polarity and phase of the Chip Select (SS) and SPI Clock (SCLK) are configurable
- Direct Memory Access (DMA) support
- Max operation frequency up to one-quarter of the reference clock frequency.

19.1.2 Modes and Operations

The CSPI supports the modes described in the indicated sections:

As described in Operations, the CSPI supports the operations described in the indicated sections:

19.2 External Signals

The following table lists conventions for representing signals.

Table 19-1. Block Signal Conventions

Category	Convention	Example(s)
Off-chip signal	Uppercase (all capital letters)	TXD
Internal signal[1]	Lowercase italics	*core_int*
Active low signal	_B (_b) suffix or overbar	RESET_EN_B or $\overline{\text{RESET\_EN}}$
Range of bussed or commonly named signals	Beginning and end points of the range are: • Separated by a colon. • Surrounded by square brackets.	ADDR[31:0] CSE_B[7:0] or $\overline{\text{CSE}}$[7:0]
Individual signal in a range of bussed or commonly named signals	Individual number in the range appears without a colon or square brackets	ADDR31 CSE0_B or $\overline{\text{CSE0}}$

1. Internal signals are for reference only in descriptions of internal block or SoC functionality.

The following table describes all CSPI signals that connect off-chip.

Table 19-2. Off-Chip Block Signals

Signal	I/O	Description	Reset State[1]	Pull-Up/ Down[1]
$\overline{\text{SS}}$[3:0]	I/O	Chip selects	1	-
SCLK	I/O	SPI clock	0	Active
MISO	I/O	Master data in; slave data out	0	Passive
MOSI	I/O	Master data out; slave data in	0	-
SPI_RDY	I	Master data out; slave data in	0	Active

1. The reset state values and pull-up/down requirements provided in this table are from the block-level perspective. To understand how the block is integrated at the SoC level, the system software developer must see discussions of the block in the appropriate SoC-level chapter(s). For example, a block signal that requires a pull-up could be integrated with a pull-up option built into the SoC. In this case, the system software developer must ensure the proper programming at the SoC level.

The following figure shows the CSPI in master mode connected to four external devices in a one-way communication link.

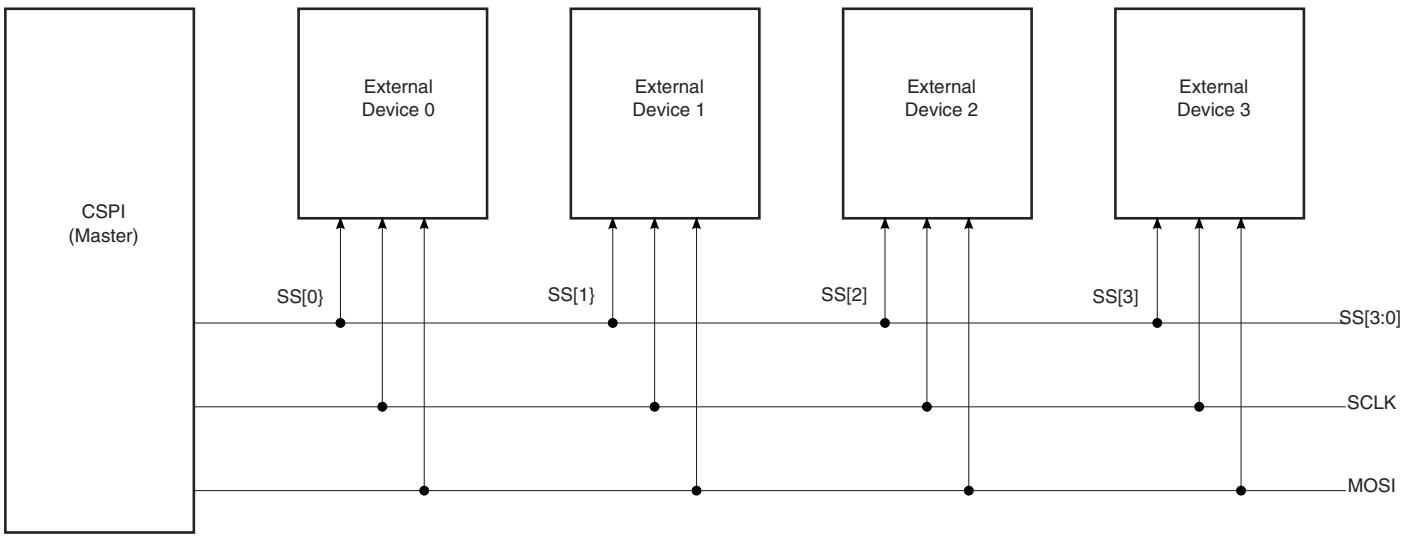


Figure 19-2. Example Connection Diagram

19.3 Functional Description

This section provides a complete functional description of the CSPI. The following figure shows the relationship of SCLK and data lines while CSPI has been configured with different POL and PHA settings.

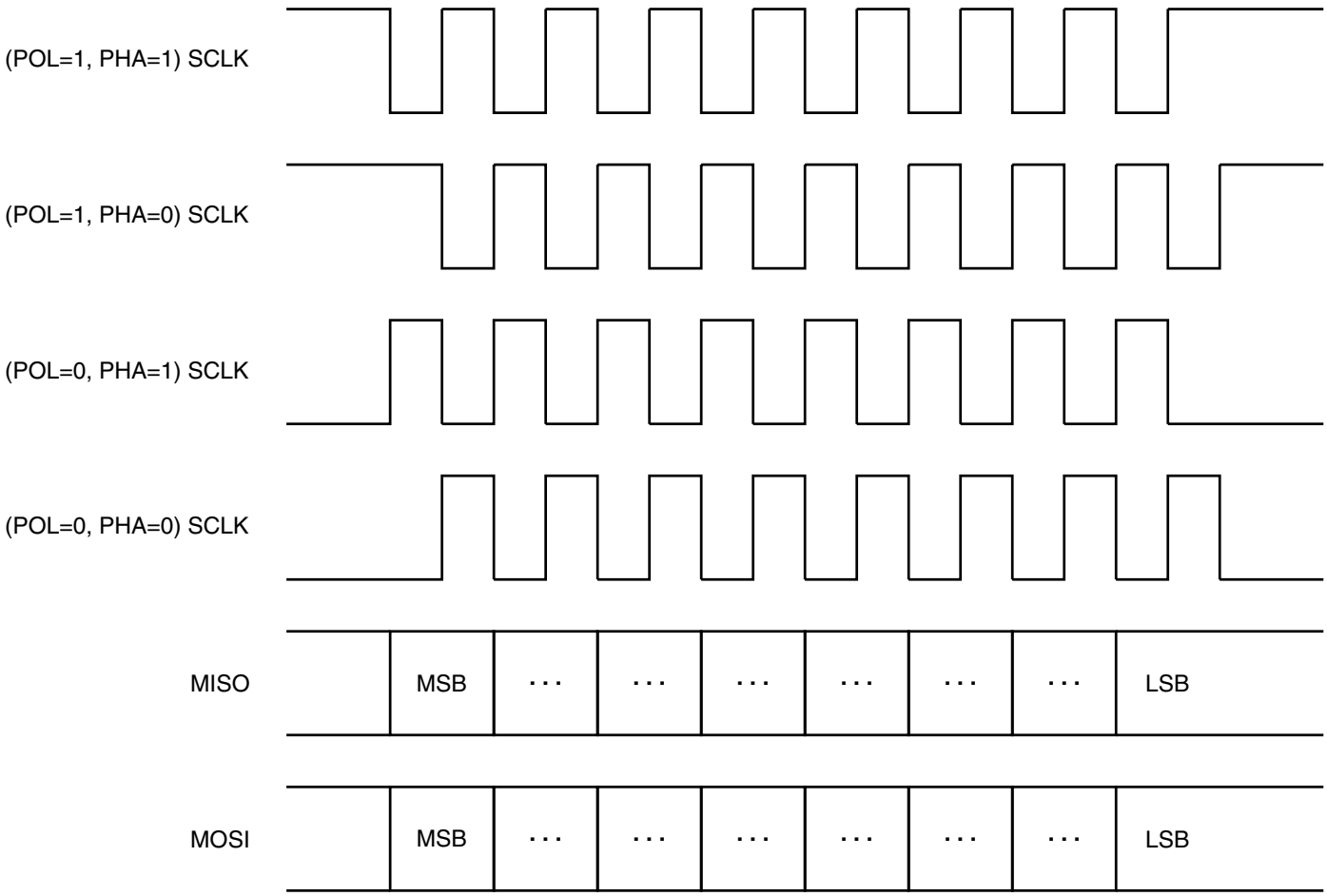


Figure 19-3. CSPI SCLK, MISO, and MOSI Relationship

19.3.1 Operating Modes

CSPI has two operating modes, master mode and slave mode. This section describes all functional operation modes of the CSPI.

19.3.1.1 Master Mode

When the CSPI is configured as a master, it uses a serial link to transfer data between the CSPI and an external slave device. One of the Chip Select (SS) signals and the clock signal (SCLK) are used to transfer data between two devices. If the external device is a transmit-only device, the CSPI master's output port can be ignored and used for other purposes. In order to use the internal TXFIFO and RXFIFO, two auxiliary output signals,

Chip Select (SS) and CSPI_CONREG[DRCTL], are used for data transfer rate control. Software can also configure the sample period control register to a fixed data transfer rate.

19.3.1.2 Slave Mode

When the CSPI is configured as a slave, software can configure the CSPI Control register to match the external SPI master's timing. In this configuration, Chip Select ($\overline{SS}$) becomes an input signal, and is used to control data transfers through the Shift register, as well as to load/store the data FIFO.

19.3.2 Low Power Modes

The CSPI does not operate under low power mode. It holds its operation when its clock is gated off in master mode. In slave mode, the CSPI does not respond when its clock is gated off.

19.3.3 Operations

This section describes the CSPI's operations.

19.3.3.1 Typical Master Mode

The CSPI master uses the Chip Select (SS) signal to enable an external SPI device, and uses the SCLK signal to transfer data in and out of the Shift register. The SPI_RDY enables fast data communication with fewer software interrupts. By programming the CSPI_PERIODREG register accordingly, the CSPI can be used for a fixed data transfer rate.

When the CSPI is in Master mode the SS, SCLK, and MOSI are output signals, and the MISO signal is an input. The following figure shows a typical SPI burst.

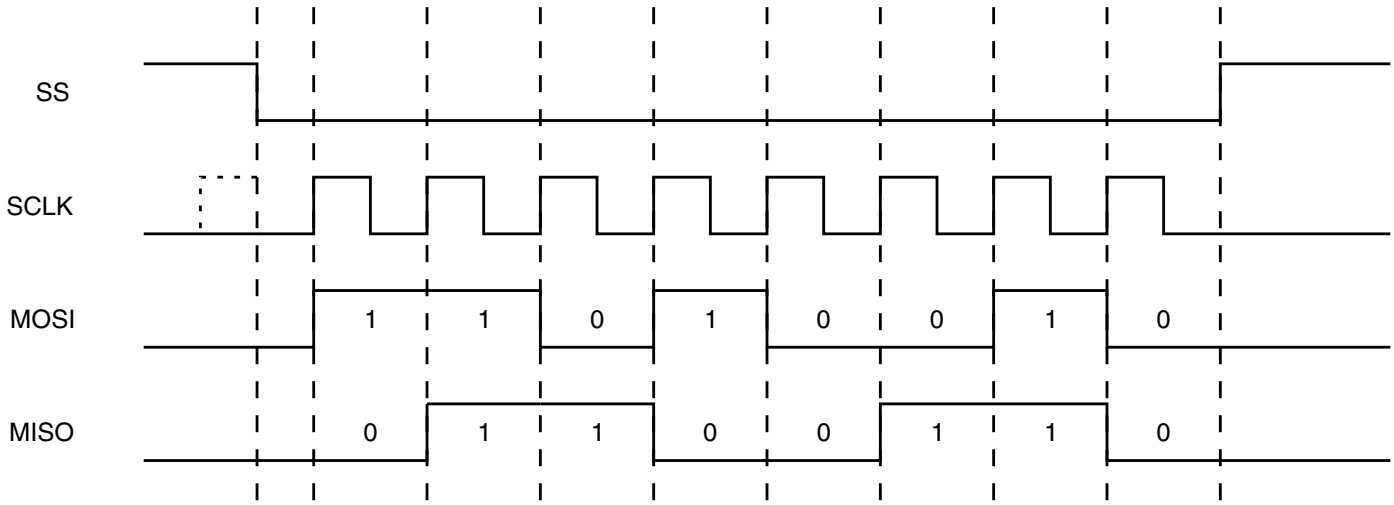


Figure 19-4. Typical SPI Burst (8-bit Transfer)

The Chip Select (SS) signal enables the selected external SPI device, and the SCLK synchronizes the data transfer. The MOSI and MISO signals change on rising edge of SCLK and the MISO signal is latched on the falling edge of the SCLK. The above figure shows a data of 0xD2 is shifted out, and a data of 0x66 is shifted in.

19.3.3.1.1 Master Mode with SPI_RDY

By default, the CSPI does not use the SPI_RDY signal in master mode (MODE =1). A SPI burst begins when the following events happen:

- The CSPI is enabled, TXFIFO has data in it, and CONREG[XCH] bit or the CONREG[SMC] bit is set.
- When the SPI Data Ready Control (CONREG[DRCTL]) bits contains either 01 or 10, the SPI_RDY signal controls when a SPI burst starts.

A SPI burst is defined as a bus transaction that starts when the slave select is asserted and ends when the slave select is negated. The Chip Select (SS) signal will remain asserted until all the bits in a SPI burst are shifted out.

If CONREG[DRCTL] is set to 01, the SPI burst can be triggered only if a falling edge of the SPI_RDY signal from CSPI_CONREG[DRCTL] register has been detected. The following figure shows the relationship between a SPI burst and the falling edge of SPI_RDY signal.

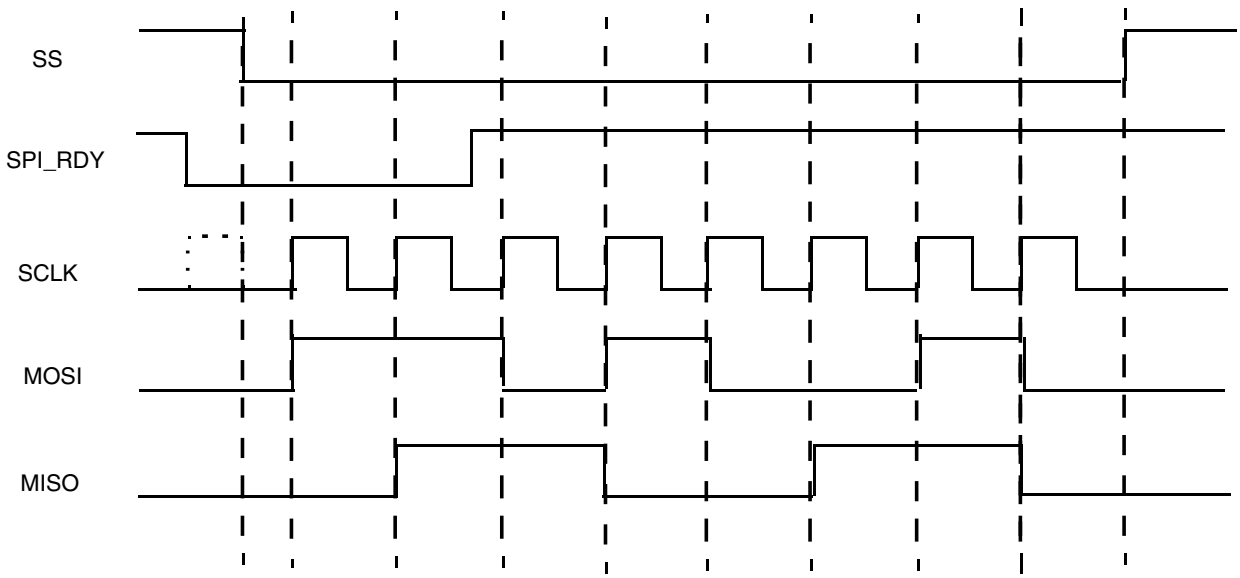


Figure 19-5. Relationship Between a SPI Burst and SPI_RDY: Falling-Edge Triggered

A SPI burst does not start until the falling edge of the SPI_RDY signal is detected. The next SPI burst starts when the next SPI_RDY falling edge is detected, after the last burst has finished.

If SPI Data Ready Control (CONREG[DRCTL]) is set to 10, the SPI burst can be triggered only if the SPI_RDY signal is low. The following figure shows the relationship between a SPI burst and the SPI_RDY signal. The SPI burst does not begin until the SPI_RDY signal goes low. The CSPI will keep transmitting SPI burst if the SPI_RDY signal remains low.

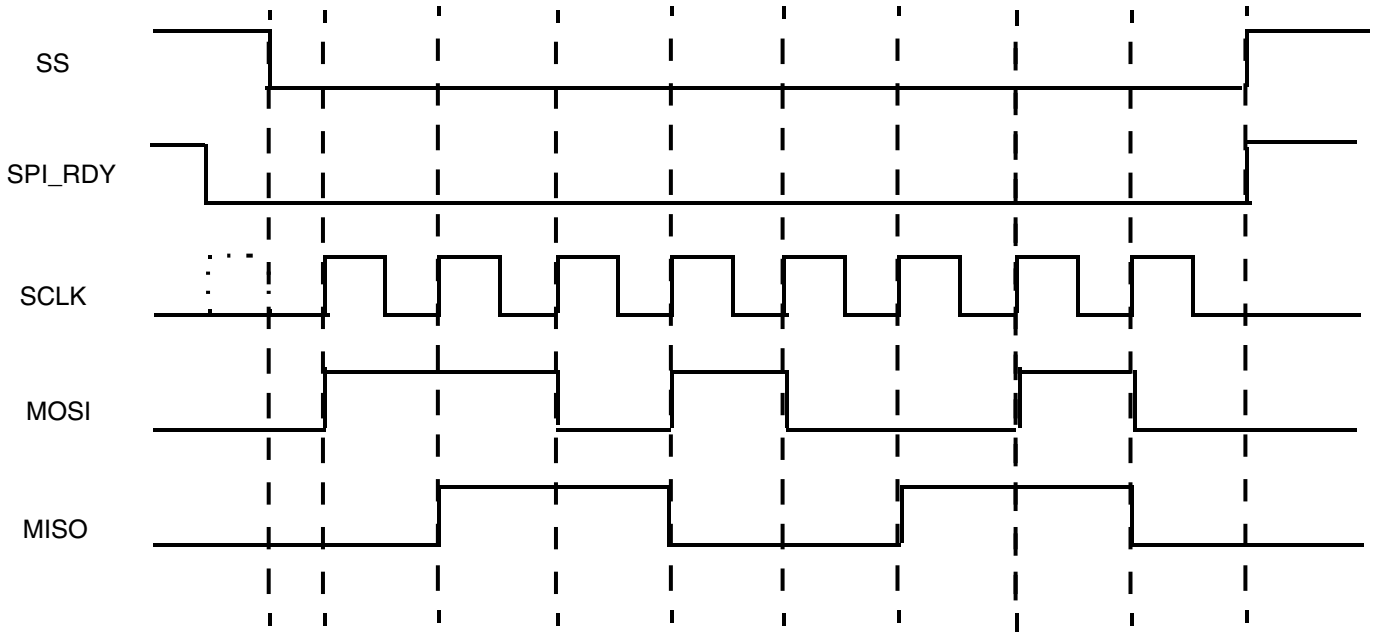


Figure 19-6. Relationship Between a SPI Burst and SPI_RDY: Low-Level Triggered

19.3.3.1.2 Master Mode with Wait States

Wait states can be inserted between SPI bursts. This provides a way for software to slow down the SPI burst to meet the timing requirements of a slower SPI device. The following figure shows wait states inserted between SPI bursts.

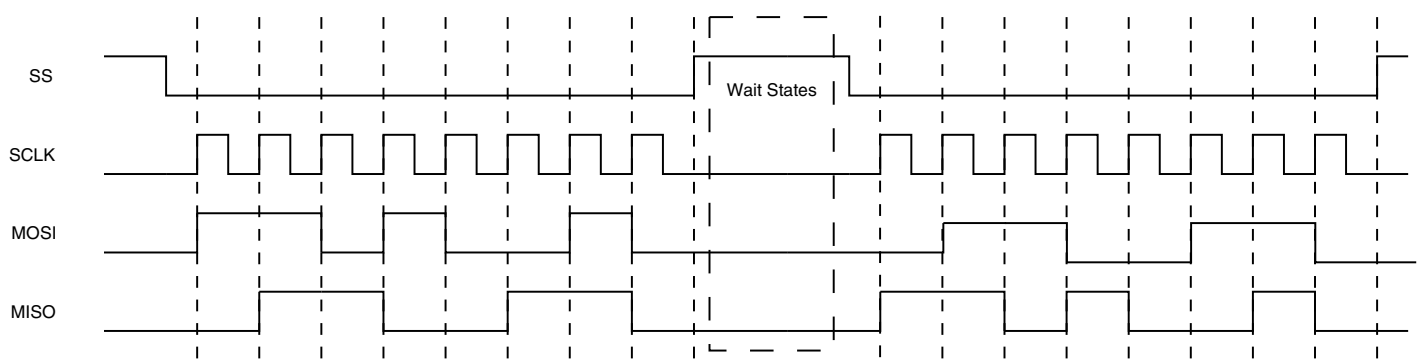


Figure 19-7. SPI Bursts with Wait States

In this case, the number of wait states is controlled by PERIODREG[SAMPLE PERIOD] and the wait states' clock source is selected by PERIODREG[CSRC].

19.3.3.1.3 Master Mode with SSCTL Control

The SPI SS Control (SSCTL) bit controls whether the current operation is single burst or multiple bursts. When the SPI SS Wave Form Select (SSCTL) bit is set, the current operation is multiple bursts transfer. When the SPI SS Wave Form Select (SSCTL) bit is cleared, the current operation is single burst transfer. A SPI burst can contains multiple words as defined in the BURST LENGTH field of the CSPI_CONREG register.

The following figure shows one SPI burst while SSCTL is clear.

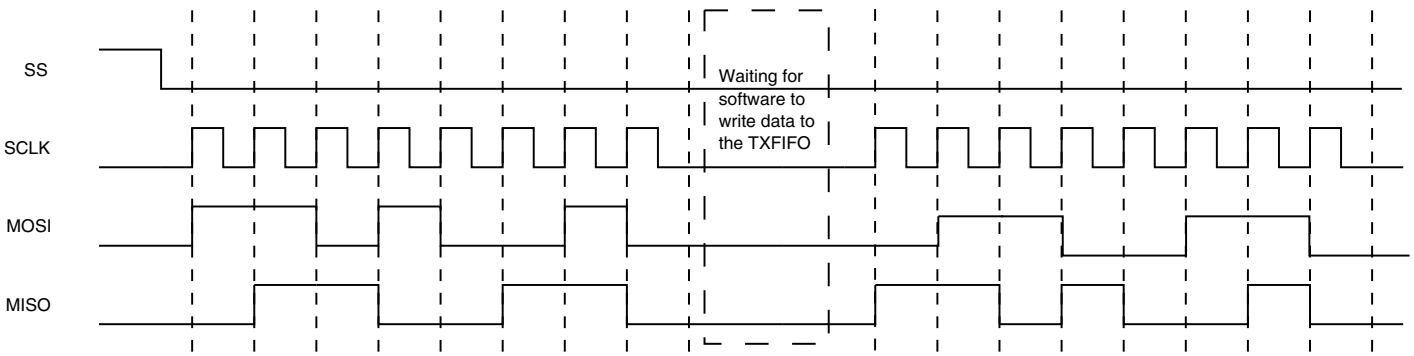


Figure 19-8. SPI Burst While SSCTL is Clear

Two 8-bit bursts in the TXFIFO have been combined and transmitted in one SPI burst. The maximum length of a single SPI burst is defined in the BURST LENGTH field of the CONREG control register. (The above figure corresponds to a BURST LENGTH of 8.) This provides a way for transferring a longer SPI burst by writing data into TXFIFO while the CSPI is transmitting.

The following figure shows two SPI bursts are transmitted while SSCTL is set.

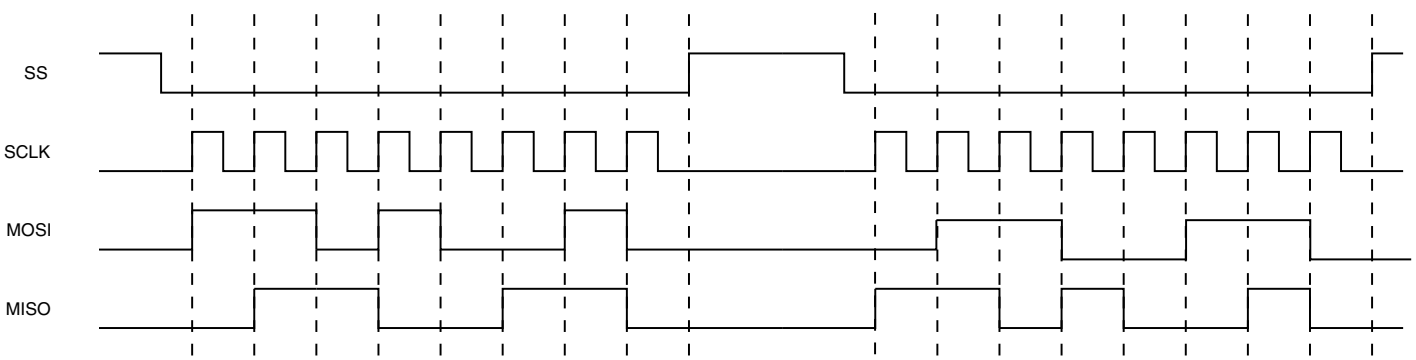


Figure 19-9. SPI Bursts While SSCTL is Set

Two FIFO entries are transmitted, one entry with each SPI burst. The CSPI will continue to transmit SPI bursts until the TXFIFO is empty. When wait states can be inserted between SPI bursts, the SS will negate between SPI bursts until the wait states finish.

19.3.3.1.4 Master Mode with Phase Control

The Phase Control (CSPI_CONREG[PHA]) bit controls how the transmit data shifts out and the receive data shifts in.

When the Phase control (CSPI_CONREG[PHA]) bit is set, the transmit data will shift out on the rising edge of SCLK, and the receive data is latched on the falling edge of SCLK. The most-significant bit is output on the first rising SCLK edge.

When CSPI_CONREG[PHA] is cleared, the transmit data is shifted out on the falling edge of SCLK and the receive data is latched on the rising edge of SCLK. The MSB is output when the host processor loads the transmitted data.

Inverting the SCLK polarity does not impact the edge-triggered operations because they are internal to the serial peripheral interface master.

The following figure shows a SPI burst using different POL and PHA configurations.

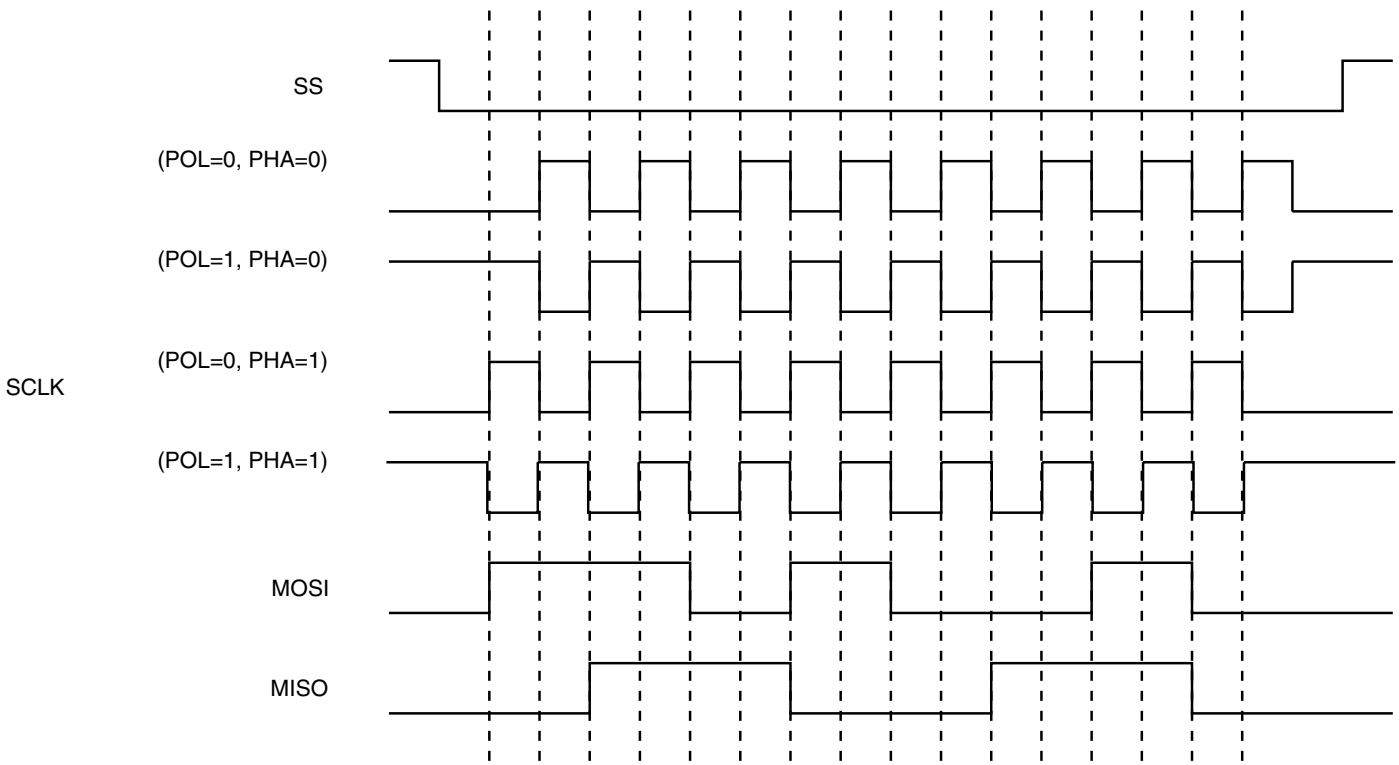


Figure 19-10. SPI Burst with Different POL and PHA Configurations

19.3.3.2 Typical Slave Mode

When the CSPI is configured as a slave (Mode = 0), software can configure the CSPI Control register to match the external SPI master's timing. In this configuration, SS becomes an input signal, and is used to latch data in and out of the internal data Shift registers, as well as to advance the data FIFO.

The SS, SCLK, and MOSI are inputs and MISO is output. Most of the timing diagrams are similar to the diagrams shown previously for the SPI in Master mode (Mode = 1), because the inputs come from a SPI master device.

However, the timing is different when SS is used to advance the data FIFO. When the SSCTL is set while the CSPI is configured in Slave mode, the data FIFO will advance on the rising edge of the SS signal. When the polarity is reversed (SSPOL = 1), the data FIFO will advance on the falling edge of the SS signal.

The following figure shows a SPI burst in which the data FIFO is advanced by the rising edge of the SS signal.

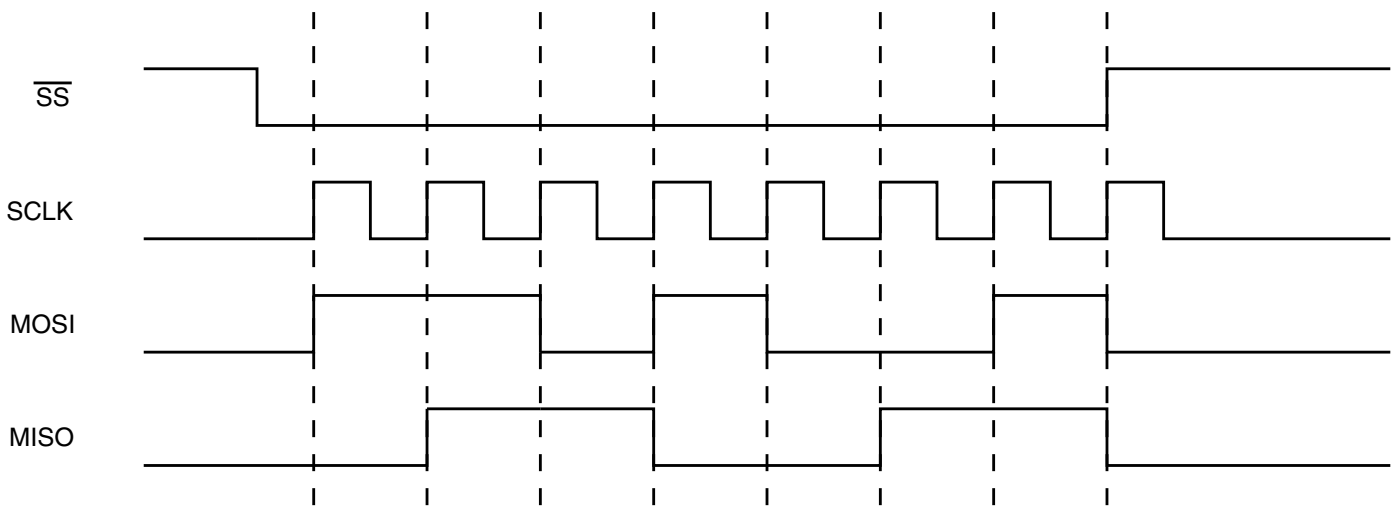


Figure 19-11. Advancing the Data FIFO on the Rising Edge of SS

In the above case, only the most significant 7 bits are loaded to the RXFIFO.

19.3.4 Clocks

This section describes clocks and special clocking requirements of the block.

CSPI has the following clock inputs:

- Reference Clock is the reference clock.
- Low-Frequency Reference Clock is a 32KHz input clock optionally used for counting wait states.

19.3.5 Reset

Whenever a device reset occurs, a reset is performed on the CSPI, resetting all registers to their default values.

Software can reset the block using the CONREG[EN] bit; see Control Register (CSPI_CONREG).

19.3.6 Interrupts

Interrupt control provides a way to manage the CSPI FIFOs:

- For transmitting data, software can enable the TXFIFO empty, TXFIFO half , and TXFIFO full interrupts to maintain the TXFIFO using an interrupt service routine.
- For receiving data, software can enable the RXFIFO ready, RXFIFO half , and RXFIFO full interrupts to retrieve data from the RXFIFO using an interrupt service routine.

Other interrupt sources can be used to control or debug the SPI bursts:

- The transfer-completed interrupt means that there is no data left in the TXFIFO and that the data in the Shift register has been shifted out.
- The RXFIFO overflow interrupt means that the RXFIFO received more than 8 words and will not accept any other words.

The following figure shows a program sequence of SPI bursts using interrupt control.

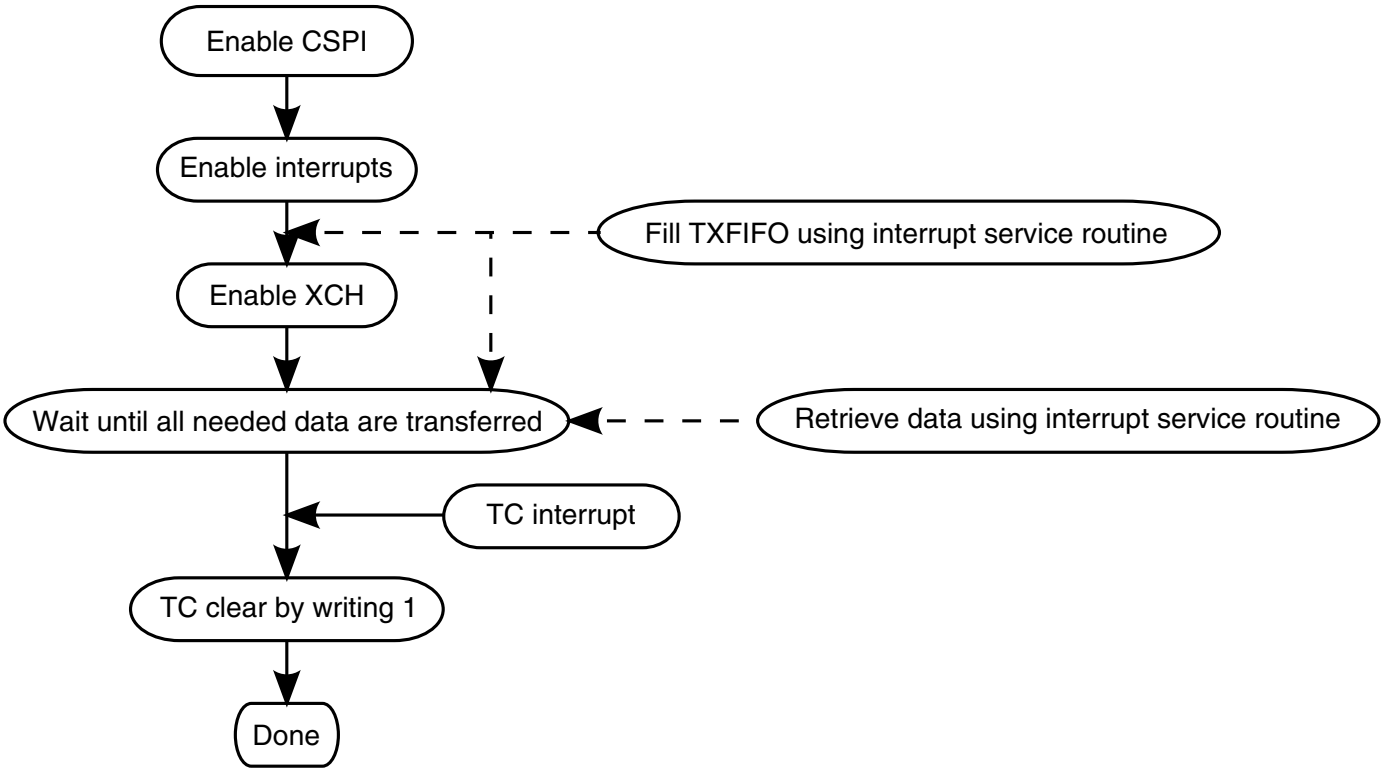


Figure 19-12. Program Sequence of SPI Burst Using Interrupt Control

19.3.7 DMA

DMA control provides another method to utilize the FIFOs in the CSPI. By using DMA request and acknowledge signals, larger amounts of data can be transferred, and will reduce interrupts and host processor loading. When the appropriate conditions are matched, the block will send out a DMA request, and the DMA can deal with the following conditions:

- TXFIFO empty
- TXFIFO half
- RXFIFO half
- RXFIFO full

The following figure shows a program sequence of SPI bursts using DMA control.

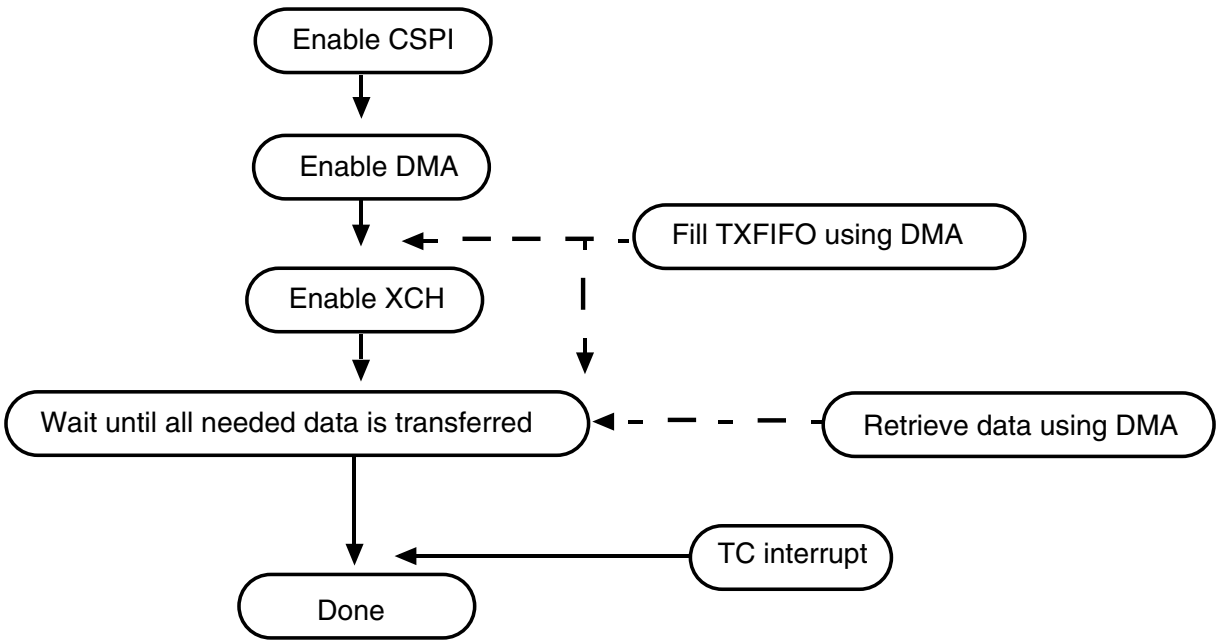


Figure 19-13. Program Sequence of SPI Burst Using DMA

19.3.8 Byte Order

Software can swap bytes for receive data using the TESTREG[SWAP] bit. See Test Control Register (CSPI_TESTREG).

19.4 Initialization

This section provides initialization information for CSPI.

To initialize the block:

1. Clear the EN bit in CSPI_CONREG to reset the block.
2. Enable the clocks for CSPI.
3. Set the EN bit in CSPI_CONREG to put CSPI out of reset.
4. Configure corresponding IOMUX for CSPI external signals.
5. Configure registers of CSPI properly according to the specifications of the external SPI device.

19.5 Applications

The following figure shows two flowcharts for the master and slave mode of operations supported by the CSPI.

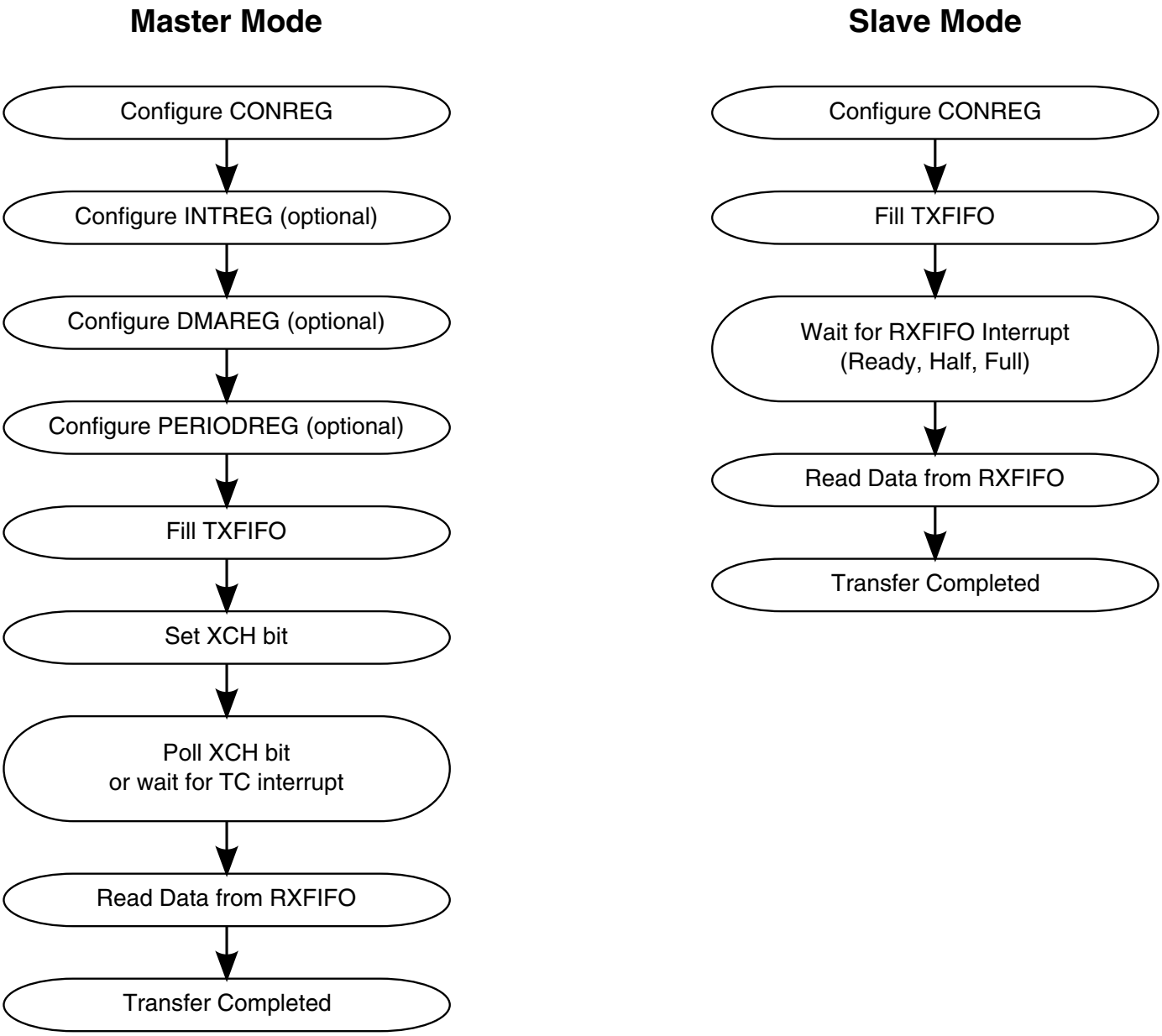


Figure 19-14. Flowchart of the CSPI Operation

The following example shows example code of CSPI operation using ARM instructions.

CSPI Operation using ARM Instructions

```
          LDR R0, =CSPI_BASE_ADDRESS                ; Load CSPI Base Address to R0

          LDR R1, =0x01F00003                       ; Master Mode, 32-bit transaction

          STR R1, [R0, #0x08]

          LDR R1, =0x00000011                       ; Enable RXFIFO half and TXFIFO empty

          STR R1, [R0, #0x0C]                          ; interrupt (Alternatively with
DMA Mode)

          LDR R1, =0x00000011                       ; Enable RXFIFO half and TXFIFO empty

          STR R1, [R0, #0x10]                          ; DMA (Alternatively with
interrupt)

          LDR R5, =0x05                             ; R5 as number of words to be
transferred.

          LDR R1, =0x11111111                       ; R1 as increment to generate the
data.

          LDR R2, =0x12345678                       ; R2 load the data to be transferred.

Loop_00

          STR R2, [R2,#0x04]                           ; Store data into TXFIFO.

          ADD R2, R2, R1                            ; Generating next data to be
transferred.

          SUB R5, R5, #1                            ; Decrease the R5.

          CMP R5, #0x00                             ; Check R5 if it is zero.

          BNE Loop_00                              ; Loop until R5 is zero.

          LDR R1, =0x01F00007                         ; set XCH bit to start transaction.

          STR R1, [R0, #0x08]

Loop_01

          LDR R1, [R0, #0x08]                          ; check XCH bit if it is cleared.

          LDR R2, =0x00000004

          AND R1, R2, R1

          CMP R1, #0x00

          BEQ PASS_00                              ; if XCH bit is cleared then finish.

          B Loop_01                                ; if it isn't cleared then continue loop

          LDR R1, [R0, #0x00]                          ; Read data from RXFIFO.
```

19.6 Programmable Registers

This section includes the block memory map and detailed descriptions of all registers. For the base address of a particular block instantiation, see the system memory map.

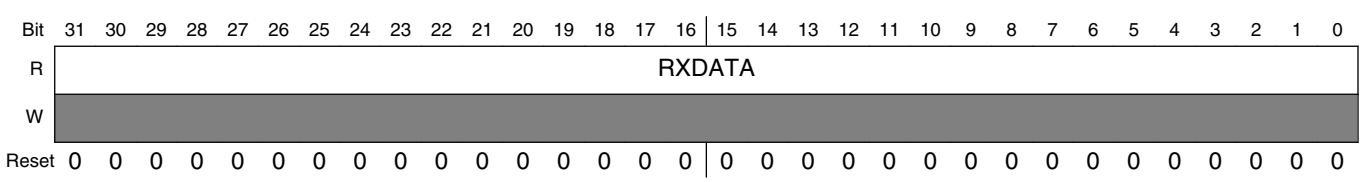
CSPI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FC_0000	Receive Data Register (CSPI_RXDATA)	32	R	0000_0000h	19.6.1/ 944
63FC_0004	Transmit Data Register (CSPI_TXDATA)	32	W (always reads zero)	0000_0000h	19.6.2/ 945
63FC_0008	Control Register (CSPI_CONREG)	32	R/W	0000_0000h	19.6.3/ 946
63FC_000C	Interrupt Control Register (CSPI_CSPI_INTREG)	32	R/W	0000_0000h	19.6.4/ 949
63FC_0010	DMA Control Register (CSPI_CSPI_DMAREG)	32	R/W	0000_0000h	19.6.5/ 950
63FC_0014	Status Register (CSPI_CSPI_STATREG)	32	R/W	0000_0003h	19.6.6/ 951
63FC_0018	Sample Period Control Register (CSPI_PERIODREG)	32	R/W	0000_0000h	19.6.7/ 952
63FC_001C	Test Control Register (CSPI_TESTREG)	32	R/W	0000_0000h	19.6.8/ 953

19.6.1 Receive Data Register (CSPI_RXDATA)

The Receive Data register (CSPI_RXDATA) is a read-only register that forms the top word of the 8 x 32 receive FIFO. This register holds the data received from an external SPI device during a data transaction. Only word-sized read operations are allowed.

Address: CSPI_RXDATA is 63FC_0000h base + 0h offset = 63FC_0000h



CSPI_RXDATA field descriptions

Field	Description
31–0 RXDATA	Receive Data. This register holds the top word of the receive data FIFO. The FIFO is advanced for each read of this register. The data read is undefined when the Receive Data Ready (RR) bit in the Interrupt Control/Status register is cleared. Zeros are read when CSPI is disabled.

19.6.2 Transmit Data Register (CSPI_TXDATA)

The Transmit Data (CSPI_TXDATA) register is a write-only data register that forms the bottom word of the 8 x 32 TXFIFO. The TXFIFO can be written to as long as it is not full, even when the SPI Exchange bit (XCH) in CSPI_CONREG is set. This allows software to write to the TXFIFO during a SPI data exchange process. Writes to this register are ignored when the CSPI block is disabled (CSPI_CONREG[EN] bit is cleared).

Address: CSPI_TXDATA is 63FC_0000h base + 4h offset = 63FC_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W	TXDATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

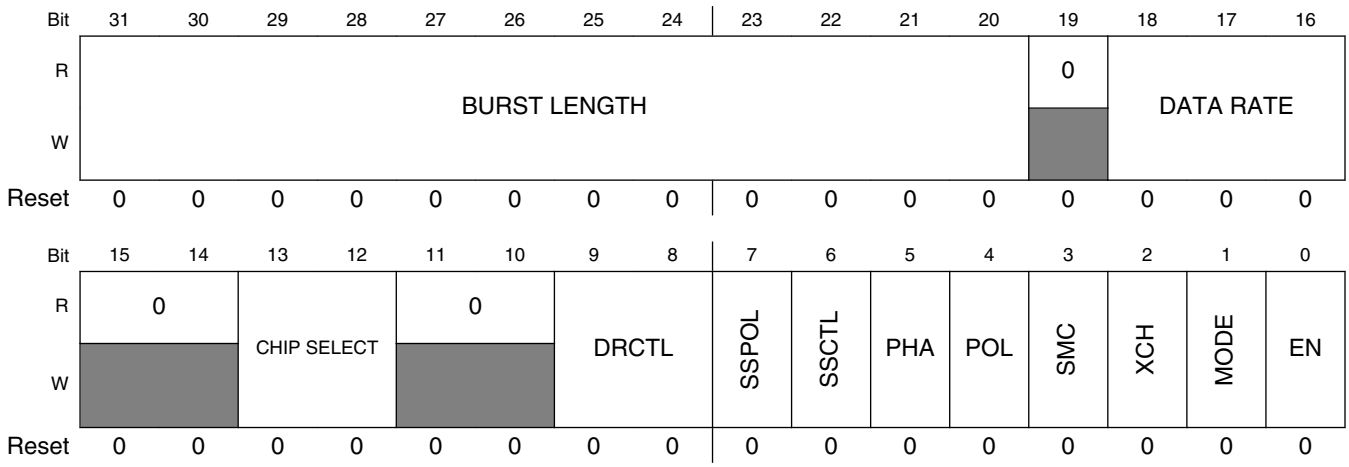
CSPI_TXDATA field descriptions

Field	Description
31–0 TXDATA	Transmit Data. This register holds the top word of data loaded into the FIFO. Data written to this register must be a word operation. The number of bits actually transmitted is determined by the BIT_COUNT field of the corresponding SPI Control register. If this field contains more bits than the number specified by BIT_COUNT, the extra bits are ignored. For example, to transfer 10 bits of data, a 32-bit word must be written to this register. Bits 9-0 are shifted out and bits 31-10 are ignored. When the CSPI is operating in Slave mode, zeros are shifted out when the FIFO is empty. Zeros are read when CSPI is disabled.

19.6.3 Control Register (CSPI_CONREG)

The Control Register (CSPI_CONREG) allows software to enable the CSPI, configure its operating modes, specify the divider value, phase, and polarity of the clock, configure the Chip Select (SS) and SPI_RDY (CSPI_CONREG[DRCTL]) control signal, and define the transfer length.

Address: CSPI_CONREG is 63FC_0000h base + 8h offset = 63FC_0008h



CSPI_CONREG field descriptions

Field	Description
31–20 BURST LENGTH	Burst Length. This field defines the length of a SPI burst to be transferred. The Chip Select (SS) will remain asserted until all bits in a SPI burst are shifted out. A maximum of 2^12 bits can be transferred in a single SPI burst. In master mode, it controls the number of bits per SPI burst. Since the shift register always loads 32-bit data from transmit FIFO, only the n least-significant (n = BURST LENGTH + 1) will be shifted out. The remaining bits will be ignored. In slave mode, only when SSCTL is cleared, this field will take effect in the transfer. Number of Valid Bits in a SPI burst. 0x000 A SPI burst contains the 1 LSB in a word. 0x001 A SPI burst contains the 2 LSB in a word. 0x002 A SPI burst contains the 3 LSB in a word. 0x01F A SPI burst contains all 32 bits in a word. 0x020 A SPI burst contains the 1 LSB in first word and all 32 bits in second word. 0x021 A SPI burst contains the 2 LSB in first word and all 32 bits in second word. 0xFFE A SPI burst contains the 31 LSB in first word and 2^7 -1 words. 0xFFF A SPI burst contains 2^7 words.
19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18–16 DATA RATE	SPI Data Rate Control. This field selects the baud rate of the SCLK based on a division of the reference clock.

Table continues on the next page...

CSPI_CONREG field descriptions (continued)

Field	Description
	These bits allow the CSPI to synchronize with different external SPI devices. The max frequency is one quarter of the reference clock. The divide ratio is determined according to the following table using the equation: $2^{(n+2)}$. SPI Data Rate Control (Master Mode only) 000 Divide by 4. 001 Divide by 8. 010 Divide by 16. 011 Divide by 32. 100 Divide by 64. 101 Divide by 128. 110 Divide by 256. 111 Divide by 512.
15–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13–12 CHIP SELECT	CHIP SELECT. Select one of four external SPI Master/Slave Devices. In master mode, these two bits select the external slave devices by asserting the Chip Select (SSn) outputs. Only the selected Chip Select (SSn) signal can be active at a given time; the remaining three signals will be negated. Chip Select 00 Chip Select 0 (SS0) will be asserted. 01 Chip Select 1 (SS1) will be asserted. 10 Chip Select 2 (SS2) will be asserted. 11 Chip Select 3 (SS3) will be asserted.
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–8 DRCTL	SPI Data Ready Control. This field selects the utilization of the $\overline{\text{SPI\_RDY}}$ signal in master mode. CSPI checks this field before it starts an SPI burst. 00 The $\overline{\text{SPI\_RDY}}$ signal is a don't care. 01 Burst will be triggered by the falling edge of the SPI_RDY signal (edge-triggered). 10 Burst will be triggered by a low level of the SPI_RDY signal (level-triggered). 11 Reserved.
7 SSPOL	SPI SS Polarity Select. In both Master and Slave modes, this bit selects the polarity of the Chip Select (SS) signal. 0 Active low. 1 Active high.
6 SSCTL	SPI SS Wave Form Select. In master mode, this bit controls the output wave form of the Chip Select (SS) signal when SMC is cleared. The SSCTL bit will be ignored if the SMC (Start Mode Control) bit is set. In slave mode, this bit controls when the SPI burst is completed. 0 Only one SPI burst will be transmitted. 1 Negate Chip Select (SS) signal between SPI bursts. Multiple SPI bursts will be transmitted. The SPI transfer will automatically stop when the TXFIFO is empty.

Table continues on the next page...

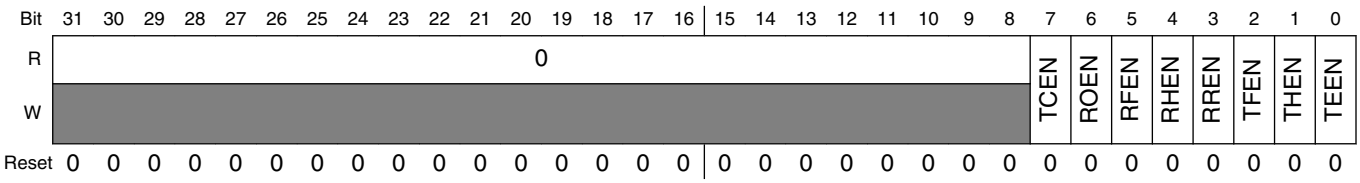
CSPI_CONREG field descriptions (continued)

Field	Description
	0 (slave mode) A SPI burst is completed when the number of bits received in the shift register is equal to BURST LENGTH + 1. Only n least-significant bits (n = BURST LENGTH[4:0] + 1) of the first received word are valid. All bits subsequent to the first received word in RXFIFO are valid. 1 (slave mode) A SPI burst is completed by the Chip Select (SS) signal edges. (SSPOL = 0: rising edge; SSPOL = 1: falling edge) The RXFIFO is advanced whenever a Chip Select (SS) signal edge is detected or the shift register contains 32-bits of valid data.
5 PHA	SPI Clock/Data Phase Control. This bit controls the clock/data phase relationship. See Typical Slave Mode for more information. 0 Phase 0 operation. 1 Phase 1 operation.
4 POL	SPI Clock Polarity Control. This bit controls the polarity of the SCLK signal. See Typical Slave Mode for more information. 0 Active high polarity (0 = Idle). 1 Active low polarity (1 = Idle).
3 SMC	Start Mode Control. This bit applies only when the block is configured in Master mode (MODE = 1). It controls how the CSPI starts a SPI burst, either through the SPI exchange bit, or immediately when the TXFIFO is written to. 0 SPI Exchange Bit (XCH) controls when a SPI burst can start. Setting the XCH bit will start a SPI burst or multiple bursts. This is controlled by the SPI SS Wave Form Select (SSCTL) bit. Refer to XCH and SSCTL bit descriptions. 1 Immediately starts a SPI burst when data is written in TXFIFO.
2 XCH	SPI Exchange Bit. This bit applies only when the block is configured in Master mode (MODE = 1). If the Start Mode Control (SMC) bit is cleared, writing a 1 to this bit starts one SPI burst or multiple SPI bursts according to the SPI SS Wave Form Select (SSCTL bit). The XCH bit remains set while either the data exchange is in progress, or when the CSPI is waiting for an active input if SPIRDY is enabled through DRCTL. This bit is cleared automatically when all data in the TXFIFO and the shift register has been shifted out. 0 Idle. 1 Initiates exchange (write) or busy (read).
1 MODE	SPI Function Mode Select. This bit selects the operating mode of the CSPI. 0 Slave mode. 1 Master mode.
0 EN	SPI Module Enable Control. This bit enables the CSPI. This bit must be set before writing to other registers or initiating an exchange. Writing zero to this bit disables the block and resets the internal logic with the exception of the CONREG. The block's internal clocks are gated off whenever the block is disabled. 0 Disable the block. 1 Enable the block.

19.6.4 Interrupt Control Register (CSPI_CSPI_INTREG)

The Interrupt Control Register (CSPI_INTREG) enables the generation of interrupts to the host processor. If the CSPI is disabled, this register reads zero.

Address: CSPI_CSPI_INTREG is 63FC_0000h base + Ch offset = 63FC_000Ch



CSPI_CSPI_INTREG field descriptions

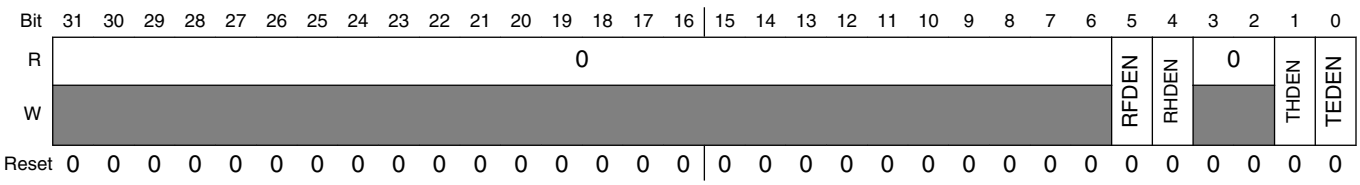
Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7 TCEN	Transfer Completed Interrupt enable. This bit enables the Transfer Completed Interrupt. 0 Disable 1 Enable
6 ROEN	RXFIFO Overflow Interrupt enable. This bit enables the RXFIFO Overflow Interrupt. 0 Disable 1 Enable
5 RFEN	RXFIFO Full Interrupt enable. This bit enables the RXFIFO Full Interrupt. 0 Disable 1 Enable
4 RHEN	RXFIFO Half Full Interrupt enable. This bit enables the RXFIFO Half Full Interrupt. 0 Disable 1 Enable
3 RREN	RXFIFO Ready Interrupt enable. This bit enables the RXFIFO Ready Interrupt. 0 Disable 1 Enable
2 TFEN	TXFIFO Full Interrupt enable. This bit enables the TXFIFO Full Interrupt. 0 Disable 1 Enable
1 THEN	TXFIFO Half Empty Interrupt enable. This bit enables the TXFIFO Half Empty Interrupt. 0 Disable 1 Enable
0 TEEN	TXFIFO Empty Interrupt enable. This bit enables the TXFIFO Empty Interrupt. 0 Disable 1 Enable

19.6.5 DMA Control Register (CSPI_CSPI_DMAREG)

The Direct Memory Access Control Register (CSPI_DMAREG) provides software a way to use an on-chip DMA controller for CSPI data. Internal DMA request signals enable direct data transfers between the CSPI FIFOs and system memory. The CSPI sends out DMA requests when the appropriate FIFO conditions are matched.

If the CSPI is disabled, this register is read as 0.

Address: CSPI_CSPI_DMAREG is 63FC_0000h base + 10h offset = 63FC_0010h



CSPI_CSPI_DMAREG field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5 RFDEN	RXFIFO Full DMA Request Enable. This bit enables/disables the RXFIFO Full DMA Request. 0 Disable 1 Enable
4 RHDEN	RXFIFO Half Full DMA Request Enable. This bit enables/disables the RXFIFO Half Full DMA Request. 0 Disable 1 Enable
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1 THDEN	TXFIFO Half Empty DMA Request Enable. This bit enables/disables the TXFIFO Half Empty DMA Request. 0 Disable 1 Enable
0 TEDEN	TXFIFO Empty DMA Request Enable. This bit enables/disables the TXFIFO Empty DMA Request. 0 Disable 1 Enable

19.6.6 Status Register (CSPI_CSPI_STATREG)

The CSPI Status Register (CSPI_STATREG) reflects the status of the CSPI block's operating condition. If the CSPI is disabled, this register reads 0x0000_0003.

Address: CSPI_CSPI_STATREG is 63FC_0000h base + 14h offset = 63FC_0014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TC	RO	RF	RH	RR	TF	TH	TE
W									w1c							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

CSPI_CSPI_STATREG field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7 TC	Transfer Completed Status bit. Writing 1 to this bit clears it. 0 Transfer in progress. 1 Transfer completed.
6 RO	RXFIFO Overflow. When set, this bit indicates that RXFIFO has overflowed. 0 RXFIFO has no overflow. 1 RXFIFO has overflowed.
5 RF	RXFIFO Full. This bit is set when the RXFIFO is full. 0 Not Full. 1 Full.
4 RH	RXFIFO Half Full. This bit is set when the RXFIFO reaches half full. 0 Less than 4 words are stored in RXFIFO. 1 Four or more words are available in RXFIFO.
3 RR	RXFIFO Ready. This bit is set when one or more words are stored in the RXFIFO. 0 No valid data in RXFIFO. 1 More than 1 word in RXFIFO.
2 TF	TXFIFO Full. This bit is set when if the TXFIFO is full. 0 TXFIFO is not Full. 1 TXFIFO is Full.
1 TH	TXFIFO Half empty. This bit is set when the TXFIFO reaches half empty.

Table continues on the next page...

CSPI_CSPI_STATREG field descriptions (continued)

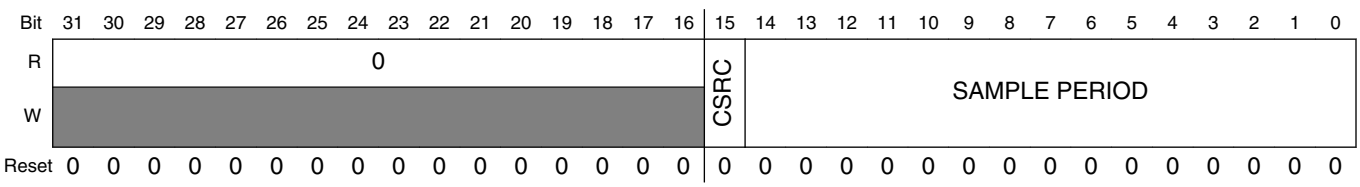
Field	Description
	0 TXFIFO holds more than 4 words. 1 TXFIFO holds 4 or fewer words.
0 TE	TXFIFO Empty. This bit is set if the TXFIFO is empty. 0 TXFIFO contains one or more words. 1 TXFIFO is empty.

19.6.7 Sample Period Control Register (CSPI_PERIODREG)

The Sample Period Control Register (CSPI_PERIODREG) provides software a way to insert delays (wait states) between consecutive SPI transfers. Control bits in this register select the clock source for the sample period counter and the delay count indicating the number of wait states to be inserted between data transfers.

The delay counts apply only when the block is configured in Master mode (CONREG[MODE] = 1).

Address: CSPI_PERIODREG is 63FC_0000h base + 18h offset = 63FC_0018h



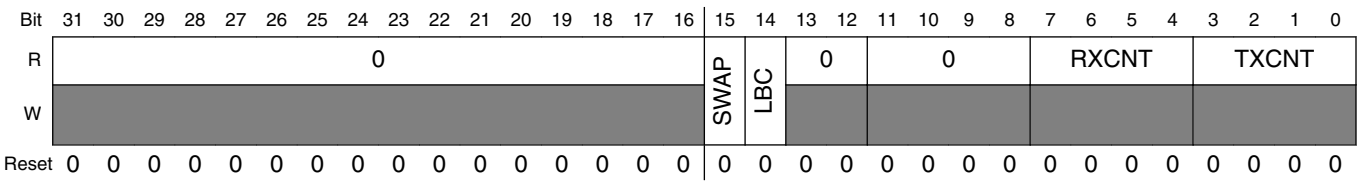
CSPI_PERIODREG field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15 CSRC	Clock Source Control. This bit selects the clock source for the sample period counter. 0 SPI Clock (SCLK) 1 Low-Frequency Reference Clock (32.768 KHz)
14–0 SAMPLE PERIOD	Sample Period Control. These bits control the number of wait states to be inserted in data transfers. During the idle clocks, the state of the SS output will operate according to the SSCTL control field in the CSPI_CONREG register. 0x00000 wait states inserted 0x00011 wait state inserted 0x7FFE32766 wait states inserted 0x7FFF32767 wait states inserted

19.6.8 Test Control Register (CSPI_TESTREG)

The Test Control Register (CSPI_TESTREG) provides software a mechanism to internally connect the receive and transmit devices of the CSPI, swap the byte order for receive data, and monitor the contents of the receive and transmit FIFO.

Address: CSPI_TESTREG is 63FC_0000h base + 1Ch offset = 63FC_001Ch



CSPI_TESTREG field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15 SWAP	Data Swap. This bit is used to swap data as it is read from the RXFIFO. When this bit is set, data read from RXFIFO is swapped. RXDATA[31:0] is swapped as follows: {RXDATA[7:0],RXDATA[15:8], RXDATA[23:16],RXDATA[31:24]} 0 Data read from RXFIFO is unchanged. 1 Data read from RXFIFO is swapped.
14 LBC	Loop Back Control. This bit is used in Master mode only. When this bit is set, the CSPI connects the transmitter and receiver sections internally, and the data shifted out from the most-significant bit of the shift register is looped back into the least-significant bit of the Shift register. In this way, a self-test of the complete transmit/receive path can be made. The output pins are not affected, and the input pins are ignored. 0 Not connected. 1 Transmitter and receiver sections internally connected for Loopback.
13–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11–8 Reserved	This read-only field is reserved and always has the value zero. Reserved, all bits should be ignored.
7–4 RXCNT	RXFIFO Counter. These bits indicate the number of words in RXFIFO. RXFIFO Counter 0000 0 word in RXFIFO 0001 1 word in RXFIFO 0111 7 words in RXFIFO 1000 8 words in RXFIFO
3–0 TXCNT	TXFIFO Counter. These bits indicate the number of words in TXFIFO. TXFIFO Counter

Table continues on the next page...

CSPI_TESTREG field descriptions (continued)

Field	Description
	0000 0 word in TXFIFO 0001 1 word in TXFIFO 0111 7 words in TXFIFO 1000 8 words in TXFIFO

Chapter 20
Central Security Unit (CSU)

20.1 Overview

The CSU enables software for setting comprehensive security policy within the platform and sharing secure information between various secure blocks.

20.2 Features

The Central Security Unit (CSU) sets access control policies between bus masters and bus slaves, allowing peripherals to be separated into distinct security domains. This protects against indirect unauthorized access to data, such as occurs when software programs a DMA bus master to access addresses that the software itself is prohibited from accessing directly. Configuring DMA bus master privileges in the CSU consistently with software privileges defends against such indirect unauthorized access. Additionally, the CSU manges system sucurity alarms.

CSU has the following security related features:

- System alarm routing policy:
  - Type of alarm triggered by the different system security violation indications
  - Type of system event generated by each of the system alarms
  - Security violation indication (Secure Real Time Clock (SRTC)) clearance (fuse driven)
- Setting emergency reset policy
- Peripheral access policy - Master privileges are required to access each peripheral.
- Masters privilege policy - CSU helps those Masters to assert user/supervisor secure/ non-secure signals which can not generate these signals on their own.

20.3 Functional Description

The CSU enables secure software for setting comprehensive security policy within the platform and for sharing secure information between the various secure modules.

Security policies may be set, and optionally locked, in the CSU registers by secure software, including the command sequence file (CSF) processed by the High Assurance Boot (HAB) or a HAB-authenticated image which executes after the boot ROM.

20.3.1 Peripheral Access Policy

The CSU helps secure software in setting the Peripheral Access Policy.

It determines which peripheral can be accessed by what master privileges.

There are four security modes of operation in the system. These security modes are distinguished by security (TrustZone/non-TrustZone) and privilege (Supervisor/User) setting of the module. Below is the list of these security modes from the highest security level to the lowest:

- TrustZone (Secure) Privilege (Supervisor) Mode - Highest Security Level
- TrustZone (Secure) non-Privilege (User) Mode - Medium Security Level
- non-TrustZone (Regular) Privilege (Supervisor) Mode - Medium Security Level
- non-TrustZone (Regular) non-Privilege (User) Mode - Lowest Security Level

This functionality is implemented as follows:

The Configure Slave Level (CSL) Register value for a specified peripheral resource defines the output signal -- csu_sec_level for that peripheral. The value of this signal determines by what master privileges the peripheral can be accessed. The relation between the value of the csu_sec_level signal and execution mode has been shown below in the table.

Table 20-1. Permission Access Table

CSU_SEC_LEVEL[2:0]	Non-Secure User Mode	Non-Secure Spvr Mode	Secure (TZ) User Mode	Secure (TZ) Spvr Mode	CSL register value
(0) 000	RD+WR	RD+WR	RD+WR	RD+WR	8'b1111_1111
(1) 001	None	RD+WR	RD+WR	RD+WR	8'b1011_1011
(2) 010	RD	RD	RD+WR	RD+WR	8'b0011_1111
(3) 011	None	RD	RD+WR	RD+WR	8'b0011_1011
(4) 100	None	None	RD+WR	RD+WR	8'b0011_0011

Table continues on the next page...

Table 20-1. Permission Access Table (continued)

CSU_SEC_LEVEL[2:0]	Non-Secure User Mode	Non-Secure Spvr Mode	Secure (TZ) User Mode	Secure (TZ) Spvr Mode	CSL register value
(5) 101	None	None	None	RD+WR	8'b0010_0010
(6) 110	None	None	RD	RD	8'b0000_0011
(7) 111	None	None	None	None	Any other value

Chapter 21
DPLL Controller (DPLLC)

21.1 Overview

The Digital Phase-Locked Loop Control (DPLLC) is the control interface to a DPLL. It generates the control signals required for the DPLL operations.

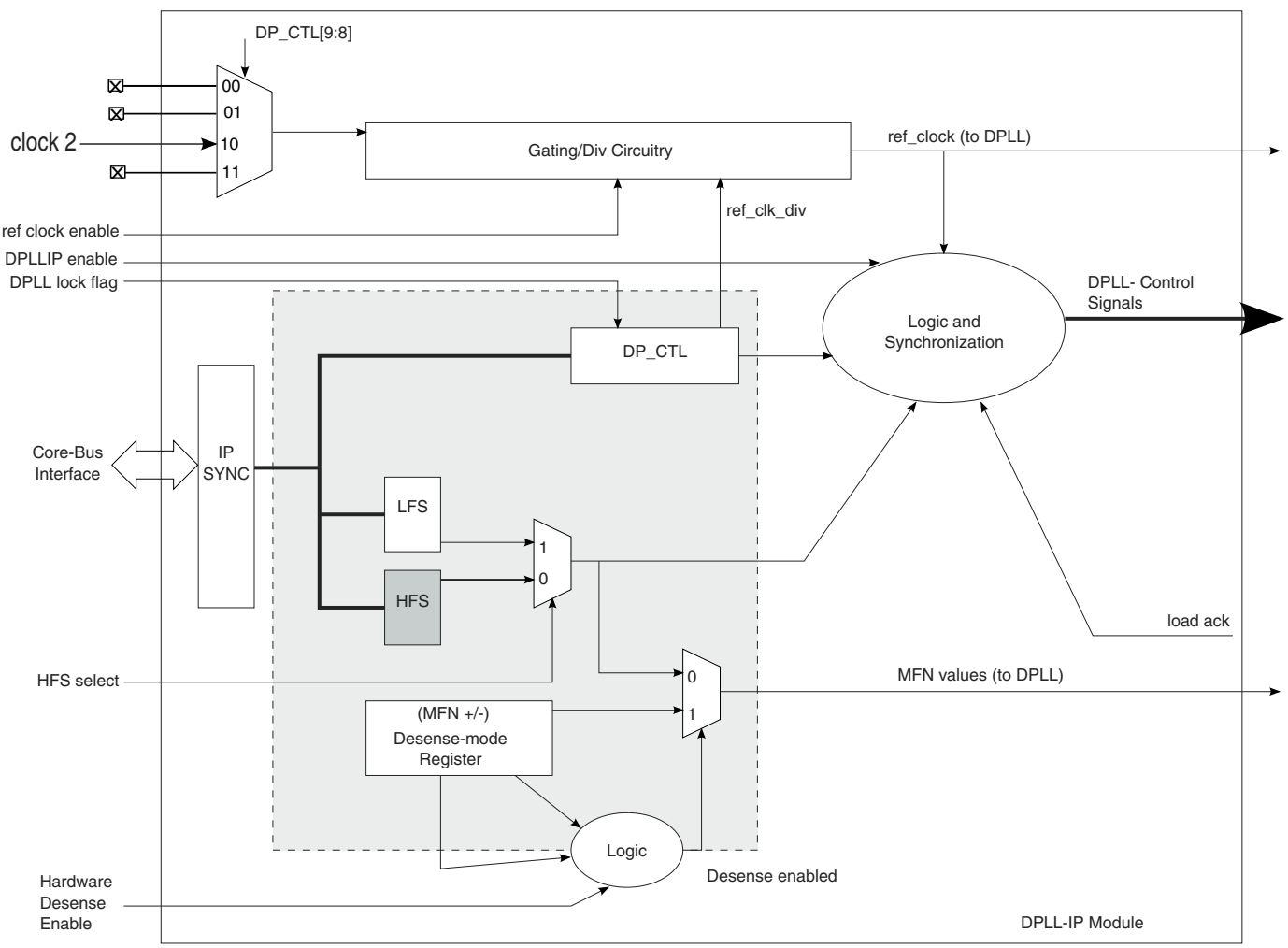


Figure 21-1. Block Diagram of the DPLLC

21.1.1 Feature Description

Key features of this module include:

- Controls the operation of DPLLs.
- Provides controlled phase modulation of clocks to reduce receiver desensitization using desense circuit for DPLLs.
- Selects clock2 input reference clock for DPLL from DPLLC_DP_CTL[ref_clk_sel]. All other setups are Reserved and shall not be changed. See DPLLC Control Register (DPLLC_CTL).
- If auto restart is enabled (AREN bit of the DPLL Config Register is '1'), a restart sequence is issued automatically whenever one of the DPLL registers (DPLLC_DP_CTL, DP_OP, DPLLC_DP_MFD or DPLLC_DP_HFS_OP, DP_HFS_MFD) is written. Otherwise, software needs to set the restart bit of DPLL Control Register manually.

21.1.2 Modes and Operation

The DPLLC supports the following modes of operation:

- Normal Mode
- DPLL Desense Mode
- DVFS Support: HFS Mode

Refer to Operations for DPLLC operations.

21.2 Functional Description

The DPLLC module is mainly divided in the following blocks as shown in Figure 21-1.

- Register read/write: Core bus reads and writes the registers of the module.
- Reference clock selection circuitry: This circuitry comprises of clock selection mux, gating circuit and divide by 2 logic. The reference clock is selected from one of the four external input clocks based on register configuration. This reference clock goes to DPLL. See Figure 21-2 for more details.
- DPLL restart logic and synchronization of data: Based on input signal, registers corresponding to low frequency or high frequency modes are selected. Data goes to the DPLL port, is synchronized on the reference clock.
- Desense mode logic: Desense mode is enabled based on DPLLC configuration. The output to DPLL toggles between predefined values (configured in

which determines the DPLL reference clock frequency. See DPLL Desense Mode for mode details.

- Desense mode logic: Desense mode is enabled based on DPLLC configuration. The output to DPLL toggles between predefined values (configured in DPLLC registers) which determines the DPLL reference clock frequency. See DPLL Desense Mode for mode details.

21.2.1 Operating Modes

21.2.1.1 Normal Mode

In DPLLC normal mode out of the four external input clocks, one is selected depending upon DPLLC register configuration. This is followed by a divider circuitry and then by a gating circuitry as shown below. If reference clock enable signal is HIGH, the clock is passed, else it is gated to a value'1'. Refer to Figure 21-2 for details.

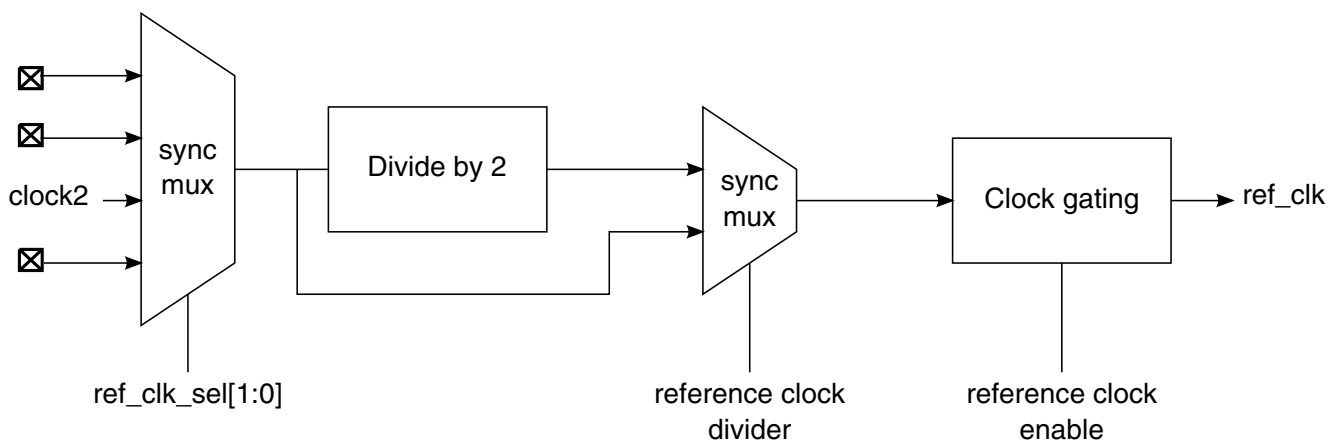


Figure 21-2. Clock MUXing and Gating/Divider Circuitry

21.2.1.2 DPLL Desense Mode

Radiation of clock harmonics causes receiver desensitization. Controlled phase modulation of the clocks coming out of the DPLL will reduce this desensitization. The desense circuit when enabled provides a MFN value that toggles between two programmed MFN_PLUS and MFN_MINUS values for a programmed toggle frequency. This will help achieve the phase modulation of the DPLL clocks.

The disable bit in the DPLLC_DP_MFN_TOGC register should be clear in order to enable the desense logic. Once the disable bit is cleared, desense can be enabled by programming the TOG_EN bit of the DPLLC_DP_MFN_TOGC register for the corresponding PLL. (The desense logic can be activated either by writing to TOG_EN, or by asserting hardware desense enable signal).

Once desense is enabled, the DPLLC module provides the DPLL with a MFN value that switches between MFN_PLUS and MFN_MINUS for n number of correct reference clock cycles. MFN_PLUS and MFN_MINUS are programmed by writing to DPLLC_DP_MFNPLUS and DPLLC_DP_MFNMINUS registers. TOG_CNT (Bits [15:0] of the DPLLC_DP_MFN_TOGC register) determines the toggle rate n. The ref_pll receives a MFN value equal to MFN_PLUS for the first n/2 cycles and MFN_MINUS value for the remaining n/2 cycles.

In the middle of the dither sequence, if the desense mode is disabled through toggle enable (disable) bit or desense hardware request, the circuit completes the ongoing MFN_PLUS, MFN_MINUS cycle before exiting the desense mode. PLL runs on normal MFN after the completion of desense mode. If desense disables and enables again in the middle of the sequence, toggle counter will not stop the current cycle. Only even values for the counter are allowed. If the counter is programmed to an odd value, the counter value will default to a value one less from the odd value.

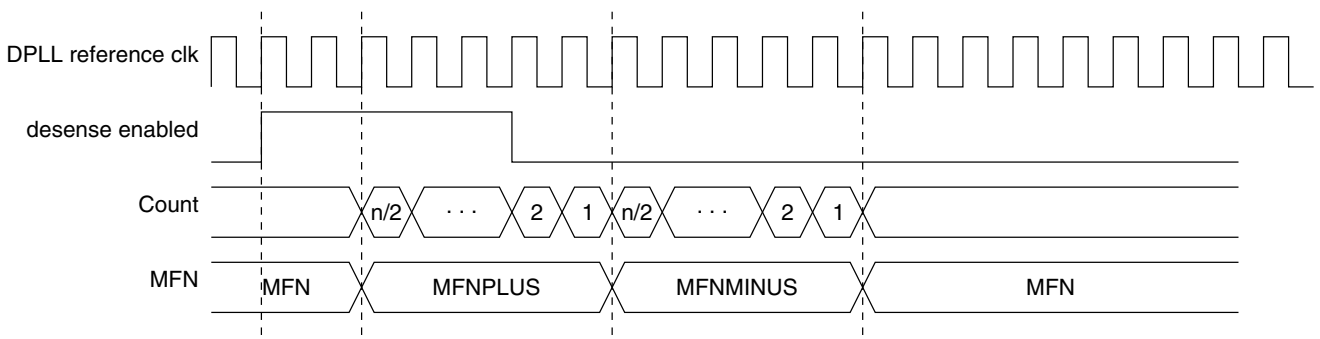


Figure 21-3. Desense Mode MFN Timing

When in desense mode, the DPLLC generates an automatic load request to the PLL whenever the MFN value changes. This signal stays asserted till the load acknowledge signal from the PLL is received. The load request asserts for a minimum of 3 reference clock cycles. Hence for a counter value of 6, where the MFN value toggles between MFN_PLUS and MFN_MINUS every 3 reference clock cycles, the load request may not reflect the updated MFN value and therefore, there is a possibility that the new MFN values are not loaded. Hence it is recommended that the counter be programmed for value greater than 8.

It is required that the toggle counter values not change after the toggle enable is asserted. If toggle counter value changes, the functioning of the desense circuit may be unpredictable.

When any of the 3 MFN values, that is, MFN_PLUS, MFN_MINUS, or MFN values change after the desense mode is enabled, they are buffered till the desense mode is disabled. Therefore to update the MFN_PLUS and MFN_MINUS from the next sequence, desense must be disabled and enabled again.

The desense status register DPLLC Desense Status Register (DPLLC_DESTAT) shows the status of the toggle_enable and the MFN value sent to the DPLL.

21.2.1.3 DVFS Support: HFS Mode

DPLL supports DVFS settings through an alternate set of registers which can be used to store high frequency settings corresponding to high voltage. This alternate set of HFS registers is used to restart the PLL with high frequency settings stored in HFS registers whenever system detects a voltage change from low to high level. Following diagram describes this configuration.

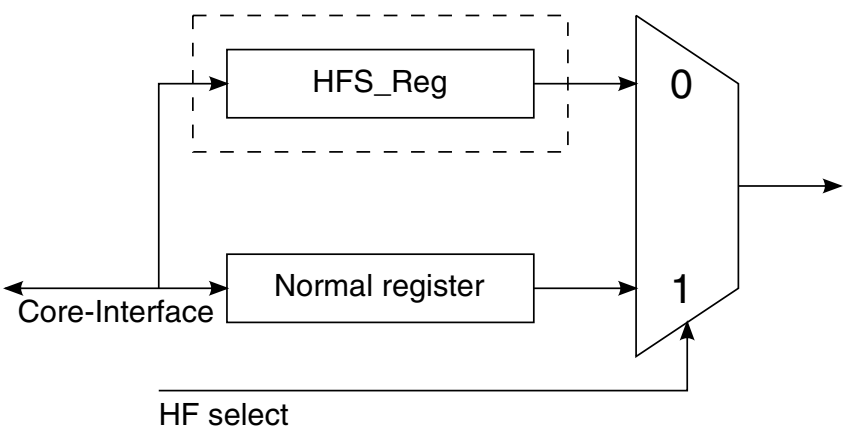


Figure 21-4. DVFS Support for High Frequency Settings

21.2.2 Operations

21.2.2.1 DPLLC_DP_CTL, DPLLC_DP_OP, DPLLC_DP_MFD Register Update

The DPLLs require special timing considerations when updating their ports. Whenever one of the DPLL control register DPLLC_DP_CTL, DPLLC_DP_OP, or DPLLC_DP_MFD, or the shadow registers DPLLC_DP_HFS_OP or DPLLC_DP_HFS_MFD is written (even if the bits don't change), a restart is issued to the DPLL before the actual control bits are sent to the DPLL. The restart assertion and port update will be synchronous to the reference input clock of the DPLL. Writing to any of these registers will cause following procedure to come into effect.

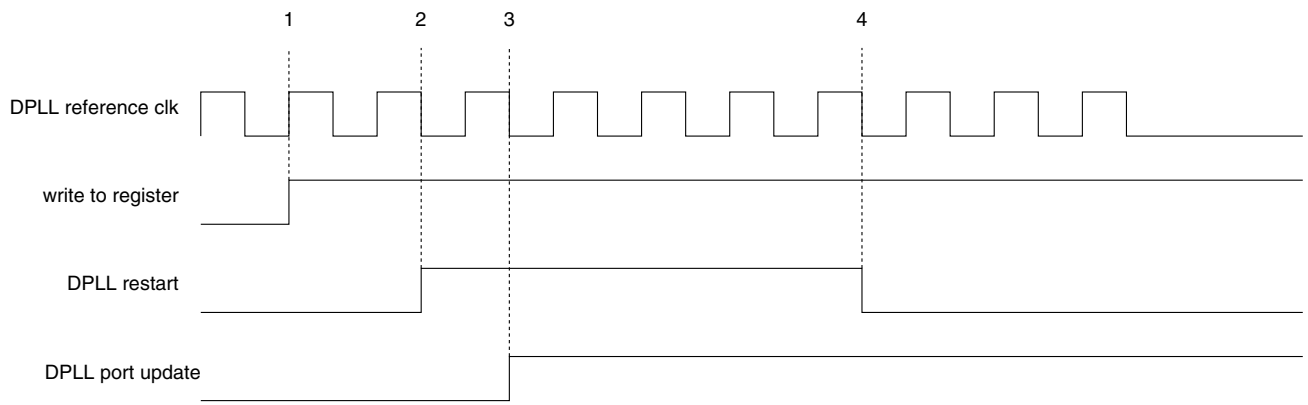


Figure 21-5. DPLL Port Timings

1. Register is written.
2. One clock cycle later restart is asserted.
3. One clock cycle later the appropriate port is updated.
4. Four clock cycles later restart is released.

NOTE

Setting the RST bit of DPLL Control Register will cause the restart (crstrt) signal to assert immediately. In case AREN bit of DPLL Config Register is LOW, the programmer will have to take care of restart of DPLL by setting the RST bit of DPLL Control Register appropriately.

21.2.2.2 DP_MFN Register Update

The MFN bits of DP_MFN register are the only bits in the DPLL that can be changed after the DPLL has been locked without a restart. As shown in diagram below, after writing a new value to MFN, the procedure to update the DPLL is as follows:

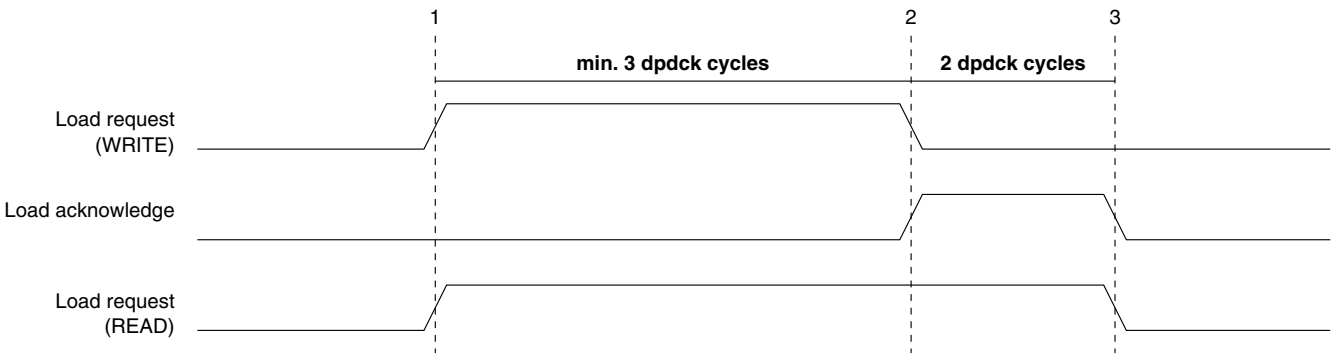


Figure 21-6. MFN Load Request Timings

1. The control bit LDREQ in DP_CONFIG is set. This sends a load request signal to the DPLL.
2. When the DPLL loads the new MFN, it sends a load acknowledge signal back to the DPLL-IP module and clears the write value of load request. At this point load request becomes a status bit, reflecting the state of the acknowledge signal.

NOTE

This write value of load request is a "sticky" bit. When set, it will remain at '1' until a load acknowledgement pulse is received.

3. The read value of load request is automatically cleared when the DPLL has completed updating MFN and has removed the load acknowledge signal, allowing the next load request to take place.

21.2.2.3 Multiple Options for DPLL Control

The DPLLC module provides multiple options for controlling the ON/OFF logic of the DPLLs from the Core and the DSM, making it possible to switch off the PLL by software and still be able to switch it on by DSM. Mul_ctrl (Bit [13]) of DPLLC_DP_CTL register provides this flexibility as shown in the following table.

Table 21-1. Options for DPLL Control

DPLLC Enable	UPEN Bit	mul_ctrl bit (Self Clearing)	DPLL Enable
X	X	1	0
X	0	X	0
posedge	1	0	1
0	X	X	0

As shown in the table above, once the DPLL is switched off by Software using the Mul_ctrl bit, the software cannot turn the PLL on thereafter, unless there is a posedge on the DPLLC enable signal, during DSM entry sequence.

21.2.2.4 Calculating the Output Frequency

The output frequency is given by the formula:

$$f_{dck=2} = 4 \bullet f_{ref} \bullet \left[\frac{\text{MultiplicationFactor}}{\text{PreDividerFactor}} \right]$$

Where:

$$\text{MultiplicationFactor} = MF_I + \frac{MF_N}{MF_D + 1}$$

$$\text{PreDividerFactor} = PDF + 1$$

For normal mode, registers DPLLC_DP_OP, DPLLC_DP_MFN, and DPLLC_DP_MFD are used.

For HFS mode, the shadow registers DPLLC_DP_HFS_OP, DPLLC_DP_HFS_MFN, and DPLLC_DP_HFS_MFD are used.

21.3 Programmable Registers

21.3.1 DPLLC Memory Map/Register Definition

DPLLC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63F8_0000	DPLLC Control Register (DPLLC-1_CTL)	32	R/W	00AA_0222h	21.31.1/ 969
63F8_0004	DPLLC Configuration Register (DPLLC-1_CONFIG)	32	R/W	00AA_0006h	21.31.2/ 972
63F8_0008	DPLLC Operation Register (DPLLC-1_OP)	32	R/W	00AA_0000h	21.31.3/ 973
63F8_000C	DPLLC Multiplication Factor Denominator Register (DPLLC-1_MFD)	32	R/W	00AA_0000h	21.31.4/ 974
63F8_0010	DPLLC Multiplication Factor Numerator Register (DPLLC-1_MFN)	32	R/W	00AA_0000h	21.31.5/ 975
63F8_0014	DPLLC Multiplication Factor Numerator PLUS/MINUS Registers (DPLLC-1_MFNMINUS)	32	R/W	0000_0000h	21.31.6/ 976
63F8_0018	DPLLC Multiplication Factor Numerator PLUS/MINUS Registers (DPLLC-1_MFNPLUS)	32	R/W	0000_0000h	21.31.6/ 976
63F8_001C	DPLLC High Frequency Support, Operation Register (DPLLC-1_HFS_OP)	32	R/W	00AA_0000h	21.31.7/ 977
63F8_0020	DPLLC High Frequency Support Multiplication Factor Denominator Register (DPLLC-1_HFS_MFD)	32	R/W	00AA_0000h	21.31.8/ 978
63F8_0024	DPLLC High Frequency Support Multiplication Factor Numerator Register (DPLLC-1_HFS_MFN)	32	R/W	00AA_0000h	21.31.9/ 978
63F8_0028	DPLLC Multiplication Factor Numerator Toggle Control Register (DPLLC-1_MFN_TOGC)	32	R/W	00AA_0000h	21.31.10/ 979
63F8_002C	DPLLC Desense Status Register (DPLLC-1_DESTAT)	32	R	00AA_0000h	21.31.11/ 981
63F8_4000	DPLLC Control Register (DPLLC-2_CTL)	32	R/W	00AA_0222h	21.31.1/ 969
63F8_4004	DPLLC Configuration Register (DPLLC-2_CONFIG)	32	R/W	00AA_0006h	21.31.2/ 972
63F8_4008	DPLLC Operation Register (DPLLC-2_OP)	32	R/W	00AA_0000h	21.31.3/ 973
63F8_400C	DPLLC Multiplication Factor Denominator Register (DPLLC-2_MFD)	32	R/W	00AA_0000h	21.31.4/ 974
63F8_4010	DPLLC Multiplication Factor Numerator Register (DPLLC-2_MFN)	32	R/W	00AA_0000h	21.31.5/ 975
63F8_4014	DPLLC Multiplication Factor Numerator PLUS/MINUS Registers (DPLLC-2_MFNMINUS)	32	R/W	0000_0000h	21.31.6/ 976
63F8_4018	DPLLC Multiplication Factor Numerator PLUS/MINUS Registers (DPLLC-2_MFNPLUS)	32	R/W	0000_0000h	21.31.6/ 976
63F8_401C	DPLLC High Frequency Support, Operation Register (DPLLC-2_HFS_OP)	32	R/W	00AA_0000h	21.31.7/ 977

Table continues on the next page...

DPLLC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63F8_4020	DPLLC High Frequency Support Multiplication Factor Denominator Register (DPLLC-2_HFS_MFD)	32	R/W	00AA_0000h	21.31.8/ 978
63F8_4024	DPLLC High Frequency Support Multiplication Factor Numerator Register (DPLLC-2_HFS_MFN)	32	R/W	00AA_0000h	21.31.9/ 978
63F8_4028	DPLLC Multiplication Factor Numerator Toggle Control Register (DPLLC-2_MFN_TOGC)	32	R/W	00AA_0000h	21.31.10/ 979
63F8_402C	DPLLC Desense Status Register (DPLLC-2_DESTAT)	32	R	00AA_0000h	21.31.11/ 981
63F8_8000	DPLLC Control Register (DPLLC-3_CTL)	32	R/W	00AA_0222h	21.31.1/ 969
63F8_8004	DPLLC Configuration Register (DPLLC-3_CONFIG)	32	R/W	00AA_0006h	21.31.2/ 972
63F8_8008	DPLLC Operation Register (DPLLC-3_OP)	32	R/W	00AA_0000h	21.31.3/ 973
63F8_800C	DPLLC Multiplication Factor Denominator Register (DPLLC-3_MFD)	32	R/W	00AA_0000h	21.31.4/ 974
63F8_8010	DPLLC Multiplication Factor Numerator Register (DPLLC-3_MFN)	32	R/W	00AA_0000h	21.31.5/ 975
63F8_8014	DPLLC Multiplication Factor Numerator PLUS/MINUS Registers (DPLLC-3_MFNMINUS)	32	R/W	0000_0000h	21.31.6/ 976
63F8_8018	DPLLC Multiplication Factor Numerator PLUS/MINUS Registers (DPLLC-3_MFNPLUS)	32	R/W	0000_0000h	21.31.6/ 976
63F8_801C	DPLLC High Frequency Support, Operation Register (DPLLC-3_HFS_OP)	32	R/W	00AA_0000h	21.31.7/ 977
63F8_8020	DPLLC High Frequency Support Multiplication Factor Denominator Register (DPLLC-3_HFS_MFD)	32	R/W	00AA_0000h	21.31.8/ 978
63F8_8024	DPLLC High Frequency Support Multiplication Factor Numerator Register (DPLLC-3_HFS_MFN)	32	R/W	00AA_0000h	21.31.9/ 978
63F8_8028	DPLLC Multiplication Factor Numerator Toggle Control Register (DPLLC-3_MFN_TOGC)	32	R/W	00AA_0000h	21.31.10/ 979
63F8_802C	DPLLC Desense Status Register (DPLLC-3_DESTAT)	32	R	00AA_0000h	21.31.11/ 981
63F8_C000	DPLLC Control Register (DPLLC-4_CTL)	32	R/W	00AA_0222h	21.31.1/ 969
63F8_C004	DPLLC Configuration Register (DPLLC-4_CONFIG)	32	R/W	00AA_0006h	21.31.2/ 972
63F8_C008	DPLLC Operation Register (DPLLC-4_OP)	32	R/W	00AA_0000h	21.31.3/ 973
63F8_C00C	DPLLC Multiplication Factor Denominator Register (DPLLC-4_MFD)	32	R/W	00AA_0000h	21.31.4/ 974

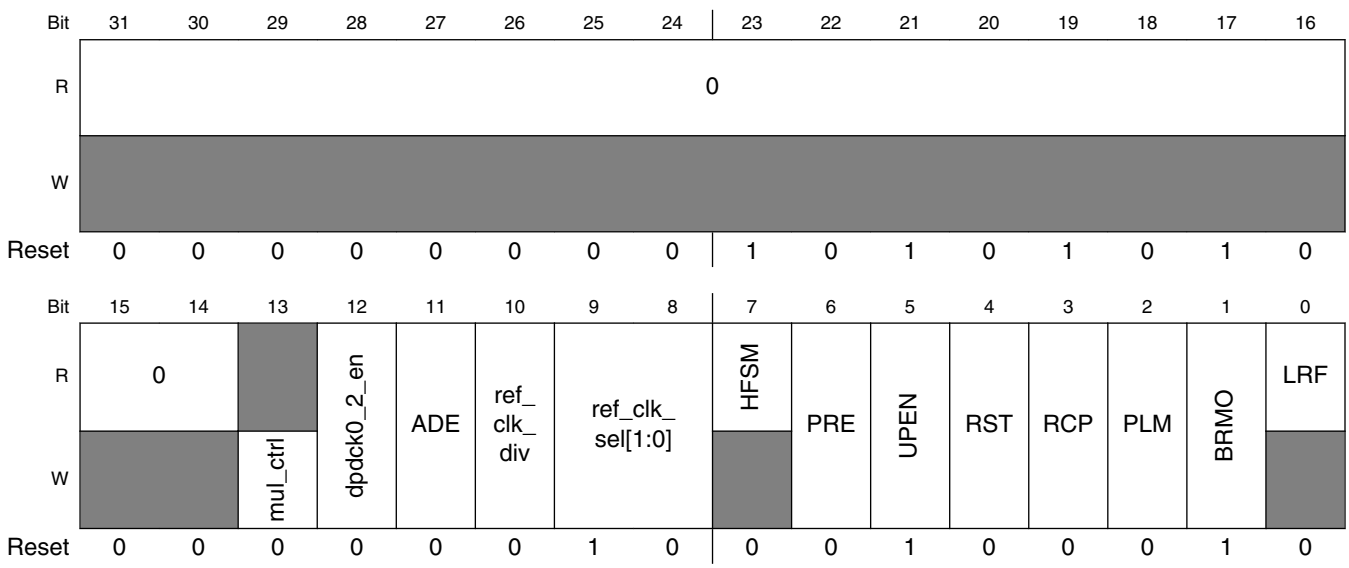
Table continues on the next page...

DPLLC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63F8_C010	DPLLC Multiplication Factor Numerator Register (DPLLC-4_MFN)	32	R/W	00AA_0000h	21.31.5/ 975
63F8_C014	DPLLC Multiplication Factor Numerator PLUS/MINUS Registers (DPLLC-4_MFNMINUS)	32	R/W	0000_0000h	21.31.6/ 976
63F8_C018	DPLLC Multiplication Factor Numerator PLUS/MINUS Registers (DPLLC-4_MFNPLUS)	32	R/W	0000_0000h	21.31.6/ 976
63F8_C01C	DPLLC High Frequency Support, Operation Register (DPLLC-4_HFS_OP)	32	R/W	00AA_0000h	21.31.7/ 977
63F8_C020	DPLLC High Frequency Support Multiplication Factor Denominator Register (DPLLC-4_HFS_MFD)	32	R/W	00AA_0000h	21.31.8/ 978
63F8_C024	DPLLC High Frequency Support Multiplication Factor Numerator Register (DPLLC-4_HFS_MFN)	32	R/W	00AA_0000h	21.31.9/ 978
63F8_C028	DPLLC Multiplication Factor Numerator Toggle Control Register (DPLLC-4_MFN_TOGC)	32	R/W	00AA_0000h	21.31.10/ 979
63F8_C02C	DPLLC Desense Status Register (DPLLC-4_DESTAT)	32	R	00AA_0000h	21.31.11/ 981

21.31.1 DPLLC Control Register (DPLLCx_CTL)

Addresses: DPLLC-1_CTL is 63F8_0000h base + 0h offset = 63F8_0000h
DPLLC-2_CTL is 63F8_4000h base + 0h offset = 63F8_4000h
DPLLC-3_CTL is 63F8_8000h base + 0h offset = 63F8_8000h
DPLLC-4_CTL is 63F8_C000h base + 0h offset = 63F8_C000h



DPLLCx_CTL field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 mul_ctrl	Multiple Control. This is a self clearing bit cleared on the next posedge of PLL reference clock. Setting this bit deasserts the enable signal to the DPLL. Using this bit instead of UPEN to disable the DPLL gives DSM the control to independently switch on the DPLL. See Multiple Options for DPLL Control for more details. 0 No Effect 1 DPLL disabled
12 dpdck0_2_en	dpdck0_2 Enable. To enable PLL output before divide by 2 flip-flop. It will enable clock dpdck0_2 which will have double the frequency of dpdck/dpgdck clock. 0 Disable dpdck0_2. 1 Enable dpdck0_2.
11 ADE	The output corresponding to this bit is left unconnected for future use.
10 ref_clk_div	Ref Clk Division factor. Decides whether selected reference clock input needs to be divided by two or one. 0 Divide by 1 1 Divide by 2
9–8 ref_clk_sel[1:0]	ref_clock_select. Used to select the reference clock for DPLL. **NOTE:** Bitfield values 00, 01 and 11 are Reserved and should not be used. 00 Reserved. 01 Reserved. 10 clock2 is selected (24MHz oscillator clock). 11 Reserved.
7 HFSM	HFS-Mode Status bit. Indicates which frequency mode of operation is active. In HFS mode all the shadow registers (DPLLC_DP_HSF_OP, DPLLC_DP_HSF_MFD, DPLLC_DP_HSF_MFN) are used whereas in LFS mode normal registers are used. Reset value of this bit is '0'. 0 LFS mode, normal registers are read. 1 HFS mode is active.
6 PRE	Power Up Reset. Asserts a hard reset to DPLL. 0 Reset cleared. 1 Reset asserted
5 UPEN	PLL Enable. Enables DPLL operation. 0 DPLL disabled. 1 DPLL enabled.
4 RST	Restart. Restarts DPLL. It should not be used as a status bit.

Table continues on the next page...

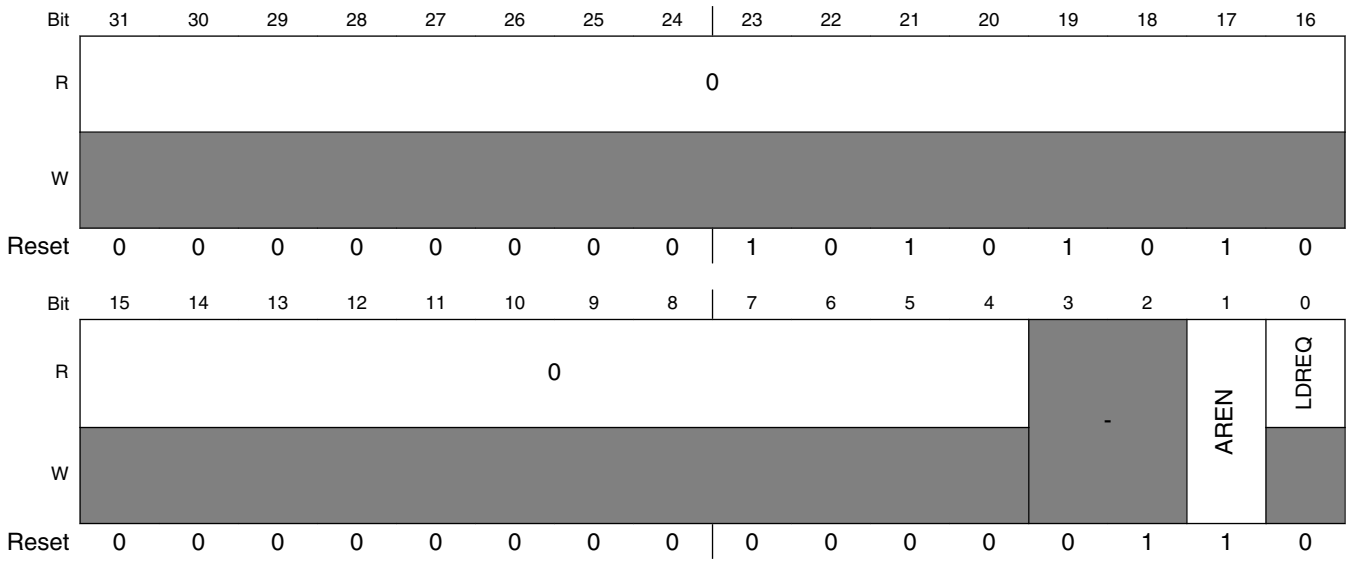
DPLLCx_CTL field descriptions (continued)

Field	Description
	0 DPLL not in restart. 1 DPLL restarted.
3 RCP	Reference Clock Polarity. Selects which edge the chip clock is adjusted to. 0 Positive edge of reference clock. 1 negative edge of reference clock.
2 PLM	Phase Lock Mode. Selects if phase is to be considered (in addition to frequency) during lock-in. 0 DPLL in frequency only lock mode. 1 DPLL in frequency and phase lock mode.
1 BRMO	BRM Order Bit. Binary Rate Modulator order 0 BRM in first order. 1 BRM in second order.
0 LRF	Lock Ready Flag. Shows if the DPLL is in lock. This is a sticky bit. Following events will break the already established lock: (1) DPLLC_DP_CTL, DPLLC_DP_OP, DPLLC_DP_MFD, DPLLC_DP_HSF_OP or DPLLC_DP_HSF_MFD is written (2) Hard reset (3) pll_lvs toggle (4) DPLL enable Reset value of this bit is '0' 0 DPLL not locked. 1 DPLL locked.

21.31.2 DPLLC Configuration Register (DPLLCx_CONFIG)

Writing to this register does not generate an automatic 'restart.'

Addresses: DPLLC-1_CONFIG is 63F8_0000h base + 4h offset = 63F8_0004h
DPLLC-2_CONFIG is 63F8_4000h base + 4h offset = 63F8_4004h
DPLLC-3_CONFIG is 63F8_8000h base + 4h offset = 63F8_8004h
DPLLC-4_CONFIG is 63F8_C000h base + 4h offset = 63F8_C004h



DPLLCx_CONFIG field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value zero. Reserved
3–2 -	Reserved. **NOTE** For FSL testing purposes only. Bit shouldn't be altered and must remain in its default value.
1 AREN	Auto Restart Enable. If auto restart is enabled, then a restart sequence will be issued automatically whenever a write is performed to one of the DPLL registers (DPLLC_DP_CTL, DPLLC_DP_OP, DPLLC_DP_MFD) in normal mode. In HFS mode, same is true if a write is performed to one of the HFS mode registers (DPLLC_DP_CTL, DPLLC_DP_HFS_OP or DPLLC_DP_HFS_MFD). Otherwise, software will need to set the restart bit manually. Bit 4 of DPLL Control Register. Default Value of this bit is '1'. 0 Auto-restart is disabled 1 Auto-restart is enabled
0 LDREQ	Load Request. Notifies the DPLL that the numerator factor of the fractional part (MFN) has changed. This bit is cleared by an acknowledge from the DPLL. Writing a '0' to it has no effect. Default Value of this bit is '0'

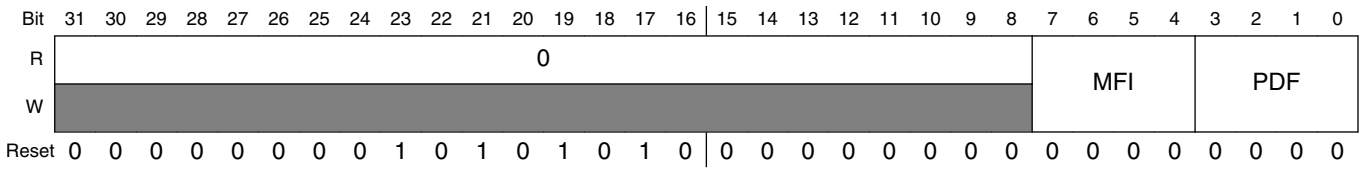
Table continues on the next page...

DPLLCx_CONFIG field descriptions (continued)

Field	Description
	0 DPLL has finished updating MFN. 1 Request that the DPLL updates MFN.

21.31.3 DPLLC Operation Register (DPLLCx_OP)

Addresses: DPLLC-1_OP is 63F8_0000h base + 8h offset = 63F8_0008h
DPLLC-2_OP is 63F8_4000h base + 8h offset = 63F8_4008h
DPLLC-3_OP is 63F8_8000h base + 8h offset = 63F8_8008h
DPLLC-4_OP is 63F8_C000h base + 8h offset = 63F8_C008h

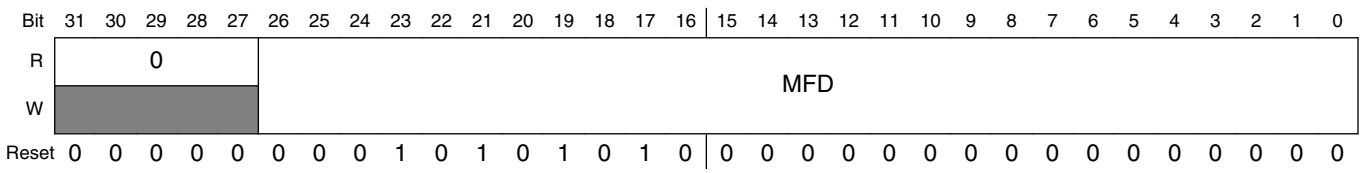


DPLLCx_OP field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–4 MFI	Multiplication Factor, Integer part (MFI). If MFI[3:0] is written with a value of less than 5, MFI will default to 5. See Calculating the Output Frequency Example settings: 0000 Integer part equals 5 0001 Integer part equals 5 0101 Integer part equals 5 0110 Integer part equals 6 1111 Integer part equals 15
3–0 PDF	Pre-division Factor (PDF). PDF determines the value of the PreDividerFactor used in the output frequency formula. See Calculating the Output Frequency Example settings: 0000 PreDividerFactor equals 1 0001 PreDividerFactor equals 2 0010 PreDividerFactor equals 3 1110 PreDividerFactor equals 15 1111 PreDividerFactor equals 16

21.31.4 DPLLC Multiplication Factor Denominator Register (DPLLCx_MFD)

Addresses: DPLLC-1_MFD is 63F8_0000h base + Ch offset = 63F8_000Ch
DPLLC-2_MFD is 63F8_4000h base + Ch offset = 63F8_400Ch
DPLLC-3_MFD is 63F8_8000h base + Ch offset = 63F8_800Ch
DPLLC-4_MFD is 63F8_C000h base + Ch offset = 63F8_C00Ch



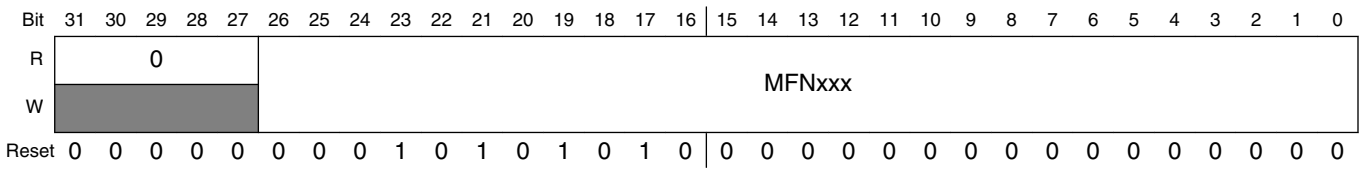
DPLLCx_MFD field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26–0 MFD	Multiplication Factor Denominator Bits. The MFD[26:0] bits, in a 2's complement format, give the denominator of the fractional part. See Calculating the Output Frequency **NOTE:** Bit 26 is always 0. MFD should be in a range from 0 to 67108863; otherwise, the output clock frequency will differ from the desired frequency. Example settings: 0x000000 MFD equals 0 0x000001 MFD equals 1 0x000002 MFD equals 2 0x3FFFFFE MFD equals 67108862 0x3FFFFFF MFD equals 67108863

21.31.5 DPLLC Multiplication Factor Numerator Register (DPLLCx_MFN)

MFNxxx denotes the registers DPLLC_DP_MFN, DPLLC_DP_MFNMINUS and DPLLC_DP_MFNPLUS.

Addresses: DPLLC-1_MFN is 63F8_0000h base + 10h offset = 63F8_0010h
DPLLC-2_MFN is 63F8_4000h base + 10h offset = 63F8_4010h
DPLLC-3_MFN is 63F8_8000h base + 10h offset = 63F8_8010h
DPLLC-4_MFN is 63F8_C000h base + 10h offset = 63F8_C010h



DPLLCx_MFN field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26–0 MFNxxx	Multiplication Factor Numerator Bits (nominal, "MINUS", or "PLUS"). The MFNxxx[26:0] bits, in a 2's complement format, give the numerator of the fractional part. See Calculating the Output Frequency **NOTE:** MFN should be in a range from -67108864 to 67108863. If the absolute value of the MFN is larger than MFD, the output clock frequency will differ from the desired frequency. Example settings: 0x4000000 MFN equals -67108864 0x4000001 MFN equals -67108863 0x4000002 MFN equals -67108862 0xFFFFFFF MFN equals -1 0x0000000 MFN equals 0 (default) 0x0000001 MFN equals 1 0x3FFFFFE MFN equals 67108862 0x3FFFFFF MFN equals 67108863

21.31.6 DPLLC Multiplication Factor Numerator PLUS/MINUS Registers (DPLLC*x*_MFN*n*)

MFNxxx denotes the registers DPLLC_DP_MFN, DPLLC_DP_MFNMINUS and DPLLC_DP_MFNPLUS.

Addresses: DPLLC-1_MFNMINUS is 63F8_0000h base + 14h offset = 63F8_0014h
DPLLC-1_MFNPLUS is 63F8_0000h base + 18h offset = 63F8_0018h
DPLLC-2_MFNMINUS is 63F8_4000h base + 14h offset = 63F8_4014h
DPLLC-2_MFNPLUS is 63F8_4000h base + 18h offset = 63F8_4018h
DPLLC-3_MFNMINUS is 63F8_8000h base + 14h offset = 63F8_8014h
DPLLC-3_MFNPLUS is 63F8_8000h base + 18h offset = 63F8_8018h
DPLLC-4_MFNMINUS is 63F8_C000h base + 14h offset = 63F8_C014h
DPLLC-4_MFNPLUS is 63F8_C000h base + 18h offset = 63F8_C018h

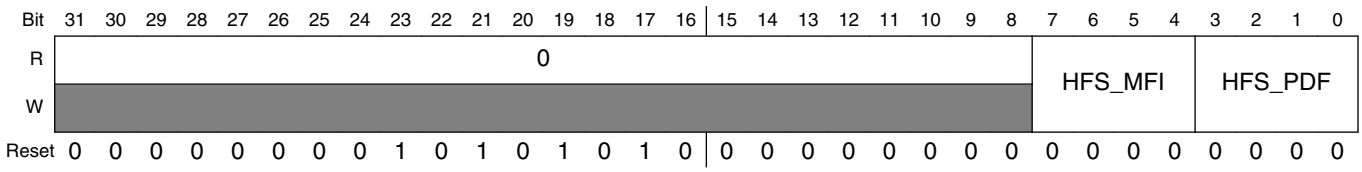
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																MFNxxx															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DPLLC*x*_MFN*n* field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26–0 MFNxxx	Multiplication Factor Numerator Bits (nominal, "MINUS", or "PLUS"). The MFNxxx[26:0] bits, in a 2's complement format, give the numerator of the fractional part. See Calculating the Output Frequency **NOTE:** MFN should be in a range from -67108864 to 67108863. If the absolute value of the MFN is larger than MFD, the output clock frequency will differ from the desired frequency. Example settings: 0x4000000 MFN equals -67108864 0x4000001 MFN equals -67108863 0x4000002 MFN equals -67108862 0xFFFFFFF MFN equals -1 0x0000000 MFN equals 0 (default) 0x0000001 MFN equals 1 0x3FFFFFE MFN equals 67108862 0x3FFFFFF MFN equals 67108863

21.31.7 DPLLC High Frequency Support, Operation Register (DPLLCx_HFS_OP)

Addresses: DPLLC-1_HFS_OP is 63F8_0000h base + 1Ch offset = 63F8_001Ch
DPLLC-2_HFS_OP is 63F8_4000h base + 1Ch offset = 63F8_401Ch
DPLLC-3_HFS_OP is 63F8_8000h base + 1Ch offset = 63F8_801Ch
DPLLC-4_HFS_OP is 63F8_C000h base + 1Ch offset = 63F8_C01Ch

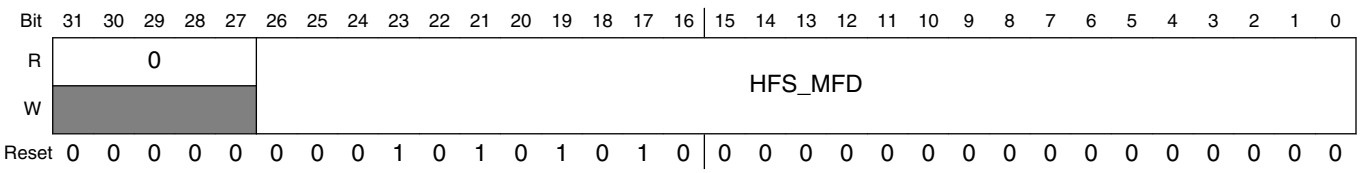


DPLLCx_HFS_OP field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved.
7–4 HFS_MFI	HFS Mode-Multiplication Factor, Integer part (MFI). If HFS_MFI[3:0] is written with a value of less than 5, MFI will default to 5. See Calculating the Output Frequency Example settings: 0000 Integer part equals 5 0101 Integer part equals 5 0110 Integer part equals 6 1111 Integer part equals 15
3–0 HFS_PDF	HFS mode-Pre-division Factor (HFS_PDF). HFS_PDF determines the value of the PreDividerFactor used in the output frequency formula. See Calculating the Output Frequency Example settings: 0000 PreDividerFactor equals 1 0001 PreDividerFactor equals 2 0010 PreDividerFactor equals 3 1110 PreDividerFactor equals 15 1111 PreDividerFactor equals 16

21.31.8 DPLLC High Frequency Support Multiplication Factor Denominator Register (DPLLCx_HFS_MFD)

Addresses: DPLLC-1_HFS_MFD is 63F8_0000h base + 20h offset = 63F8_0020h
DPLLC-2_HFS_MFD is 63F8_4000h base + 20h offset = 63F8_4020h
DPLLC-3_HFS_MFD is 63F8_8000h base + 20h offset = 63F8_8020h
DPLLC-4_HFS_MFD is 63F8_C000h base + 20h offset = 63F8_C020h

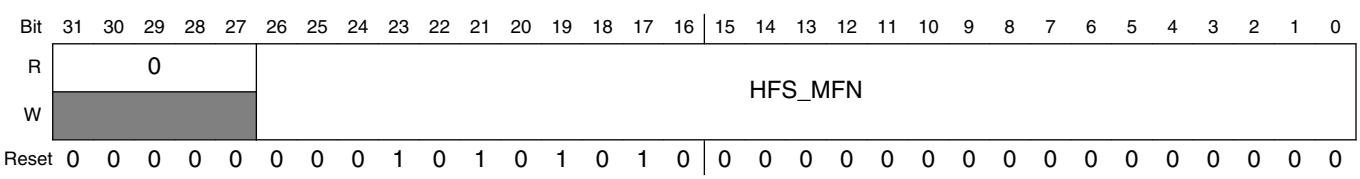


DPLLCx_HFS_MFD field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26–0 HFS_MFD	Multiplication Factor Denominator Bits. The HFS_MFD[26:0] bits, in a 2's complement format, give the denominator of the fractional part. See Calculating the Output Frequency **NOTE:** Bit 26 is always 0. HFS_MFD should be in a range from 0 to 67108863 otherwise the output clock frequency will differ from the desired frequency. Example settings: 0x000000 MFD equals 0 0x000002 MFD equals 1 0x000010 MFD equals 2 0x3FFFFFE MFD equals 67108862 0x3FFFFFF MFD equals 67108863

21.31.9 DPLLC High Frequency Support Multiplication Factor Numerator Register (DPLLCx_HFS_MFN)

Addresses: DPLLC-1_HFS_MFN is 63F8_0000h base + 24h offset = 63F8_0024h
DPLLC-2_HFS_MFN is 63F8_4000h base + 24h offset = 63F8_4024h
DPLLC-3_HFS_MFN is 63F8_8000h base + 24h offset = 63F8_8024h
DPLLC-4_HFS_MFN is 63F8_C000h base + 24h offset = 63F8_C024h

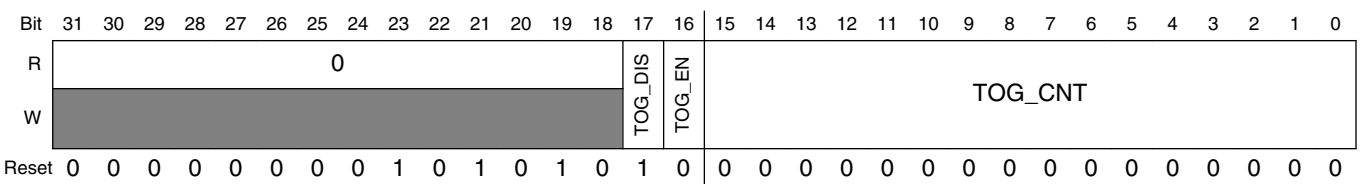


DPLLCx_HFS_MFN field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–0 HFS_MFN	Multiplication Factor Numerator Bits. The HFS_MFN[26:0] bits, in a 2's complement format, give the numerator of the fractional part. See Calculating the Output Frequency **NOTE:** HFS_MFN should be in a range from -67108864 to 67108863. If the absolute value of the HFS_MFN is larger than HFS_MFD, the output clock frequency will differ from the desired frequency. Example settings: 0x4000000 MFN equals -67108864 0x4000001 MFN equals -67108863 0x4000002 MFN equals -67108862 0xFFFFFFF MFN equals -1 0x0000000 MFN equals 0 (default) 0x0000001 MFN equals 1 0x3FFFFFE MFN equals 67108862 0x3FFFFFF MFN equals 67108863

21.31.10 DPLLC Multiplication Factor Numerator Toggle Control Register (DPLLCx_MFN_TOGC)

Addresses: DPLLC-1_MFN_TOGC is 63F8_0000h base + 28h offset = 63F8_0028h
DPLLC-2_MFN_TOGC is 63F8_4000h base + 28h offset = 63F8_4028h
DPLLC-3_MFN_TOGC is 63F8_8000h base + 28h offset = 63F8_8028h
DPLLC-4_MFN_TOGC is 63F8_C000h base + 28h offset = 63F8_C028h



DPLLCx_MFN_TOGC field descriptions

Field	Description
31–18 Reserved	This read-only field is reserved and always has the value zero. Reserved.
17 TOG_DIS	Desense Global Disable. If TOG_DIS is set, then the desense logic for the DPLL is disabled, and software (TOG_EN) or hardware desense requests to enable the logic will be ignored. (default value of this bit is '1')

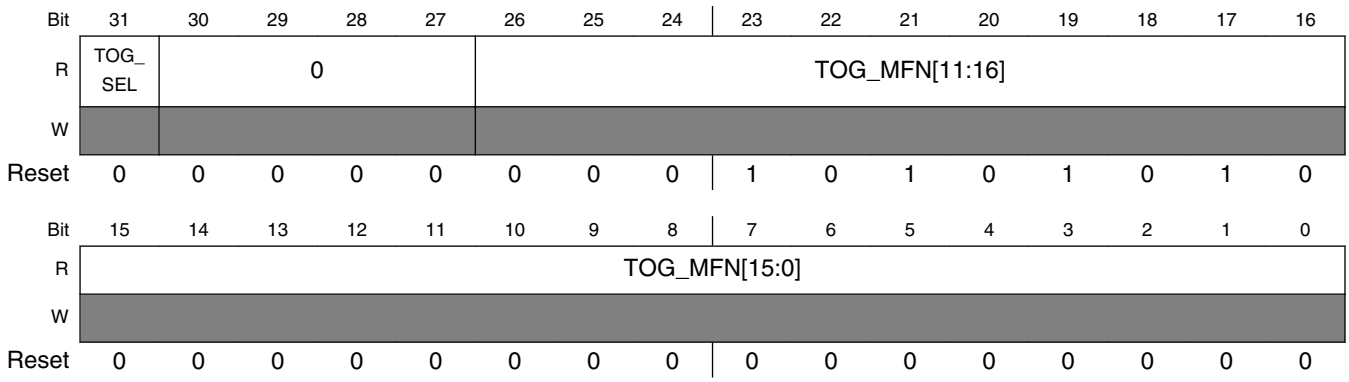
Table continues on the next page...

DPLLCx_MFN_TOGC field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Hardware</th><th>TOG_
EN</th><th>TOG_
DIS</th><th>Desense logic</th></tr>
<tr><td>x</td><td>x</td><td>1</td><td>OFF</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>OFF</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>on</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>on</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>on</td></tr>
</table>
</td></tr>
<tr><td>16
TOG_EN</td><td>Desense On. Software request to activate desense logic.
(default value of this bit is '0')
NOTE: Desense logic may also be activated by a hardware request ( )
<table>
<tr><th>Hardware</th><th>TOG_
EN</th><th>TOG_
DIS</th><th>Desense logic</th></tr>
<tr><td>x</td><td>x</td><td>1</td><td>OFF</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>OFF</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>on</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>on</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>on</td></tr>
</table>
</td></tr>
<tr><td>15–0
TOG_CNT</td><td>Toggle counter value. If the value programmed for the count is odd, the next lowest even value is used (original value minus 1).
(default value of tog_counter is '0')
Example settings:
0x0000 Count value equals 0
0x0001 Count value equals 0
0x0002 Count value equals 2
0x0003 Count value equals 2</td></tr>
</table>

21.31.11 DPLLC Desense Status Register (DPLLCx_DESTAT)

Addresses: DPLLC-1_DESTAT is 63F8_0000h base + 2Ch offset = 63F8_002Ch
DPLLC-2_DESTAT is 63F8_4000h base + 2Ch offset = 63F8_402Ch
DPLLC-3_DESTAT is 63F8_8000h base + 2Ch offset = 63F8_802Ch
DPLLC-4_DESTAT is 63F8_C000h base + 2Ch offset = 63F8_C02Ch



DPLLCx_DESTAT field descriptions

Field	Description
31 TOG_SEL	Toggle set status 0 Desense circuit is inactive. 1 Desense circuit is active.
30–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–0 TOG_MFN	Indicates the current MFN value that is sent to the DPLL in desense mode.

Chapter 22
Dynamic Voltage and Frequency Scaling Core (DVFSC)

22.1 Introduction

The DVFSC allows simple dynamic voltage frequency scaling. The frequency of the core clock domain and the voltage of the core power domain can be changed on the fly while all blocks (including the ARM platform) continue their normal operation. The frequency of the core clock domain can be changed by switching temporally to an alternate PLL clock, and then get back to the updated PLL, already locked at a specific frequency, or by merely changing the post dividers division factors.

The DVFSC load tracking block allows hardware tracking on the core load and a generation of an interrupt when a frequency change is requested.

DVFSC is a sub-block of the General Power Controller (GPC). See GPC chapter for more details.

NOTE

DVFSC is a monitor that only provides an interrupt when counting exceeds the predefined value and doesn't actually send request to make a change a change of voltage and frequency. This can be done by the user in the interrupt routine or smart direct memory access (SDMA) code by using GPC or clock control module (CCM) (relocking the PLL or changing the post dividers at the CCM).

22.1.1 Overview

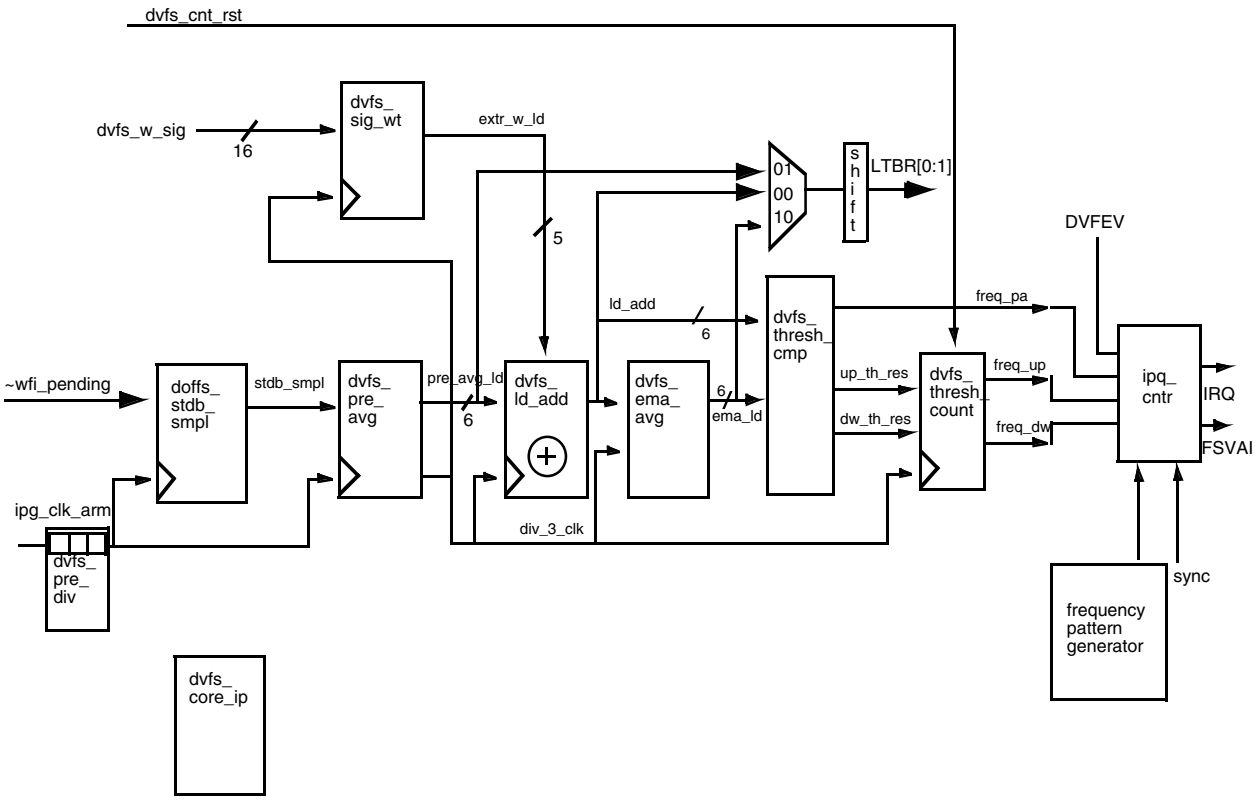


Figure 22-1. Block Diagram

22.1.2 Features

The DVFSC load tracking sub-block includes the following features:

- Configurable include/exclude of input signals:
  - ARM standby signal (idle / non-idle)
  - 16 general purpose bits
    - Configurable weight of each bit.
- Configurable generated clocks and averaging time slicing (respond time).
- Configurable panic mode respond logic (for frequency up).
- Programmable buffer for last 4,8,12, or 16 load tracking samples. Based on value in LBCF in DVFSC_CNTR. There is also a buffer full signal - LBFL.

22.2 Functional Description of DVFS Load Tracking

The DVFS load tracking sub-block includes the following features:

Configurable include/exclude of input signals:

- ARM standby signal (idle / non-idle)
- 16 general purpose load tracking signals
- Configurable weight and (level-sensitive) of GeP signals.
- Configurable generated clocks and averaging time slicing (respond time).
- Configurable panic mode respond logic (for frequency up).
- Programmable buffer for last 4,8,12, or 16 load tracking samples. Two Load Tracking Buffers are working in ping-pong mode with two buffer full signals - LBFL1, LBFL0.

22.3 Component Blocks Description

22.3.1 dvfs_stdb_smpl Block

This block samples the ccm_arm_stdb signal (ARM11 STANDBYWFI signal - idle state indicating) according to the edge of the ipg_clk_arm (ARM11 system clock) signal.

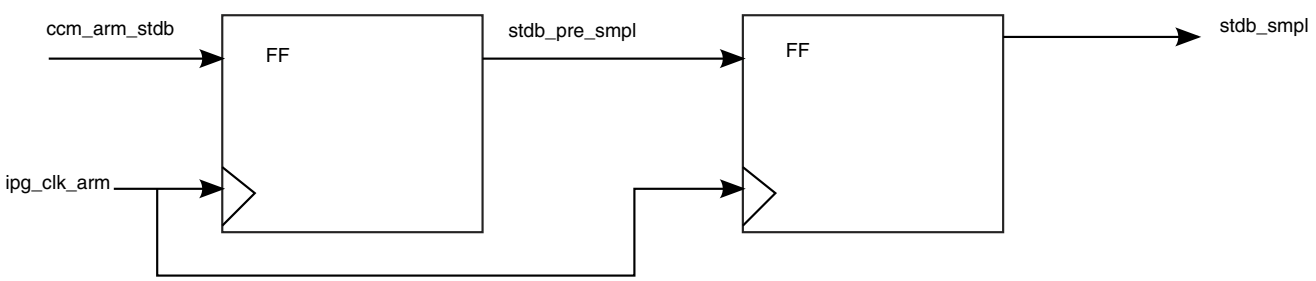


Figure 22-2. dvfs_stdb_smpl block diagram

This block synchronizes the ccm_arm_stdb signal with the ipg_clk_arm clock. The two signals enter the Flip-Flops (twice), and by doing so, are synchronized. The resulting synchronized signal is stdb_smpl.

22.3.2 dvfs_sig_wt Block

The purpose of this block is to sample the 16 GeP (general purpose) load signals, multiply each one of them by appropriate weight and sum products. The sampling is done by the slow clock div_3_clk. These signals have the only options of detection: level sampling.

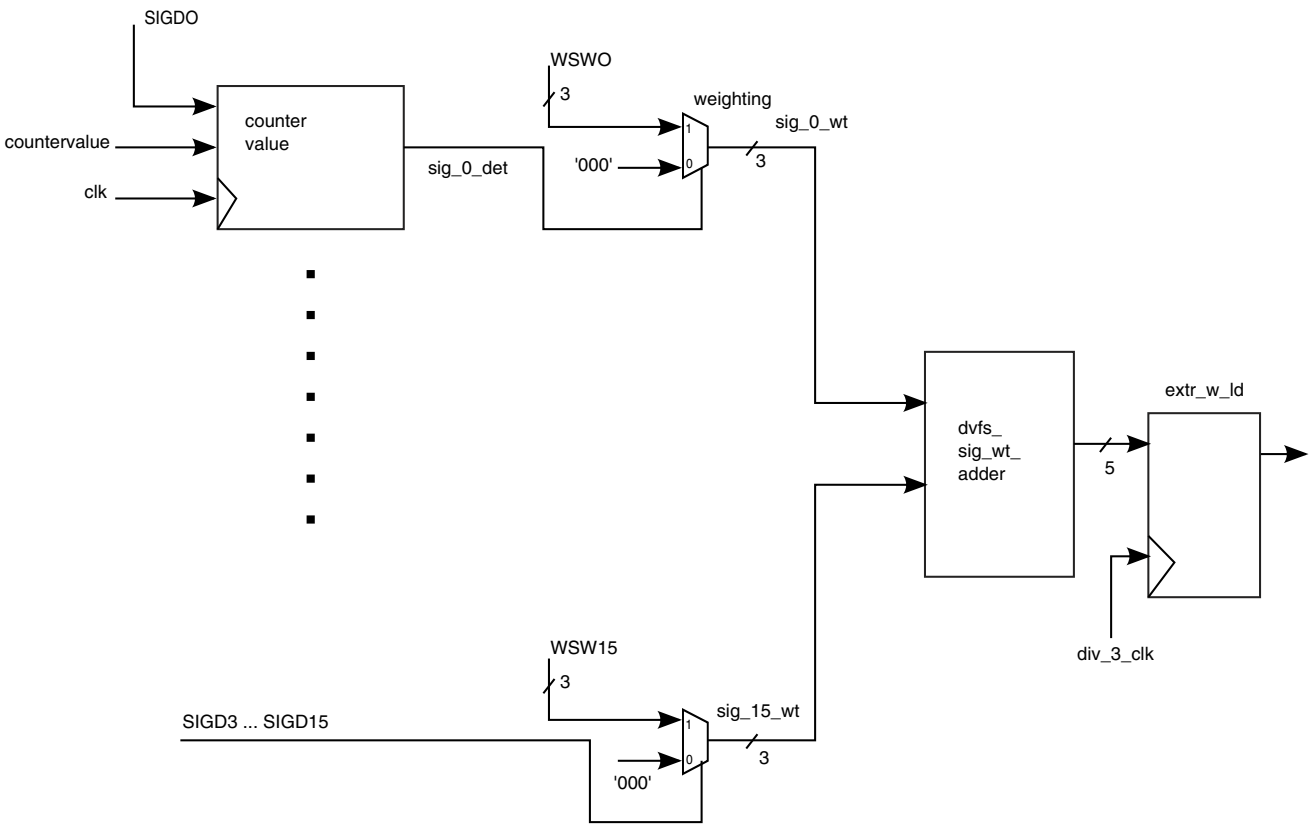


Figure 22-3. dvfs_sig_wt block diagram

There are two paths for weighted general purpose load signals to be calculated.

22.3.3 dvfs_pre_avg Block

The purpose of the dvfs_pre_avg block is to perform simple, non-overlapping averaging, reducing the sampling clock frequency and providing a level-based average index of the tracked ARM Platform load.

External signals:

- stdb_sampl - (input) sampled ARM11 standby signal
- ipg_clk_arm - (input) ARM11 system clk
- pre_avg_ld[4..0] - (output) averaged load

- div_3_clk - (output) clock, generated (reduced) from div_2_clk
- div_2_clk - (output) clock, generated (reduced) from div_1_clk
- div_1_clk - (output) clock, generated (reduced) from ipg_clk_arm

In the current implementation, the averaging is performed in three stages:

1. The first stage IDLE is sampled by divided sys_clock.
2. The second stage (counter2) performs an averaging operation with constant parameters. The counter is an nine (9) bit adder whose output is sampled by the div_2_clk clock. Five (5) highest bits of the sum_2 signal are passed to counter3 (equal to shift right operation).
3. The third (last) stage (counter3) performs an averaging operation with configurable parameters, set by the DIV3CK. The counter3 cell is a fifteen(15) bit adder with output sampled by div_3_clk. The Pre_avg_shifter_5 cell passes configurable bits from the sum_3 signal (see table1_3).

The output 5 bits avg_load signal provides a result with a tolerance of ~3%. (supported values range is 0-0.97)

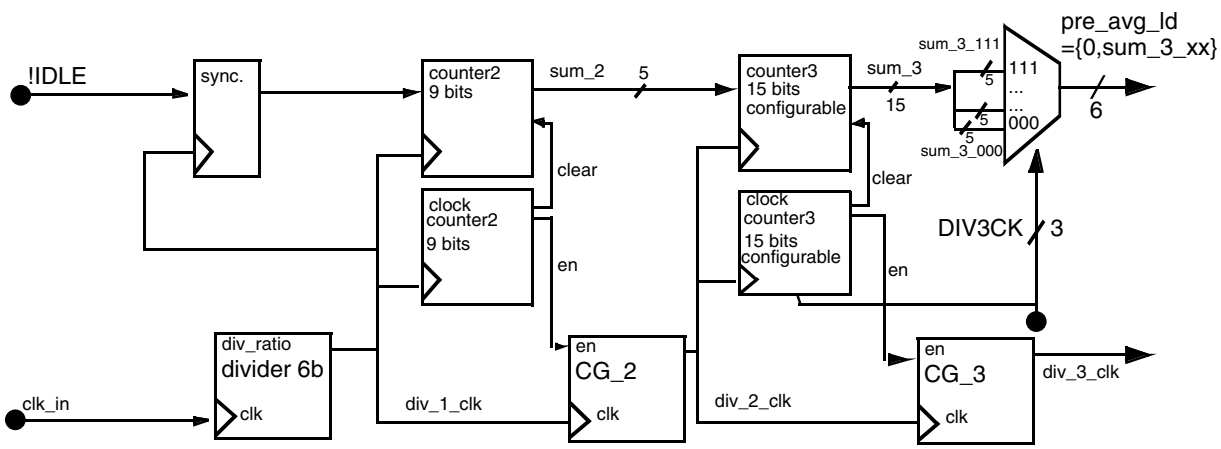


Figure 22-4. dvfs_pre_avg block diagram

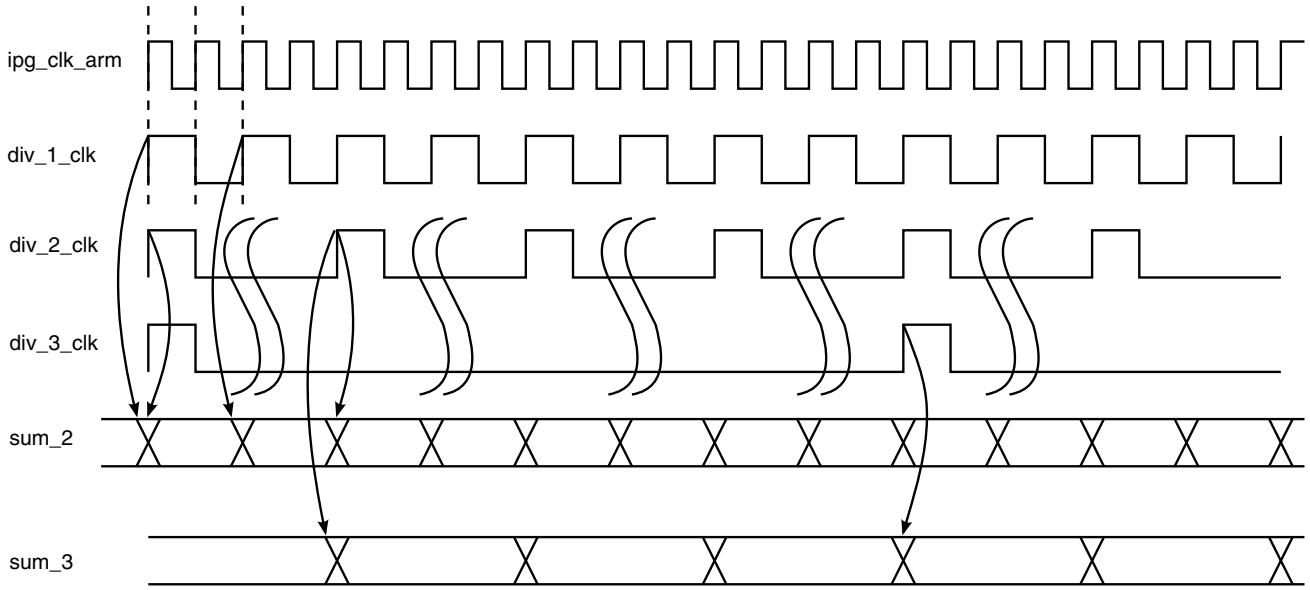


Figure 22-5. dvfs_pre_avg Clocks

The internal clocks in dvfs_avg_pre block are generated from ipg_clk_arm. The details are in the tables below:

Table 22-1. dvfs_pre_avg Signals and its Values

signal	binary max value	binary min value	decimal max value	decimal min value
sum_2	11111	0000	31	0
sum_3	111'1100'0000'0000	000'0000'0000'0000	31744	0
pre_avg_load	11111	0000	31	0

Table 22-2. dvfs_pre_avg generated clocks

clock name	generated from	ratio to source	ratio to ipg_clk_arm	max clk freq.	note
ipg_clk_arm	N/A	N/A	1	66MHz	source clock
div_1_clk	ipg_clk_arm	configurable	configurable	66MHz	configurable
div_2_clk (gated)	div_1_clk	512	configurable	128KHz	none
div_3_clk (gated)	div_2_clk	configurable	configurable	128KHz	configurable

Table 22-3. iv_3_clk Configurable Averaging

DIV3CK setting	dividing ratio	sum_3 passing bits	div_1_clk cumulative divider
00	1	4-0	1*512=512

Table continues on the next page...

Table 22-3. iv_3_clk Configurable Averaging (continued)

DIV3CK setting	dividing ratio	sum_3 passing bits	div_1_clk cumulative divider
001	4	6-2	4*512=2048
010	16	8-4	16*512=8192
011	64	10-6	64*512=32768
100	256	12-8	256*512=131072
101	1024	16-10	1024*512=524288

22.3.4 dvfs_ld_add Block

The dvfs_ld_add block sums the ARM Platform load, tracked by idle/non-idle signal and the load, detected from the additional load signals, weighted by signal_weighting block. The adder should perform the following operation:

extr_w_ld[4..0] + {0,pre_avg_ld[4..0]}

The input signals of five (5) bits produce output signal of six (6) bits, providing 3% resolution in the range of 0-1.97 of load indication. (1 is equal to 100% load tracking with no additional signals include). The output ld_add[5..0] is sampled by div_3_clk signal.

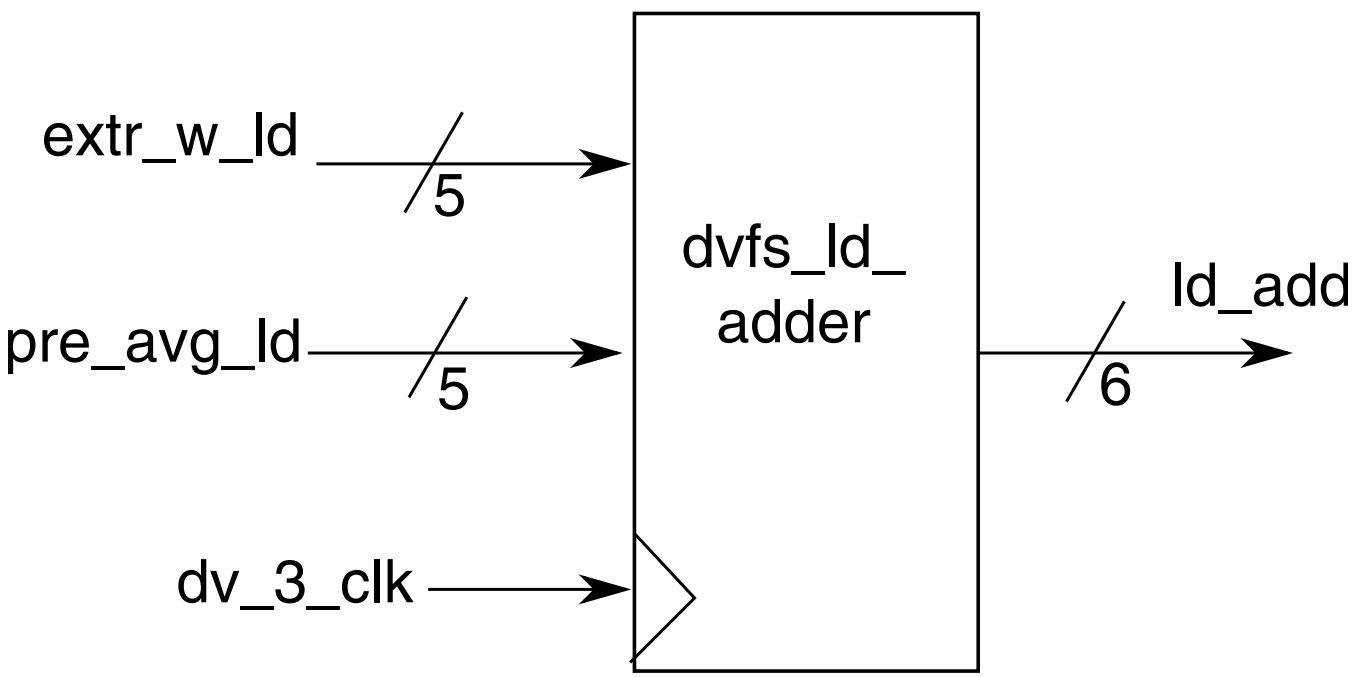


Figure 22-6. dvfs_ld_add block diagram

22.3.5 dvfs_ema_avg Block

The purpose of dvfs_ema_avg (EMA - Exponential Moving Average) block is to calculate an exponential moving average of the tracked ARM Platform load. The parameters of EMA are defined by EMAC field of the DVFSC_EMAC register. These parameters set how many samples of simple average will be taken into account.

The EMA formula is: EMA(i) = a*X(i) + (1-a)*EMA(i-1)

where:

a = 2 / (N+1);

N is the number of samples taken into account.

X(i) = current input sample;

EMA(i-1) = previous value of EMA.

By setting the value of 'a', the behavior of EMA is defined.

In the following table, the parameter 'a' is listed relative to the amount of the X(i) samples taken into account (('N')) in the equation above. (The resolution of the digital multiplier is limited, hence the lowest values of the 'a' can be linked to a range of included samples in EMA instead of single number.)

Also included in the following table is the amount of cycles in which the lower frequency will be masked from the moment the DVFS is enabled (and not between frequency switches), so that enough information about the history can be gained before the frequency is lowered.

EMA settings

samples included in EMA calculation (N)	Number of cycles while lower frequency interrupt is masked	'a' value (decimal)	'a' value, adjusted by binary resolution	'a' value (binary)								
----------		----------	----------	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7	bit 8
1	6	1.000	1.000	1	0	0	0	0	0	0	0	0
2	11	0.667	0.668	0	1	0	1	0	1	0	1	1
3	11	0.500	0.500	0	1	0	0	0	0	0	0	0
4	22	0.400	0.398	0	0	1	1	0	0	1	1	0
5	22	0.333	0.332	0	0	1	0	1	0	1	0	1
6	22	0.286	0.285	0	0	1	0	0	1	0	0	1
7	22	0.250	0.250	0	0	1	0	0	0	0	0	0

Table continues on the next page...

samples included in EMA calculation (N)	Number of cycles while lower frequency interrupt is masked	'a' value (decimal)	'a' value, adjusted by binary resolution	'a' value (binary)								
8	43	0.222	0.223	0	0	0	1	1	1	0	0	1
9	43	0.200	0.199	0	0	0	1	1	0	0	1	1
10	43	0.182	0.180	0	0	0	1	0	1	1	1	0
11	43	0.167	0.168	0	0	0	1	0	1	0	1	1
12	43	0.154	0.152	0	0	0	1	0	0	1	1	1
13	43	0.143	0.141	0	0	0	1	0	0	1	0	1
14	43	0.133	0.133	0	0	0	1	0	0	0	1	0
15	43	0.125	0.125	0	0	0	1	0	0	0	0	0
16	86	0.118	0.117	0	0	0	0	1	1	1	1	0
17	86	0.111	0.109	0	0	0	0	1	1	1	0	0
18	86	0.105	0.105	0	0	0	0	1	1	0	1	1
19	86	0.100	0.102	0	0	0	0	1	1	0	1	0
20	86	0.095	0.094	0	0	0	0	1	1	0	0	0
21	86	0.091	0.090	0	0	0	0	1	0	1	1	1
22	86	0.087	0.086	0	0	0	0	1	0	1	1	0
23	86	0.083	0.082	0	0	0	0	1	0	1	0	1
25	86	0.077	0.078	0	0	0	0	1	0	1	0	0
26	86	0.074	0.074	0	0	0	0	1	0	0	1	1
28	86	0.069	0.070	0	0	0	0	1	0	0	1	0
30	86	0.065	0.066	0	0	0	0	1	0	0	0	1
31	86	0.063	0.063	0	0	0	0	1	0	0	0	0
33	173	0.059	0.059	0	0	0	0	0	1	1	1	1
35	173	0.056	0.055	0	0	0	0	0	1	1	1	0
38	173	0.051	0.051	0	0	0	0	0	1	1	0	1
42	173	0.047	0.047	0	0	0	0	0	1	1	0	0
45	173	0.043	0.043	0	0	0	0	0	1	0	1	1
50	173	0.039	0.039	0	0	0	0	0	1	0	1	0
56	173	0.035	0.035	0	0	0	0	0	1	0	0	1
~64	173	0.031	0.031	0	0	0	0	0	1	0	0	0
~74	256	0.027	0.027	0	0	0	0	0	0	1	1	1
~86	256	0.023	0.023	0	0	0	0	0	0	1	1	0
~100	256	0.020	0.020	0	0	0	0	0	0	1	0	1
~128	256	0.016	0.016	0	0	0	0	0	0	1	0	0

Table continues on the next page...

samples included in EMA calculation (N)	Number of cycles while lower frequency interrupt is masked	'a' value (decimal)	'a' value, adjusted by binary resolution	'a' value (binary)								
~170	512	0.012	0.012	0	0	0	0	0	0	0	1	1
~260	512	0.008	0.008	0	0	0	0	0	0	0	1	0
~500	512	0.004	0.004	0	0	0	0	0	0	0	0	1
N/A		0.000	0.000	0	0	0	0	0	0	0	0	0

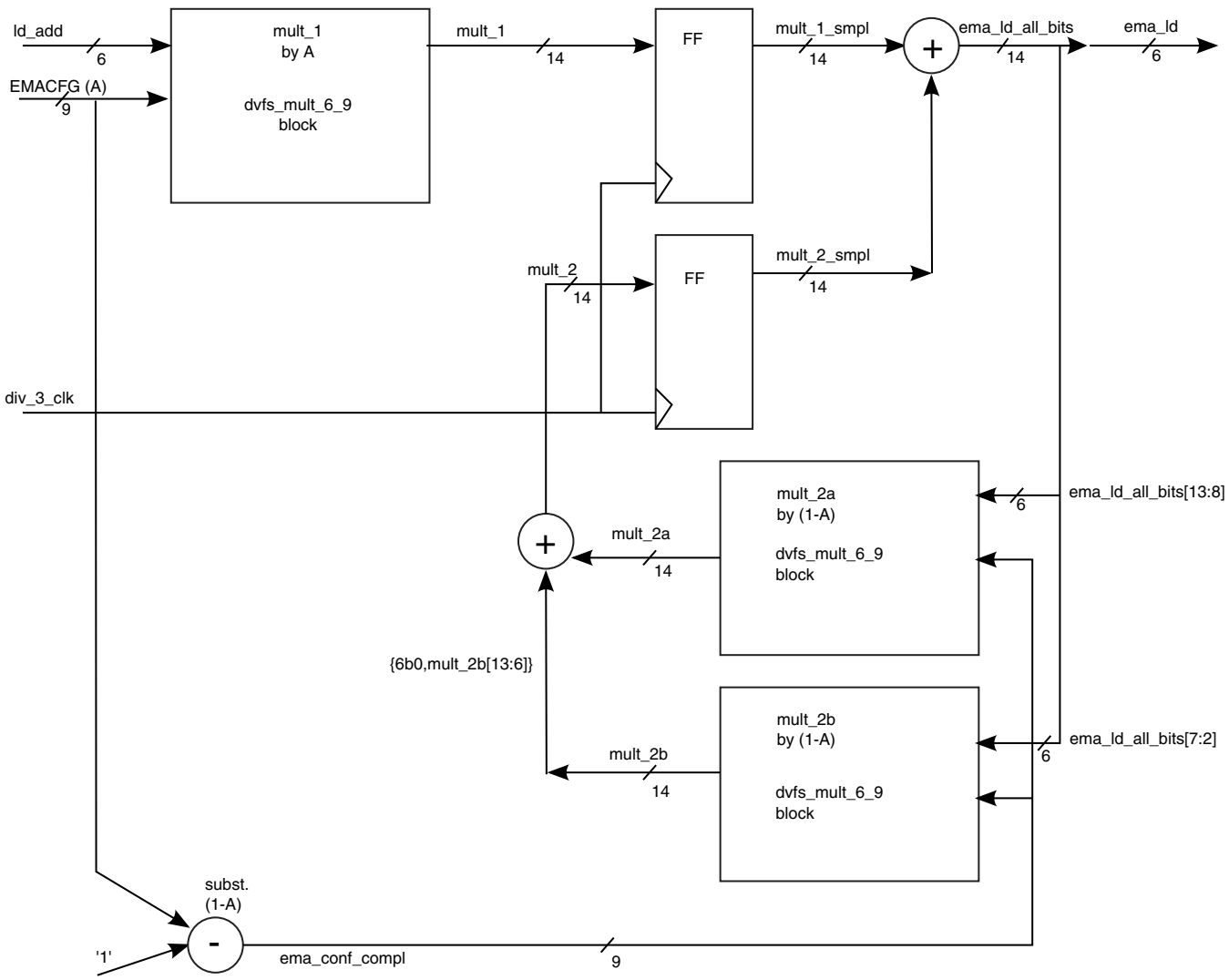


Figure 22-7. dvfs_ema_avg block diagram

The EMA block inputs are as follows:

- ld_add[5..0] - load level
- EMACFG[8..0] - 'a' parameter of EMA algorithm

- div_2_clk - fast clock don't see this in the diagram
- div_3_clk - slow clock

The EMA block outputs the following:

- ema_ld[5..0] - result of EMA algorithm (by div_3_clk)

The mult_a block provides multiplying between comp_load (6 bits) and EMACFG (9 bits). For the multiply operation, a faster clock (for internal sum) is required- this is provided by div_2_clk. Div_2_clk is faster than the div_3_clk by a factor of at least 16, which is enough for 6*9 or 9*6 operations. Only the highest 6 bits are taken from the result of mult_a.

The mult_b block operates in a similar manner.

The adder block's output is sampled with div_3_clk clock.

22.3.6 dvfs_thres_cmp Block

The dvfs_thres_cmp block compares the ARM Platform load value to programmable threshold levels. The comparators as shown in the drawing below are used:

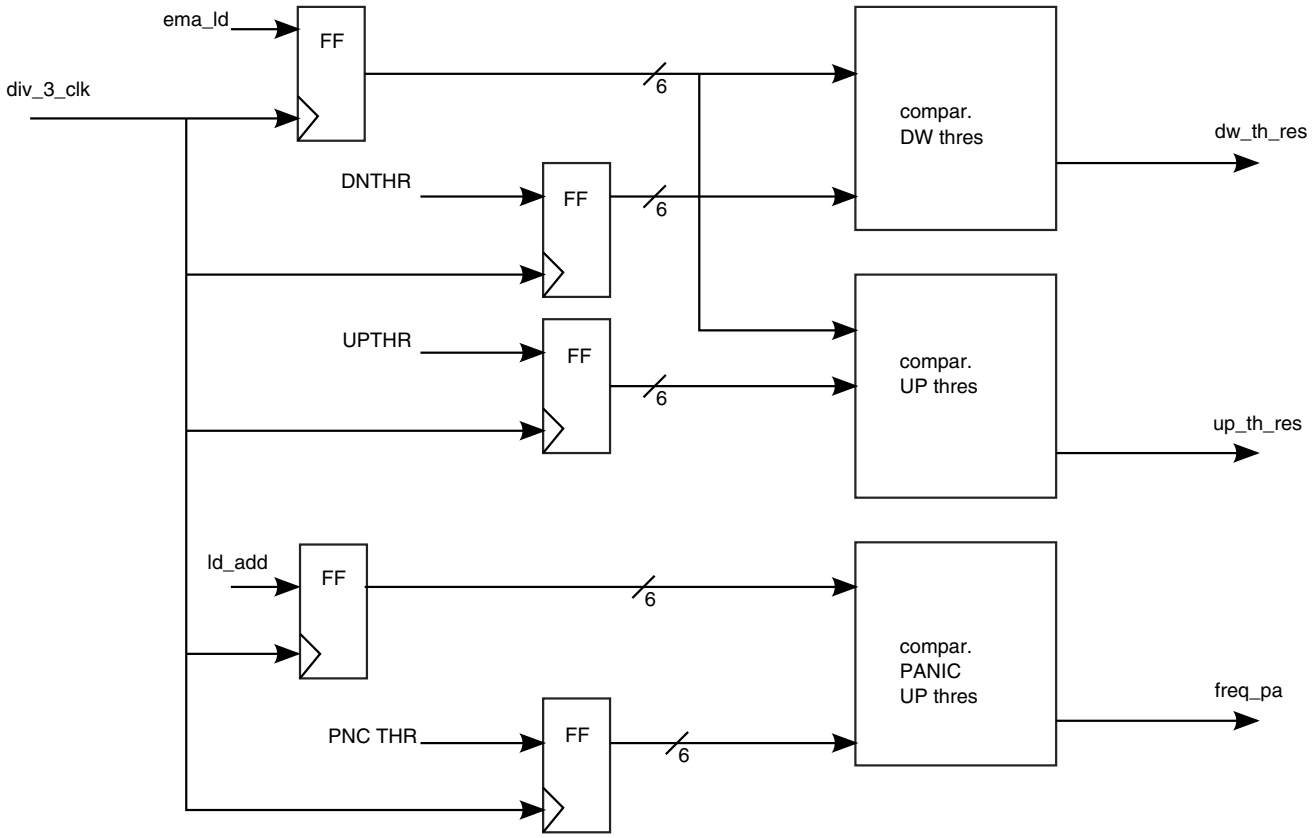


Figure 22-8. dvfs_tresh_cmp Diagram

The block is composed of three (3) comparators:

1. compar_dw_thres comparator, which compares between ema_ld signal (output of EMA block) and DNTHR signal (taken from config register, bits DNTHR).
2. compar_up_thres comparator, which compares between ema_ld signal (output of EMA block) and UPTHR signal (taken from config register, bits UPTHR).
3. compar_panic_up_thres comparator, which compares between ld_add signal (output of load_adder block) and PNCTHR signal (taken from config register, bits PNCTHR).

NOTE

The current implementation of this block enables step-by-step down frequency change. For more advanced option, the number of the DW threshold comparators should be increased (up to three for four-level DVFS).

22.3.7 dvfs_thresh_count Block

The purpose of the dvfs_thres_count block is to count consecutive threshold overcomes of dw_th_res and up_th_res (outputs of threshold_comp block). If any of the counters (see Figure 22-9 below) receives a null (zero) input synchronous with the clk3 signal, the counter is reset.

If the counter reaches a user defined value, its output is set to an active level. These counters are reset each time frequency scaling occurs or if the threshold overcomes are consecutive.

This block's output signals are saved in configuration register 0, bits [3,2].

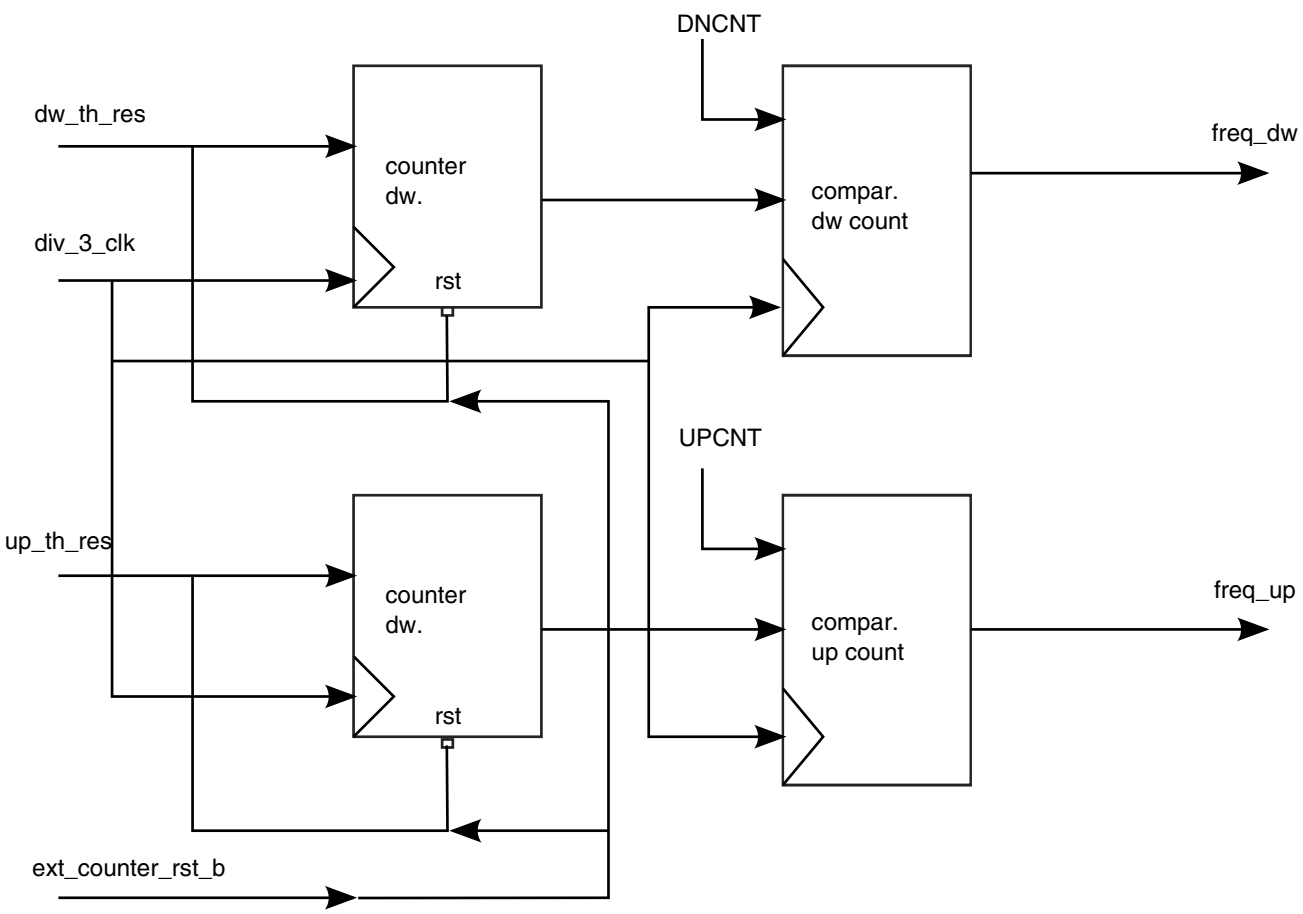


Figure 22-9. Thresh_counters block diagram

22.3.8 Load Tracking Buffer Register

The purpose of the load tracking buffer register is to save last 16 samples of the tracked load (before EMA operation). Hence, the 4 bits of ld_add signal (depending on the LTBRSH) are saved continuously, overwriting each time the latest sample. Each save is carried upon detecting an edge of the div_3_clk signal.

22.3.9 Frequency Pattern Generator

Frequency pattern generator is able to manage the frequency update requests periodically, additionally to main frequency update requests (if bit FPUE is set).

The periodic requests are created following the DVFSPT0, DVFSPT1, DVFSPT2 and DVFSPT3 register values, as described in Figure 22-10.

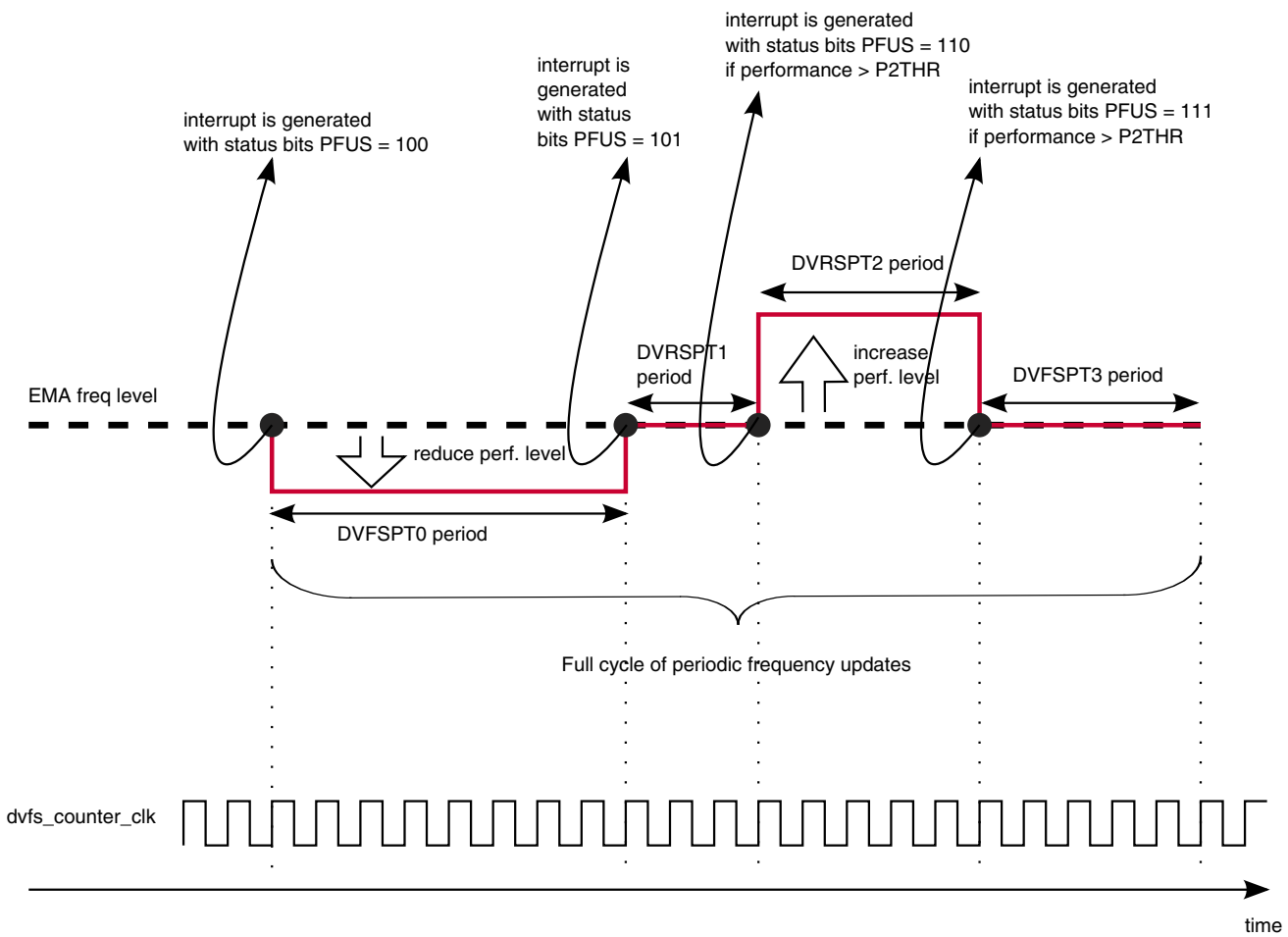


Figure 22-10. DVFS periodic frequency update requests

In case that one of the DVSPT0-DVFSPT3 period is set to "0", such frequency update will be skipped.

The periodic frequency update status is reflected in PFUS bits (reduce/increase performance request is an example - the actual steps taken upon period expiration are defined by s/w routine).

dvfs_counter_clk is ckih divided 64: 26MHz/64=0.40625MHz. The DVFSPT0, DVFSPT1, DVFSPT2 and DVFSPT3 counter are selected for 17 bits each to provide delay of 2^{18}-1=262143 counts, that are equal to 645ms. On the other hand, clk cycle of 0.40625MHz is ~2.46us, that is fast enough to provide high resolution for frequency management for tasks.

The DVFSPT2 and DVFSPT3 begin indication (interrupt) are conditional - only if the current performance is greater than P2THR bits at DVFSPT2 start, the interrupts will be created. Otherwise, the pattern delay will be counted, but without interrupt generation.

22.4 DVFS Output Event/interrupt Configuration

Event/Interrupt will be always high as long as LBFL is'1' and was not cleared by SW. Unless DVFEV (always event) is asserted. Then the event/interrupted will be toggled up and down every toggle of div_3_clk.

22.4.1 Interrupts

DVFS generates an interrupt that indicates that frequency and voltage update is needed. The user has to read the FSVAIM bits in order to know which change needs to be done.

22.5 Initialization Information

The user has to configure threshold values for load tracking and for counters and then enable the DVFS. When an interrupt is received, the user will change the frequency and the voltage in the interrupt handler.

NOTE

DVFS is a monitor that only provides an interrupt when counting exceeds the predefined value and doesn't actually send request to make a change a change of voltage and frequency. This can be done by the user in the interrupt routine or SDMA code by using GPC or CCM (re locking the PLL or changing the post dividers at the CCM).

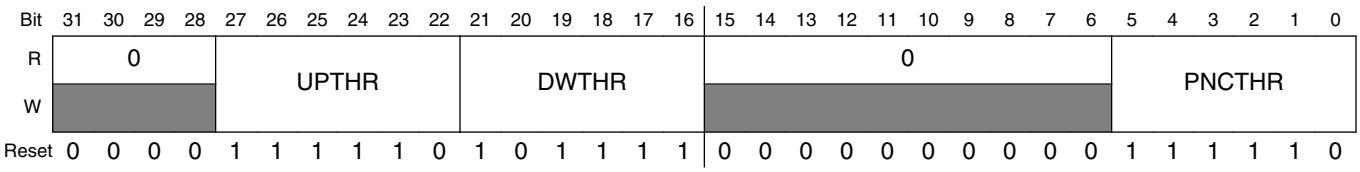
22.6 Programmable Registers

DVFSC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FD_8180	DVFSC Thresholds (DVFSC_THRS)	32	R/W	0FAF_003Eh	22.6.1/ 999
53FD_8184	DVFSC Counters thresholds (DVFSC_COUN)	32	R/W	0007_0020h	22.6.2/ 999
53FD_8188	DVFSC general purpose bits weight (DVFSC_SIG1)	32	R/W	0000_0000h	22.6.3/ 1000
53FD_818C	DVFSC general purpose bits weight (DVFSC_SIG0)	32	R/W	0000_0000h	22.6.4/ 1001
53FD_8190	DVFSC general purpose bit 0 weight counter (DVFSC_GPC0)	32	R/W	0000_0000h	22.6.5/ 1002
53FD_8194	DVFSC general purpose bit 1 weight counter (DVFSC_GPC1)	32	R/W	0000_0000h	22.6.6/ 1003
53FD_8198	DVFSC general purpose bits enables (DVFSC_GPBT)	32	R/W	0000_0000h	22.6.7/ 1004
53FD_819C	DVFSC EMAC settings (DVFSC_EMAC)	32	R/W	0000_0008h	22.6.8/ 1005
53FD_81A0	DVFSC Control (DVFSC_CNTR)	32	R/W	A900_0808h	22.6.9/ 1005
53FD_81A4	DVFSC Load Tracking Register 0, portion 0 (DVFSC_LTR0_0)	32	R	0000_0000h	22.6.10/ 1008
53FD_81A8	DVFSC Load Tracking Register 0, portion 1 (DVFSC_LTR0_1)	32	R	0000_0000h	22.6.11/ 1009
53FD_81AC	DVFSC Load Tracking Register 1, portion 0 (DVFSC_LTR1_0)	32	R	0000_0000h	22.6.12/ 1010
53FD_81B0	DVFS Load Tracking Register 3, portion 1 (DVFSC_LTR1_1)	32	R	0000_0000h	22.6.13/ 1011
53FD_81B4	DVFSC pattern 0 length (DVFSC_PT0)	32	R/W	0000_0010h	22.6.14/ 1011
53FD_81B8	DVFSC pattern 1 length (DVFSC_PT1)	32	R	0000_0010h	22.6.15/ 1012
53FD_81BC	DVFSC pattern 2 length (DVFSC_PT2)	32	R/W	0000_0010h	22.6.16/ 1013
53FD_81C0	DVFSC pattern 3 length (DVFSC_PT3)	32	R/W	0000_0010h	22.6.17/ 1013

22.6.1 DVFSC Thresholds (DVFSC_THRS)

Address: DVFSC_THRS is 53FD_8180h base + 0h offset = 53FD_8180h

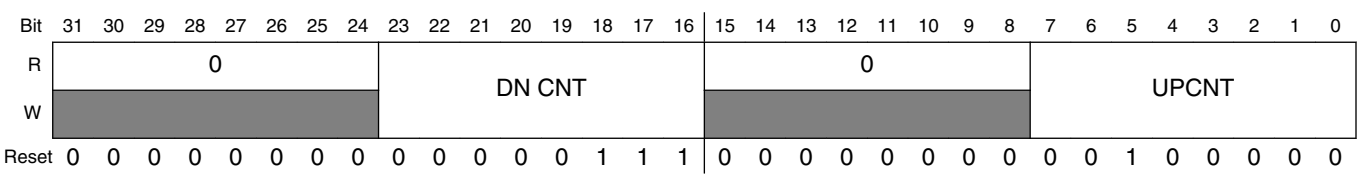


DVFSC_THRS field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved
27–22 UPTHR	Upper threshold for load tracking
21–16 DWTHR	Down threshold for load tracking
15–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5–0 PNCTHR	Panic threshold for load tracking

22.6.2 DVFSC Counters thresholds (DVFSC_COUN)

Address: DVFSC_COUN is 53FD_8180h base + 4h offset = 53FD_8184h

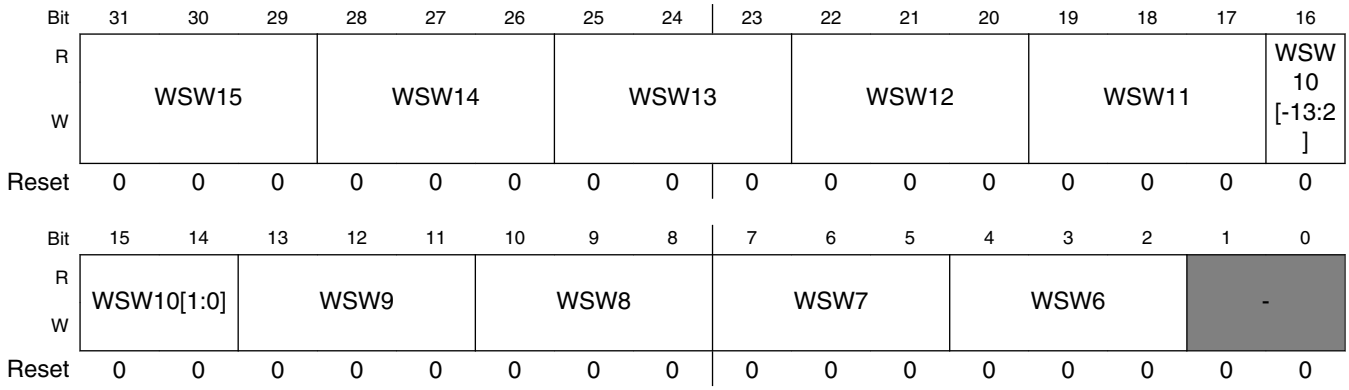


DVFSC_COUN field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23–16 DN CNT	Down counter threshold value
15–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–0 UPCNT	UP counter threshold value

22.6.3 DVFSC general purpose bits weight (DVFSC_SIG1)

Address: DVFSC_SIG1 is 53FD_8180h base + 8h offset = 53FD_8188h

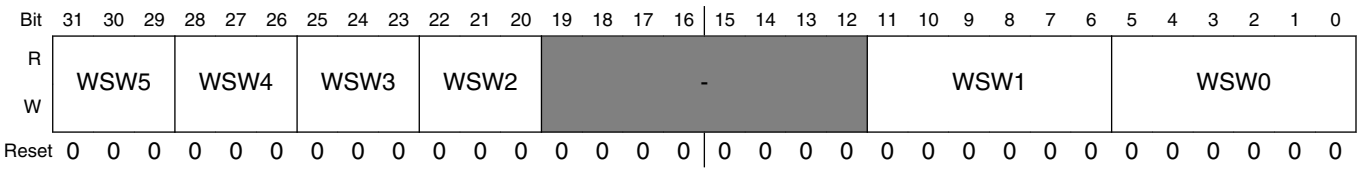


DVFSC_SIG1 field descriptions

Field	Description
31–29 WSW15	General purpose load tracking signal weight dvfs_w_sig[15]
28–26 WSW14	General purpose load tracking signal weight dvfs_w_sig[14]
25–23 WSW13	General purpose load tracking signal weight dvfs_w_sig[13]
22–20 WSW12	General purpose load tracking signal weight dvfs_w_sig[12]
19–17 WSW11	General purpose load tracking signal weight dvfs_w_sig[11]
16–14 WSW10	General purpose load tracking signal weight dvfs_w_sig[10]
13–11 WSW9	General purpose load tracking signal weight dvfs_w_sig[9]
10–8 WSW8	General purpose load tracking signal weight dvfs_w_sig[8]
7–5 WSW7	General purpose load tracking signal weight dvfs_w_sig[7]
4–2 WSW6	General purpose load tracking signal weight dvfs_w_sig[6]
1–0 -	Reserved

22.6.4 DVFSC general purpose bits weight (DVFSC_SIG0)

Address: DVFSC_SIG0 is 53FD_8180h base + Ch offset = 53FD_818Ch



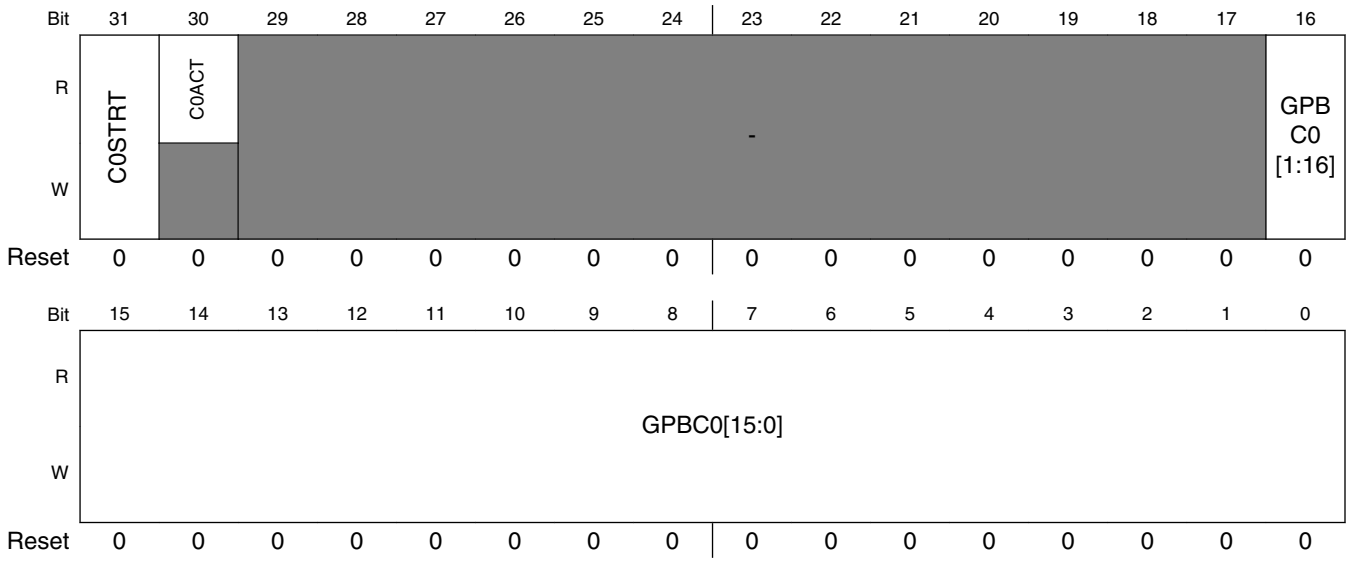
DVFSC_SIG0 field descriptions

Field	Description
31–29 WSW5	General purpose load tracking signal weight dvfs_w_sig[5]
28–26 WSW4	General purpose load tracking signal weight dvfs_w_sig[4]
25–23 WSW3	General purpose load tracking signal weight dvfs_w_sig[3]
22–20 WSW2	General purpose load tracking signal weight dvfs_w_sig[2]
19–12 -	Reserved
11–6 WSW1	General purpose load tracking signal weight dvfs_w_sig[1]. This value is relevant during GPC1 counting period or when GPB1 is set.
5–0 WSW0	General purpose load tracking signal weight dvfs_w_sig[0]. This value is relevant during GPC0 counting period or when GPB0 is set.

22.6.5 DVFSC general purpose bit 0 weight counter (DVFSC_GPC0)

DVFS general purpose bits weight counter.

Address: DVFSC_GPC0 is 53FD_8180h base + 10h offset = 53FD_8190h



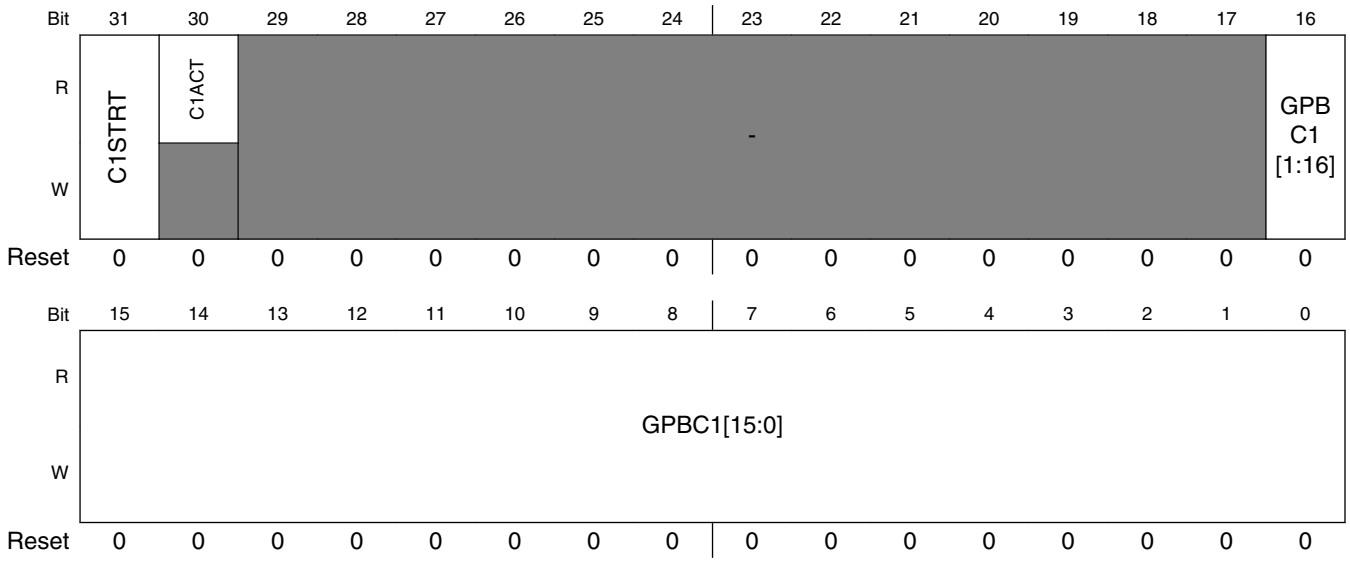
DVFSC_GPC0 field descriptions

Field	Description
31 C0STRT	C0STRT-Counter 0 start Setting of this bit will initialize down counting of the GPC0 value. Bit is self-cleared next cycle after setting. Any setting of this bit will re-start GPC0 counter to the GPC0 value. GPB0 bit disables (overrides) GPC0 counter-WSW0 weight is applicable continuously
30 C0ACT	C0ACT-Counter 0 active indicator 1 General Purpose bit0 counter didn't reach value of "0" - the WSW0 is provided to DVFS calculation 0 General Purpose bit0 counter reached value of "0" - the instead of WSW0, "0" (zero) is provided to DVFS calculation
29–17 -	Reserved
16–0 GPBC0	GPBC0 - General Purpose bits Counter 0 During period of this counter the GeP bit 0 will be set and WSW0 will be added to the calculations.

22.6.6 DVFSC general purpose bit 1 weight counter (DVFSC_GPC1)

DVFS general purpose bits weight counter1.

Address: DVFSC_GPC1 is 53FD_8180h base + 14h offset = 53FD_8194h



DVFSC_GPC1 field descriptions

Field	Description
31 C1STRT	C1STRT - Counter 1start Setting of this bit will initialize down counting of the GPC1 value. Bit is self-cleared next cycle after setting. Any setting of this bit will re-start GPC1 counter to the GPC1 value. GPB1 bit disables (overrides) GPC1 counter - WSW1 weight is applicable continuously
30 C1ACT	C1ACT - Counter 1 active indicator 1 General Purpose bit1 counter didn't reach value of "0" - the WSW1 is provided to DVFS calculation 0 General Purpose bit1 counter reached value of "0" - the instead of WSW1, "0" (zero) is provided to DVFS calculation
29–17 -	Reserved
16–0 GPBC1	GPBC1 - General Purpose bits Counter 1 During period of this counter the GeP bit 1 will be set and WSW1 will be added to the calculations.

22.6.7 DVFSC general purpose bits enables (DVFSC_GPBT)

Address: DVFSC_GPBT is 53FD_8180h base + 18h offset = 53FD_8198h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved																GPB15	GPB14	GPB13	GPB12	GPB11	GPB10	GPB9	GPB8	GPB7	GPB6	GPB5	GPB4	GPB3	GPB2	GPB1	GPB0
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DVFSC_GPBT field descriptions

Field	Description
31–16 Reserved	This field is reserved. N/A
15 GPB15	General purpose bit 15. Its weight is set by WSW15 value.
14 GPB14	General purpose bit 14. Its weight is set by WSW14 value.
13 GPB13	General purpose bit 13. Its weight is set by WSW13 value.
12 GPB12	General purpose bit 12. Its weight is set by WSW12 value.
11 GPB11	General purpose bit 11. Its weight is set by WSW11 value.
10 GPB10	General purpose bit 10. Its weight is set by WSW10 value.
9 GPB9	General purpose bit 9. Its weight is set by WSW9 value.
8 GPB8	General purpose bit 8. Its weight is set by WSW8 value.
7 GPB7	General purpose bit 7. Its weight is set by WSW7 value.
6 GPB6	General purpose bit 6. Its weight is set by WSW6 value.
5 GPB5	General purpose bit 5. Its weight is set by WSW5 value.
4 GPB4	General purpose bit 4. Its weight is set by WSW4 value.
3 GPB3	General purpose bit 3. Its weight is set by WSW3 value.
2 GPB2	General purpose bit 2. Its weight is set by WSW2 value.
1 GPB1	General purpose bit 1. Its weight is set by WSW1 value. IF set (1), the GPBC1 operation is disregarded, WSW1 value is applied continuously.

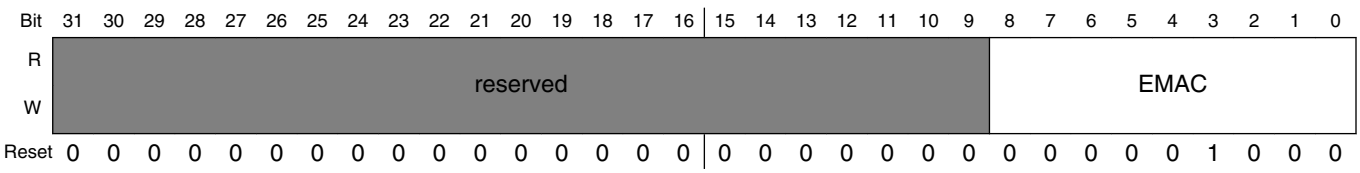
Table continues on the next page...

DVFSC_GPBT field descriptions (continued)

Field	Description
0 GPB0	General purpose bit 0. Its weight is set by WSW0 value. IF set (1), the GPBC0 operation is disregarded, WSW0 value is applied continuously.

22.6.8 DVFSC EMAC settings (DVFSC_EMAC)

Address: DVFSC_EMAC is 53FD_8180h base + 1Ch offset = 53FD_819Ch



DVFSC_EMAC field descriptions

Field	Description
31–9 Reserved	This field is reserved. Reserved
8–0 EMAC	EMAC - EMA control value

22.6.9 DVFSC Control (DVFSC_CNTR)

Table 22-14. DIV3CK division

DIV3CK setting	dividing ratio	sum_3 passing bits	div_1_clk cumulative divider
00	1	4-0	1*512=512
001	4	6-2	4*512=2048
010	16	8-4	16*512=8192
011	64	10-6	64*512=32768
100	256	12-8	256*512=131072
101	1024	16-10	1024*512=524288

Table 22-15. Preliminary Divider definition

DIV_RATIO value	ARM clk division ratio
000000	1
000001	2

Table continues on the next page...

Table 22-15. Preliminary Divider definition (continued)

DIV_RATIO value	ARM clk division ratio
000010	3
...	...

Address: DVFSC_CNTR is 53FD_8180h base + 20h offset = 53FD_81A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DIV3CK			DVFEV	LBMI	LBFL1	LBFL0	DVFIS	PIRQS	FSVAIM	FSVAI[1:0]		WFIM	MAXF	MINF	DIV_RATIO[-10:5]
W																
Reset	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DIV_RATIO[4:0]					-	PFUE	PFUS			LTBRSH	LTBRSR		-		DVFEN
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

DVFSC_CNTR field descriptions

Field	Description
31–29 DIV3CK	DIV3CK - div_3_clk division ratio inside the DVFSC. According to the DVFSC Control (DVFSC_CNTR)
28 DVFEV	Always give a DVFS event. 0 Do not give an event always. 1 Always give event.
27 LBMI	Load buffer full mask interrupt. This bit masks the generation of this interrupt. Load buffer full interrupt is masked (LBFL0 and LBFL1 bits still will be updated, but interrupt won't be generated) 0 Load buffer full interrupt is enabled.
26 LBFL1	Load buffer 1 - full status bit. This bit indicates that log buffer registers are full. The bit is set to 1 automatically. An interrupt will be generated if LBMI bit is set to '0' Load buffer0 is full.Write '1' to clear. (write '0' leaves bit unchanged)

Table continues on the next page...

DVFSC_CNTR field descriptions (continued)

Field	Description
	0 Load buffer0 is not full.
25 LBFL0	Load buffer 0 - full status bit. This bit indicates that log buffer registers are full. The bit is set to 1 automatically. An interrupt will be generated if LBMI bit is set to "0" Load buffer1 is full. Write '1' to clear. (write '0' leaves bit unchanged) 0 Load buffer1 is not full.
24 DVFIS	DVFS Interrupt select. These bits define destination of DVFS interrupts. ARM platform interrupt will be generated for DVFS events. 0 SDMA interrupt will be generated for DVFS events.
23 PIRQS	PIRQS - Pattern IRQ Source * write '1' to clear. Writing '1' will clear interrupt if interrupt was from pattern 1 DVFS IRQ source was from pattern 0 DVFS IRQ source was not from pattern
22 FSVAIM	DVFS Frequency adjustment interrupt mask. This bit masks the DVFS frequency adjustment interrupt. FSVAI status bits will be still asserted in relevant cases. interrupt is masked. 0 Interrupt is enabled.
21–20 FSVAI[1:0]	FSVAI DVFS Frequency adjustment interrupt. These status bits indicate that the system frequency should be changed. 00 no interrupt 01 frequency should be increased. Low priority interrupt. Interrupt is asserted, if FSVAIM=0. Interrupt is masked if MAXF = 1 (highest frequency). 10 frequency should be decreased. Interrupt is asserted, if FSVAIM=0. Interrupt is masked if MINF= 1 (lowest frequency). 11 frequency should be increased immediately. High priority interrupt. Interrupt is asserted, if FSVAIM=0. Interrupt is masked if MAXF = 1 (highest frequency).
19 WFIM	DVFS Wait for Interrupt mask bit 0 Wait for interrupt not masked 1 Wait for interrupt masked.
18 MAXF	Maximum frequency reached. Interrupt will not be created in maximum frequency reached and frequency increase required. 1 max frequency reached 0 max frequency not reached
17 MINF	Minimum frequency reached. Interrupt will not be created in minimum frequency reached and frequency decrease required. 1 min frequency reached 0 min frequency not reached

Table continues on the next page...

DVFSC_CNTR field descriptions (continued)

Field	Description
16–11 DIV_RATIO	DIV_RATIO - Divider value. Divider divides the input ARM clock, following the table DVFSC Control (DVFSC_CNTR)
10 -	Reserved
9 PFUE	PFUE - Period Frequency Update Enable 1 enabled 0 disabled
8–6 PFUS	PFUS - Periodic Frequency Update Status 000 no update 100 DVFSPT0 period, previous finished (can be performance level decrease) 101 DVFSPT1 period, previous finished (can be EMA-detected performance level) 110 DVFSPT2 period, previous finished (can be performance level increase) 111 DVFSPT3 period, previous finished (can be EMA-detected performance level)
5 LTBRSH	LTBRSH - Load Tracking Buffer Register Shift: 0 values of [5:2] of the selected input are saving in Load Tracking Buffer 1 values of [4:1] of the selected input are saving in Load Tracking Buffer
4–3 LTBRSR	LTBRSR - Load Tracking Buffer Register Source: 00 pre_ld_add 01 ld_add 10 after ema 11 reserved
2–1 -	Reserved
0 DVFEN	DVFEN DVFS enable. This bit enables the DVFSC. **NOTE:** Between disable and enable there has to be at least 3 cycles of div_3_clk. 1 DVFSC enabled. 0 DVFSC disabled.

22.6.10 DVFSC Load Tracking Register 0, portion 0 (DVFSC_LTR0_0)

Address: DVFSC_LTR0_0 is 53FD_8180h base + 24h offset = 53FD_81A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LTS0_7				LTS0_6				LTS0_5				LTS0_4				LTS0_3				LTS0_2				LTS0_1				LTS0_0			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DVFSC_LTR0_0 field descriptions

Field	Description
31–28 LTS0_7	Load Tracking Sample 7
27–24 LTS0_6	Load Tracking Sample 6
23–20 LTS0_5	Load Tracking Sample 5
19–16 LTS0_4	Load Tracking Sample 4
15–12 LTS0_3	Load Tracking Sample 3
11–8 LTS0_2	Load Tracking Sample 2
7–4 LTS0_1	Load Tracking Sample 1
3–0 LTS0_0	Load Tracking Sample 0

22.6.11 DVFSC Load Tracking Register 0, portion 1 (DVFSC_LTR0_1)

Address: DVFSC_LTR0_1 is 53FD_8180h base + 28h offset = 53FD_81A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LTS0_15				LTS0_14				LTS0_13				LTS0_12				LTS0_11				LTS0_10				LTS0_9				LTS0_8			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DVFSC_LTR0_1 field descriptions

Field	Description
31–28 LTS0_15	Load Tracking Sample 15
27–24 LTS0_14	Load Tracking Sample 14
23–20 LTS0_13	Load Tracking Sample 13
19–16 LTS0_12	Load Tracking Sample 12
15–12 LTS0_11	Load Tracking Sample 11
11–8 LTS0_10	Load Tracking Sample 10

Table continues on the next page...

DVFSC_LTR0_1 field descriptions (continued)

Field	Description
7–4 LTS0_9	Load Tracking Sample 9
3–0 LTS0_8	Load Tracking Sample 8

22.6.12 DVFSC Load Tracking Register 1, portion 0 (DVFSC_LTR1_0)

Address: DVFSC_LTR1_0 is 53FD_8180h base + 2Ch offset = 53FD_81ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LTS1_7				LTS1_6				LTS1_5				LTS1_4				LTS1_3				LTS1_2				LTS1_1				LTS1_0			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DVFSC_LTR1_0 field descriptions

Field	Description
31–28 LTS1_7	Load Tracking Sample 7
27–24 LTS1_6	Load Tracking Sample 6
23–20 LTS1_5	Load Tracking Sample 5
19–16 LTS1_4	Load Tracking Sample 4
15–12 LTS1_3	Load Tracking Sample 3
11–8 LTS1_2	Load Tracking Sample 2
7–4 LTS1_1	Load Tracking Sample 1
3–0 LTS1_0	Load Tracking Sample 0

22.6.13 DVFS Load Tracking Register 3, portion 1 (DVFSC_LTR1_1)

Address: DVFSC_LTR1_1 is 53FD_8180h base + 30h offset = 53FD_81B0h

Bit	31–28	27–24	23–20	19–16	15–12	11–8	7–4	3–0
R	LTS1_15	LTS1_14	LTS1_13	LTS1_12	LTS1_11	LTS1_10	LTS1_9	LTS1_8
W								
Reset	0000	0000	0000	0000	0000	0000	0000	0000

DVFSC_LTR1_1 field descriptions

Field	Description
31–28 LTS1_15	Load Tracking Sample 15
27–24 LTS1_14	Load Tracking Sample 14
23–20 LTS1_13	Load Tracking Sample 13
19–16 LTS1_12	Load Tracking Sample 12
15–12 LTS1_11	Load Tracking Sample 11
11–8 LTS1_10	Load Tracking Sample 10
7–4 LTS1_9	Load Tracking Sample 9
3–0 LTS1_8	Load Tracking Sample 8

22.6.14 DVFSC pattern 0 length (DVFSC_PT0)

Address: DVFSC_PT0 is 53FD_8180h base + 34h offset = 53FD_81B4h

Bit	31–24	23–18	17	16
R	-		PT0A	FPTN0 [1:16]
W				
Reset	00000000	000000	0	0

Bit	15–8	7–0
R	FPTN0[15:0]	
W		
Reset	00000000	00010000

DVFSC_PT0 field descriptions

Field	Description
31–18 -	Reserved
17 PT0A	PT0A - Pattern 0 currently active (read-only) 1 active 0 non-active
16–0 FPTN0	FPTN0 - Frequency pattern 0 counter During period of this counter the frequency will be reduced from the EMA-detected level.

22.6.15 DVFSC pattern 1 length (DVFSC_PT1)

Address: DVFSC_PT1 is 53FD_8180h base + 38h offset = 53FD_81B8h

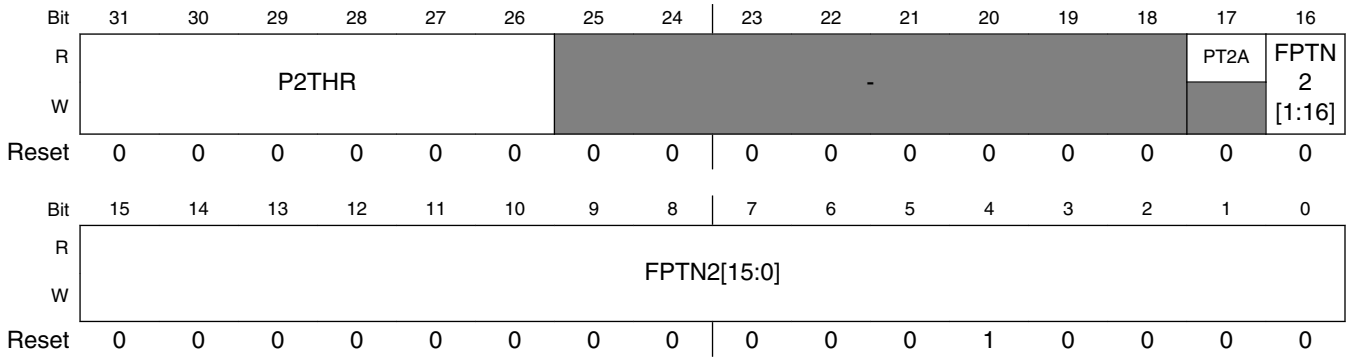
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-														PT1A		FPTN1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

DVFSC_PT1 field descriptions

Field	Description
31–18 -	Reserved
17 PT1A	PT1A - Pattern 1 currently active (read-only) 1 active 0 non-active
16–0 FPTN1	FPTN1 - Frequency pattern 1 counter During period of this counter the frequency will be set to the EMA-detected level.

22.6.16 DVFSC pattern 2 length (DVFSC_PT2)

Address: DVFSC_PT2 is 53FD_8180h base + 3Ch offset = 53FD_81BCh

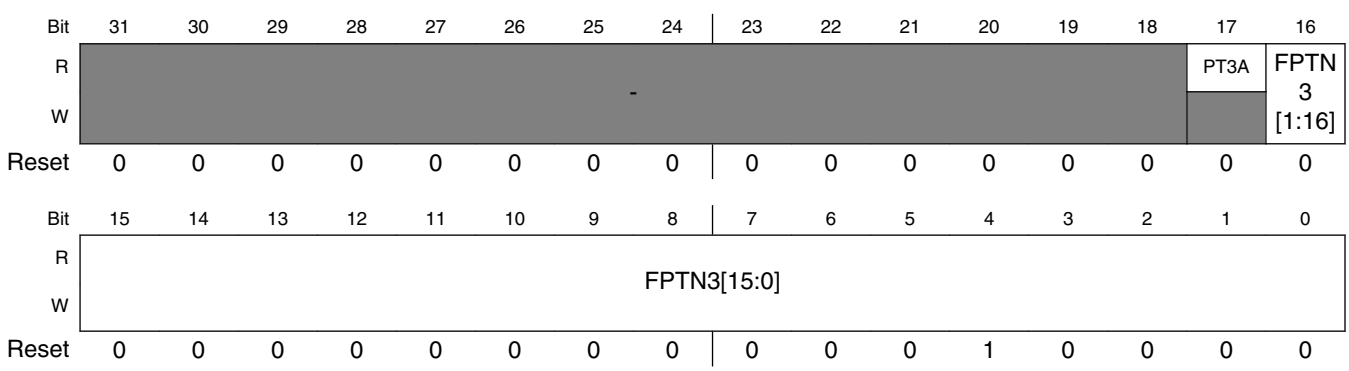


DVFSC_PT2 field descriptions

Field	Description
31–26 P2THR	P2THR - Pattern 2 Threshold Threshold of current DVFS load (after EMA), for generating interrupts with PFUS indicators 110, 111. If the current performance is greater than the P2THR value, the interrupts will be generated. Otherwise, pattern delay will be counted, but without interrupt generation.
25–18 -	Reserved
17 PT2A	PT2A - Pattern 2 currently active (read-only) 1 active 0 non-active
16–0 FPTN2	FPTN2 - Frequency pattern 2 counter During period of this counter the frequency will be increased to higher, than detected by the EMA-detected level.

22.6.17 DVFSC pattern 3 length (DVFSC_PT3)

Address: DVFSC_PT3 is 53FD_8180h base + 40h offset = 53FD_81C0h



DVFSC_PT3 field descriptions

Field	Description
32–18 -	Reserved
17 PT3A	PT3A - Pattern 3 currently active (read-only) 1 active 0 non-active
16–0 FPTN3	FPTN3 - Frequency pattern 3 counter During period of this counter the frequency will be set to the EMA-detected level.

Chapter 23
Dynamic Voltage and Frequency Scaling for Peripherals (DVFSP)

23.1 Introduction

23.1.1 Overview

Dynamic Voltage and Frequency Scaling for Peripherals (DVFSP) enables the frequency of the system and the voltage of the power domain to be changed on the fly while all blocks continue their normal operation on reduced frequency. The frequency of the clock domain is changed by dividing the clock source by DVFSP divider. Serial clocks will remain the same.

The DVFSP load tracking block allows hardware tracking on the system load and a generation of an interrupt when a frequency change is requested.

The statistics for DVFSP are done on peripherals such as External Memory Controller (EXTMC) (PER1), accelerators such as Image Processing Unit (IPU), and so on.

Frequency and voltages changes should comply with the specific limitations of each peripheral, for example lowest allowed frequency of memory device or the ability to change voltages on the fly. Those limitations should be considered when configuring the DVFSP monitors.

DVFSP is sub-block of the Global Power Controller (GPC) block. See GPC chapter for more details.

NOTE

DVFSP is a monitor that provides an interrupt when a count exceeds a predefined value. It does not request a change of voltage and frequency. This can be done by the user in the interrupt routine or Smart Direct Memory Access (SDMA)

code by using GPC or Clock Control Module (CCM) (by locking the PLL or changing the post dividers at the CCM).

A diagram of the DVFSP block is shown in Figure 23-1.

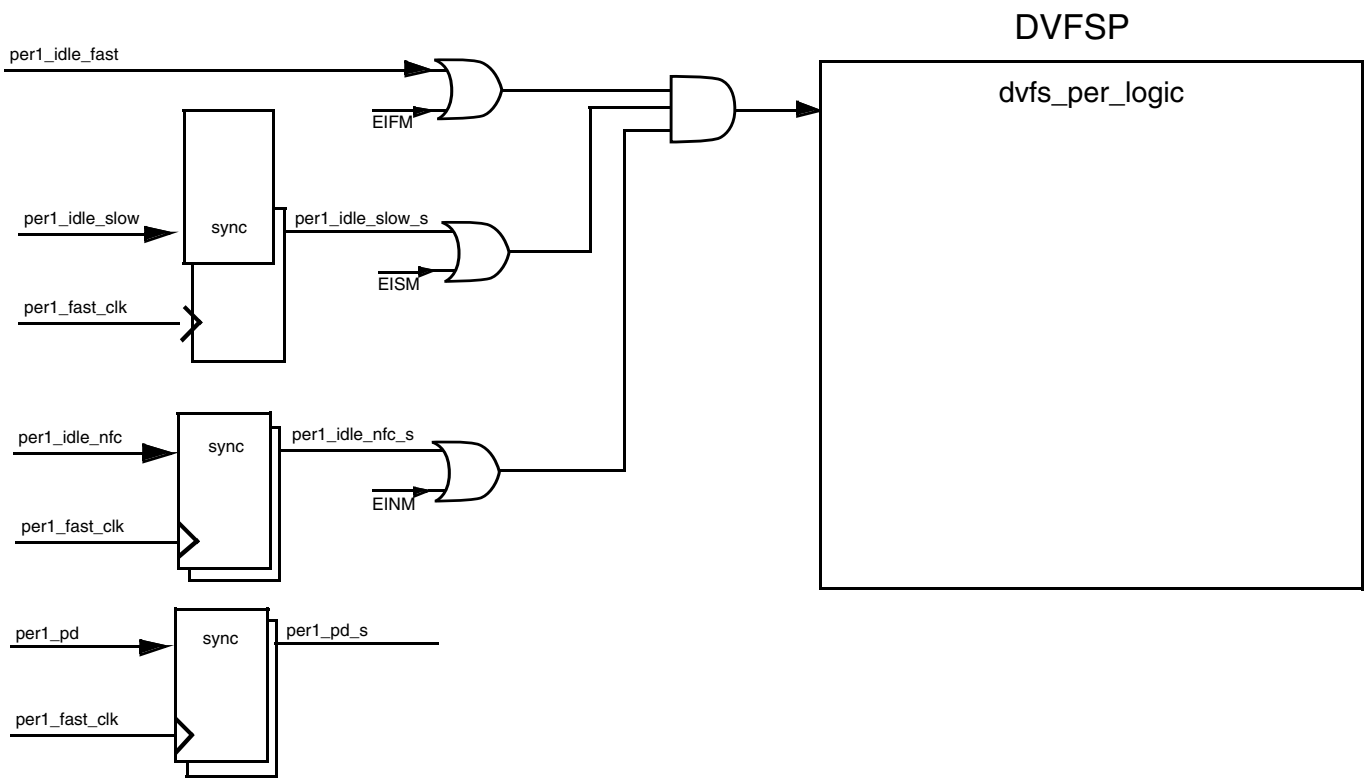


Figure 23-1. DVFSP Block Diagram

The DVFSP logic is shown in Figure 23-2.

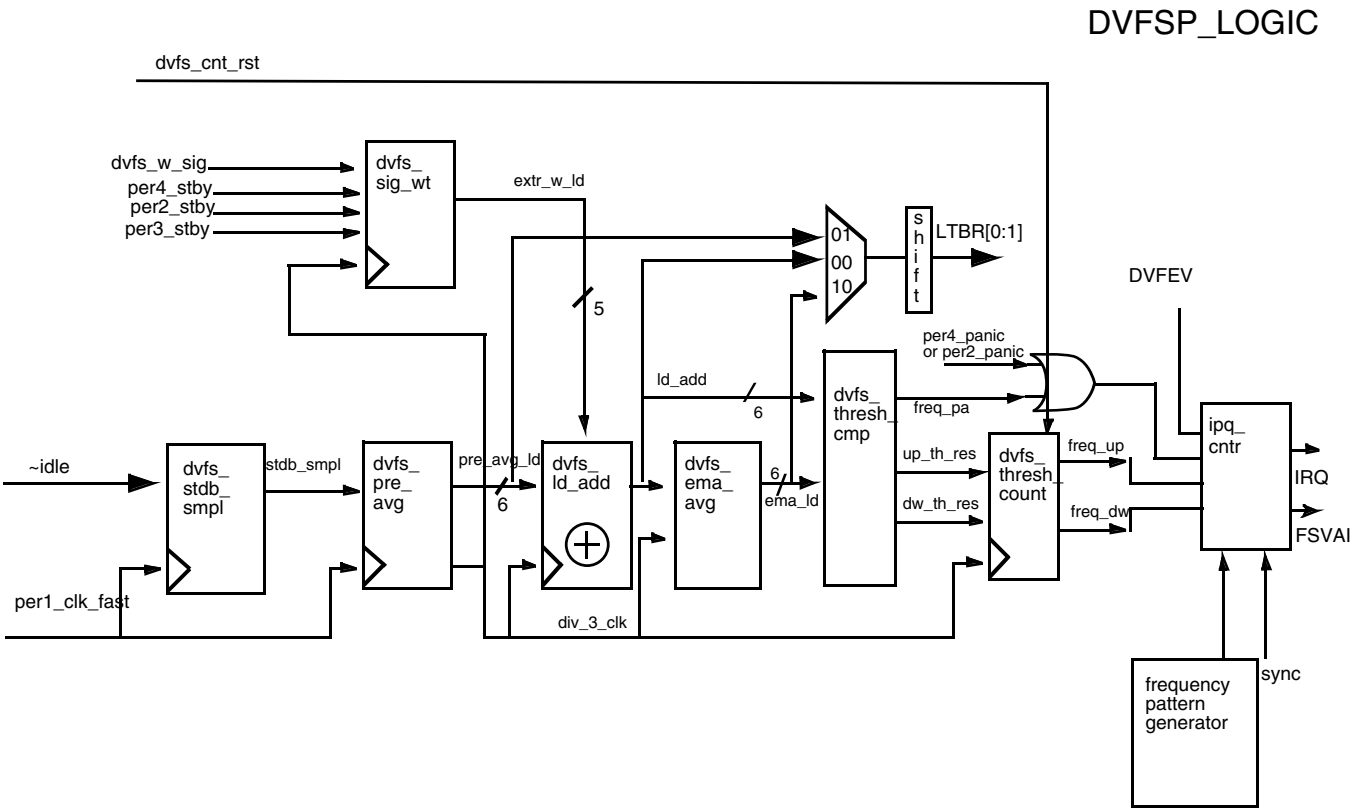


Figure 23-2. DVFSP_LOGIC Block Diagram

23.1.2 Features

The DVFSP load tracking block includes the following features:

- Configurable include/exclude of input signals:
  - Peripheral_1 (PER1) standby signal (idle / non-idle)
  - 10 general purpose bits
  - Peripheral_2 (PER2) standby
  - Peripheral_3 (PER3) standby
  - Peripheral_4 (PER4) standby
    - Configurable weight of each bit.
- Configurable generated clocks and averaging time slicing (respond time).
- Configurable panic mode respond logic (for frequency up).
- Programmable buffer for last 4, 8, 12 or 16 load tracking samples. Based on value in 'LBCF' in DVFSP_PMCR0. There is also a buffer full signal 'LBFL'.

23.2 Functional Description of DVFSP Core Load Tracking

The DVFSP load tracking block includes the following features:

Configurable include/exclude of input signals:

- PER1 standby signal (idle / non-idle)
- 10 general purpose load tracking signals
- per2 idle
- per3 idle
- per4 idle
- per2 panic
- per4 panic
- Configurable weight and (level-sensitive) of GeP signals.
- Configurable generated clocks and averaging time slicing (respond time).
- Configurable panic mode respond logic (for frequency up).
- Programmable buffer for last 4,8,12, or 16 load tracking samples.

23.3 Component Blocks Description

23.3.1 dvfs_stdb_smpl Block

This block samples the per1_non_idle, composed form per1_fast_idle, per1_slo_idle, per1_nfc_idle

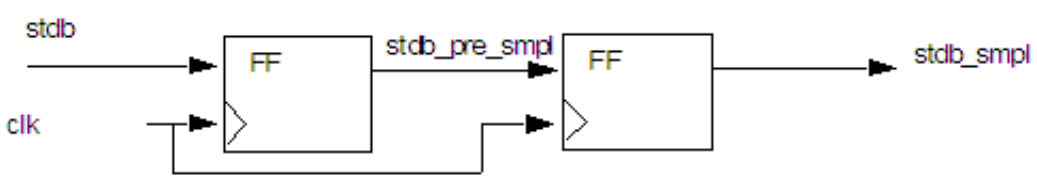


Figure 23-3. dvfs_stdb_smpl block diagram

This block synchronizes the stdb signal with the per1_clk_fast clock. The two signals enter the Flip-Flops (twice), and by doing so, are synchronized. The resulting synchronized signal is stdb_smpl.

23.3.2 dvfs_sig_wt Block

The purpose of this block is to sample the 16 GeP (general purpose) load signals, multiply each one of them by appropriate weight and sum products. The sampling is done by the slow clock div_3_clk. These signals have the only options of detection: level sampling. There are two paths for general purpose bits weight.

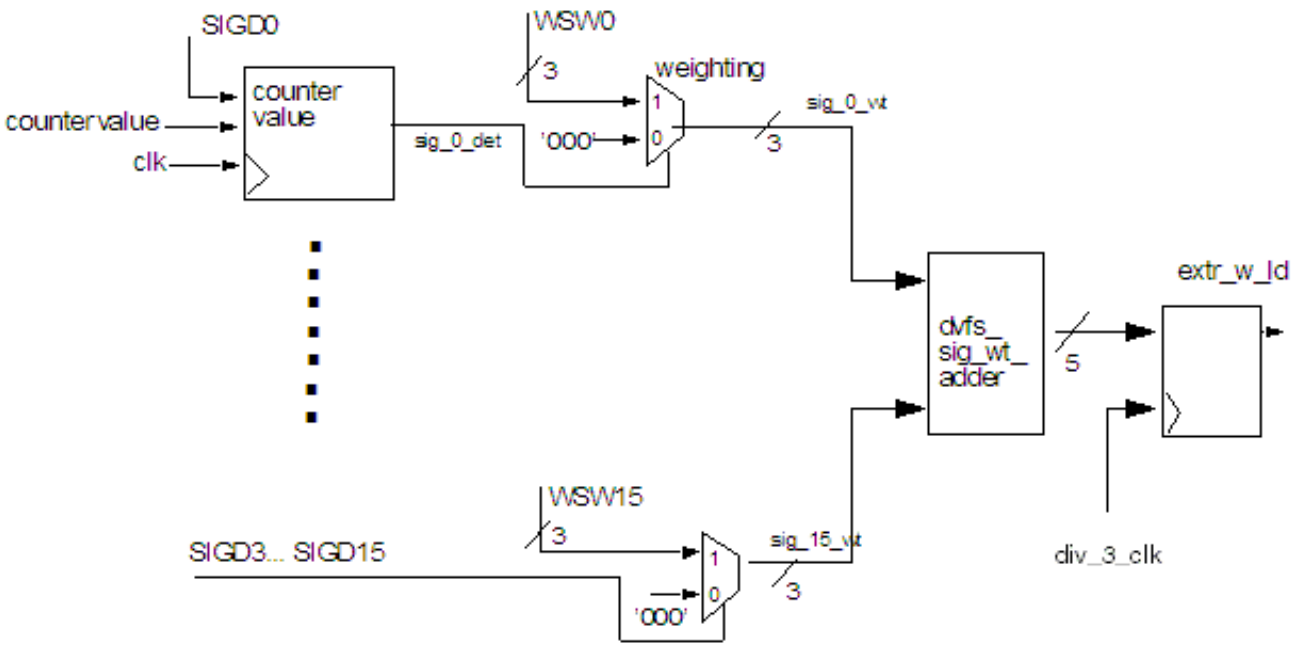


Figure 23-4. dvfs_sig_wt block diagram

23.3.3 dvfs_pre_avg Block

The purpose of the dvfs_pre_avg block is to perform simple, non-overlapping averaging, reducing the sampling clock frequency and providing a level-based average index of the tracked ARM platform load.

The output 5 bits avg_load signal provides a result with a tolerance of ~3%. (Supported values range is 0-0.9.)

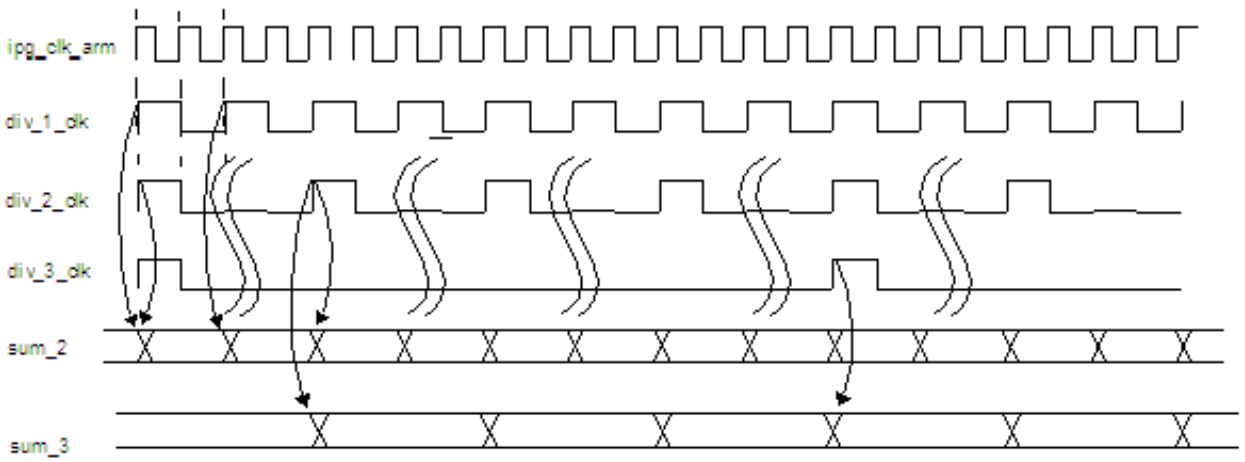


Figure 23-5. dvfs_pre_avg clocks

The internal clocks in dvfs_avg_pre block are generated from ipg_clk_arm.

Table 23-1. Value Ranges of dvfs_pre_avg Signals

signal	binary max value	binary min value	decimal max value	decimal min value

Table 23-2. dvfs_pre_avg Generated Clocks

clock name	generated from	ratio to source	ratio to ipg_clk_arm	max clk freq.	note

23.3.4 dvfs_ld_add Block

The dvfs_ld_add block sums the ARM platform load, tracked by idle/non-idle signal and the load, detected from the additional load signals, weighted by signal_weighting block.

The adder should perform the following operation:

extr_w_ld[4-0] + {0,pre_avg_ld[4-0]}

The input signals of five (5) bits produce output signal of six (6) bits, providing 3% resolution in the range of 0-1.97 of load indication. (1 is equal to 100% load tracking with no additional signals include).

The output ld_add[5-0] is sampled by div_3_clk signal.

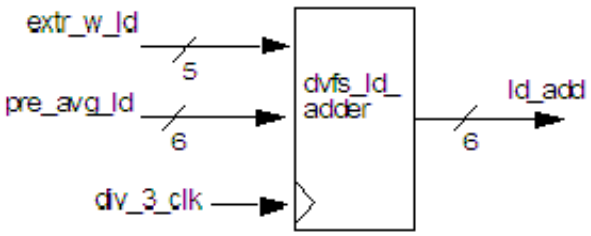


Figure 23-6. dvfs_ld_add block diagram

23.3.5 dvfs_ema_avg Block

The purpose of dvfs_ema_avg, or Exponential Moving Average (EMA), block is to calculate an exponential moving average of the tracked ARM platform load. The parameters of EMA are defined by EMAC field of the LTR2 register. These parameters set how many samples of simple average will be taken into account.

The EMA formula is: EMA(i) = a*X(i) + (1-a)*EMA(i-1)

where:

a = 2 / (N+1);

N is the number of samples taken into account.

X(i) = current input sample;

EMA(i-1) = previous value of EMA.

By setting the value of "a", the behavior of EMA is defined.

In the following table, the parameter "a" is listed relatively to the amount of the X(i) samples taken into account ("N") in the equation above: (the resolution of the digital multiplier is limited, hence the lowest values of the "a" can be linked to a range of included samples in EMA instead of single number).

In the following table there is also the amount of cycles in which the lower frequency will be masked from the moment that DVFSP is enabled (and not between frequency switches), so that enough information about the history can be gained before the frequency is lowered.

EMA settings

samples included in EMA calculation (N)	Number of cycles while lower frequency interrupt is masked	"a" value (decimal)	"a" value, adjusted by binary resolution	"a" value (binary)								
----------		----------	----------	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7	bit 8
1	6	1.000	1.000	1	0	0	0	0	0	0	0	0
2	11	0.667	0.668	0	1	0	1	0	1	0	1	1
3	11	0.500	0.500	0	1	0	0	0	0	0	0	0
4	22	0.400	0.398	0	0	1	1	0	0	1	1	0
5	22	0.333	0.332	0	0	1	0	1	0	1	0	1
6	22	0.286	0.285	0	0	1	0	0	1	0	0	1
7	22	0.250	0.250	0	0	1	0	0	0	0	0	0
8	43	0.222	0.223	0	0	0	1	1	1	0	0	1
9	43	0.200	0.199	0	0	0	1	1	0	0	1	1
10	43	0.182	0.180	0	0	0	1	0	1	1	1	0
11	43	0.167	0.168	0	0	0	1	0	1	0	1	1
12	43	0.154	0.152	0	0	0	1	0	0	1	1	1
13	43	0.143	0.141	0	0	0	1	0	0	1	0	1
14	43	0.133	0.133	0	0	0	1	0	0	0	1	0
15	43	0.125	0.125	0	0	0	1	0	0	0	0	0
16	86	0.118	0.117	0	0	0	0	1	1	1	1	0
17	86	0.111	0.109	0	0	0	0	1	1	1	0	0
18	86	0.105	0.105	0	0	0	0	1	1	0	1	1
19	86	0.100	0.102	0	0	0	0	1	1	0	1	0
20	86	0.095	0.094	0	0	0	0	1	1	0	0	0
21	86	0.091	0.090	0	0	0	0	1	0	1	1	1
22	86	0.087	0.086	0	0	0	0	1	0	1	1	0
23	86	0.083	0.082	0	0	0	0	1	0	1	0	1
25	86	0.077	0.078	0	0	0	0	1	0	1	0	0
26	86	0.074	0.074	0	0	0	0	1	0	0	1	1
28	86	0.069	0.070	0	0	0	0	1	0	0	1	0
30	86	0.065	0.066	0	0	0	0	1	0	0	0	1
31	86	0.063	0.063	0	0	0	0	1	0	0	0	0
33	173	0.059	0.059	0	0	0	0	0	1	1	1	1

Table continues on the next page...

samples included in EMA calculation (N)	Number of cycles while lower frequency interrupt is masked	"a" value (decimal)	"a" value, adjusted by binary resolution	"a" value (binary)								
35	173	0.056	0.055	0	0	0	0	0	1	1	1	0
38	173	0.051	0.051	0	0	0	0	0	1	1	0	1
42	173	0.047	0.047	0	0	0	0	0	1	1	0	0
45	173	0.043	0.043	0	0	0	0	0	1	0	1	1
50	173	0.039	0.039	0	0	0	0	0	1	0	1	0
56	173	0.035	0.035	0	0	0	0	0	1	0	0	1
~64	173	0.031	0.031	0	0	0	0	0	1	0	0	0
~74	256	0.027	0.027	0	0	0	0	0	0	1	1	1
~86	256	0.023	0.023	0	0	0	0	0	0	1	1	0
~100	256	0.020	0.020	0	0	0	0	0	0	1	0	1
~128	256	0.016	0.016	0	0	0	0	0	0	1	0	0
~170	512	0.012	0.012	0	0	0	0	0	0	0	1	1
~260	512	0.008	0.008	0	0	0	0	0	0	0	1	0
~500	512	0.004	0.004	0	0	0	0	0	0	0	0	1
N/A		0.000	0.000	0	0	0	0	0	0	0	0	0

dvfs_ema_avg block diagram

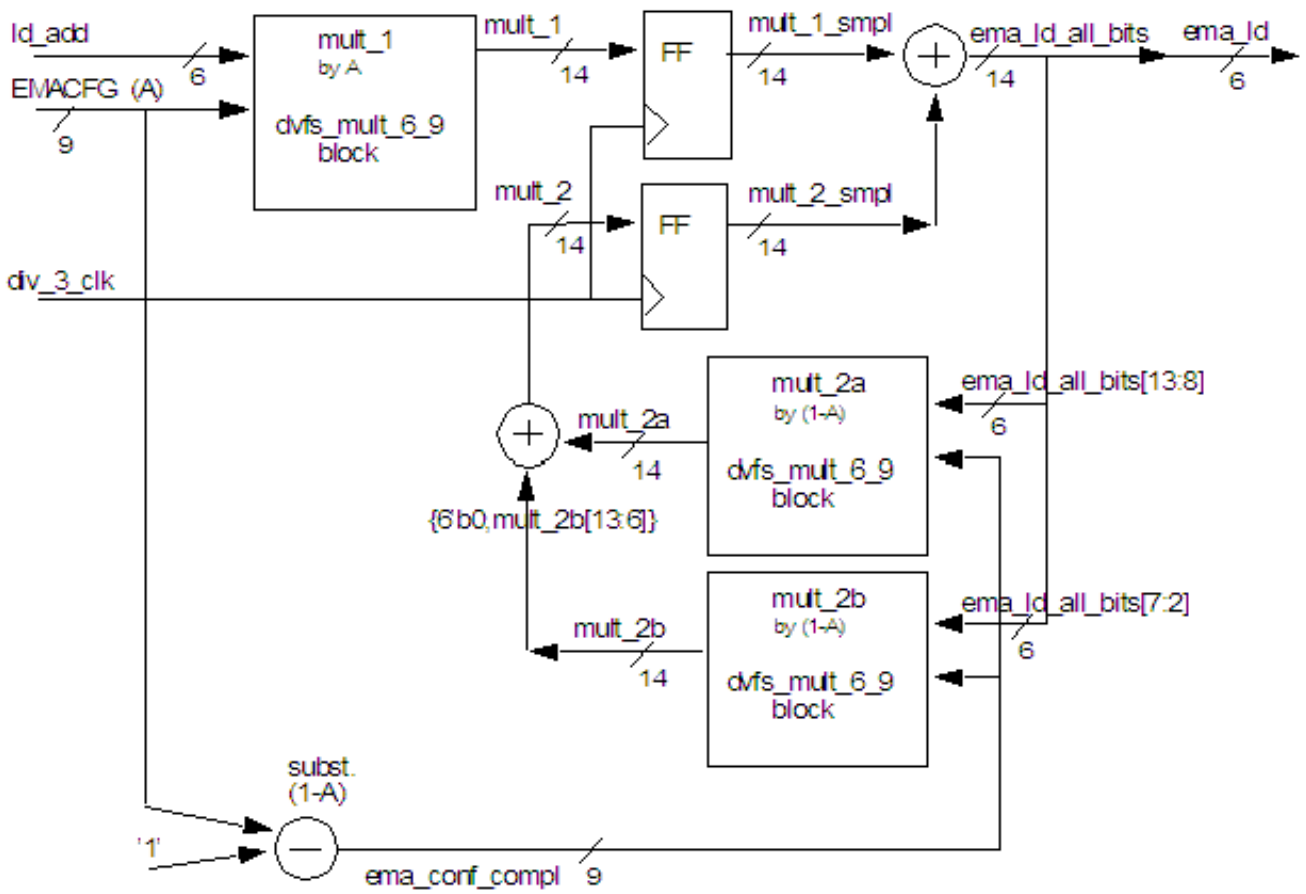


Figure 23-7. dvfs_ema_avg block diagram

The EMA inputs are as follows:

- ld_add[5-0] - load level
- EMACFG[8-0] - "a" parameter of EMA algorithm
- div_2_clk - fast clock don't see this in the diagram
- div_3_clk - slow clock

The EMA outputs the following:

- ema_ld[5-0] - result of EMA algorithm (by div_3_clk)

The mult_a block provides multiplying between comp_load (6 bits) and EMACFG (9 bits). For the multiply operation, a faster clock (for internal sum) is required- this is provided by div_2_clk. Div_2_clk is faster than the div_3_clk by a factor of at least 16, which is enough for 6*9 or 9*6 operations. Only the highest 6 bits are taken from the result of mult_a.

The mult_b block operates in a similar manner.

The adder block's output is sampled with div_3_clk clock.

23.3.6 dvfs_thres_cmp Block

The dvfs_thres_cmp block compares the ARM platform load value to programmable threshold levels. The comparators as shown in the drawing below are used:

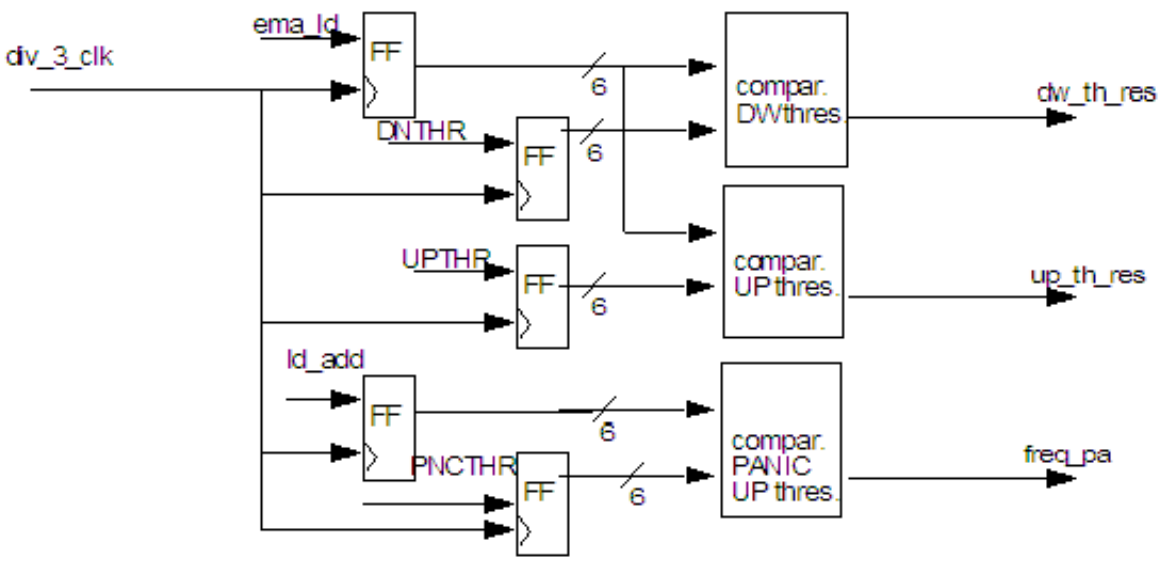


Figure 23-8. dvfs_tresh_cmp diagram

The block is composed of three (3) comparators:

1. compar_dw_thres comparator, which compares between ema_ld signal (output of EMA) and DNTHR signal (taken from config register, bits DNTHR).
2. compar_up_thres comparator, which compares between ema_ld signal (output of EMA) and UPTHR signal (taken from config register, bits UPTHR).
3. compar_panic_up_thres comparator, which compares between ld_add signal (output of load_adder block) and PNCTHR signal (taken from config register, bits PNCTHR).

NOTE

The current implementation of this block enables step-by-step down frequency change. For more advanced option, the number of the DW threshold comparators should be increased (up to three for four-level DVFSP).

23.3.7 dvfs_thresh_count Block

The purpose of the dvfs_thres_count block is to count consecutive threshold overcomes of dw_th_res and up_th_res (outputs of threshold_comp block). If any of the counters (see Figure Figure 23-9 below) receives a null (zero) input synchronous with the clk3 signal, the counter is reset.

If the counter reaches a user defined value, its output is set to an active level. These counters are reset each time frequency scaling occurs or if the threshold overcomes are consecutive.

This block's output signals are saved in configuration register 0, bits [3,2].

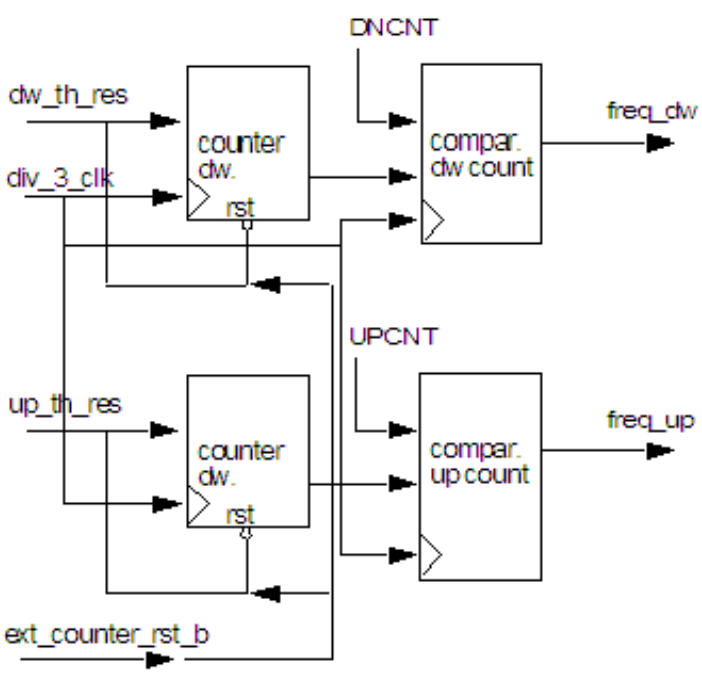


Figure 23-9. Thresh_counters block diagram

23.3.8 Load Tracking Buffer Register

The purpose of the load tracking buffer register is to save last 16 samples of the tracked load (before EMA operation). Hence, the 4 bits of ld_add signal (depending on the LTBRSH) are saved continuously, overwriting each time the latest sample. Each save is carried upon detecting an edge of the div_3_clk signal.

23.4 DVFSP Output Event/Interrupt Configuration

Event/Interrupt will be always high as long as LBFL is'1' and was not cleared by SW. Unless DVFEV (always event) is asserted. Then the event/interrupt will be toggled up and down every toggle of div_3_clk.

23.4.1 Interrupts

DVFSP generates an interrupt that indicates that frequency and voltage update is needed. The FSVAIM bits must be read in order to know which changes should be made.

NOTE

DVFSP is a monitor that provides an interrupt when a count exceeds a predefined value. It does not request a change of voltage and frequency. This can be done by the user in the interrupt routine or SDMA code by using GPC or CCM (by locking the PLL or changing the post dividers at the CCM).

23.5 Programmable Registers

DVFSP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FD_81C8	DVFSP Load Tracking Register 0 (DVFSP_LTR0)	32	R/W	0714_0005h	23.5.1/ 1028
53FD_81CC	DVFSP Load Tracking Register 1 (DVFSP_LTR1)	32	R/W	040F_D53Eh	23.5.2/ 1030
53FD_81D0	DVFSP Load Tracking Register 2 (DVFSP_LTR2)	32	R/W	0000_0000h	23.5.3/ 1031

Table continues on the next page...

DVFSP memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FD_81D4	DVFSP Load Tracking Register 3 (DVFSP_LTR3)	32	R/W	0000_0000h	23.5.4/ 1032
53FD_81D8	LTBR0 (DVFSP_LTBR0)	32	R	0000_0000h	23.5.5/ 1032
53FD_81DC	LTBR1 (DVFSP_LTBR1)	32	R	0000_0000h	23.5.6/ 1033
53FD_81E0	PMCR0 (DVFSP_PMCR0)	32	R/W	082C_0000h	23.5.7/ 1034
53FD_81E4	PMCR1 (DVFSP_PMCR1)	32	R/W	0000_0000h	23.5.8/ 1036

23.5.1 DVFSP Load Tracking Register 0 (DVFSP_LTR0)

DIV3CK setting	dividing ratio	sum_3 passing bits	div_1_clk cumulative divider
00	1	4-0	1*512=512
001	4	6-2	4*512=2048
010	16	8-4	16*512=8192
011	64	10-6	64*512=32768
100	256	12-8	256*512=131072
101	1024	16-10	1024*512=524288

Address: DVFSP_LTR0 is 53FD_81C4h base + 4h offset = 53FD_81C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SIGD15	SIGD14	SIGD13	0	UPTHR						DWTHR						SIGD12	SIGD11	SIGD10	SIGD9	SIGD8	SIGD7	SIGD6	SIGD5	SIGD4	SIGD3	SIGD2	SIGD1	SIGD0	DIV3CK		
W																																
Reset	0	0	0	0	0	1	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

DVFSP_LTR0 field descriptions

Field	Description
31 SIGD15	detect way of general signal #15 (per2_not_idle): 1=edge, 0=level
30 SIGD14	detect way of general signal #14 (per2_not_idle): 1=edge, 0=level
29 SIGD13	detect way of general signal #13 (per3_not_idle): 1=edge, 0=level

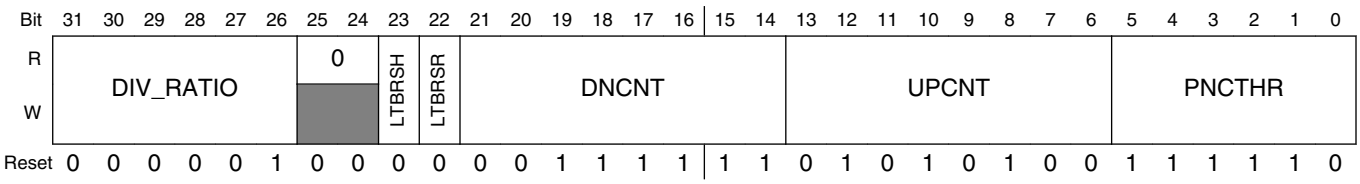
Table continues on the next page...

DVFSP_LTR0 field descriptions (continued)

Field	Description
28 Reserved	This read-only field is reserved and always has the value zero. reserved
27–22 UPTHR	Upper threshold for load tracking
21–16 DWTHR	Down threshold for load tracking
15 SIGD12	detect way of general signal #12 (per3_not_idle): 1=edge, 0=level
14 SIGD11	detect way of general signal #11 (per4_not_idle): 1=edge, 0=level
13 SIGD10	detect way of general signal #10 (per4_not_idle): 1=edge, 0=level
12 SIGD9	detect way of general signal #9(register): 1=edge, 0=level
11 SIGD8	detect way of general signal #8(register): 1=edge, 0=level
10 SIGD7	detect way of general signal #7(register): 1=edge, 0=level
9 SIGD6	detect way of general signal #6(register): 1=edge, 0=level
8 SIGD5	detect way of general signal #5(register): 1=edge, 0=level
7 SIGD4	detect way of general signal #4(register): 1=edge, 0=level
6 SIGD3	detect way of general signal #3(register): 1=edge, 0=level
5 SIGD2	detect way of general signal #2(register): 1=edge, 0=level
4 SIGD1	detect way of general signal #1(register): 1=edge, 0=level
3 SIGD0	detect way of general signal #0(register): 1=edge, 0=level
2–0 DIV3CK	DIV3CK - div_3_clk division ratio inside the DVFSP block. See PMCR1LTBR1 for the significance of different field values.

23.5.2 DVFSP Load Tracking Register 1 (DVFSP_LTR1)

Address: DVFSP_LTR1 is 53FD_81C4h base + 8h offset = 53FD_81CCh



DVFSP_LTR1 field descriptions

Field	Description
31–26 DIV_RATIO	DIV_RATIO - Divider value. Divider divides the input ARM clock: 000000 ARM clock division ratio is 1 000001 ARM clock division ratio is 2 000010 ARM clock division ration is 3 000011-111111 Reserved
25–24 Reserved	This read-only field is reserved and always has the value zero. Reserved.
23 LTBRSH	LTBR source shift: (source by ltbrsr bit)
22 LTBRSR	LTBR source signal 0 pre_avg_l 1 ld_add
21–14 DNCNT	Down counter threshold value
13–6 UPCNT	UP counter threshold value
5–0 PNCTHR	Panic threshold value

23.5.3 DVFSP Load Tracking Register 2 (DVFSP_LTR2)

Address: DVFSP_LTR2 is 53FD_81C4h base + Ch offset = 53FD_81D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WSW15			WSW14			WSW13		WSW12				WSW11			WSW10 [-13:2]
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WSW10[1:0]		WSW9			-			EMAC							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DVFSP_LTR2 field descriptions

Field	Description
31–29 WSW15	General purpose load tracking signal weight dvfs_w_sig[15]
28–26 WSW14	General purpose load tracking signal weight dvfs_w_sig[14]
25–23 WSW13	General purpose load tracking signal weight dvfs_w_sig[13]
22–20 WSW12	General purpose load tracking signal weight dvfs_w_sig[12]
19–17 WSW11	General purpose load tracking signal weight dvfs_w_sig[11]
16–14 WSW10	General purpose load tracking signal weight dvfs_w_sig[10]
13–11 WSW9	General purpose load tracking signal weight dvfs_w_sig[9]
10–9 -	reserved
8–0 EMAC	EMAC - EMA algorithm value

23.5.4 DVFSP Load Tracking Register 3 (DVFSP_LTR3)

Address: DVFSP_LTR3 is 53FD_81C4h base + 10h offset = 53FD_81D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WSW8			WSW7			WSW6			WSW5			WSW4			WSW3			WSW2			WSW1			WSW0			-				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DVFSP_LTR3 field descriptions

Field	Description
31–29 WSW8	General purpose load tracking signal weight dvfs_w_sig[8]
28–26 WSW7	General purpose load tracking signal weight dvfs_w_sig[7]
25–23 WSW6	General purpose load tracking signal weight dvfs_w_sig[6]
22–20 WSW5	General purpose load tracking signal weight dvfs_w_sig[5]
19–17 WSW4	General purpose load tracking signal weight dvfs_w_sig[4]
16–14 WSW3	General purpose load tracking signal weight dvfs_w_sig[3]
13–11 WSW2	General purpose load tracking signal weight dvfs_w_sig[2]
10–8 WSW1	General purpose load tracking signal weight dvfs_w_sig[1]
7–5 WSW0	General purpose load tracking signal weight dvfs_w_sig[0]
4–0 -	Reserved

23.5.5 LTBR0 (DVFSP_LTBR0)

Address: DVFSP_LTBR0 is 53FD_81C4h base + 14h offset = 53FD_81D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LTS0_7				LTS0_6				LTS0_5				LTS0_4				LTS0_3				LTS0_2				LTS0_1				LTS0_0			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DVFSP_LTBR0 field descriptions

Field	Description
31–28 LTS0_7	Load Tracking Sample 7
27–24 LTS0_6	Load Tracking Sample 6
23–20 LTS0_5	Load Tracking Sample 5
19–16 LTS0_4	Load Tracking Sample 4
15–12 LTS0_3	Load Tracking Sample 3
11–8 LTS0_2	Load Tracking Sample 2
7–4 LTS0_1	Load Tracking Sample 1
3–0 LTS0_0	Load Tracking Sample 0

23.5.6 LTBR1 (DVFSP_LTBR1)

Address: DVFSP_LTBR1 is 53FD_81C4h base + 18h offset = 53FD_81DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LTS0_15				LTS0_14				LTS0_13				LTS0_12				LTS0_11				LTS0_10				LTS0_9				LTS0_8			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DVFSP_LTBR1 field descriptions

Field	Description
31–28 LTS0_15	Load Tracking Sample 15
27–24 LTS0_14	Load Tracking Sample 14
23–20 LTS0_13	Load Tracking Sample 13
19–16 LTS0_12	Load Tracking Sample 12
15–12 LTS0_11	Load Tracking Sample 11
11–8 LTS0_10	Load Tracking Sample 10

Table continues on the next page...

DVFSP_LTBR1 field descriptions (continued)

Field	Description
7–4 LTS0_9	Load Tracking Sample 9
3–0 LTS0_8	Load Tracking Sample 8

23.5.7 PMCR0 (DVFSP_PMCR0)

Address: DVFSP_PMCR0 is 53FD_81C4h base + 1Ch offset = 53FD_81E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	-				UDSC	-			DVFEV	DVFIS	LBMI	LBFL	LBCF		-	
Reset	0	0	0	0	1	0	0	0	0	0	1	0	1	1	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FSVAIM	FSVAI[1:0]		-		WFIM	-					DVFEN	-			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DVFSP_PMCR0 field descriptions

Field	Description
31–28 -	reserved
27 UDSC	Up-down scaling. This bit indicates the direction of current frequency scaling. 1 Frequency is increased. 0 Frequency is decreased.
26–24 -	reserved
23 DVFEV	Always give a DVFSP event. 0 Do not always give event. 1 Always give event.
22 DVFIS	DVFSP Interrupt select. These bits define destination of DVFSP interrupts. 1 ARM platform interrupt will be generated for DVFSP events. 0 SDMA interrupt will be generated for DVFSP events.
21 LBMI	Load buffer full mask interrupt. This bit masks the generation of this interrupt.

Table continues on the next page...

DVFSP_PMCR0 field descriptions (continued)

Field	Description
	1 Load buffer full interrupt is masked. 0 Load buffer full interrupt is enabled.
20 LBFL	Load buffer - full status bit. This bit indicates that log buffer registers are full. The bit is set to 1 automatically. An interrupt will be generated if LBMI bit is set to "0" 1 Load buffer is full. 0 Load buffer is not full.
19–18 LBCF	load tracking configuration: 00 4 samples 01 8 samples 10 12 samples 11 16 samples
17–16 -	reserved
15 FSVAIM	DVFSP Frequency adjustment interrupt mask. This bit masks the DVFSP frequency adjustment interrupt. FSVAI status bits will be still asserted in relevant cases. 1 interrupt is masked. 0 interrupt is enabled.
14–13 FSVAI[1:0]	FSVAI DVFSP Frequency adjustment interrupt. These status bits indicate that the system frequency should be changed. 00 no interrupt 01 frequency should be increased. Low priority interrupt. Interrupt is asserted, if FSVAIM=0. Interrupt is masked if MAXF = 1 (highest frequency). 10 frequency should be decreased. Interrupt is asserted, if FSVAIM=0. Interrupt is masked if MINF= 1 (lowest frequency). 11 frequency should be increased immediately. High priority interrupt. Interrupt is asserted, if FSVAIM=0. Interrupt is masked if MAXF = 1 (highest frequency).
12–11 -	reserved
10 WFIM	WFIM - WFI ARM signal masking. 1 WFI ARM is masked 0 WFI ARM is not masked
9–5 -	reserved
4 DVFEN	DVFEN DVFSP enable. This bit enables the DVFSP. **NOTE:** Between disable and enable there has to be at least 3 cycles of div_3_clk. 1 DVFSP enabled. 0 DVFSP disabled.

Table continues on the next page...

DVFSP_PMCR0 field descriptions (continued)

Field	Description
3–0 -	reserved

23.5.8 PMCR1 (DVFSP_PMCR1)

Address: DVFSP_PMCR1 is 53FD_81C4h base + 20h offset = 53FD_81E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-											P1INM	P1ISM	P1IFM	P4PM	P2PM	DVGP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DVFSP_PMCR1 field descriptions

Field	Description
31–21 -	reserved
20 P1INM	PER1 Idle NFC Mask 1 per1_idle_nfc is masked 0 not masked
19 P1ISM	PER1 Idle Slow Mask 1 per1_idle_slow is masked 0 not masked
18 P1IFM	PER1 Idle Fast Mask 1 per1_idle_fast is masked 0 not masked
17 P4PM	PER4 panic mask 1 PER4 panic input is masked
16 P2PM	PER2 panic mask 1 PER2 panic input is masked
15–0 DVGP	General purpose bits, used for WSW

Chapter 24
Enhanced Configurable SPI (ECSPI)

24.1 Overview

This chapter describes a block integrated into an SoC. The chapter is intended for a block driver software developer. It describes block-level operation and programming. To understand how the block is integrated at the SoC level, a system software developer should see discussions of the block in the appropriate SoC-level chapter(s).

The Enhanced Configurable Serial Peripheral Interface (ECSPI) is a full-duplex, synchronous, four-wire serial communication block. The ECSPI contains a 64 x 32 receive buffer (RXFIFO) and a 64 x 32 transmit buffer (TXFIFO). With data FIFOs, the ECSPI allows rapid data communication with fewer software interrupts. The following figure shows a block diagram of the ECSPI.

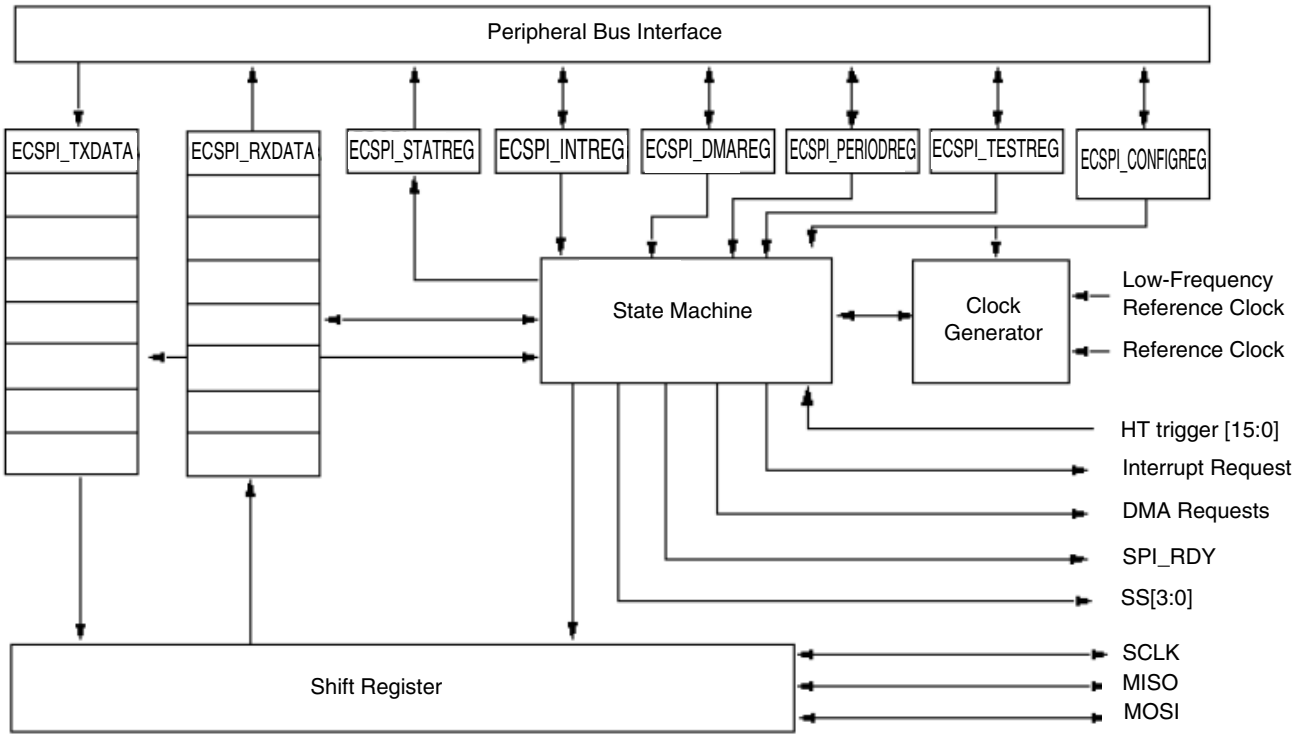


Figure 24-1. ECSPI Block Diagram

24.1.1 Features

Key features of the ECSPI include:

- Full-duplex synchronous serial interface
- Master/Slave configurable
- Four Chip Select (SS) signals to support multiple peripherals
- Transfer continuation function allows unlimited length data transfers
- 32-bit wide by 64-entry FIFO for both transmit and receive data
- 32-bit wide by 16-entry FIFO for HT message data
- Polarity and phase of the Chip Select (SS) and SPI Clock (SCLK) are configurable
- Direct Memory Access (DMA) support
- Max operation frequency up to the reference clock frequency.

24.1.2 Modes and Operations

The ECSPI supports the modes described in the indicated sections:

- Master Mode

- Slave Mode
- Low Power Modes

As described in Operations, the ECSPI supports the operations described in the indicated sections:

- Typical Master Mode
  - Master Mode with SPI_RDY
  - Master Mode with Wait States
  - Master Mode with SS_CTL[3:0] Control
  - Master Mode with Phase Control
- Typical Slave Mode

24.2 External Signals

The table below describes all ECSPI Block I/Os.

Table 24-1. Block I/O Signals

Signal	I/O	Description	Reset State[1]	Pull-Up/ Down[1]
$\overline{SS}$[3:0]	I/O	Chip selects	1	-
SCLK	I/O	SPI clock	0	Active
MISO	I/O	Master data in; slave data out	0	Passive
MOSI	I/O	Master data out; slave data in	0	-
SPI_RDY	I	Master data out; slave data in	0	Active

1. The reset state values and pull-up/down requirements provided in this table are from the block-level perspective. To understand how the block is integrated at the SoC level, the system software developer must see discussions of the block in the appropriate SoC-level chapter(s). For example, a block signal that requires a pull-up could be integrated with a pull-up option built into the SoC. In this case, the system software developer must ensure the proper programming at the SoC level.

The figure below shows the ECSPI in master mode connected to four external devices in a one-way communication link.

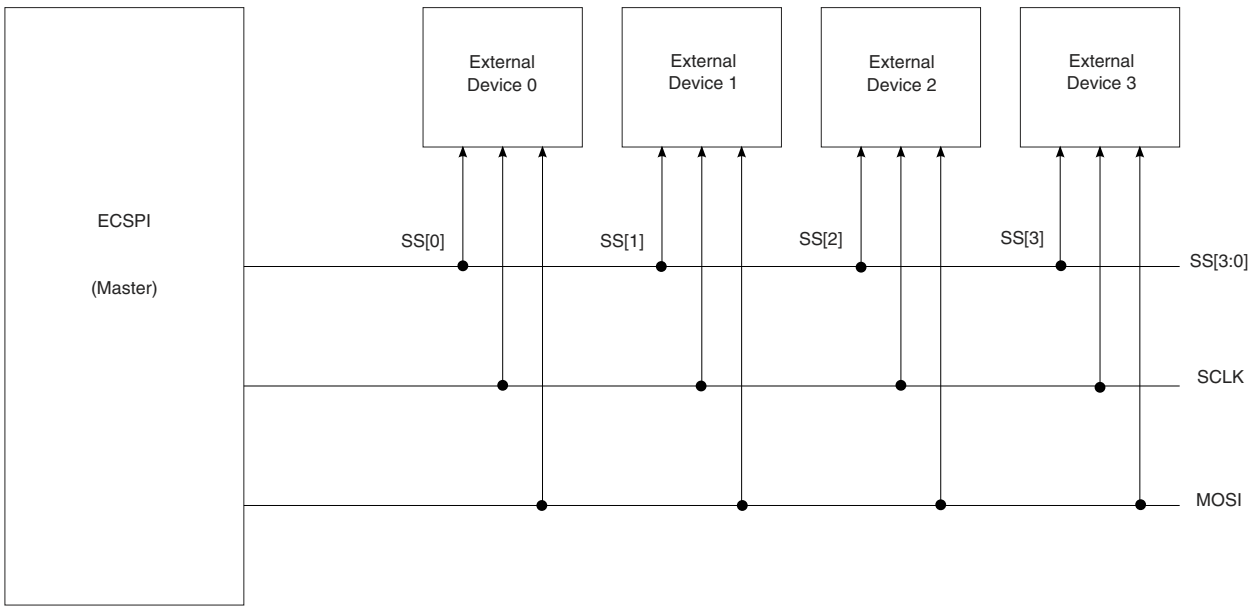


Figure 24-2. Example Connection Diagram

24.3 Functional Description

This section provides a complete functional description of the ECSPI. The figure below shows the relationship of SCLK and data lines while ECSPI has been configured with different POL and PHA settings.

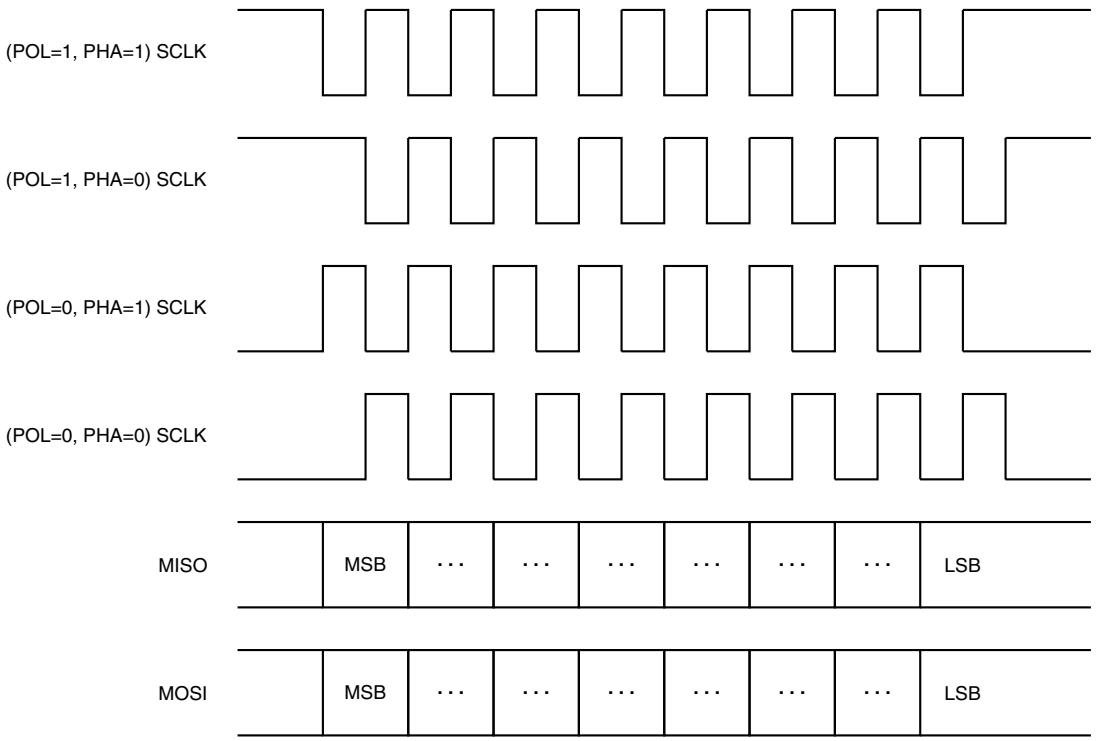


Figure 24-3. ECSPI SCLK, MISO, and MOSI Relationship

24.3.1 Master Mode

When the ECSPI is configured as a master, it uses a serial link to transfer data between the ECSPI and an external slave device. One of the Chip Select (SS) signals and the clock signal (SCLK) are used to transfer data between two devices.

If the external device is a transmit-only device, the ECSPI master's output port can be ignored and used for other purposes. In order to use the internal TXFIFO and RXFIFO, two auxiliary output signals, Chip Select (SS) and SPI_RDY, are used for data transfer rate control. Software can also configure the sample period control register to a fixed data transfer rate.

24.3.2 Slave Mode

When the ECSPI is configured as a slave, software can configure the ECSPI Control register to match the external SPI master's timing. In this configuration, Chip Select ($\overline{SS}$) becomes an input signal, and is used to control data transfers through the Shift register, as well as to load/store the data FIFO.

24.3.3 Hardware Trigger (HT) Mode

In hardware trigger (HT) mode, the behavior of the ECSPI is similar to master mode. The differences are in the trigger mechanism and the burst length.

When the block is in HT mode, the SPI burst is no longer triggered by software, setting the XCH bit or writing to the TXFIFO. Instead, there are 16 input signals HT trigger[15:0] associated with each slot of the 16-entry ECSPI_MSGDATA FIFO, and a positive pulse will trigger the ECSPI to transmit a SPI burst with data in ECSPI_MSGDATA FIFO. And in this case, the burst length is limited in 32-bit, but also configurable. Only SPI channel 0 can be used in HT mode.

24.3.4 Low Power Modes

The ECSPI does not operate under low power mode. It holds its operation when its clock is gated off in master mode. In slave mode, the ECSPI does not respond when its clock is gated off.

24.3.5 Operations

This section describes the ECSPI's operations.

24.3.5.1 Typical Master Mode

The ECSPI master uses the Chip Select (SS) signal to enable an external SPI device, and uses the SCLK signal to transfer data in and out of the Shift register. The SPI_RDY enables fast data communication with fewer software interrupts. By programming the ECSPI_PERIODREG register accordingly, the ECSPI can be used for a fixed data transfer rate.

When the ECSPI is in Master mode the SS, SCLK, and MOSI are output signals, and the MISO signal is an input.

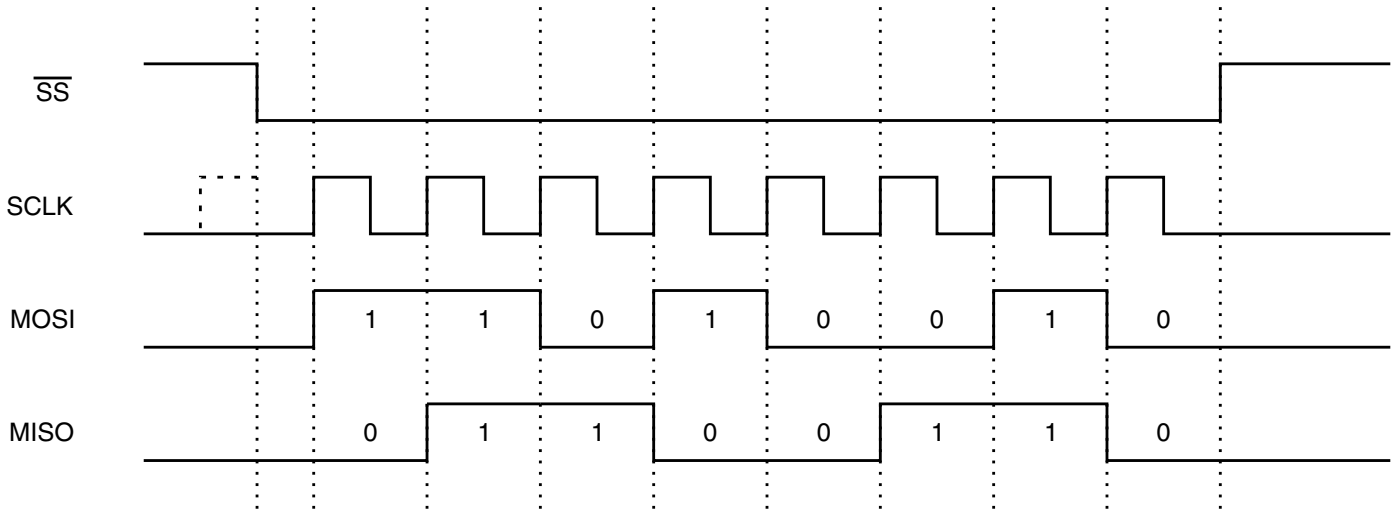


Figure 24-4. Typical SPI Burst (8-bit Transfer)

In the above figure, the Chip Select (SS) signal enables the selected external SPI device, and the SCLK synchronizes the data transfer. The MOSI and MISO signals change on rising edge of SCLK and the MISO signal is latched on the falling edge of the SCLK. The figure above shows a data of 0xD2 is shifted out, and a data of 0x66 is shifted in.

24.3.5.1.1 Master Mode with SPI_RDY

By default, the ECSPI does not use the SPI_RDY signal in master mode (MODE =1).

A SPI burst begins when the following events happen:

- The ECSPI is enabled, TXFIFO has data in it, and ECSPI_CONREG[XCH] bit or the ECSPI_CONREG[SMC] bit is set.
- When the SPI Data Ready Control (ECSPI_CONREG[DRCTL]) bits contains either 01 or 10, the SPI_RDY signal controls when a SPI burst starts.

A SPI burst is defined as a bus transaction that starts when the slave select is asserted and ends when the slave select is negated. The Chip Select (SS) signal will remain asserted until all the bits in a SPI burst are shifted out.

If ECSPI_CONREG[DRCTL] is set to 01, the SPI burst can be triggered only if a falling edge of the SPI_RDY signal has been detected.

The following figure shows the relationship between a SPI burst and the falling edge of SPI_RDY signal.

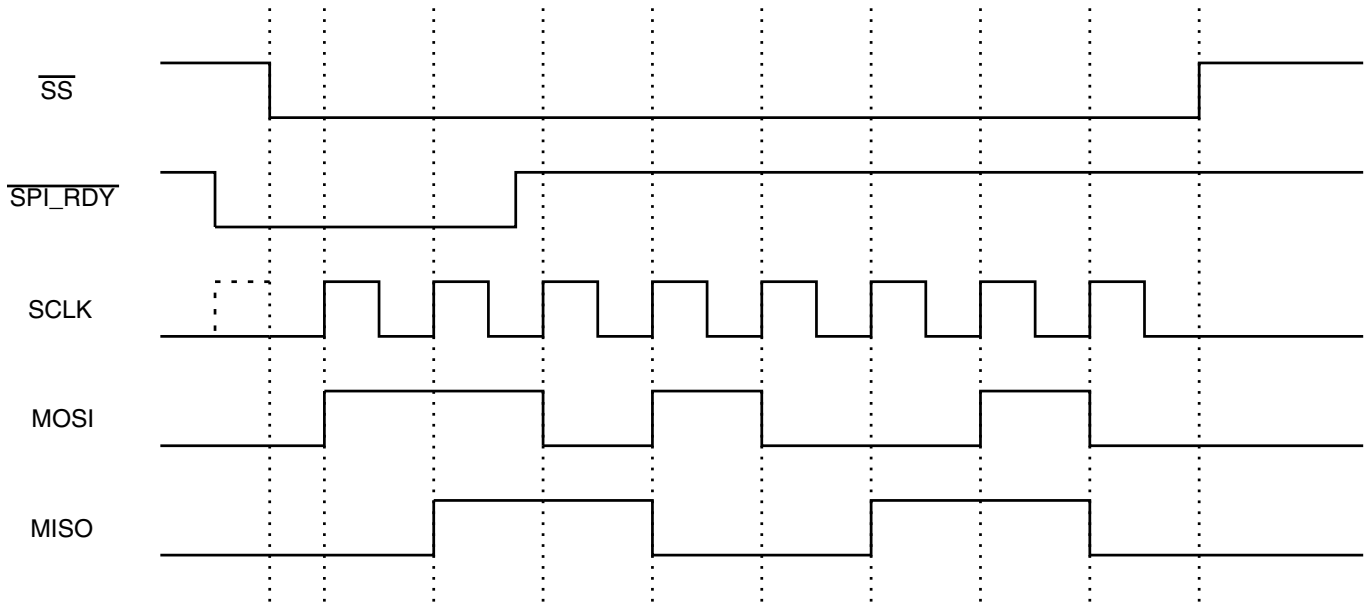


Figure 24-5. Relationship Between a SPI Burst and SPI_RDY: Falling-Edge Triggered

A SPI burst does not start until the falling edge of the SPI_RDY signal is detected. The next SPI burst starts when the next SPI_RDY falling edge is detected, after the last burst has finished.

If SPI Data Ready Control (ECSPI_CONREG[DRCTL]) is set to 10, the SPI burst can be triggered only if the SPI_RDY signal is low.

The following figure shows the relationship between a SPI burst and the SPI_RDY signal. The SPI burst does not begin until the SPI_RDY signal goes low. The ECSPI will keep transmitting SPI burst if the SPI_RDY signal remains low.

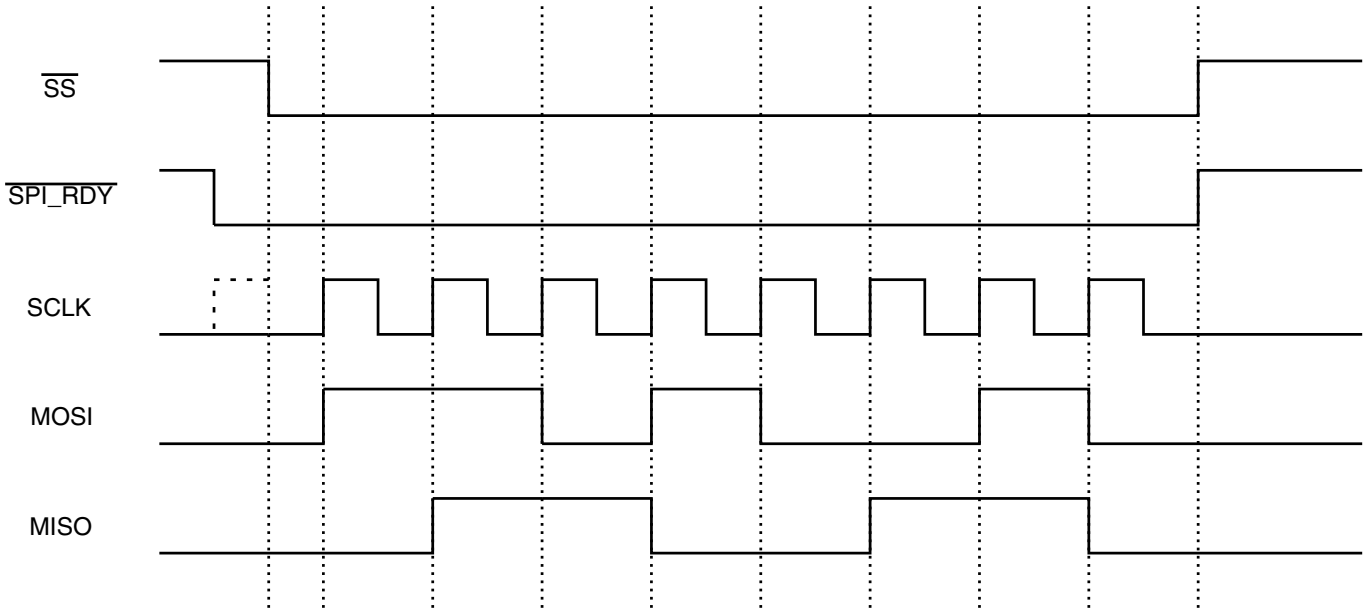


Figure 24-6. Relationship Between a SPI Burst and SPI_RDY: Low-Level Triggered

24.3.5.1.2 Master Mode with Wait States

Wait states can be inserted between SPI bursts. This provides a way for software to slow down the SPI burst to meet the timing requirements of a slower SPI device.

The following figure shows wait states inserted between SPI bursts.

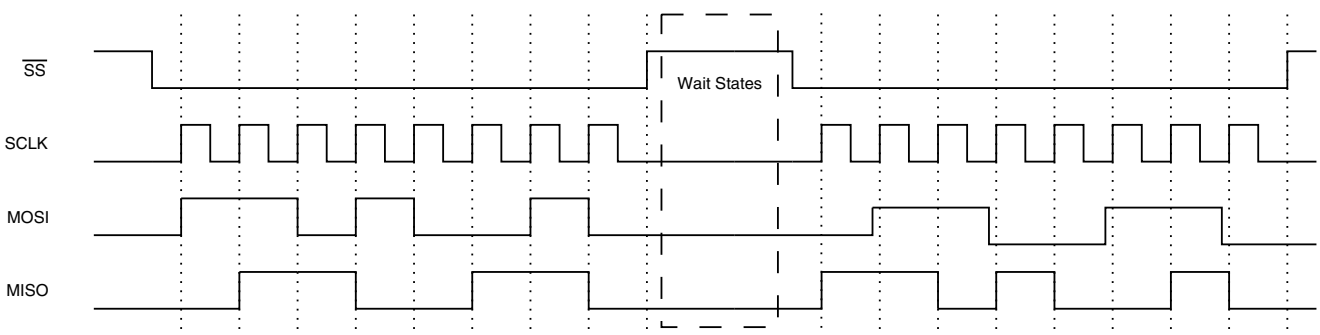


Figure 24-7. SPI Bursts with Wait States

In this case, the number of wait states is controlled by ECSPI_PERIODREG[SAMPLE PERIOD] and the wait states' clock source is selected by ECSPI_PERIODREG[CSRC].

24.3.5.1.3 Master Mode with SS_CTL[3:0] Control

The SPI SS Control (SS_CTL[3:0]) controls whether the current operation is single burst or multiple bursts. When the SPI SS Wave Form Select (SS_CTL[3:0]) is set, the current operation is multiple bursts transfer. When the SPI SS Wave Form Select (SS_CTL[3:0]) bit is cleared, the current operation is single burst transfer. A SPI burst can contains multiple words as defined in the BURST LENGTH field of the ECSPI_CONREG register.

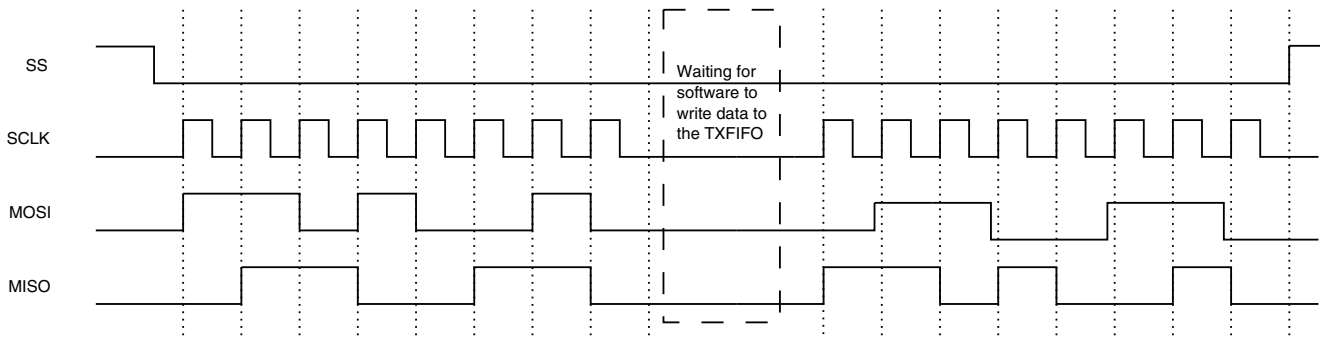


Figure 24-8. SPI Burst While SS_CTL[3:0] is Clear

In Figure 24-8, two 8-bit bursts in the TXFIFO have been combined and transmitted in one SPI burst. The maximum length of a single SPI burst is defined in the BURST LENGTH field of the ECSPI_CONREG control register. (Figure 24-8 corresponds to a BURST LENGTH of 8.) This provides a way for transferring a longer SPI burst by writing data into TXFIFO while the ECSPI is transmitting.

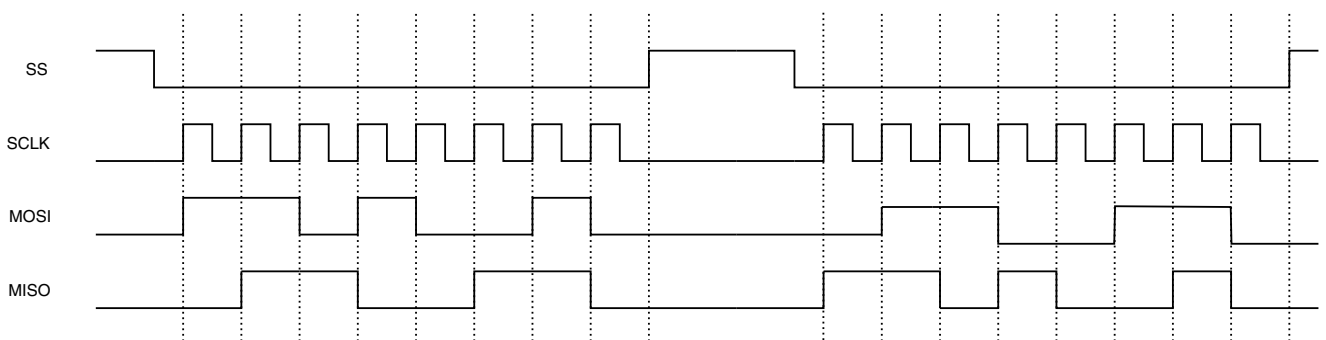


Figure 24-9. SPI Bursts While SS_CTL[3:0] is Set

In Figure 24-9, two FIFO entries are transmitted, one entry with each SPI burst. The ECSPI will continue to transmit SPI bursts until the TXFIFO is empty. When wait states can be inserted between SPI bursts, the SS will negate between SPI bursts until the wait states finish.

24.3.5.1.4 Master Mode with Phase Control

The Phase Control (ECSPI_CONREG[PHA]) bit controls how the transmit data shifts out and the receive data shifts in.

When the Phase control (ECSPI_CONREG[PHA]) bit is set, the transmit data will shift out on the rising edge of SCLK, and the receive data is latched on the falling edge of SCLK. The most-significant bit is output on the first rising SCLK edge.

When ECSPI_CONREG[PHA] is cleared, the transmit data is shifted out on the falling edge of SCLK and the receive data is latched on the rising edge of SCLK. The MSB is output when the host processor loads the transmitted data.

Inverting the SCLK polarity does not impact the edge-triggered operations because they are internal to the serial peripheral interface master.

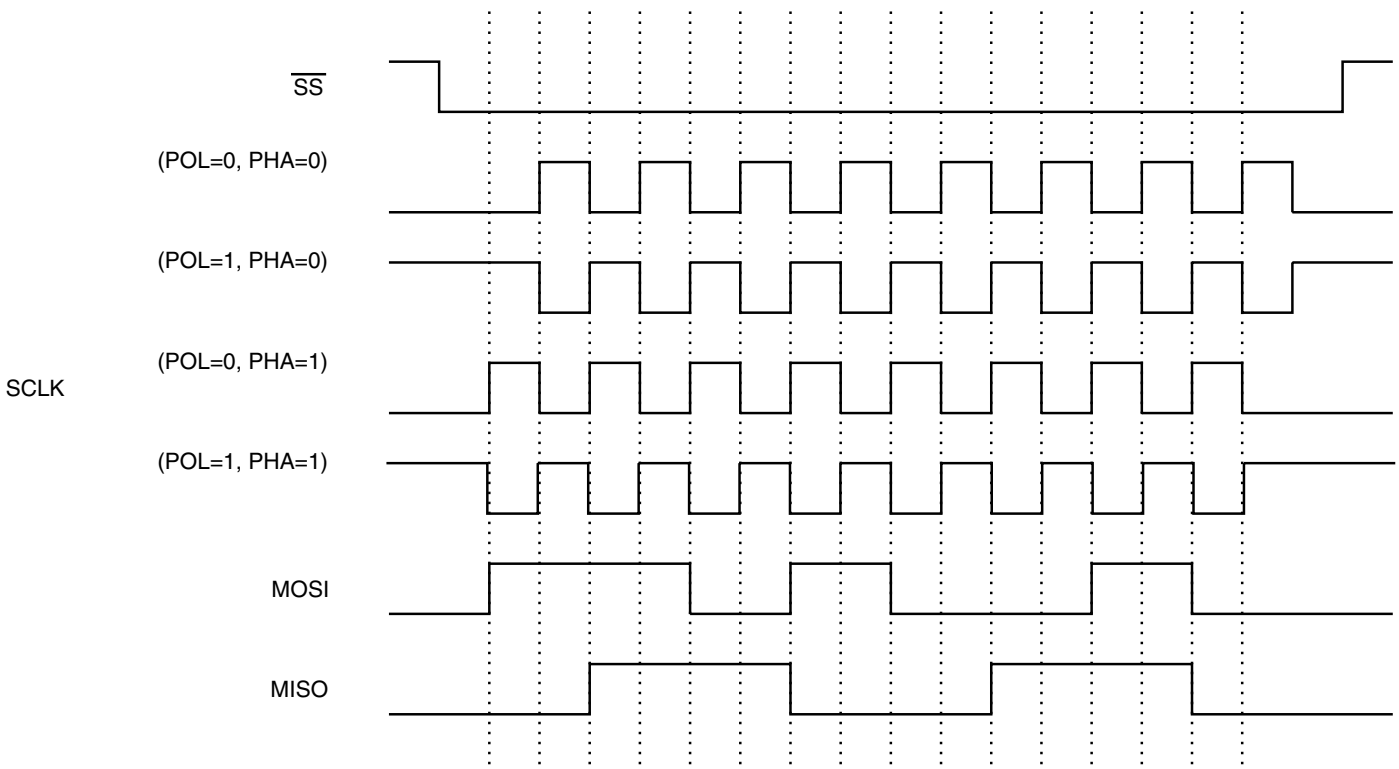


Figure 24-10. SPI Burst with Different POL and PHA Configurations

24.3.5.2 Typical Slave Mode

When the ECSPI is configured as a slave (Mode = 0), software can configure the ECSPI Control register to match the external SPI master's timing. In this configuration, SS becomes an input signal, and is used to latch data in and out of the internal data Shift registers, as well as to advance the data FIFO.

The SS, SCLK, and MOSI are inputs and MISO is output. Most of the timing diagrams are similar to the diagrams shown previously for the SPI in Master mode (Mode = 1), because the inputs come from a SPI master device.

However, the timing is different when SS is used to advance the data FIFO. When the SS_CTL[3:0] is set while the ECSPI is configured in Slave mode, the data FIFO will advance on the rising edge of the SS signal. When the polarity is reversed (SSPOL = 1), the data FIFO will advance on the falling edge of the SS signal.

The figure below shows a SPI burst in which the data FIFO is advanced by the rising edge of the SS signal.

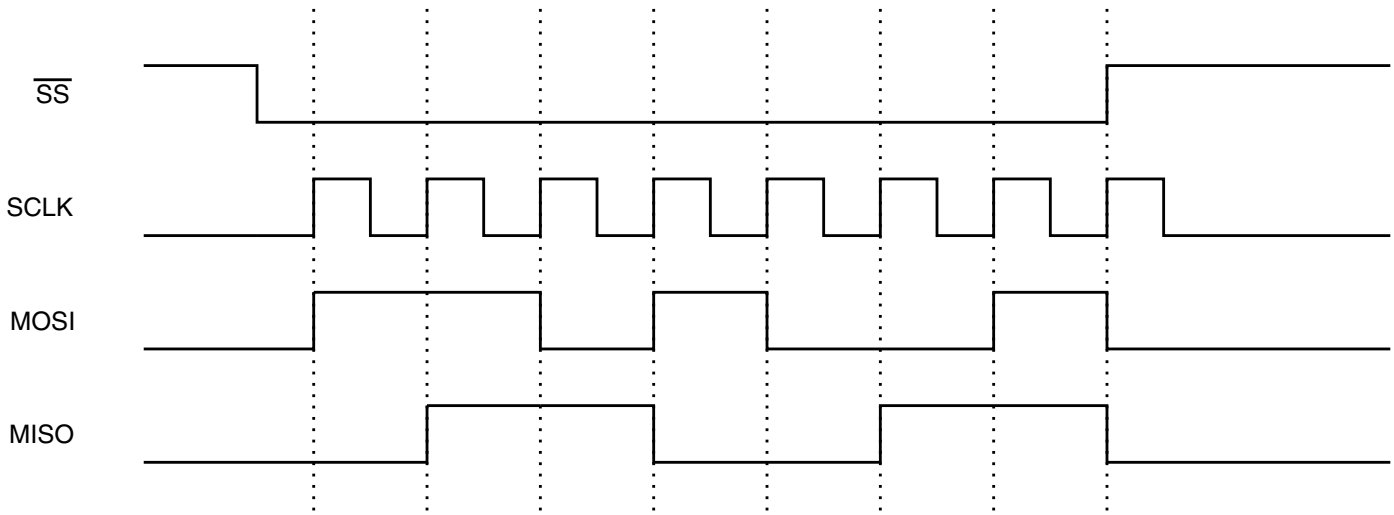


Figure 24-11. Advancing the Data FIFO on the Rising Edge of SS

In the above case, only the most significant 7 bits are loaded to the RXFIFO.

24.3.6 Clocks

This section describes clocks and special clocking requirements of the block.

ECSPI has the following clock inputs:

- Reference Clock is the reference clock.
- Low-Frequency Reference Clock is a 32KHz input clock optionally used for counting wait states.

24.3.7 Reset

Whenever a device reset occurs, a reset is performed on the ECSPI, resetting all registers to their default values.

Software can reset the block using the CONREG[EN] bit; see Control Register (ECSPI_CONREG).

24.3.8 Interrupts

Interrupt control provides a way to manage the ECSPI FIFOs:

- For transmitting data, software can enable the TXFIFO empty, TXFIFO data request, and TXFIFO full interrupts to maintain the TXFIFO using an interrupt service routine.
- For receiving data, software can enable the RXFIFO ready, RXFIFO data request, and RXFIFO full interrupts to retrieve data from the RXFIFO using an interrupt service routine.

Other interrupt sources can be used to control or debug the SPI bursts:

- The transfer-completed interrupt means that there is no data left in the TXFIFO and that the data in the Shift register has been shifted out.
- The RXFIFO overflow interrupt means that the RXFIFO received more than 64 words and will not accept any other words.

.

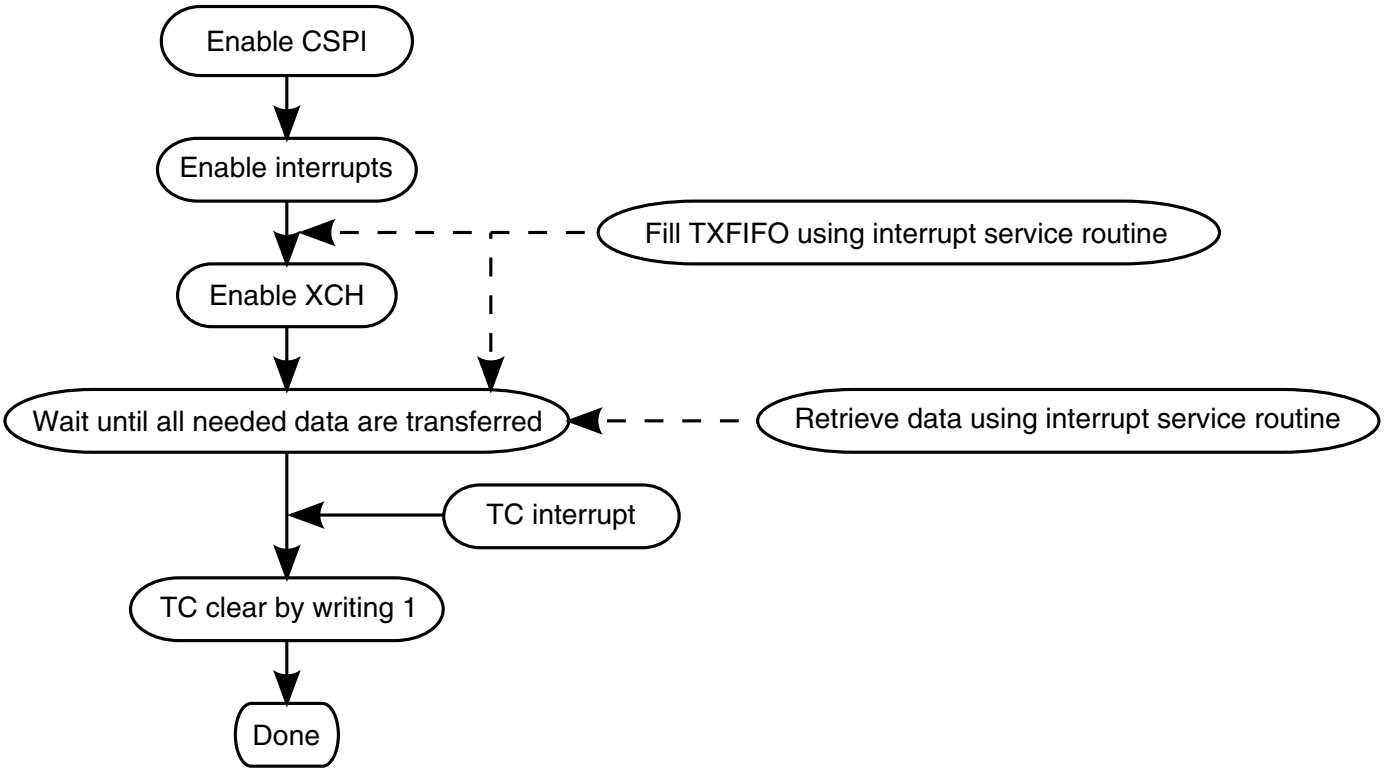


Figure 24-12. Program Sequence of SPI Burst Using Interrupt Control

24.3.9 DMA

DMA control provides another method to utilize the FIFOs in the ECSPI. By using DMA request and acknowledge signals, larger amounts of data can be transferred, and will reduce interrupts and host processor loading. When the appropriate conditions are matched, the block will send out a DMA request.

The DMA can deal with the following conditions:

- TXFIFO empty
- TXFIFO data request
- RXFIFO data request
- RXFIFO full

The figure below shows a program sequence of SPI bursts using DMA control.

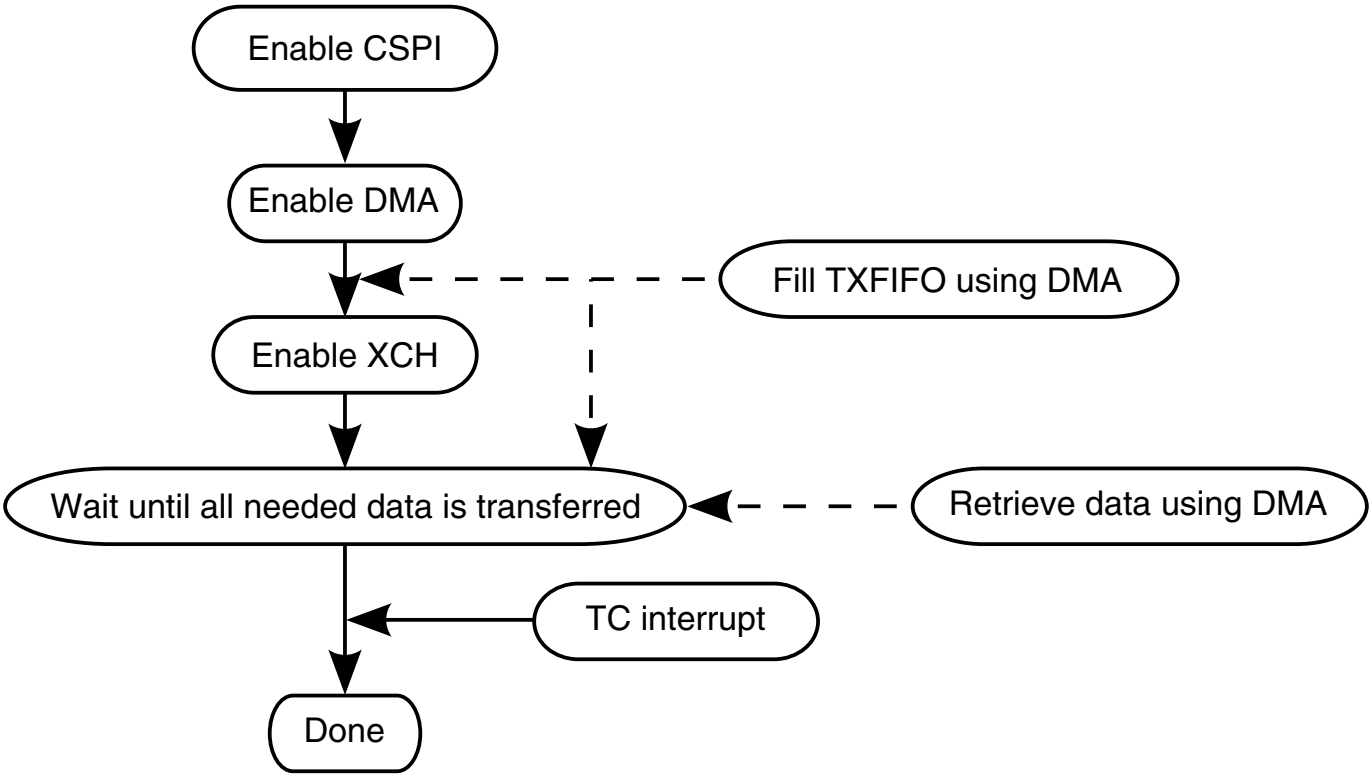


Figure 24-13. Program Sequence of SPI Burst Using DMA

24.3.10 Byte Order

The ECSPI does not support byte re-ordering in hardware.

24.4 Initialization

This section provides initialization information for ECSPI.

To initialize the block:

1. Clear the EN bit in ECSPI_CONREG to reset the block.
2. Enable the clocks for ECSPI.
3. Set the EN bit in ECSPI_CONREG to put ECSPI out of reset.
4. Configure corresponding IOMUX for ECSPI external signals.
5. Configure registers of ECSPI properly according to the specifications of the external SPI device.

24.5 Applications

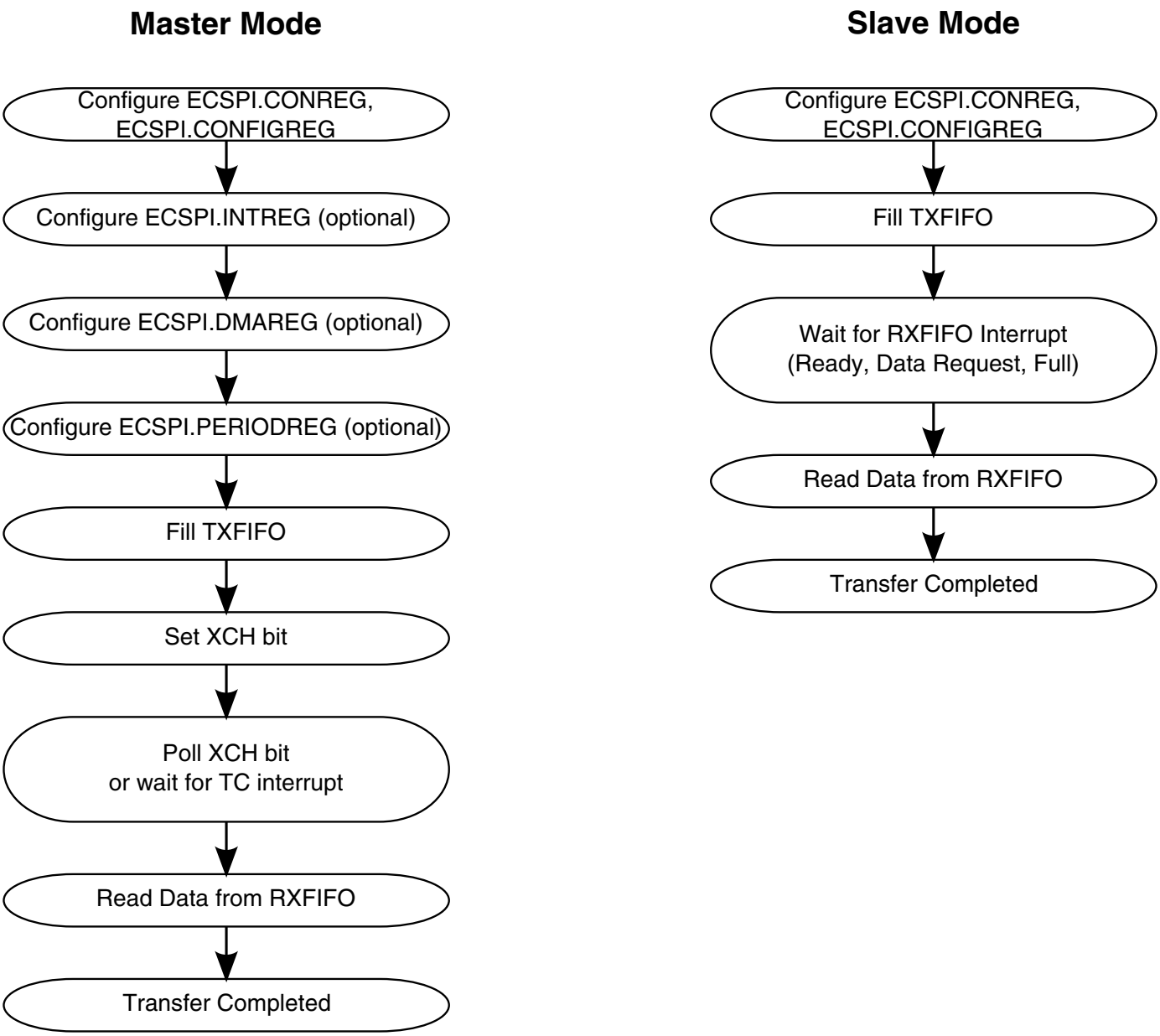


Figure 24-14. Flowchart of the ECSPI Operation

24.6 Programmable Registers

This section includes the block memory map and detailed descriptions of all registers. For the base address of a particular block instantiation, see the system memory map.

ECSPI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5001_0000	Receive Data Register (ECSPI-1_RXDATA)	32	R	0000_0000h	24.6.1/ 1054
5001_0004	Transmit Data Register (ECSPI-1_TXDATA)	32	W	0000_0000h	24.6.2/ 1055
5001_0008	Control Register (ECSPI-1_CONREG)	32	R/W	0000_0000h	24.6.3/ 1056
5001_000C	Config Register (ECSPI-1_CONFIGREG)	32	R/W	0000_0000h	24.6.4/ 1058
5001_0010	Interrupt Control Register (ECSPI-1_INTREG)	32	R/W	0000_0000h	24.6.5/ 1060
5001_0014	DMA Control Register (ECSPI-1_DMAREG)	32	R/W	0000_0000h	24.6.6/ 1061
5001_0018	Status Register (ECSPI-1_STATREG)	32	R/W	0000_0003h	24.6.7/ 1063
5001_001C	Sample Period Control Register (ECSPI-1_PERIODREG)	32	R/W	0000_0000h	24.6.8/ 1064
5001_0020	Test Control Register (ECSPI-1_TESTREG)	32	R/W	0000_0000h	24.6.9/ 1065
5001_0040	Message Data Register (ECSPI-1_MSGDATA)	32	W	0000_0000h	24.6.10/ 1066
63FA_C000	Receive Data Register (ECSPI-2_RXDATA)	32	R	0000_0000h	24.6.1/ 1054
63FA_C004	Transmit Data Register (ECSPI-2_TXDATA)	32	W	0000_0000h	24.6.2/ 1055
63FA_C008	Control Register (ECSPI-2_CONREG)	32	R/W	0000_0000h	24.6.3/ 1056
63FA_C00C	Config Register (ECSPI-2_CONFIGREG)	32	R/W	0000_0000h	24.6.4/ 1058
63FA_C010	Interrupt Control Register (ECSPI-2_INTREG)	32	R/W	0000_0000h	24.6.5/ 1060
63FA_C014	DMA Control Register (ECSPI-2_DMAREG)	32	R/W	0000_0000h	24.6.6/ 1061

Table continues on the next page...

ECSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FA_C018	Status Register (ECSPI-2_STATREG)	32	R/W	0000_0003h	24.6.7/ 1063
63FA_C01C	Sample Period Control Register (ECSPI-2_PERIODREG)	32	R/W	0000_0000h	24.6.8/ 1064
63FA_C020	Test Control Register (ECSPI-2_TESTREG)	32	R/W	0000_0000h	24.6.9/ 1065
63FA_C040	Message Data Register (ECSPI-2_MSGDATA)	32	W	0000_0000h	24.6.10/ 1066

24.6.1 Receive Data Register (ECSPIx_RXDATA)

The Receive Data register (ECSPI_RXDATA) is a read-only register that forms the top word of the 64 x 32 receive FIFO. This register holds the data received from an external SPI device during a data transaction. Only word-sized read operations are allowed.

Addresses: ECSPI-1_RXDATA is 5001_0000h base + 0h offset = 5001_0000h

ECSPI-2_RXDATA is 63FA_C000h base + 0h offset = 63FA_C000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ECSPI_RXDATA																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_RXDATA field descriptions

Field	Description
31–0 ECSPI_RXDATA	Receive Data. This register holds the top word of the receive data FIFO. The FIFO is advanced for each read of this register. The data read is undefined when the Receive Data Ready (RR) bit in the Interrupt Control/Status register is cleared. Zeros are read when ECSPI is disabled.

24.6.2 Transmit Data Register (ECSPIx_TXDATA)

The Transmit Data (ECSPI_TXDATA) register is a write-only data register that forms the bottom word of the 64 x 32 TXFIFO. The TXFIFO can be written to as long as it is not full, even when the SPI Exchange bit (XCH) in ECSPI_CONREG is set. This allows software to write to the TXFIFO during a SPI data exchange process. Writes to this register are ignored when the ECSPI is disabled (ECSPI_CONREG[EN] bit is cleared).

Addresses: ECSPI-1_TXDATA is 5001_0000h base + 4h offset = 5001_0004h

ECSPI-2_TXDATA is 63FA_C000h base + 4h offset = 63FA_C004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																																
W	ECSPI_TXDATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_TXDATA field descriptions

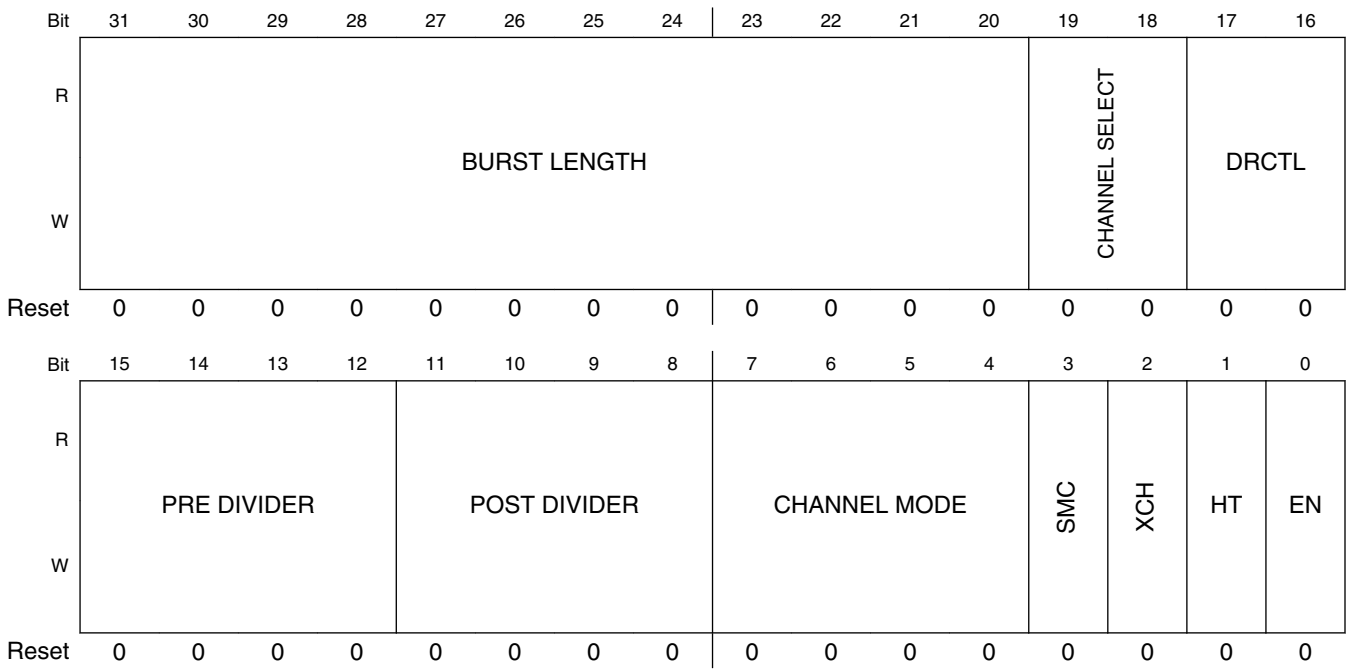
Field	Description
31–0 ECSPI_TXDATA	Transmit Data. This register holds the top word of data loaded into the FIFO. Data written to this register must be a word operation. The number of bits actually transmitted is determined by the BIT_COUNT field of the corresponding SPI Control register. If this field contains more bits than the number specified by BIT_COUNT, the extra bits are ignored. For example, to transfer 10 bits of data, a 32-bit word must be written to this register. Bits 9-0 are shifted out and bits 31-10 are ignored. When the ECSPI is operating in Slave mode, zeros are shifted out when the FIFO is empty. Zeros are read when ECSPI is disabled.

24.6.3 Control Register (ECSPIx_CONREG)

The Control Register (ECSPI_CONREG) allows software to enable the ECSPI , configure its operating modes, specify the divider value, and SPI_RDY control signal, and define the transfer length.

Addresses: ECSPI-1_CONREG is 5001_0000h base + 8h offset = 5001_0008h

ECSPI-2_CONREG is 63FA_C000h base + 8h offset = 63FA_C008h



ECSPIx_CONREG field descriptions

Field	Description
31–20 BURST LENGTH	Burst Length. This field defines the length of a SPI burst to be transferred. The Chip Select (SS) will remain asserted until all bits in a SPI burst are shifted out. A maximum of 2^12 bits can be transferred in a single SPI burst. In master mode, it controls the number of bits per SPI burst. Since the shift register always loads 32-bit data from transmit FIFO, only the n least-significant (n = BURST LENGTH + 1) will be shifted out. The remaining bits will be ignored. In slave mode, only when SS_CTL is cleared, this field will take effect in the transfer. Number of Valid Bits in a SPI burst. 0x000 A SPI burst contains the 1 LSB in a word. 0x001 A SPI burst contains the 2 LSB in a word. 0x002 A SPI burst contains the 3 LSB in a word. 0x01F A SPI burst contains all 32 bits in a word. 0x020 A SPI burst contains the 1 LSB in first word and all 32 bits in second word. 0x021 A SPI burst contains the 2 LSB in first word and all 32 bits in second word.

Table continues on the next page...

ECSPIx_CONREG field descriptions (continued)

Field	Description
	0xFFE A SPI burst contains the 31 LSB in first word and 2^7 -1 words. 0xFFF A SPI burst contains 2^7 words.
19–18 CHANNEL SELECT	SPI CHANNEL SELECT bits. Select one of four external SPI Master/Slave Devices. In master mode, these two bits select the external slave devices by asserting the Chip Select (SSn) outputs. Only the selected Chip Select (SSn) signal can be active at a given time; the remaining three signals will be negated. 00 Channel 0 is selected. Chip Select 0 (SS0) will be asserted. 01 Channel 1 is selected. Chip Select 1 (SS1) will be asserted. 10 Channel 2 is selected. Chip Select 2 (SS2) will be asserted. 11 Channel 3 is selected. Chip Select 3 (SS3) will be asserted.
17–16 DRCTL	SPI Data Ready Control. This field selects the utilization of the $\overline{\text{SPI\_RDY}}$ signal in master mode. ECSPI checks this field before it starts an SPI burst. 00 The $\overline{\text{SPI\_RDY}}$ signal is a don't care. 01 Burst will be triggered by the falling edge of the SPI_RDY signal (edge-triggered). 10 Burst will be triggered by a low level of the SPI_RDY signal (level-triggered). 11 Reserved.
15–12 PRE DIVIDER	SPI Pre Divider. ECSPI uses a two-stage divider to generate the SPI clock. This field defines the pre-divider of the reference clock. 0000 Divide by 1. 0001 Divide by 2. 0010 Divide by 3. 1101 Divide by 14. 1110 Divide by 15. 1111 Divide by 16.
11–8 POST DIVIDER	SPI Post Divider. ECSPI uses a two-stage divider to generate the SPI clock. This field defines the post-divider of the reference clock using the equation: 2^{n} . 0000 Divide by 1. 0001 Divide by 2. 0010 Divide by 4. 1110 Divide by 2^{14} . 1111 Divide by 2^{15} .
7–4 CHANNEL MODE	SPI CHANNEL MODE selects the mode for each SPI channel. CHANNEL MODE[3] is for SPI channel 3. CHANNEL MODE[2] is for SPI channel 2. CHANNEL MODE[1] is for SPI channel 1. CHANNEL MODE[0] is for SPI channel 0. 0 Slave mode. 1 Master mode.
3 SMC	Start Mode Control. This bit applies only to channels configured in Master mode (CHANNEL MODE = 1).

Table continues on the next page...

ECSPIx_CONREG field descriptions (continued)

Field	Description
	It controls how the ECSPI starts a SPI burst, either through the SPI exchange bit, or immediately when the TXFIFO is written to. 0 SPI Exchange Bit (XCH) controls when a SPI burst can start. Setting the XCH bit will start a SPI burst or multiple bursts. This is controlled by the SPI SS Wave Form Select (SS_CTL). Refer to XCH and SS_CTL descriptions. 1 Immediately starts a SPI burst when data is written in TXFIFO.
2 XCH	SPI Exchange Bit. This bit applies only to channels configured in Master mode (CHANNEL MODE = 1). If the Start Mode Control (SMC) bit is cleared, writing a 1 to this bit starts one SPI burst or multiple SPI bursts according to the SPI SS Wave Form Select (SS_CTL). The XCH bit remains set while either the data exchange is in progress, or when the ECSPI is waiting for an active input if SPIRDY is enabled through DRCTL. This bit is cleared automatically when all data in the TXFIFO and the shift register has been shifted out. 0 Idle. 1 Initiates exchange (write) or busy (read).
1 HT	Hardware Trigger Enable. This bit is used in master mode only. It enables hardware trigger (HT) mode. 0 Disable HT mode. 1 Enable HT mode.
0 EN	SPI Block Enable Control. This bit enables the ECSPI. This bit must be set before writing to other registers or initiating an exchange. Writing zero to this bit disables the block and resets the internal logic with the exception of the ECSPI_CONREG. The block's internal clocks are gated off whenever the block is disabled. 0 Disable the block. 1 Enable the block.

24.6.4 Config Register (ECSPIx_CONFIGREG)

The Config Register (ECSPI_CONFIGREG) allows software to configure each SPI channel, configure its operating modes, specify the phase and polarity of the clock, configure the Chip Select (SS), and define the HT transfer length.

Addresses: ECSPI-1_CONFIGREG is 5001_0000h base + Ch offset = 5001_000Ch

ECSPI-2_CONFIGREG is 63FA_C000h base + Ch offset = 63FA_C00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-			HT LENGTH					SCLK CTL				DATA CTL				SS POL				SS CTL				SCLK POL				SCLK PHA			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_CONFIGREG field descriptions

Field	Description
31–29 -	Reserved
28–24 HT LENGTH	HT LENGTH. This field defines the message length in HT Mode. The length in bits of one message is (HT LENGTH + 1).
23–20 SCLK CTL	SCLK CTL. This field controls the inactive state of SCLK for each SPI channel. SCLK CTL[3] is for SPI channel 3. SCLK CTL[2] is for SPI channel 2. SCLK CTL[1] is for SPI channel 1. SCLK CTL[0] is for SPI channel 0. 0 Stay high. 1 Stay low.
19–16 DATA CTL	DATA CTL. This field controls inactive state of the data line for each SPI channel. DATA CTL[3] is for SPI channel 3. DATA CTL[2] is for SPI channel 2. DATA CTL[1] is for SPI channel 1. DATA CTL[0] is for SPI channel 0. 0 Stay high. 1 Stay low.
15–12 SS POL	SPI SS Polarity Select. In both Master and Slave modes, this field selects the polarity of the Chip Select (SS) signal. SS POL[3] is for SPI channel 3. SS POL[2] is for SPI channel 2. SS POL[1] is for SPI channel 1. SS POL[0] is for SPI channel 0. 0 Active low. 1 Active high.
11–8 SS CTL	SPI SS Wave Form Select. In master mode, this field controls the output wave form of the Chip Select (SS) signal when the SMC (Start Mode Control) bit is cleared. The SS_CTL are ignored if the SMC bit is set. SS CTL[3] is for SPI channel 3. SS CTL[2] is for SPI channel 2. SS CTL[1] is for SPI channel 1. SS CTL[0] is for SPI channel 0. In slave mode, this bit controls when the SPI burst is completed. 1 An SPI burst is completed by the Chip Select (SS) signal edges. (SSPOL = 0: rising edge; SSPOL = 1: falling edge) The RXFIFO is advanced whenever a Chip Select (SS) signal edge is detected or the shift register contains 32-bits of valid data. 0 In master mode - only one SPI burst will be transmitted.

Table continues on the next page...

ECSPIx_CONFIGREG field descriptions (continued)

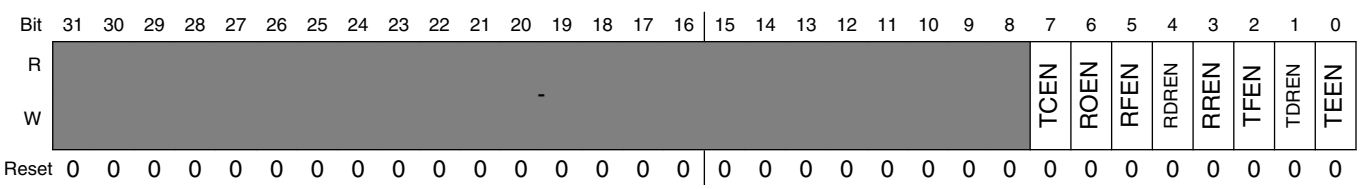
Field	Description
	1 In master mode - Negate Chip Select (SS) signal between SPI bursts. Multiple SPI bursts will be transmitted. The SPI transfer will automatically stop when the TXFIFO is empty. 0 In slave mode - an SPI burst is completed when the number of bits received in the shift register is equal to (BURST LENGTH + 1). Only the n least-significant bits (n = BURST LENGTH[4:0] + 1) of the first received word are valid. All bits subsequent to the first received word in RXFIFO are valid. 1 In slave mode - an SPI burst is completed by the Chip Select (SS) signal edges. (SSPOL = 0: rising edge; SSPOL = 1: falling edge) The RXFIFO is advanced whenever a Chip Select (SS) signal edge is detected or the shift register contains 32-bits of valid data.
7–4 SCLK POL	SPI Clock Polarity Control. This field controls the polarity of the SCLK signal. See Figure 24-10 for more information. SCLK POL[3] is for SPI channel 3. SCLK POL[2] is for SPI channel 2. SCLK POL[1] is for SPI channel 1. SCLK POL[0] is for SPI channel 0. 0 Active high polarity (0 = Idle). 1 Active low polarity (1 = Idle).
3–0 SCLK PHA	SPI Clock/Data Phase Control. This field controls the clock/data phase relationship. See Figure 24-10 for more information. SCLK PHA[3] is for SPI channel 3. SCLK PHA[2] is for SPI channel 2. SCLK PHA[1] is for SPI channel 1. SCLK PHA[0] is for SPI channel 0. 0 Phase 0 operation. 1 Phase 1 operation.

24.6.5 Interrupt Control Register (ECSPIx_INTREG)

The Interrupt Control Register (ECSPI_INTREG) enables the generation of interrupts to the host processor. If the ECSPI is disabled, this register reads zero.

Addresses: ECSPI-1_INTREG is 5001_0000h base + 10h offset = 5001_0010h

ECSPI-2_INTREG is 63FA_C000h base + 10h offset = 63FA_C010h



ECSPIx_INTREG field descriptions

Field	Description
31–8 -	Reserved
7 TCEN	Transfer Completed Interrupt enable. This bit enables the Transfer Completed Interrupt. 0 Disable 1 Enable
6 ROEN	RXFIFO Overflow Interrupt enable. This bit enables the RXFIFO Overflow Interrupt. 0 Disable 1 Enable
5 RFEN	RXFIFO Full Interrupt enable. This bit enables the RXFIFO Full Interrupt. 0 Disable 1 Enable
4 RDREN	RXFIFO Data Request Interrupt enable. This bit enables the RXFIFO Data Request Interrupt when the number of data entries in the RXFIFO is greater than RX THRESHOLD. 0 Disable 1 Enable
3 RREN	RXFIFO Ready Interrupt enable. This bit enables the RXFIFO Ready Interrupt. 0 Disable 1 Enable
2 TFEN	TXFIFO Full Interrupt enable. This bit enables the TXFIFO Full Interrupt. 0 Disable 1 Enable
1 TDREN	TXFIFO Data Request Interrupt enable. This bit enables the TXFIFO Data Request Interrupt when the number of data entries in the TXFIFO is less than or equal to TX THRESHOLD. 0 Disable 1 Enable
0 TEEN	TXFIFO Empty Interrupt enable. This bit enables the TXFIFO Empty Interrupt. 0 Disable 1 Enable

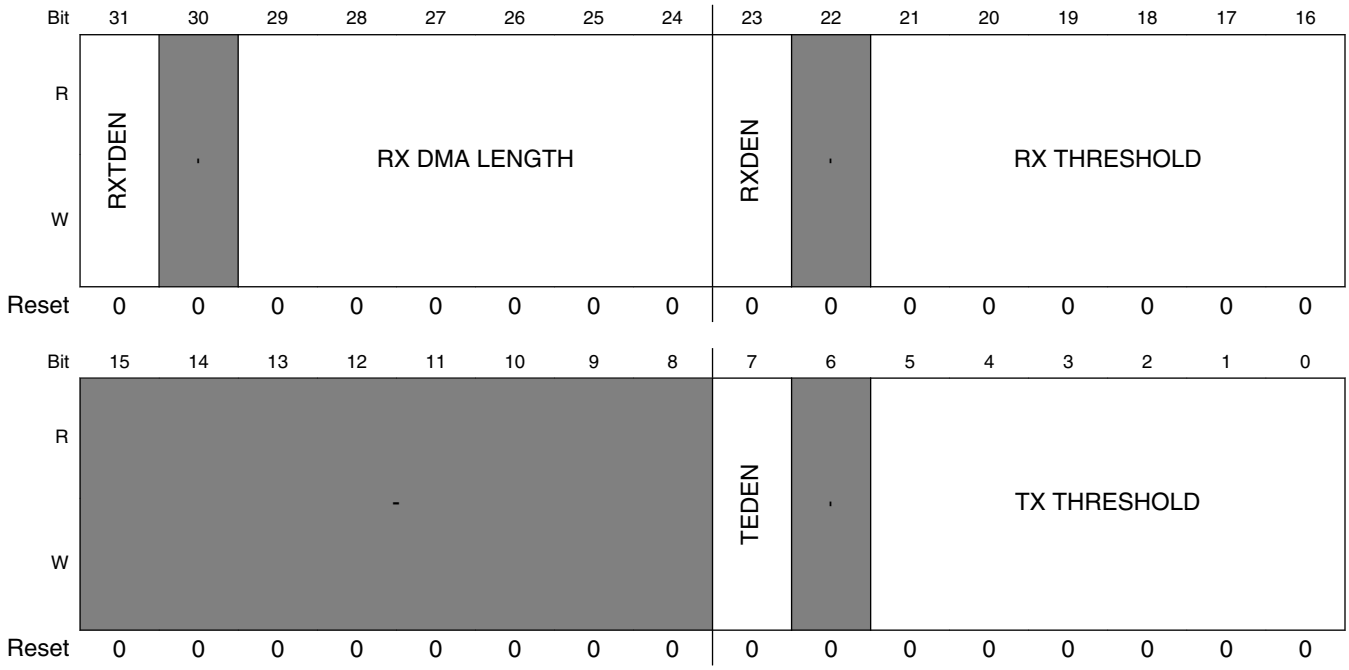
24.6.6 DMA Control Register (ECSPIx_DMAREG)

The Direct Memory Access Control Register (ECSPI_DMAREG) provides software a way to use an on-chip DMA controller for ECSPI data. Internal DMA request signals enable direct data transfers between the ECSPI FIFOs and system memory. The ECSPI sends out DMA requests when the appropriate FIFO conditions are matched.

If the ECSPI is disabled, this register is read as 0.

Addresses: ECSPI-1_DMAREG is 5001_0000h base + 14h offset = 5001_0014h

ECSPI-2_DMAREG is 63FA_C000h base + 14h offset = 63FA_C014h



ECSPIx_DMAREG field descriptions

Field	Description
31 RXTDEN	RXFIFO TAIL DMA Request Enable. This bit enables an internal counter that is increased at each read of the RXFIFO. This counter is cleared automatically when it reaches RX DMA LENGTH. If the number of words remaining in the RXFIFO is greater than or equal to RX DMA LENGTH, a DMA request is generated even if it is less than or equal to RX THRESHOLD. 0 Disable 1 Enable
30 -	Reserved
29–24 RX DMA LENGTH	RX DMA LENGTH. This field defines the burst length of a DMA operation. Applies only when RXTDEN is set.
23 RXDEN	RXFIFO DMA Request Enable. This bit enables/disables the RXFIFO DMA Request. 0 Disable 1 Enable
22 -	Reserved
21–16 RX THRESHOLD	RX THRESHOLD. This field defines the FIFO threshold that triggers a RX DMA/INT request. A RX DMA/INT request is issued when the number of data entries in the RXFIFO is greater than RX THRESHOLD.
15–8 -	Reserved
7 TEDEN	TXFIFO Empty DMA Request Enable. This bit enables/disables the TXFIFO Empty DMA Request.

Table continues on the next page...

ECSPIx_DMAREG field descriptions (continued)

Field	Description
	0 Disable 1 Enable
6 -	Reserved
5–0 TX THRESHOLD	TX THRESHOLD. This field defines the FIFO threshold that triggers a TX DMA/INT request. A TX DMA/INT request is issued when the number of data entries in the TXFIFO is greater than TX THRESHOLD.

24.6.7 Status Register (ECSPIx_STATREG)

The ECSPI Status Register (ECSPI_STATREG) reflects the status of the ECSPI's operating condition. If the ECSPI is disabled, this register reads 0x0000_0003.

Addresses: ECSPI-1_STATREG is 5001_0000h base + 18h offset = 5001_0018h

ECSPI-2_STATREG is 63FA_C000h base + 18h offset = 63FA_C018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-								TC	RO	RF	RDR	RR	TF	TDR	TE
W									w1c	w1c						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

ECSPIx_STATREG field descriptions

Field	Description
31–8 -	Reserved
7 TC	Transfer Completed Status bit. Writing 1 to this bit clears it. 0 Transfer in progress. 1 Transfer completed.
6 RO	RXFIFO Overflow. When set, this bit indicates that RXFIFO has overflowed. Writing 1 to this bit clears it. 0 RXFIFO has no overflow. 1 RXFIFO has overflowed.
5 RF	RXFIFO Full. This bit is set when the RXFIFO is full. 0 Not Full. 1 Full.

Table continues on the next page...

ECSPIx_STATREG field descriptions (continued)

Field	Description
4 RDR	RXFIFO Data Request. 0 When RXTDE is set - Number of data entries in the RXFIFO is not greater than RX THRESHOLD. 1 When RXTDE is set - Number of data entries in the RXFIFO is greater than RX THRESHOLD or a DMA TAIL DMA condition exists. 0 When RXTDE is clear - Number of data entries in the RXFIFO is not greater than RX THRESHOLD. 1 When RXTDE is clear - Number of data entries in the RXFIFO is greater than RX THRESHOLD.
3 RR	RXFIFO Ready. This bit is set when one or more words are stored in the RXFIFO. 0 No valid data in RXFIFO. 1 More than 1 word in RXFIFO.
2 TF	TXFIFO Full. This bit is set when if the TXFIFO is full. 0 TXFIFO is not Full. 1 TXFIFO is Full.
1 TDR	TXFIFO Data Request. 0 Number of empty slots in TXFIFO is greater than TX THRESHOLD. 1 Number of empty slots in TXFIFO is not greater than TX THRESHOLD.
0 TE	TXFIFO Empty. This bit is set if the TXFIFO is empty. 0 TXFIFO contains one or more words. 1 TXFIFO is empty.

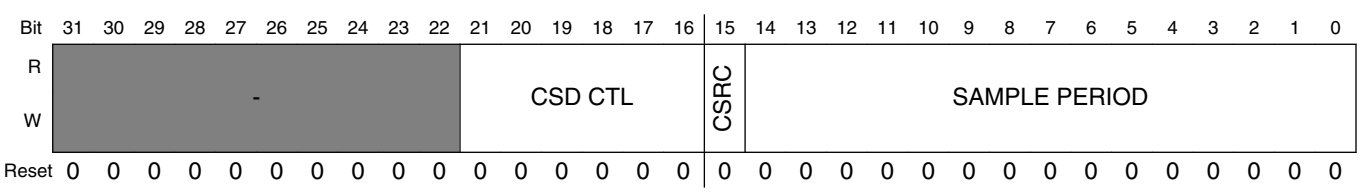
24.6.8 Sample Period Control Register (ECSPIx_PERIODREG)

The Sample Period Control Register (ECSPI_PERIODREG) provides software a way to insert delays (wait states) between consecutive SPI transfers. Control bits in this register select the clock source for the sample period counter and the delay count indicating the number of wait states to be inserted between data transfers.

The delay counts apply only when the current channel is operating in Master mode (ECSPI_CONREG[CHANNEL MODE] = 1).ECSPI_PERIODREG also contains the CSD CTRL field used to insert a delay between the Chip Select's active edge and the first SPI Clock edge.

Addresses: ECSPI-1_PERIODREG is 5001_0000h base + 1Ch offset = 5001_001Ch

ECSPI-2_PERIODREG is 63FA_C000h base + 1Ch offset = 63FA_C01Ch



ECSPIx_PERIODREG field descriptions

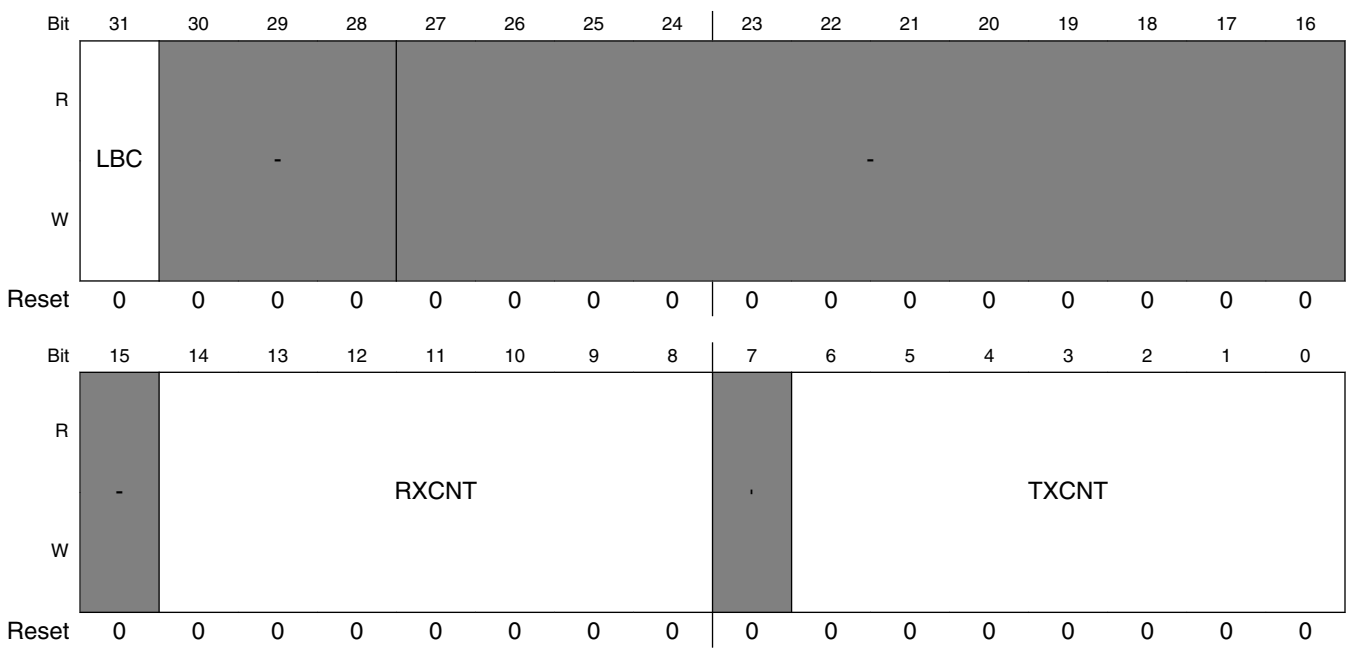
Field	Description
31–22 -	Reserved
21–16 CSD CTL	Chip Select Delay Control bits. This field defines how many SPI clocks will be inserted between the chip select's active edge and the first SPI clock edge. The range is from 0 to 63.
15 CSRC	Clock Source Control. This bit selects the clock source for the sample period counter. 0 SPI Clock (SCLK) 1 Low-Frequency Reference Clock (32.768 KHz)
14–0 SAMPLE PERIOD	Sample Period Control. These bits control the number of wait states to be inserted in data transfers. During the idle clocks, the state of the SS output will operate according to the SS_CTL control field in the ECSPI_CONREG register. 0x0000 0 wait states inserted 0x0001 1 wait state inserted ... ... 0x7FFE 32766 wait states inserted 0x7FFF 32767 wait states inserted

24.6.9 Test Control Register (ECSPIx_TESTREG)

The Test Control Register (ECSPI_TESTREG) provides software a mechanism to internally connect the receive and transmit devices of the ECSPI, and monitor the contents of the receive and transmit FIFOs.

Addresses: ECSPI-1_TESTREG is 5001_0000h base + 20h offset = 5001_0020h

ECSPI-2_TESTREG is 63FA_C000h base + 20h offset = 63FA_C020h



ECSPIx_TESTREG field descriptions

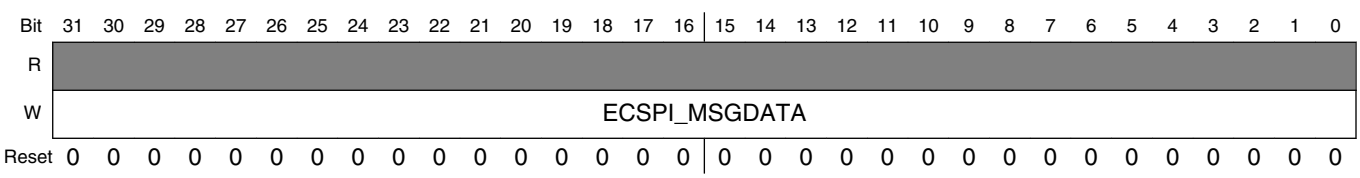
Field	Description
31 LBC	Loop Back Control. This bit is used in Master mode only. When this bit is set, the ECSPI connects the transmitter and receiver sections internally, and the data shifted out from the most-significant bit of the shift register is looped back into the least-significant bit of the Shift register. In this way, a self-test of the complete transmit/receive path can be made. The output pins are not affected, and the input pins are ignored. 0 Not connected. 1 Transmitter and receiver sections internally connected for Loopback.
30–28 -	Reserved, all bits should be ignored.
27–15 -	Reserved
14–8 RXCNT	RXFIFO Counter. This field indicates the number of words in the RXFIFO.
7 -	Reserved
6–0 TXCNT	TXFIFO Counter. This field indicates the number of words in the TXFIFO.

24.6.10 Message Data Register (ECSPIx_MSGDATA)

The Message Data Register (ECSPI_MSGDATA) forms the top word of the 16 x 32 MSG Data FIFO. Only word-size accesses are allowed for this register. Reads to this register return zero, and writes to this register store data in the MSG Data FIFO. See Hardware Trigger (HT) ModeIn hardware trigger (HT) mode, the behavior of the ECSPI is similar to master mode. The differences are in the trigger mechanism and the burst length. .

Addresses: ECSPI-1_MSGDATA is 5001_0000h base + 40h offset = 5001_0040h

ECSPI-2_MSGDATA is 63FA_C000h base + 40h offset = 63FA_C040h



ECSPIx_MSGDATA field descriptions

Field	Description
31–0 ECSPI_ MSGDATA	ECSPI_MSGDATA holds the top word of MSG Data FIFO. The MSG Data FIFO is advanced for each write of this register. The data read is zero. The data written to this register is stored in the MSG Data FIFO.

Chapter 25
External Interface Module (EIM)

25.1 Overview

The EIM handles the interface to devices external to the chip, including generation of chip selects, clock and control for external peripherals and memory. It provides asynchronous access to devices with SRAM-like interface and synchronous access to devices with NOR-Flash-like or PSRAM-like interface.

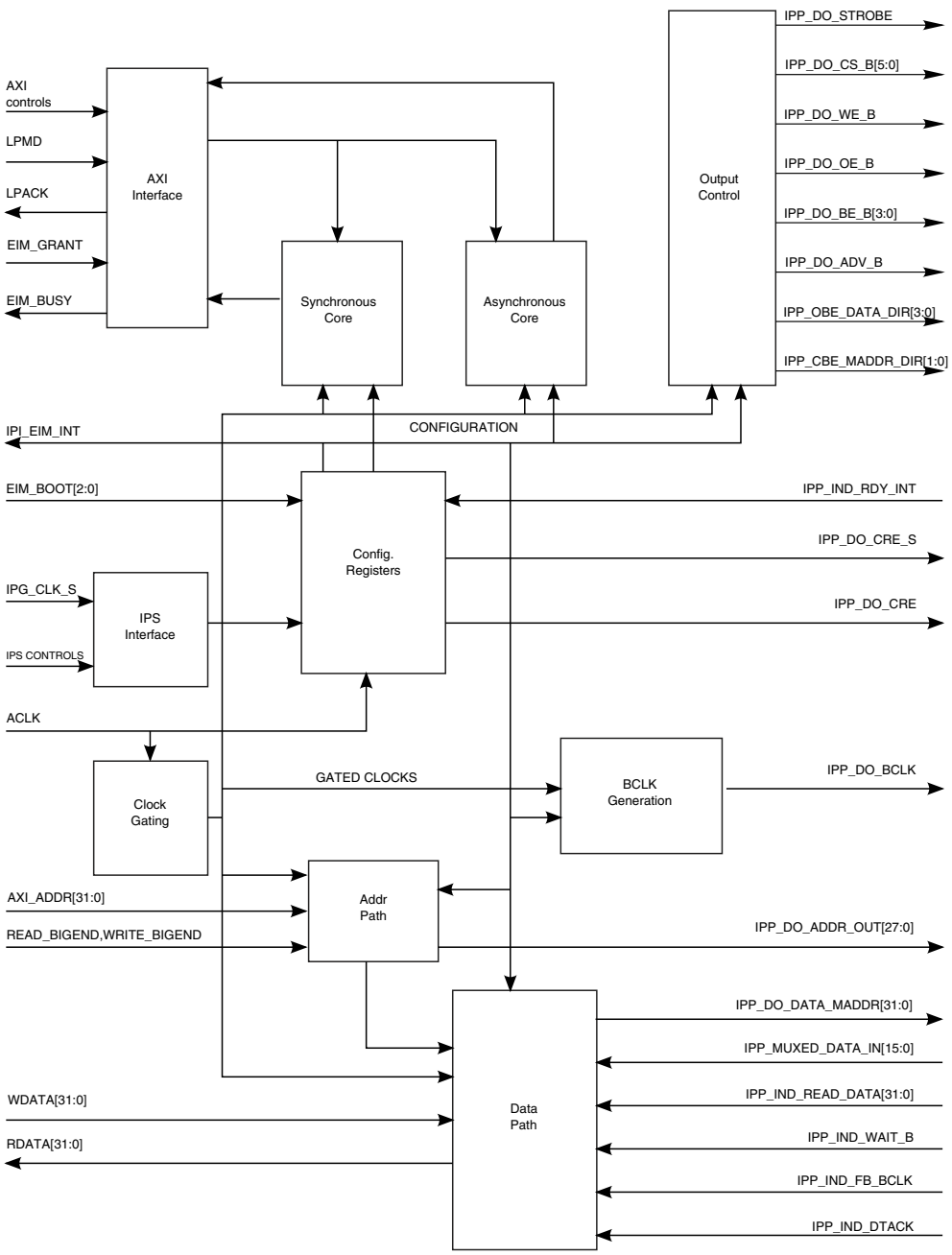


Figure 25-1. EIM Diagram

25.2 Features

- Up to four chip selects for external devices
  - Flexible address decoding. Each chip select memory space determined separately, according to VIA port configuration (see Chip Select Memory Map). Configurable Chip Select 0 base address (by VIA)
  - Individual select signal for each one of the memory space defined. Up to 6 memory spaces may be defined and programmed individually.
  - 28-bit external address bus, max memory size can be 256MByte (2 Gigabit).
- Selectable Write Protection for each Chip Select
- Support for multiplexed address / data bus operation x16 and x32 port size
- Programmable Data Port Size for each Chip Select (x8, x16 and x32)
- Programmable Wait-State generator for each Chip Select, for write and read accesses separately
- Asynchronous accesses with programmable setup and hold times for control signals
- Support for Asynchronous page mode accesses (x16 and x32 port size)
- Independent synchronous Memory Burst Read Mode support for NOR-Flash and PSRAM memories (x16 and x32 port size)
- Independent synchronous Memory Burst Write Mode support for PSRAM and NOR-Flash like memories (CellularRAM™ from Micron, Infineon, and Cypress, OneNAND™ and utRAM™ from Samsung, and COSMORAM™ from Toshiba)
- Support of NAND-Flash devices with NOR-Flash like interface - MDOC™ (M-Systems), OneNAND™ (Samsung)
- Independent programable variable/fix Latency support for read and write synchronous (burst) mode
- Support for Big Endian and Little Endian operation modes per access
- ARM AXI slave interface. One ID at a time support.
- External Interrupt support, RDY_INT signal function as external interrupt
- Boot from external device support according to boot signals, using RDY_INT signal
  - RDY signal support assertion after reset for MDOC™ (M-Systems) device
  - INT signal support assertion after reset for OneNAND™ (Samsung) device

25.3 Modes of Operation

The EIM has the following modes of operation:

- Asynchronous Mode
- Asynchronous Page Mode
- Multiplexed Address/Data mode

- Burst Clock Mode
- Low Power Modes
- Boot Mode

See details in the EIM Operational Modes.

25.3.1 Asynchronous Mode

This is a non-burst mode that is used for SRAM access. In this mode, a single data is read/written with each access (asserted address).

All controls' timings are controlled by preset values in Chip Select Configuration Registers.

25.3.2 Asynchronous Page Read Mode

Setting the APR bit causes the EIM to perform memory burst accesses by emulating page mode operation.

The external address asserts for each piece of data. The initial access timing is according to RWSC field, and the next address assertions timing is according to PAT field. When APR bit is set, RCSN OEN, RADVN and RBEN fields are ignored for burst access to the external device.

The page size can be set via the BL field to 2, 4, 8, 16, or 32 words (the word size is determined by the DSZ field).

25.3.3 Multiplexed Address/Data Mode

In this mode, multiplexing addresses and data bits on the same pins is supported for synchronous/asynchronous accesses to x8/x16/ x32 data width memory devices.

For more information about the pins that drive data/address in 8/16/32 non-muxed mode and 16/32 muxed mode, refer to the EIM Internal Module Multiplexing table in the EIM Internal Pads Allocation chapter of the datasheet.

Table 25-1. EIM multiplexing

Setup	Non Multiplexed Address/Data Mode						Multiplexed Address/ Data mode	
	8 Bit			16 Bit		32 Bit	16 Bit	32 Bit
	MUM = 0, DSZ = 100	MUM = 0, DSZ = 101	MUM = 0, DSZ = 111	MUM = 0, DSZ = 001	MUM = 0, DSZ = 010	MUM = 0, DSZ = 011	MUM = 1, DSZ = 001	MUM = 1, DSZ = 011
A[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]
A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[24:16][1]	EIM_A[25:16]	EIM_A[8:0]
D[7:0], EIM_EB0	NANDF_D[7:0]	-	-	NANDF_D[7:0][2]	-	NANDF_D[7:0]	EIM_DA[7:0]	EIM_DA[7:0]
D[15:8], EIM_EB1	-	NANDF_D[15:8]	-	NANDF_D[15:8][3]	-	NANDF_D[15:8]	EIM_DA[15:8]	EIM_DA[15:8]
D[23:16], EIM_EB2	-	-	-	-	EIM_D[23:16]	EIM_D[23:16]	-	NANDF_D[7:0]
D[31:24], EIM_EB3	-	-	EIM_D[31:24]	-	EIM_D[31:24]	EIM_D[31:24]	-	NANDF_D[15:8]

1. For 32-bit, the address range is A[24:0], due to address space allocation in memory map.
2. NANDF_D[7:0] multiplexed on ALT3 mode of PATA_DATA[7:0]
3. NANDF_D[15:8] multiplexed on ALT3 mode of PATA_DATA[15:8]

25.3.4 Burst Clock Mode

The controller has the ability to support a burst synchronous operations in a various frequencies, depending on the frequency of the input clock supplied by the system (EIM clock).

The EIM clock can be divided by one, two, three or four, and its frequency can be changed according to the requirements. Variable and fix latency are supported for this mode, according to the external device requirements.

- Synchronous read mode. This is a burst mode, which is used for reading from Flash/ PSRAM memory devices. In this mode, after address assertion a burst of sequential data can be read. Data exchange is carried out according to BCLK being generated by EIM. An access is delayed according to external WAIT_B signal assertion (signal from the memory device).
- Synchronous write mode. A burst mode used for accessing external devices, which support synchronous write type of access (PSRAM protocol). In this mode, after address assertion a burst of sequential data can be written to the external device. Access may be delayed according to WAIT_B signal assertion (signal from the memory device) before first piece of data arrived to the external device.

NOTE

Maximum frequency of the EIM main clock is 133Mhz. It may be reduced by the system for special cases of external devices, which demand a different frequency then integer division of the 133MHz clock.

25.3.5 Low Power Modes

The Input Clock is gated by ACT_CS bits. When all the ACT_CS are negated (all CS disable) the internal clock is turned off; awready/wready & arready signal are de-asserted and the master can't access the EIM.

25.3.6 Boot Mode

It is possible to perform a boot operation from external device located on CS0. The configuration of the relevant bits are done with boot Mode signals according to the external device parameters (for example, port size and protocol assertion).

See System Boot for more details.

Table 25-2. EIM Boot configuration

EIM_BOOT bits	EIM affected bits	EIM Register	Boot Value
2	MUM OEA	EIM_CS0GCR1 EIM_CS0RCR1	Configurable by fuses; If SRC.SBMR[BOOT_CFG2[7:6]] = 0b00, value = 0x100 If SRC.SBMR[BOOT_CFG2[7:6]] = 0b00, value = 0x010
[1:0]	DSZ[1:0]	EIM_CS0GCR1	

25.4 External Signal Description

25.4.1 Signals Overview

Table 25-3. Block I/Os

Name	Function	Input / Output	Notes
WEIM_A[26:16]	Address Bus MSB	Output	

Table continues on the next page...

Table 25-3. Block I/Os (continued)

Name	Function	Input / Output	Notes
WEIM_DA[15:0]	Address/Data Bus LSB	I/O	NFC data lines can be multiplexed on these lines as well
EIM_NFC_D[15:0]	Data LSB	I/O	The EIM LSB data signals can be shared with the NFC data lines through a multiplexor in the EXTMC module, so the EIM LSB data signals can be routed out through either the EIM_NFC_D[15:0] path OR the EIM_DA[15:0] path
WEIM_BCLK	Burst Clock	Output	
WEIM_CRE	Memory Register Set	Output	
WEIM_CS[5:0]	Chip Selects	Output	
WEIM_DTACK_B	Data Acknowledge	Input	
WEIM_D[31:16]	Data MSB	I/O	
WEIM_EB[3:0]	Enable Byte Signals	Output	
WEIM_LBA	Load Burst Address	Output	
WEIM_OE	Output Enable	Output	
WEIM_RW	Read/Write	Output	
WEIM_WAIT	Memory Ready/Busy/Wait	Input	

25.4.2 Detailed Signal Descriptions

The following is a detailed description of the EIM signals mentioned in Table 25-3.

Table 25-4. Block I/O Descriptions

Name	I/O	Description
WEIM_A[26:16]	O	MSB Address Bus
WEIM_DA[15:0]	I/O	LSB multiplexed Address/Data Bus
EIM_NFC_D[15:0]	I/O	LSB Data Bus (multiplexed with NFC data bus signals)
WEIM_BCLK	O	Burst Clock (BCLK). This active-high output signal is used to clock external burst-capable devices to synchronize the loading and incrementing of addresses and delivery of burst read and write data to/from the EIM. Its behavior is affected by the BCM field in the EIM_WCR and the SWR, SRD, BCD, and BCS fields of the EIM_CSxGCR1.
WEIM_CRE	O	Used as CRE/PS for CellularRam memory. It is used for the Mode Register Set command. This signal can be configured as active low or active high. See CRE and CREP field descriptions of the EIM_CSxGCR1 registers.
WEIM_CS[5:0]	O	Chip Selects. These signals are active-low. Behavior is affected by the RCSA and RCSN fields of the EIM_CSxRCR1 registers and the WCSA and WCSN fields of the EIM_CSxWCR1registers.
WEIM_DTACK_B	I	Data Acknowledge, asynchronous access. This input is used as a data acknowledge signal for single asynchronous accesses.

Table continues on the next page...

Table 25-4. Block I/O Descriptions (continued)

Name	I/O	Description
WEIM_D[31:16]	I/O	MSB Data Bus
WEIM_EB[3:0]	O	Byte Enable. These active-low output signals indicate valid data bytes for the current access. They may be configured to assert for write cycles only. EIM_EB[0] corresponds to DATA_OUT[7:0] EIM_EB[1] corresponds to DATA_OUT[15:8] EIM_EB[2] corresponds to DATA_OUT[23:16] EIM_EB[3] corresponds to DATA_OUT[31:24]. For asynchronous write accesses, behavior is affected by the WBEA and WBEN fields of the EIM_CS1WCR1-EIM_CS5WCR1 Registers. On synchronous or asynchronous read accesses, these signals are always asserted at the start of the access and negated at end of the access.
WEIM_LBA	O	Address Valid. This active-low output signal is asserted during burst mode accesses to cause the external burst capable device to load a new starting burst address. Assertion of LBA indicates that a valid address is present on the address bus. Its behavior is affected by the SWR, SRD, BCD, and BCS fields of the EIM_CSxGCR1 registers, the RADVA and RADVN fields of the EIM_CSxRCR1 registers, and the WADVA and WADVN fields of the EIM_CSxWCR1 registers. In asynchronous mode, LBA length is affected by the RADVA, WADVA, RADVN, and WADVN fields. Minimum length of LBA signal in all modes is one EIM clock cycle.
WEIM_OE	O	Output Enable. This active-low output signal indicates the bus access is a read and enables external devices to drive the data bus with read data. Its behavior is affected by the OEA and OEN bit fields in the Chip Select Configuration Registers.
WEIM_RW	O	Memory Write Enable. This active-low output signal indicates the bus access is a write and enables external devices to sample the data bus. Its behavior is affected by the WEA and WEN bit fields in the Chip Select Configuration Registers.
WEIM_WAIT	I	Ready/Busy/Wait signal. This active-low input signal is asserted by external burst capable devices which support fixed or variable latency of data. It is serviced in synchronous mode only (EIM_CSxGCR1[SWR, SRD] =1). WAIT will have a pull up resistor in I/O. The signal indicates whether the External device is ready for data transaction or not. Busy cycles (or wait cycles) of the external device can occur at the start of a Burst access or at page boundary crossover. **NOTE:** For burst devices, WAIT output should be configured to change one cycle before data is ready (before delay). **NOTE:** Some External devices may not use this input signal for ready state indication (fix latency without WAIT signal monitoring). For these devices EIM should be configured accordingly (see RFL, WFL, and PSZ field descriptions). **NOTE:** This is same as what is shown in IP_IND_WAIT_B

25.4.3 Other Important Block I/O Signals Internal to the SoC

The following table provides a description of other signals which are internal to the i.MX53 that are important to understand the function of EIM.

Name	I/O	Description
WEIM_FB_BCLK	Input	Burst Clock Feedback. This block input is used to sample read data during high transfer speeds. The signal provides feedback from the I/O pad of the BCLK output pin and tends to align more closely with data from the external memory device.
EIM_BOOT[2:0]	Input	EIM Boot Configuration. These block inputs determine the reset state of DSZ[1:0] and MUM. See Table 5-4 for detailed description.
ACLK	Input	AXI clock, maximum frequency 133 Mhz
IPG_CLK_S	Input	EIM module IPG clock
RST_B	Input	Active low HW reset
EIM_WARM_RESET	Input	Warm Reset. If this signal is asserted the rst_b will reset only the internal FF and state machine while S/W registers will keep their current state.This signal is active high signal.

25.5 Chip Select Memory Map

Table 25-6. EIM Chip Select Memory Map

Address	Space Size	Use	Access
EIM_NFC_BASE/ACT_CS/ADDRS0 inputs	0MB-256MB	CS0 memory region	R/W
ACT_CS/ADDRS1 inputs	0MB-256MB	CS1 memory region	R/W
ACT_CS/ADDRS2 inputs	0MB-256MB	CS2 memory region	R/W
ACT_CS/ADDRS3 inputs	0MB-256MB	CS3 memory region	R/W

25.6 Functional Description

This section provides the functional description for the EIM.

25.6.1 Clocks

The EIM has four input clocks.

- ACLK: EIM clock (main clock, AXI clock) with a Max frequency of 133Mhz. Can be gated externally when there is no active AXI access.
- ACLK_SLOW: EIM all time running ACLK. Used for FF that must be active even when EIM is in low power down mode to provide clock for lpack/lpmd registers, IP registers and IP to AXI sync registers.
- IPG_CLK_S: IPG clock for IP accesses. IP registers are activated by ACLK_SLOW clock. BCLK is a clock created from EIM clock for External device usage. Integer division by 1, 2, 3 and 4 of the clock can be use with BCD bit field configuration,

according to external devices demands. EIM clock frequency may be reduced for lower frequency support which cannot be achieved via BCD bit field.
- FB_CLK: BCLK after pads delays.

25.6.2 Bus Sizing Configuration

The EIM supports byte, half word and word operands allowing access to x8, x16, x32 ports. It can be address/data multiplexed in x16, x32 ports. The port size is programmable via the DSZ bit field in the corresponding Chip Select Configuration Register. An 8-bit port can reside in each one of the bytes of the data bus. A 16-bit port can reside on the lower 16 bits of the data bus, DATA_IN/OUT[15:0] or on the higher 16 bits of the data bus, DATA_IN/OUT[31:16] (see Table 25-1).

In the case of a multi-cycle transfer, the lower two address bits (ADDR[1:0]) are incremented appropriately. The EIM address bus is configured according to DSZ bit field and AUS bits. There is either one bit (for x16 port size) or two bits (for x32 port size) right shift of the address bits (only when AUS=0) and no bit shift when AUS = 1 or DSZ[2] = 1.

The EIM has a data multiplexer which takes the four bytes of the AXI data bus and routes them to their required positions to properly interface to memory.

NOTE

A word access to or from a x16 port requires two external bus cycles to complete the transfer.

A word access to or from a x8 port requires four external bus cycles to complete the transfer.

25.6.2.1 8 BIT PORT SUPPORT

EIM has limited support for mot68000 & intel 386 protocols.

25.6.2.1.1 MOTOROLA 68000

EIM has limited support for mot68000 protocol. Only basic read or write asynchronous operations are supported.

The following operations are not supported:

- Read modify write
- Sync access

- All special accesses (ARM platform space, bus arbitration, bus control, bus error & reset operations)
- FC outputs

25.6.2.1.2 INTEL 386

EIM has limited support for intel 386 protocol. Only basic read or write async non-pipelined operations are supported.

The following operations are not supported:

- Other bus cycles (interrupt, halt & refresh)
- Bus lock
- M/IO, DC, LBA, NA, REFRESH & BS8 signals

25.6.3 EIM Operational Modes

EIM has the following main operational modes selected by control bit fields settings. For details, see the bit field descriptions of SWR / SRD / MUM. All modes are supported in with 8-, 16- or 32-bit port configuration, according to DSZ bit field.

Table 25-7. EIM Operation Modes Field Settings

Control bit fields			Brief mode description
MUM	SRD	SWR	
0	0	0	Asynchronous write / Asynchronous read for APR=0 / Asynchronous page read for APR=1, none multiplexed
		1	Synchronous write/ Asynchronous read or APR=0 / Asynchronous page read for APR=1,none multiplexed
	1	0	Asynchronous write/Synchronous read none multiplexed
		1	Synchronous write/read none multiplexed
1	0	0	Asynchronous write/read multiplexed
		1	Synchronous write/ Asynchronous read multiplexed
	1	0	Asynchronous write/Synchronous read multiplexed
		1	Synchronous write/read multiplexed

25.6.4 Burst Mode (Synchronous) Memory Operation

This mode is enabled for read or write access. Bit SWR sets the burst mode for write operations at the corresponding chip select and bit SRD sets it for read operation.

When this mode is set, the controller attempts to translate the Master burst accesses to memory burst accesses, being limited by the memory burst length, predefined by BL value, or memory and Master WRAP/INCR boundary crossing non-matching. Only the first address accessed is displayed by the controller on the external address bus in a memory burst sequence.

EIM may translate from some Master sequential accesses to one or few memory bursts, but not from two Master individual accesses to one memory burst.

For the first access in a memory burst sequence, the EIM asserts $\overline{\text{ADV}}$, causing the external burst device to latch the starting burst address; then toggle the burst clock (BCLK) a predefined number of cycles in order to latch the first unit of data. Subsequent accessed data units can then be burst in fewer clock cycles, realizing an overall increase in bus bandwidth.

NOTE

The BCLK signal toggles only when burst access is executed toward the external device (BCM=1'b0 for normal mode use). It runs with a 50% duty cycle until the end of access is reached. When access is terminated, BCLK stops toggling.

Memory burst accesses are terminated by the EIM whenever it detects the following:

- The specific burst length has executed completely (end of access)
- Write access - missing data in write buffer (Master is delaying the data transfer toward the EIM)
- Next sequential access crosses boundary with unequal condition (wrap/increment, burst length) on the Master and memory
- Current memory burst length reached

25.6.5 Burst Clock Divisor (BCD)

In some cases, it may be necessary to slow the external bus in relation to the internal bus to allow accesses to burst devices that have a maximum operating frequency less than the operating frequency of the internal bus.

The internal bus frequency can be divided by one, two, three or four for presentation on the external bus in burst mode operation.

To achieve frequency of BCLK other then an integer division, the cycle time of the EIM clock entering the controller must be changed accordingly.

By programming the BCD bit field to various values, two signals on the external bus are affected; $\overline{ADV}$ and BCLK. The $\overline{ADV}$ signal is asserted according to RADVA or WADVA bit fields programming, and is negated according to the formula mentioned in RADVN and WADVN bit fields description. The BCLK signal runs with a 50% duty cycle until the end of access is reached.

If BCM = 1, the BCLK runs at frequency according to GBCD bit field settings on every async memory access, regardless of the SWR and SRD bits configuration. Caution should be exercised when using BCM bit; GBCD bit field should be updated once and should not change when BCLK is toggling. The BCM bit is used mainly for system debug mode. It has no functional use of the EIM in normal mode.

25.6.6 Burst Clock Start (BCS)

In an effort to allow greater flexibility in achieving the minimum number of wait states on burst accesses, you can determine when you want the BCLK to start toggling after the start of access. This allows the BCLK to be skewed from point of data capture on the EIM clock by any number of EIM clock cycles.

Care must be exercised when setting BCS bit field in conjunction with the BCD and RWSC/WWSC bit fields. See the external timing diagrams in "Burst Memory Accesses Timing Diagrams" for examples of how to use the BCS, BCD and RWSC/WWSC bit fields together.

25.6.7 Multiplexed Address/Data Mode Support

The control bit MUM allows support memory with multiplexed address/data bus both in asynchronous and in synchronous modes.

Caution should be exercised for using OEA/WEA & ADH bit fields. They should be configured according to the external device requirements, as it determines the time point of end of address phase and start of data phase.

25.6.8 Mixed Master/Memory Burst Modes Support

To provide mixed sequential/wrap accesses with different length, EIM interprets burst signal and generate additional $\overline{ADV}$ signals whenever there appear unequal address or burst boundary crossing condition.

BL bit field is used to notify EIM about current memory burst and wrap condition for properly external address generation. In case of non-matching boundaries in both the memory and Master access, EIM starts a new memory burst access by updating address from Master on address bus and generating $\overline{\text{ADV}}$ signal.

25.6.9 AXI (Master) Bus Cycles Support

The EIM uses an ARM AXI slave interface. It has a 32-bit bus and supports one access (one ID) at a time. No out of order or parallel accesses are supported.

The following AXI protocol signals are not supported:

- AWLOCK
- AWCACHE
- ARLOCK
- ARCACHE

ARID bus is sampled when:

- new read access is valid on the read address channel and is reflected on the RID bus output toward the master.

AWID bus is sampled when:

- new write access is valid on the write address channel and is reflected on the WID/ BID bus output toward the master.

ARPROT and AWPROT signal are partially used. ARPROT[0] and AWPROT[0] bits are used for normal/privileged access detection. ARPROT[2:1] and AWPROT[2:1] are not used.

When sampling a valid access on both of the address channels, the read access will be performed first while write access is pending. After last data transfer completed, the pending write will be executed.

A new access may be executed one cycle after sampling a valid access on the read or write address channels. Assuming there is no current access (back to back) which can cause a recovery or end of access penalty cycles; for write access, data should be present at write buffer for fast execution.

NOTE

- Only 32-bit word size accesses are supported for burst mode accesses.
- Only 8-bit (1 byte), 16-bit (2 byte) and 32-bit (4 byte) word size supported for single access.

- Maximum number of burst length is 16.
- According to AXI protocol, burst access should not cross 4 KB blocks. In case EIM gets an access that crosses the 4 KB, memory address calculation is invalid.

AXI transfers shown in the table below are also supported. These AXI cycles will be translated into the necessary cycles on the memory side. For example, for optimal operation in case ARM cache is configured to 8 beat burst with wrap, a synchronous flash and cellular RAM memory should be configured in 16 word wrap burst mode when using a 16-bit data port, and in 8 word wrap burst mode when using a 32-bit data port. EIM uses BL bit field to support different memory configurations. The controller splits the transaction when needed in some cases. See Table 25-9.

Table 25-8. AXI Burst Cycles Supported

Burst Length - Number of data transfers	Burst size - Bytes in transfer	Burst type	Description
1	1	INCR	Single transfer
1	2	INCR	Single transfer
1	4	INCR	Single transfer
2	4	WRAP	2-beat wrapping burst
4	4	WRAP	4-beat wrapping burst
8	4	WRAP	8-beat wrapping burst
16	4	WRAP	16-beat wrapping burst
2	4	INCR	2-beat incrementing burst
3	4	INCR	3-beat incrementing burst
4	4	INCR	4-beat incrementing burst
5	4	INCR	5-beat incrementing burst
6	4	INCR	6-beat incrementing burst
7	4	INCR	7-beat incrementing burst
8	4	INCR	8-beat incrementing burst
9	4	INCR	9-beat incrementing burst
10	4	INCR	10-beat incrementing burst
11	4	INCR	11-beat incrementing burst
12	4	INCR	12-beat incrementing burst
13	4	INCR	13-beat incrementing burst
14	4	INCR	14-beat incrementing burst
15	4	INCR	15-beat incrementing burst
16	4	INCR	16-beat incrementing burst

Table 25-9. AXI to Memory Burst Splits Number

AXI Burst Type	Memory Burst Type Config.	# of accesses to X8 Memory Port size	# of accesses to X16 Memory Port size	# of accesses to X32 Memory Port size
INC16 Aligned Addr.	WRAP4	16	8	4
	Cont.	1	1	1
INC16 Unaligned Addr.	WRAP4	17	9	5
	Cont.	1	1	1
WRAP16 Aligned Addr.	WRAP16	4	2	1
	Cont.	1	1	1
WRAP16 Unaligned Addr.	WRAP16	5	3	1
	Cont.	2	2	2
INC8 Aligned Addr.	WRAP8	4	2	1
	WRAP16	2	1	1
INC8 Unaligned Addr.	WRAP8	4 or 5	2 or 3	2
	WRAP16	2 or 3	2	1 or 2
WRAP8 Aligned Addr.	WRAP16	2	1	1
	Cont.	1	1	1
WRAP8 Unaligned Addr.	WRAP16	2 or 3	1	2
	Cont.	2	2	2

25.6.10 WAIT_B Signal, RWSC and WWSC bit fields Usage

Most of the external devices supporting burst mode for write or read accesses providing signal which indicates for data valid on the memory bus (a.k.a. handshake mode). For this mode, RFL and WFL bits should be cleared and RWSC/ WWSC bit fields indicate when the controller should start sampling this signal from the external device or, in other words, how many BCLK cycles should be masked.

For devices which are not using this signal or have a fixed latency ability, the RFL and WFL bits may be set for internal calculation regarding BCLK cycles penalty until data is valid (memory initial access time). For this mode, RWSC/ WWSC indicates when the data is ready for sampling by the controller (read access) or the external device (write access). There is separation between read and write accesses wait-state control. For read access, RWSC bit field is valid and WWSC bit field is ignored; for write access, WWSC is valid and RWSC is ignored.

25.6.11 IPS Register Interface

Access to the registers of the EIM, read or write, is made with IPS protocol signals. The system should avoid changing the registers while master/memory transaction is valid, as this can cause an unknown behavior of the controller.

Register access size is 32-bit as the register size definition, other size of access (byte or half word) is not supported.

25.6.12 MRS Set for PSRAM

Memory registers of PSRAM devices can be configured according to external signal, which indicates whether the access is to a memory array or memory register domain.

When the CRE bit is set, the following transactions to the external device will assert the CRE signal. The polarity of this signal is determined by the CREP bit for active low or active high assertion of the signal.

25.6.13 EIM Access Termination

EIM is monitoring the corresponding CSx control signal every time variable latency access or dtack access is performed toward the external device.

In variable latency accesses, the Watchdog Timer (WDOG-1) counts BCLK cycles. If it reaches the wdog_limit (according to the WDOG_LIMIT bit field in the WCR) before the device signals can drive/sample new data, the controller will terminate the access and generate an error response transfer toward the Master.

In dtack access, WDOG-1 counts ACLK cycles instead of BCLK and it reaches the wdog_limit before the device asserts the dtack signal, the controller will terminate the access and generate an error response transfer toward the Master.

WDOG-1 can by disabled by WDOG_EN bit in the WCR.

25.6.14 Error Conditions

The following conditions cause an error (AXI error or IPS error) response signal:

- AXI errors
  - Access to a disabled chip select - access to a mapped chip select address space where the CSEN bit in the corresponding chip select Configuration Register is clear

- Access to a non mapped address - access to an address that is not mapped to any CS.
- User access to a supervisor-protected chip select address space (the SP bit in the corresponding chip select Configuration Register is set)
- User access in fixed mode access
- User preform write access to write protected chip select
- First write data ID and write address ID do not match. (No data is written to the memory.)
- First Write Data ID and write address ID match but one or more of the other Write data IDs does not match the First Write data ID (data is written to memory according)
- Access duration to external device from CSx signal assertion is 128/256/512/1024 cycles (access is terminated by the controller) - This error can be disabled be software.

- IPS errors
  - User read or write access to a reserved/non-valid address in the EIM Configuration Register

25.6.15 DTACK Mode

In DTACK mode, the EIM uses DTACK signal as an indication of when to end the access.

DTACK is an asynchronous edge/level sensitive signal. DTACK polarity is configurable by the DAP bit in CsxGCR2 (default value is 0).

In this case, EIM begins the access and after a few cycles (according DAPS field) and waits until DTACK (after synchronization) becomes asserted, then samples the data in read access and completes the current data access (see Figure 25-13, Figure 25-14 & Figure 25-15).

If more than one data is needed, CS will be negated between access (csrec field is not zero) and the AXI burst access will be split into single accesses (see Figure 25-17).

25.6.16 RDY_INT Signal as Interrupt

The EIM has an external interrupt support. When INTEN bit in the WCR is set, signal RDY_INT is used as interrupt; its status is being reflected by INT bit and output signal.

It is cleared by writing one to the INT bit. When INTEN is cleared, the interrupt is disabled. This interrupt is a level interrupt and its polarity can be configured by the INTPOL bit in the WCR.

25.6.17 RDY_INT Signal as Ready After Reset Indication

This feature is used for boot propose from external devices based on NANDFlash array memory with NORFlash interface.

When ERRST bit is set, RDY_INT signal is monitored to determine ready after reset of the external device located on CS0.

The monitoring is taking place when CS0 is accessed for the first time. The access will be pending until assertion of the signal is detected. When detection occurs, ERRST bit is self-cleared and pending access is executed to the external device on CS0.

25.6.18 EIM_GRANT / EIM_BUSY Handshake Description

Prior to executing command to one of the external device (chip select), EIM assert EIM_BUSY signal (1'b1) and checks the EIM_GRANT signal status.

If EIM_GRANT signal is high, it indicates external data bus is not used by other slaves (NAND Flash Controller) and EIM may start to execute the access. If EIM_GRANT is low, EIM waits until it is set (1'b1) before executing the access.

EIM keeps EIM_BUSY signal set until it completes the access toward the external device.

Once EIM_GRANT signal is set, it can not be reset until EIM_BUSY signal is cleared by EIM.

NOTE

In 16-bit Muxed EIM doesn't use the data bus, therefore there is no sharing of the data bus with NFC. EIM doesn't wait for EIM_GRANT signal from NFC and doesn't assert the EIM_BUSY signal.

25.6.19 LPMD / LPACK Handshake Description

These signals are used for frequency and/or voltage change, and for entering low power mode during normal operation of the EIM. Before any change can take place, the controller and all the relevant external devices should be in idle state, which means no access or data transfer is in process.

LPMD input signal is asserted once EIM detects the assertion of LPMD, all ready signals of the AXI channels are negated, and EIM is not sampling new accesses. It finishes all the ongoing accesses and already pending ones. When EIM is in idle state, the LPACK output signal is asserted. EIM will stay in idle state and the LPACK signal will stay asserted until the LPMD signal is negated.

25.6.20 Endianness

Big and Little endianness are supported by the controller according to the following table.

Table 25-10. EIM Out/in Data in Case AXI Out/in Data is 0xB3B2B1B0

Endian mode	AXI access	AXI address [1:0]	Port size and used bits								
			Word port				Half word port			Byte port	
			[31:24]	[23:16]	[15:8]	[7:0]	External address [0]	[31:24] ([15:8])	[23:16] ([7:0])	External address [1:0]	[31:24] ([23:16]) ([15:8]) ([7:0])
Big	Word	0	0xB3	0xB2	0xB1	0xB0	0	0xB3	0xB2	0	0xB3
										1	0xB2
							1	0xB1	0xB0	2	0xB1
										3	0xB0
	Half Word	0			0xB1	0xB0	0	0xB3	0xB2	0	0xB3
										1	0xB2
		2	0xB3	0xB2			1	0xB1	0xB0	2	0xB1
										3	0xB0
	Byte	0				0xB0	0		0xB3	0	0xB3
		1			0xB1			0xB2		1	0xB2
		2		0xB2			1		0xB1	2	0xB1
		3	0xB3					0xB0		3	0xB0

Table continues on the next page...

Table 25-10. EIM Out/in Data in Case AXI Out/in Data is 0xB3B2B1B0 (continued)

Endian mode	AXI access	AXI address [1:0]	Port size and used bits								
			Word port				Half word port			Byte port	
			[31:24]	[23:16]	[15:8]	[7:0]	External address [0]	[31:24] ([15:8])	[23:16] ([7:0])	External address [1:0]	[31:24] ([23:16]) ([15:8]) ([7:0])
Little	Word	0	0xB3	0xB2	0xB1	0xB0	0	0xB1	0xB0	0	0xB0
										1	0xB1
							1	0xB3	0xB2	2	0xB2
										3	0xB3
	Half Word	0			0xB1	0xB0	0	0xB1	0xB0	0	0xB0
										1	0xB1
		2	0xB3	0xB2			1	0xB3	0xB2	2	0xB2
										3	0xB3
	Byte	0				0xB0	0		0xB0	0	0xB0
		1			0xB1			0xB1		1	0xB1
		2		0xB2			1		0xB2	2	0xB2
		3	0xB3					0xB3		3	0xB3

25.6.21 Strobe Signal Use

The strobe signal is toggling according to address/data valid condition on the external bus for read and write accesses, and for both synchronous and asynchronous modes.

At any time point when address/data is valid on the external bus, the strobe signal will generate a positive edge, which can be used to sample the external data and control signal.

NOTE

Strobe signal for read data is active (RL + 1) cycles after data on external bus is valid.

25.7 Initialization Information

25.7.1 Booting from EIM

EIM is ready to work with CS0 after the hardware reset, but it has been configured for very slow access (for boot purposes), with additional setup and hold time.

Other CSs are disabled by hardware reset. Therefore, all CSs must be properly initialized before use in writing values to the corresponding chip select configuration registers.

DSZ[1:0] and MUM fields are set according to EIM_BOOT [2:0] block inputs.

25.8 Application Note

Application note uses next functions to illustrate EIM and memory accesses:

- WR16(address, data) is a 16 bit write access
- WR32(address, data) is a 32 bit write access
- RD16(address, data) is a 16 bit read access
- RD32(address, data) is a 32 bit read access
- WR_I(address, data, delta, counter) is a write data sequence, there data(i+1) = data(i) + delta
- COMMAND_SEQUENCE as described on page 1-53
- CHECK_STATUS as described on page 1-53

All addresses are byte address. 'CS0 is a Chip Select 0 base address. 'EIM_ is a prefix of EIM's registers. 'h is a prefix of hexadecimal constant. // is a comment beginning. csba[cs] is a dimension of CS base addresses. "addr" means an address offset in current CS address space. Examples use CS0 address space, but it may be any except boot mode functionality.

Configuration examples were checked with some below defined memory models and may require some adjustment for other family members.

25.8.1 Access to AMD Flash

Below mentioned configurations intended for AMD Simultaneous Flash Memory w/ VersatileIO™ Control.

25.8.1.1 AMD Flash Asynchronous Mode Configuration

Next EIM register configuration used for AMD flash asynchronous access:

- WR32('EIM_CS0RCR1,'h0a018000);

- WR32('EIM_CS0WCR1,'h0704a240);
- WR32('EIM_CS0GCR1,'h00430081); // for 32 bit memory or
- WR32('EIM_CS0GCR1,'h00410081); // for 16 bit memory

25.8.1.2 AMD Flash Utility

```
       // Single data word programming to addr, 32-bit memory
           port_size = 32;
           WR32('CS0+('h0555<<2),'h00aa); // command sequence
           WR32('CS0+('h02aa<<2),'h0055);
           WR32('CS0+('h0555<<2),'ha0); // command code
           WR32('CS0+addr, data); // addr & data
// Polling end of programming
          edata = data; // expected data
          data = ~edata;
          errst = 0;
      while(!(data == edata) &&!errst)
             begin: BR_EN
             RD16('CS0+addr, data);          // Read status
             if(data[7]!== edata[7])
                begin
                if(data[5] == 1)
                    begin
                    RD16('CS0+addr, data3);
                    RD16('CS0+addr, data);
                    if(data[6] !== data3[6])
                        begin
                        $display("CHECK_STATUS: Error timout on single data program");
                        errst = 1;
                        disable BR_EN;
                        end
                    end
               end
             else
                begin
                RD16('CS0+addr, data3);
                if(port_size == 32)
                    RD32('CS0+addr, data);
                else
                    begin
                    RD16('CS0+addr, data);
                    edata[31:16] = 16'h0;
                    end
                if(data !== edata)
                    begin
                    $display("CHECK_STATUS: Error in data write on single data program");
                      errst =2;
                    disable BR_EN;
                    end
                end
             end
      // Single data word programming to addr, 16-bit memory
          port_size = 16;
          WR32('CS0+('h0555<<1),'h00aa); // command sequence
          WR32('CS0+('h02aa<<1),'h0055);
          WR32('CS0+('h0555<<1),'ha0); // commund code
          WR32('CS0+addr, data); // addr & data
      // Polling end of programming - the same as for 32 bit
```

25.8.2 Access to Intel Sibley Flash

Below mentioned configurations intended to Sibley family muxed and non-muxed devices.

25.8.2.1 Intel Sibley Flash Asynchronous Mode Configuration

- WR32('EIM_CS0GCR1,'h00210081);
- WR32('EIM_CS0RCR1,'h0e020000);
- WR32('EIM_CS0RCR2,'h00000000);
- WR32('EIM_CS0WCR1,'h0704a040);

25.8.2.2 Intel Sibley Flash Synchronous Mode Configuration

Next configuration used for 133 MHz synchronous access to flash:

```
// Set memory to synchronous read mode
     WR16('CS0+('h5903<<1),'h0060);
     WR16('CS0+('h5903<<1),'h0003);
     WR16('CS0+('h0000<<1),'h00ff);
// Set EIM configuration to synchronous timing
     WR32('EIM_CS0GCR1,'h50214225);                    // 133 MHz
     WR32('EIM_CS0RCR1,'h0c000000);                    // 12 cycles on memory
```

Next configuration used for 66 MHz synchronous access to muxed flash:

```
// Set memory to synchronous read mode
     WR16('CS0+('h3103<<1),'h0060);
     WR16('CS0+('h3103<<1),'h0003);
     WR16('CS0+('h0000<<1),'h00ff);
//-------------------------------------------------------------------------
// Set EIM configuration to synchronous timing
     WR32('EIM_CS0GCR1,'h5021122d);                    // 66 MHz
     WR32('EIM_CS0RCR1,'h07000000);                    // 7cycles on memory
```

25.8.2.3 Intel Sibley Flash Utility

```
// Single data word programming to addr
     WR16('CS0+addr,'h0060);                 // Unlock
     WR16('CS0+addr,'h00d0);
     WR16('CS0+addr, 'h0041);
     WR16('CS0+addr, data);
     WR16('CS0+caddr, 'h0070);                 // Read Status command
     while('CS0+data[7] == 0)                             // Wait / Polling
         RD16('CS0+addr, data);              // Read status
     RD16('CS0+addr, data);                  // Read status
     WR16('CS0+'h0000,'h00ff);
// Write buffer programming
     WR16('CS0+addr,'h0060);                 // Unlock
     WR16('CS0+addr,'h00d0);
     data    = 0;
     WR16('CS0+addr, 'h0070);                // Read Status command
     while(data[7] == 0)                           // Wait
         RD16('CS0+addr, data);              // Read status
```

```
WR16('CS0+'h0000,'h00ff);
           WR16('CS0+addr,'h00e9);           // Write Buffer command
           WR16('CS0+addr,255);              // Word counter (<256)
           for(i=0; i<'h200; i = i + 'h40)
              WR_I('CS0+addr+i, data+((i>2)*'h0010_0001), 'h0010_0001, 16); // Data
           WR16('CS0+addr,'h00d0);           // Write Confirm command
           data  = 0;
           while(data[7] == 0)                     // Wait
              RD16('CS0+addr, data);         // Read status
           RD16('CS0+addr, data);            // Read status
           WR16('CS0+'h0000,'h00ff);
```

25.8.3 Access to MDOC Device

Below mentioned configurations intended to MDOC H3 device.

25.8.3.1 MDOC Device Boot

To boot from the MDOC device the ERRST bit should be configured to 1, so that EIM will hold the first read access to CS0 until the MDOC asserts the RDY signal.

25.8.3.2 MDOC Device Asynchronous Mode Configuration

```
// Non-muxed mode
WR32('EIM_CS0GCR1,'h00410081);
WR32('EIM_CS0RCR1,'h0e121010);
WR32('EIM_CS0RCR2,'h00000000);
WR32('EIM_CS0WCR1,'h12092492);
// Muxed mode
WR32('EIM_CS0GCR1,'h00410081);
WR32('EIM_CS0RCR1,'h0e121010);
WR32('EIM_CS0RCR2,'h00000000);
WR32('EIM_CS0WCR1,'h12092492);
```

25.8.3.3 MDOC Device Utility

```
// Read Manufacturer ID and Device ID
    RE16('CS0+'h9400,'h4833);
    RE16('CS0+'h9422,'hb7cc);
```

25.8.4 Access to Micron PSRAM

Below mentioned configurations intended to mt45w4mw16bfb_706.

25.8.4.1 Micron PSRAM Asynchronous Mode Configuration

```
// 16 bit memory
    WR32('EIM_CS0GCR1,'h403104b1);
    WR32('EIM_CS0RCR1,'h0b010000);
    WR32('EIM_CS0RCR2,'h00000008);
    WR32('EIM_CS0WCR1,'h0b040040);
// 32 bit memory
    WR32('EIM_CS0GCR1,'h403304b1);
    WR32('EIM_CS0RCR1,'h0f010000);
    WR32('EIM_CS0RCR2,'h00000008);
    WR32('EIM_CS0WCR1,'h0f040040);
```

25.8.4.2 Micron PSRAM Synchronous Mode Configuration

```
// 16 bit memory
    WR32('EIM_CS0GCR1,'h403104b1);
    WR32('EIM_CS0WCR1,'h0b040000);
    WR16('CS0+('h85947<<1),'h0040); // memory configuration
    WR32('EIM_CS0GCR1,'h4021_5487);// fixed latency memory wrap 4
    WR32('EIM_CS0RCR1,'h04000000);
    WR32('EIM_CS0RCR2,'h00000008);
    WR32('EIM_CS0WCR1,'h04000000);
// 32 bit memory
    WR32('EIM_CS0GCR1,'h6003_04f1);
    WR32('EIM_CS0WCR1,'h0b04_0000);
    WR32('CS0+('h85947<<2),'h0040); // memory configuration
    WR32('EIM_CS0GCR1,'h4003_1487); // var latency memory inc. page size 128
    WR32('EIM_CS0RCR1,'h04000000);
    WR32('EIM_CS0RCR2,'h00000008);
    WR32('EIM_CS0WCR1,'h04000000);
```

25.8.5 Access to Samsung OneNAND

Mentioned below are the configurations intended for Samsung OneNAND muxed and non-muxed devices.

25.8.5.1 Samsung OneNAND Boot

There are two ways to boot from Samsung OneNAND. In the first way, the ERRST bit is set to 0 and the user has to poll the interrupt status in the OneNAND interrupt register (or set interrupt handler there). In the second way, the ERRST bit is set to 1 and the user should enable the device interrupt output before the first read from CS0 access is issued.

Load sectors 2,3 to DataRAM, page 0 done in the next example:

- WR16('CS0+('hF241<<1),'h0); // Clear interrupt status
- WR16('CS0+('hF100<<1),'h0); // block[8:0] address
- WR16('CS0+('hF107<<1),'h2); // sector[1:0] and page[7:2] addresses
- WR16('CS0+('hF200<<1),'h802); // buffer[11:8] address and counter[1:0]

- WR16('CS0+('hF101<<1),'h0); // DDP choose
- WR16('CS0+('hF220<<1),'h0); // Set command

25.8.5.2 Samsung OneNAND Asynchronous Mode Configuration

```
// Non-muxed memory
    WR32('EIM_CS0GCR1,'h00410081);
    WR32('EIM_CS0RCR1,'h0b010000);
    WR32('EIM_CS0RCR2,'h00000000);
    WR32('EIM_CS0WCR1,'h0c092480);
// Muxed memory
    WR32('EIM_CS0GCR1,'h00410089);
    WR32('EIM_CS0RCR1,'h0b010000);
    WR32('EIM_CS0RCR2,'h00000000);
    WR32('EIM_CS0WCR1,'h0c092480);
```

25.8.5.3 Samsung OneNAND Synchronous Mode Configuration

Set memory and EIM to synchronous read mode is shown in the next example:

```
    WR16('CS0+('hF221<<1),'hc0e0); // Synchronous read, 4 clk latency
    WR32('EIM_CS0GCR1,'h50412405); // 44 MHz (non-muxed)
    WR32('EIM_CS0RCR1,'h05010000);
```

The muxed Samsung OneNAND supports synchronous write, too:

```
// Set memory & EIM to synchronous read and write mode
    WR16('CS0+('hF221<<1),'hc0f2); // Sync. read and write, 4 clk latency
    WR32('EIM_CS0GCR1,'h5041240f); // 44 MHz
    WR32('EIM_CS0RCR1,'h05010000);
    WR32('EIM_CS0WCR1,'h05040000);
```

25.8.5.4 Samsung OneNAND Utility

The following utility algorithms are used on the Samsung OneNAND:

```
// Unlock Block command
    WR16('CS0+('hF100<<1),'h0);         // DFS
    WR16('CS0+('hF100<<1),'h0);         // DBS
    WR16('CS0+('hF24c<<1),'h2);         // SBA - block number (2)
    WR16('CS0+('hF241<<1),'h0);         // Clear interrupt status
    WR16('CS0+('hF220<<1),'h23);        // Unlock command
    data ='h0;
    while(!(data &'h0004))              // Polling
       RD32('WIAR, data); // Read status
// Erase block command
    WR16('CS0+('hF100<<1),'h2);         // DFS and block ([8:0]) address
    WR16('CS0+('hF101<<1),'h0);         // DBS
    WR16('CS0+('hF241<<1),'h0);         // Clear interrupt status
              WR16('CS0+('hF220<<1),'h94);             // Erase command
    data ='h0;
    while(!(data &'h0004))                    // Wait
        RD32('WIAR, data); // Read status
// Program page command
    WR16('CS0+('hF100<<1),'h2);               // DFS and block[8:0] address
    WR16('CS0+('hF107<<1),'h0);               // sector[1:0] and pag
```

```
WR16('CS0+('hF200<<1),'h800);           // buffer[11:8] address and counter[1:0]
WR16('CS0+('hF241<<1),'h0);             // Clear interrupt status
WR16('CS0+('hF220<<1),'h80);            // Program command
data ='h0;
while(!(data &'h0004))                  // Wait
    RD32('WIAR, data); // Read status
```

25.8.6 Access to Samsung UtRAM

Below mentioned configurations are intended for Samsung UtRAM.

25.8.6.1 Samsung UtRAM Asynchronous Mode Configuration

```
WR32('EIM_CS0GCR1,'h400104b1);
WR32('EIM_CS0RCR1,'h0a010000);
WR32('EIM_CS0RCR2,'h00000008);
WR32('EIM_CS0WCR1,'h0b040040);
```

25.8.6.2 Samsung UtRAM Synchronous Mode Configuration

```
RD16('CS0+('hff_ffff<<1),data);             // command sequence
RD16('CS0+('hff_ffff<<1),data);
RD16('CS0+('hff_ffff<<1),data);
RD16('CS0+('hff_feff<<1),data);
RD16('CS0+('h00_82a0<<1),data);             // memory sync. configuration
WR32('EIM_CS0GCR1,'h4021_53b7);            // fixed latency memory wrap 32
WR32('EIM_CS0RCR1,'h0500_0000);
WR32('EIM_CS0WCR1,'h0300_0000);
```

25.8.7 Access to Spansion Flash

Below mentioned configurations are intended for Spansion Flash.

25.8.7.1 Spansion Flash Asynchronous Mode Configuration

```
WR32('EIM_CS0GCR1,'h00410081);
WR32('EIM_CS0RCR1,'h0a018000);
WR32('EIM_CS0RCR2,'h00000000);
WR32('EIM_CS0WCR1,'h0704a240);
WR16('CS0+('hF220<<1),'h94);            // Erase command
data ='h0;
while(!(data &'h0004))                  // Wait
    RD32('WIAR, data); // Read status
// Program page command
WR16('CS0+('hF100<<1),'h2);             // DFS and block[8:0] address
WR16('CS0+('hF107<<1),'h0);             // sector[1:0] and page[7:2] addresses
WR16('CS0+('hF200<<1),'h800);           // buffer[11:8] address and counter[1:0]
WR16('CS0+('hF241<<1),'h0);             // Clear interrupt status
WR16('CS0+('hF220<<1),'h80);            // Program command
data ='h0;
```

```
        while(!(data &'h0004))                  // Wait
            RD32('WIAR, data); // Read status
```

25.8.7.2 Spansion Flash Synchronous Mode Configuration

```
        WR16('CS0+('h0555<<1),'h00aa);   // command sequence
        WR16('CS0+('h02aa<<1),'h0055);
        WR16('CS0+('h0555<<1),'hd0);
        WR16('CS0+('h0000<<1),'h1ec4);   // memory sync. configuration
        WR32('EIM_CS0GCR1,'h50411325); // 66 MHz
        WR32('EIM_CS0RCR1,'h05000000); // 5 cycles on memory
```

25.8.7.3 Spansion Flash Utility

```
    // Single word programming
        COMMAND_SEQUENCE(cs,16,'ha0);           // single word programming
        WR16('CS0+addr, data);
        CHECK_STATUS('CS0+addr,data,16,1,errst);
    // Write buffer programming
        COMMAND_SEQUENCE(0,16,'h25);              // write buffer programming
        WR16('CS0+addr,'h001f);                     // counter-1
        WR_I('CS0+addr, data, 'h0010_0001, 16); // data
        WR16('CS0+addr,'h0029);                     // write buffer to flash
        CHECK_STATUS('CS0+addr+'h3e,data[31:16]+'h00f0,16,1,errst);
```

There COMMAND_SEQUENCE and CHECK_STATUS are next functions:

```
    task COMMAND_SEQUENCE;
        input[2:0]    cs;
        input[7:0]    port_size;
        input[31:0]   code;
    begin
        if(port_size == 16)
           begin
           WR16(csba[cs]+('h0555<<1),'h00aa);
           WR16(csba[cs]+('h02aa<<1),'h0055);
           WR16(csba[cs]+('h0555<<1),code);
           end
        else
           begin
           WR32(csba[cs]+('h0555<<2),'h00aa);
           WR32(csba[cs]+('h02aa<<2),'h0055);
           WR32(csba[cs]+('h0555<<2),code);
           end
    end
    endtask
    task     CHECK_STATUS;
        input[31:0] addr;
        input[31:0] edata;
        input[7:0]   port_size;
        input[7:0]   opcode;
        output[7:0] errst;
        reg[31:0]    data;
        reg[31:0]    data3;
    begin
        errst = 0;
        data  = 0;
        data3 = 0;
while(!(data == edata) && !errst) // Wait operation
      begin: BR_EN
      RD16(addr, data);             // Read status
      if(data[7] !== edata[7])
```

```
            begin
            if(data[5] == 1)
                 begin
                 RD16(addr, data3);
                 RD16(addr, data);
                 if(data[6] !== data3[6])
                      begin
                      $display("CHECK_STATUS: Error timout on single data program");
                      errst = 1;
                      disable BR_EN;
                      end
                 end
            else
                 begin
                 if(opcode == 2)
                      if(data[1] == 1)
                          begin
                          RD16(addr, data3);
                          if(port_size == 32)
                             RD32(addr, data);
                          else
                             RD16(addr, data);
                          if(data[1] == 1 && data !== edata)
                             begin
                             $display("CHECK_STATUS: Error on write buffer");
                             errst =3;
                             disable BR_EN;
                             end
                          end
                 end
            end
        else
            begin
            RD16(addr, data3);
            if(port_size == 32)
                 RD32(addr, data);
            else
                 begin
                 RD16(addr, data);
                 edata[31:16] = 16'h0;
                 end
            if(data !== edata)
                 begin
                 $display("CHECK_STATUS: Error in data write on single data program");
                 errst =2;
                 disable BR_EN;
                 end
            end
        end
end
endtask
```

25.8.8 8 bit support

This section details the pin connections for Intel mode and Motorola mode.

Intel Mode - For intel mode use the following connection:

Table 25-11. Intel Mode pin connections

ARM platform Pin	EIM Pin	Notes
ADS#	IPP_DO_ADV_B	WAL = 1,RAL = 1

Table continues on the next page...

Table 25-11. Intel Mode pin connections (continued)

ARM platform Pin	EIM Pin	Notes
W/R	IPP_DO_BE_B	WBED = 1
WR#	WE#	
RD#	OE#	

Mot. Mode - For intel mode use the following connection:

Table 25-12. Motorola Mode pin connections

ARM platform Pin	EIM Pin	Notes
AS#	IPP_DO_CS_B	
R/W#	WE#	
LDS#	BE#	

25.9 Booting from OneNAND and NOR Flash devices

External Bus Timing Diagrams

The following timing diagrams show the timing of accesses to memory or a peripheral with different timing parameters. All examples done for CS0, but are valid for any others chip select. BE means one from current used BE[3:0].

25.9.1 Asynchronous Read Memory Accesses Timing Diagram

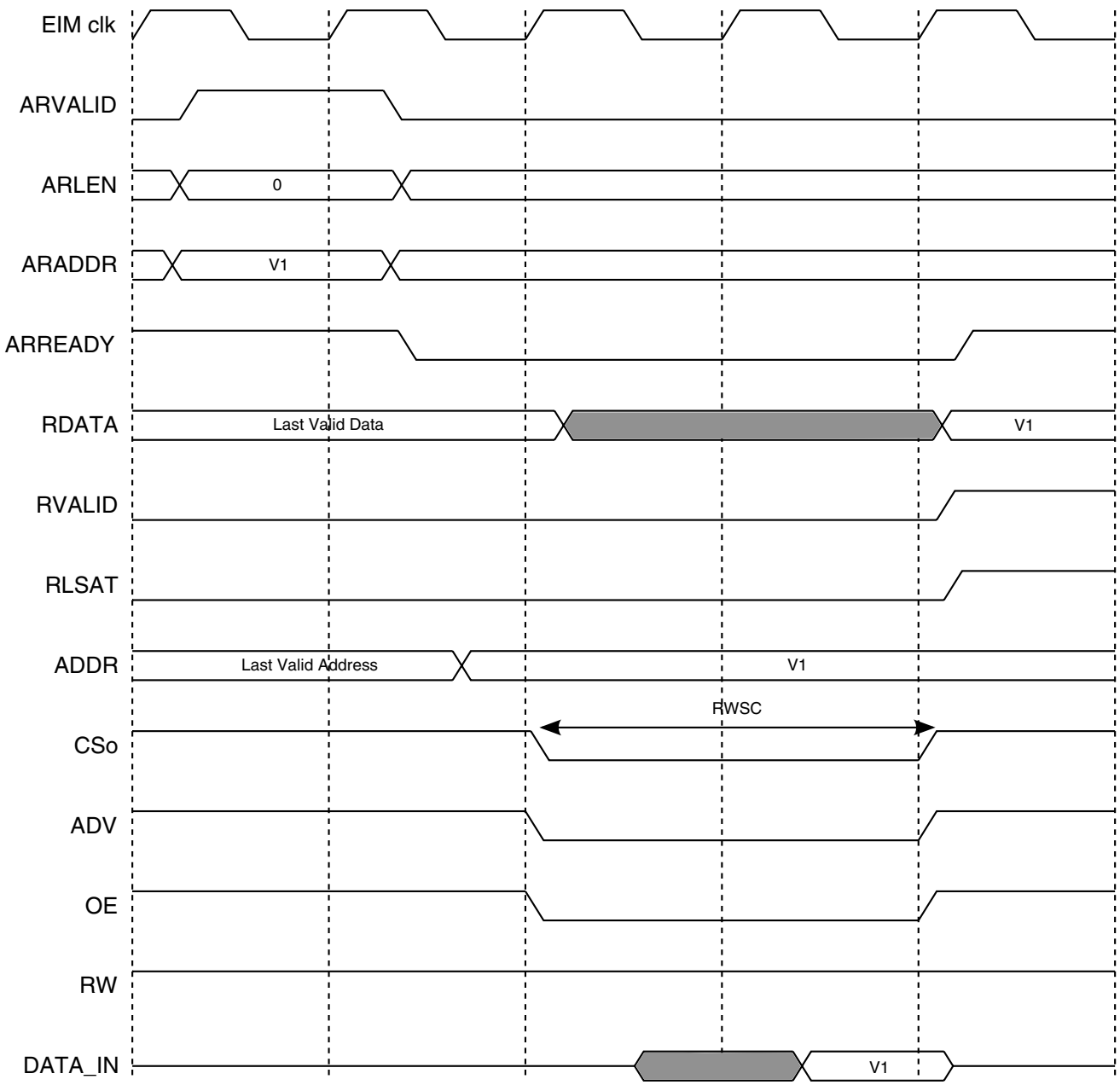


Figure 25-2. Read Access, RWSC=2,RCSA=0,OEA=0,RCSN=0,OEN=0, RAL=1

25.9.2 Asynchronous Write Memory Accesses Timing Diagram

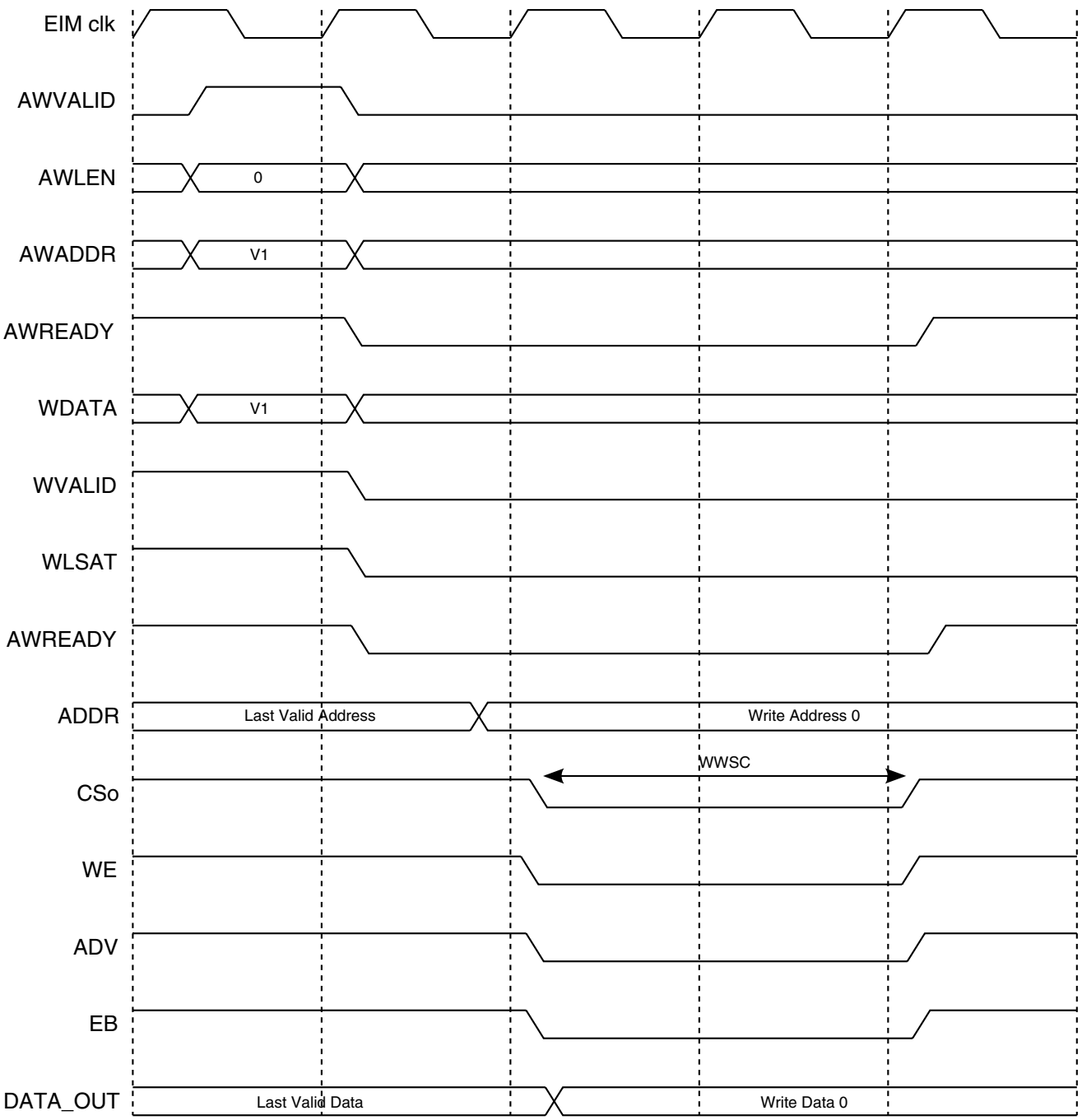


Figure 25-3. Write Access, WWSC=2,WCSA=0,WEA=0,WCSN=0,WEN=0,BEA=0, BEN=0, WAL=1

25.9.3 Asynchronous Read/Write Memory Accesses Timing Diagram

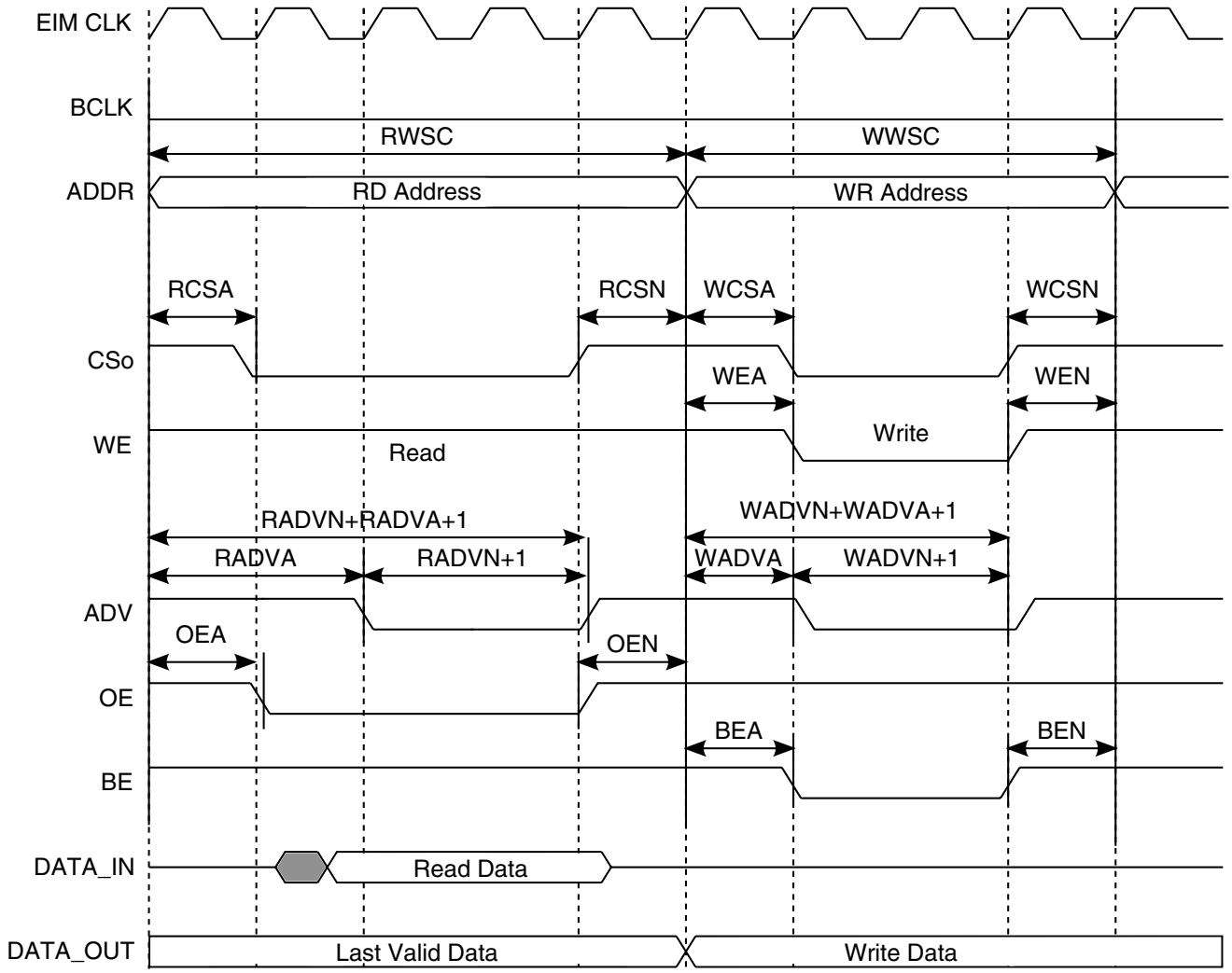


Figure 25-4. RCSA=1,RADVA=2,OEA=1,RADVN=1,RCSN=1,OEN=1,WCSA=1,WEA=1,WADVA=1,BEA=1,WADVN=1,WCSN=1,WEN=1,BEN=1

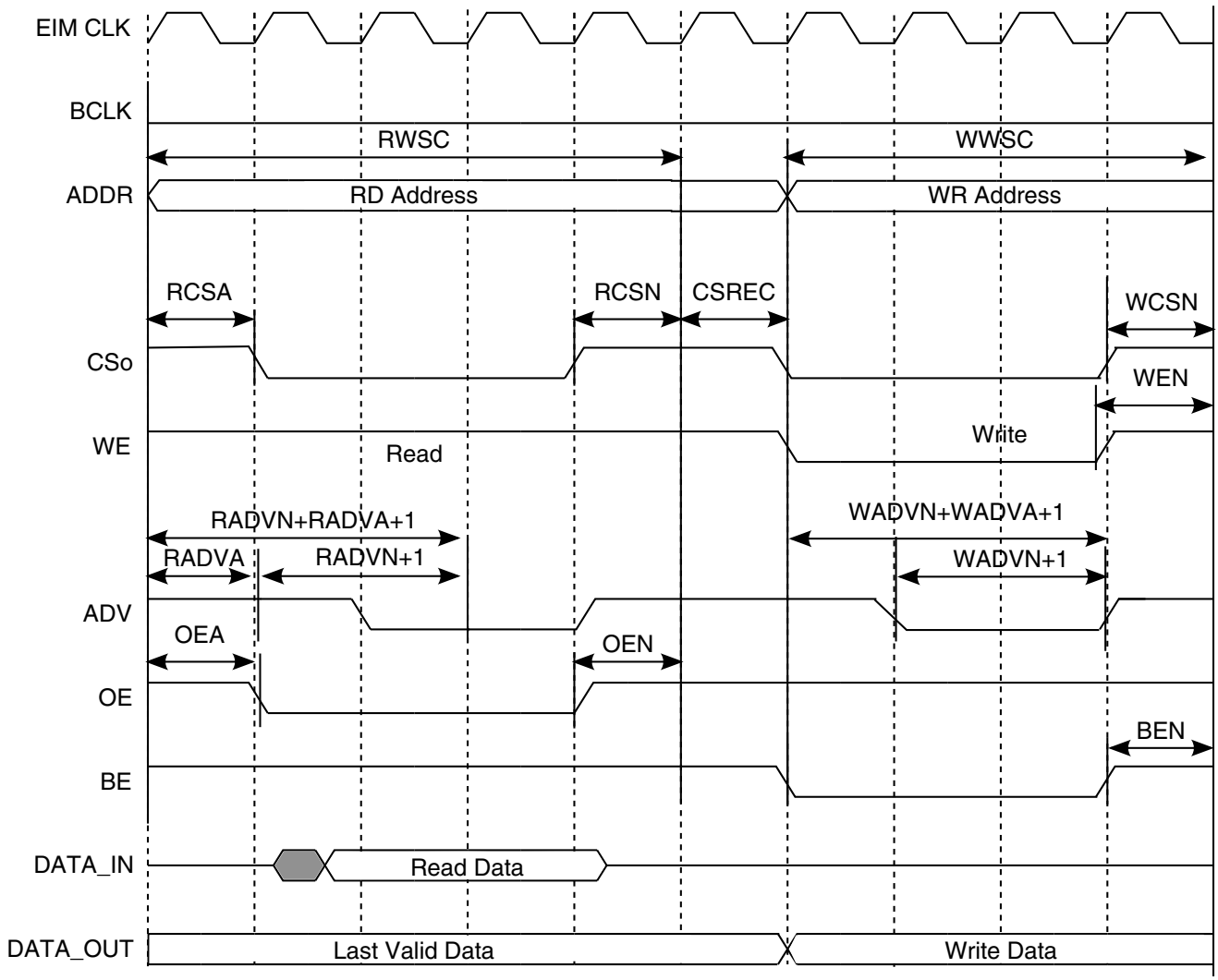


Figure 25-5.
RWSC=5,RCSA=1,RCSN=1,RADVA=1,RADVN=1,OEA=1,OEN=1,WWSC=4,WCSA=0,WCSN=1,WEA=0,WEN=1,WADVA=1,WADVN=1,BEA=0,BEN=1,CSREC=1

25.9.4 Asynchronous Read/Write Using RAL, WAL and CSREC

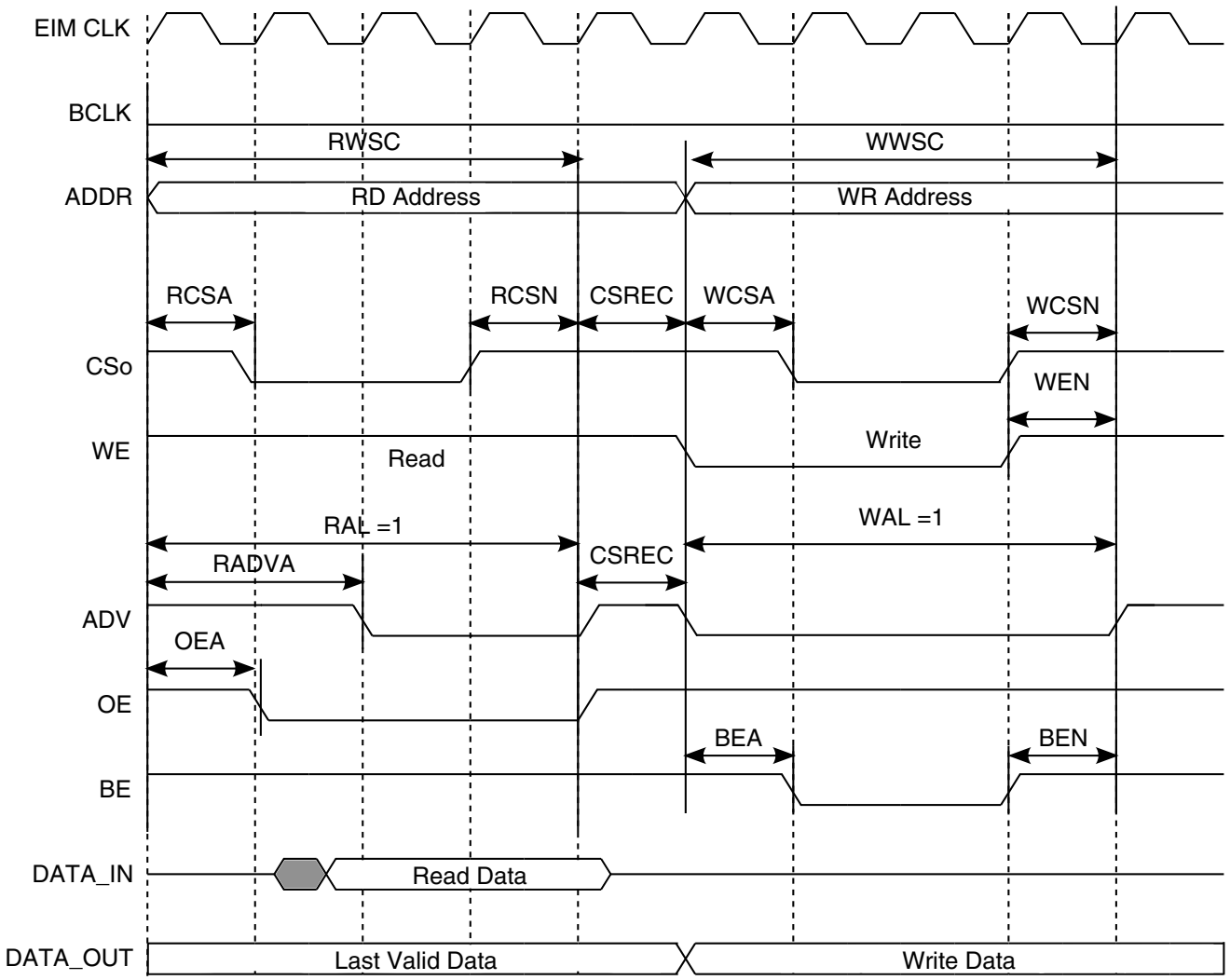


Figure 25-6.
RAL=1,RCSN=1,RADVA=2,OEA=1,RCSN=1,CSREC=1,WCSA=1,WEA=0,WADVA=0,BEA=1,WAL=1,WCSN=1,WEN=1,BEN=1

25.9.5 Consecutive Asynchronous Write Memory Accesses Timing Diagram

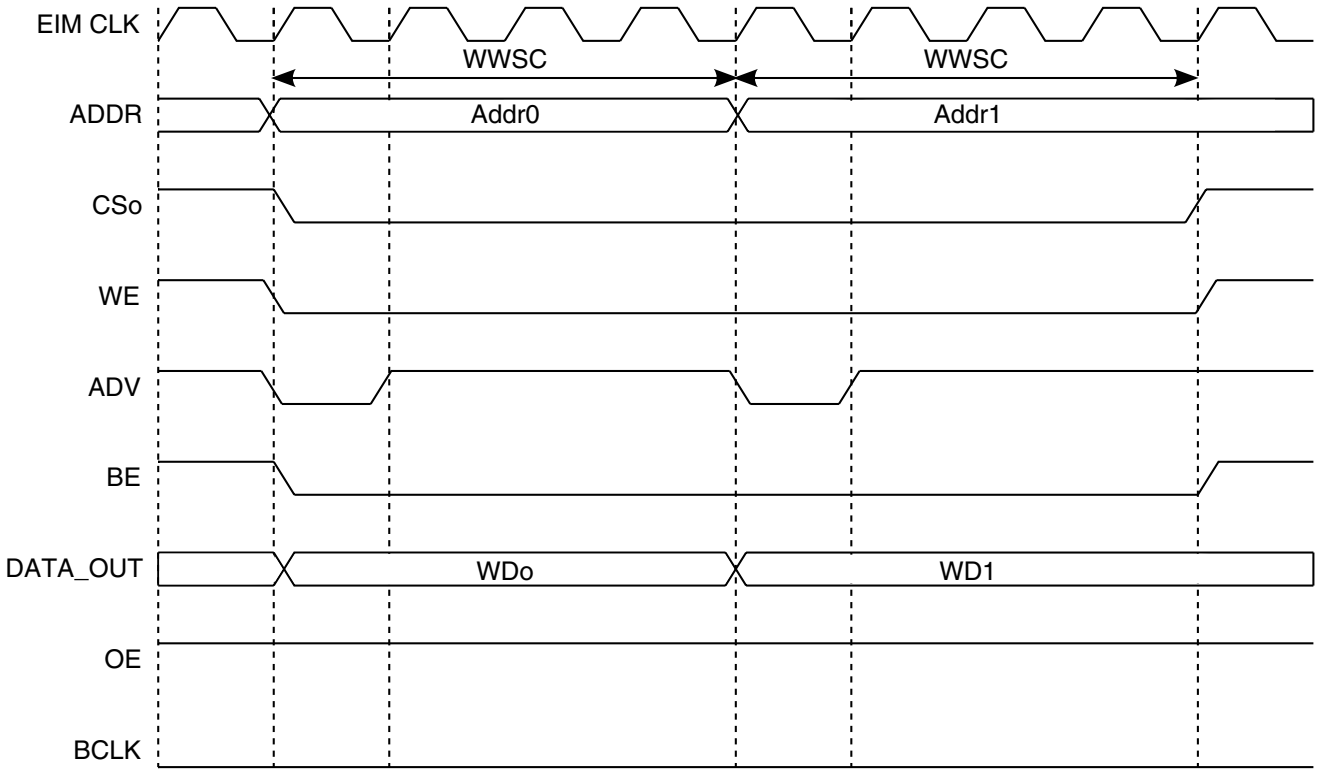


Figure 25-7. WWSC=4,WCSA=0,WEA=0,WADVA=0,BEA=0,WCSN=0,WEN=0,WADVN=0,BEN=0,CSREC=0

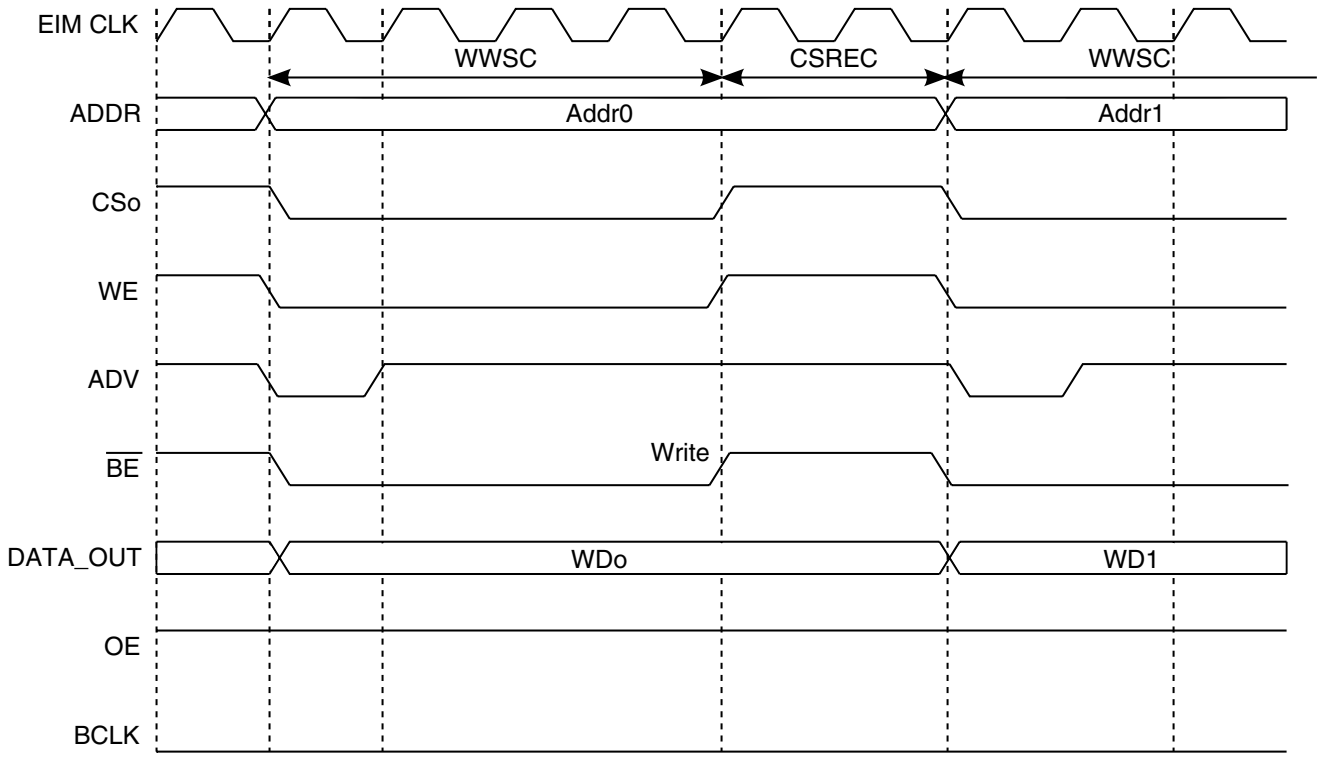


Figure 25-8. WWSC=4,WCSA=0,WEA=0,WADVA=0,BEA=0,WCSN=0,WEN=0,WADVN=0,BEN=0,CSREC=2

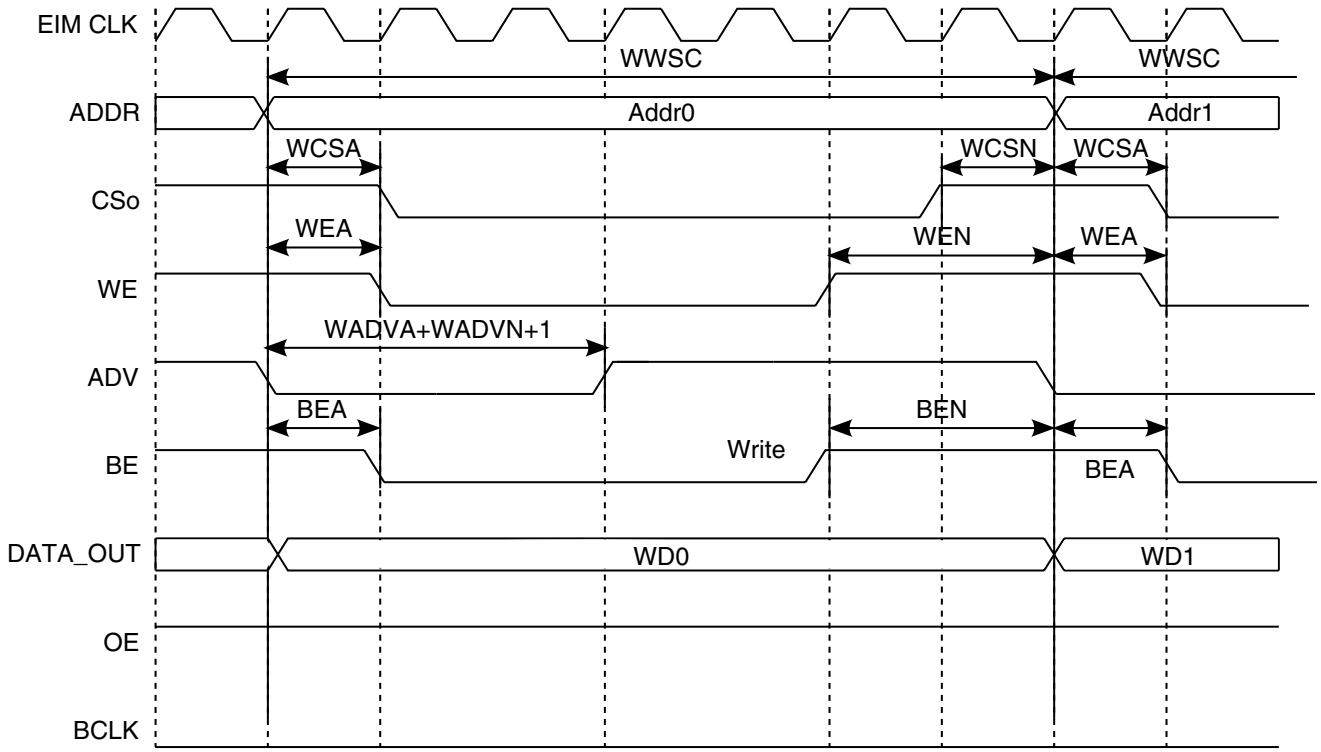


Figure 25-9.
WWSC=7,WCSA=1,WCSN=1,WEA=1,WEN=2,WADVA=0,WADVN=2,BEA=1,BEN=2

25.9.6 Consecutive Asynchronous Read Memory Accesses Timing Diagram

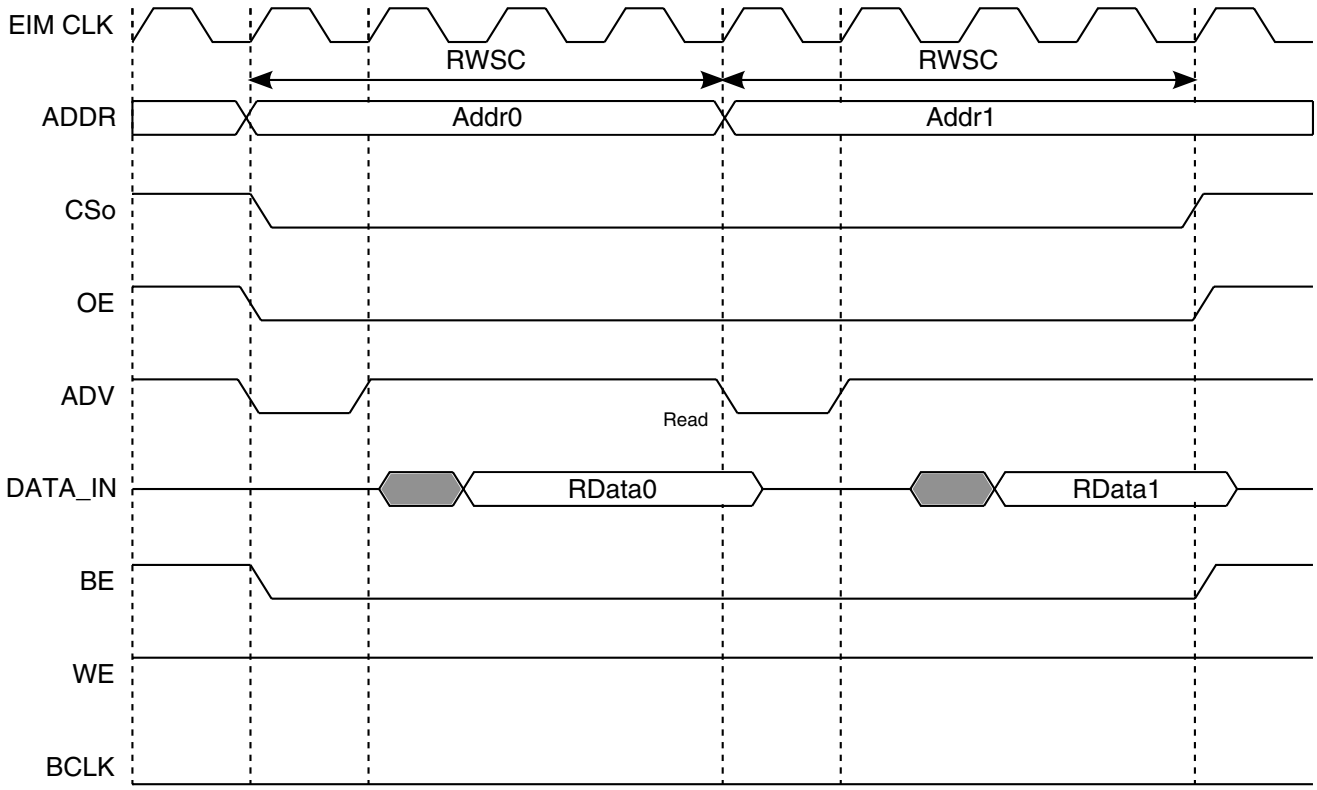


Figure 25-10. RWSC=4,RCSA=0,OEA=0,RADVA=0,RCSN=0,OEN=0,RADVN=0,CSREC=0

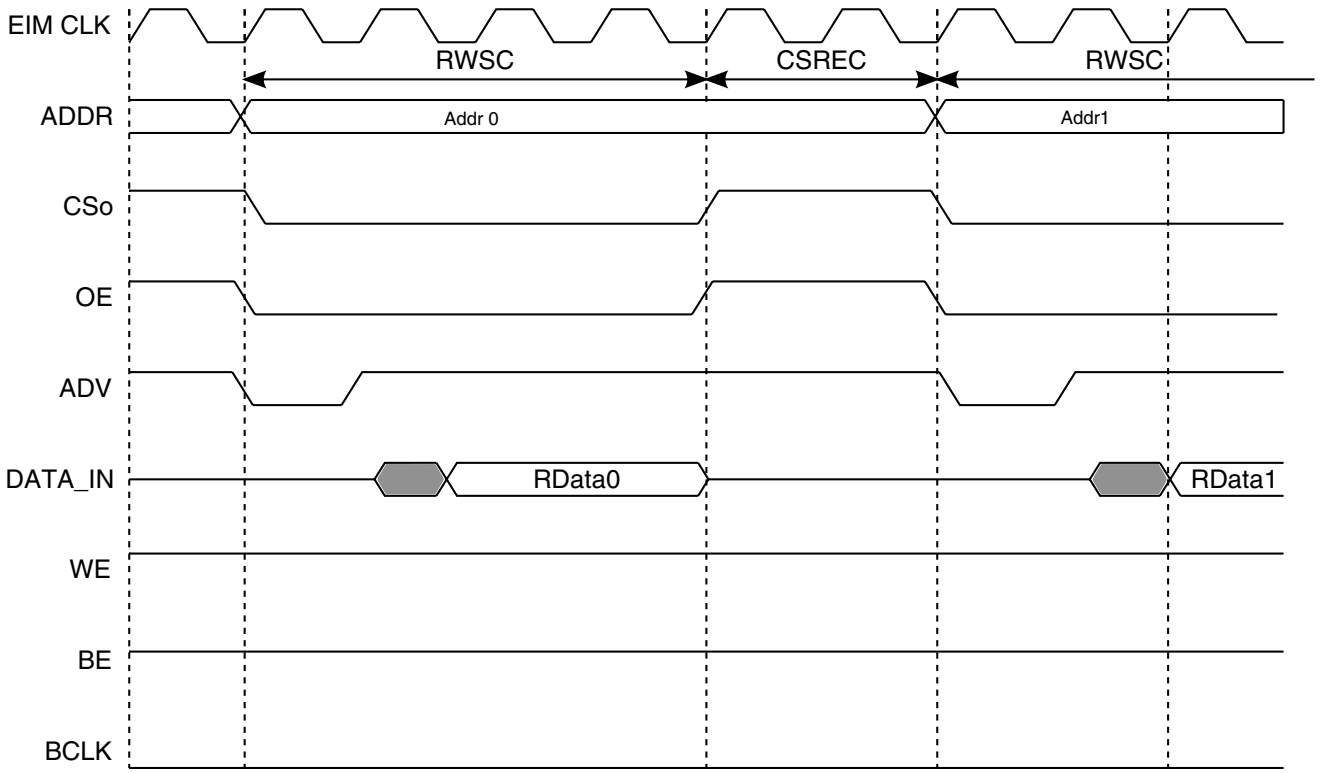


Figure 25-11. RWSC=4,RCSA=0,OEA=0,RADVA=0,RCSN=0,OEN=0,RADVN=0,CSREC=2

25.9.7 Async. Page Mode Access

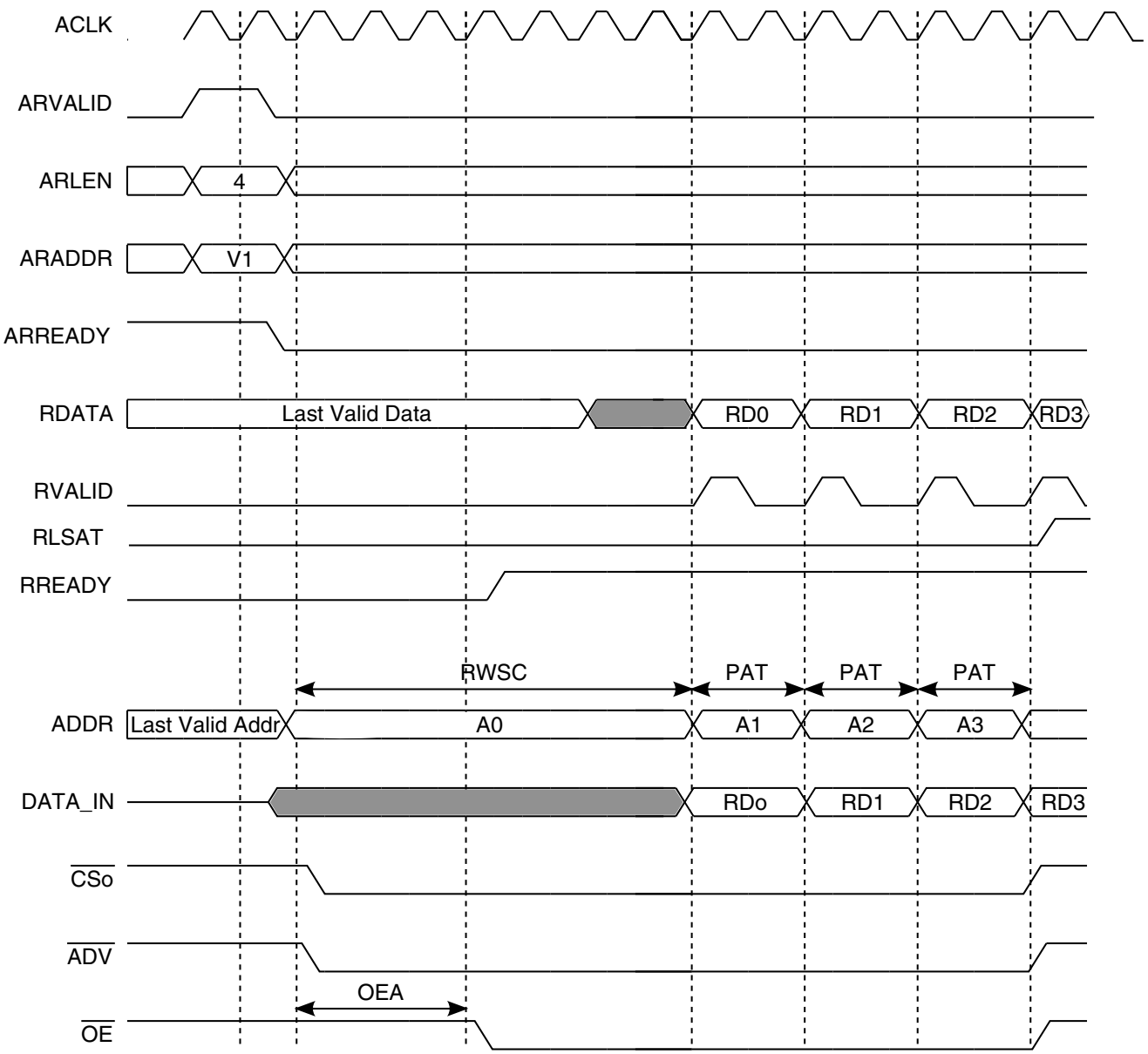


Figure 25-12. PAT = 2

25.9.8 DTACK Mode - AXI Single Access

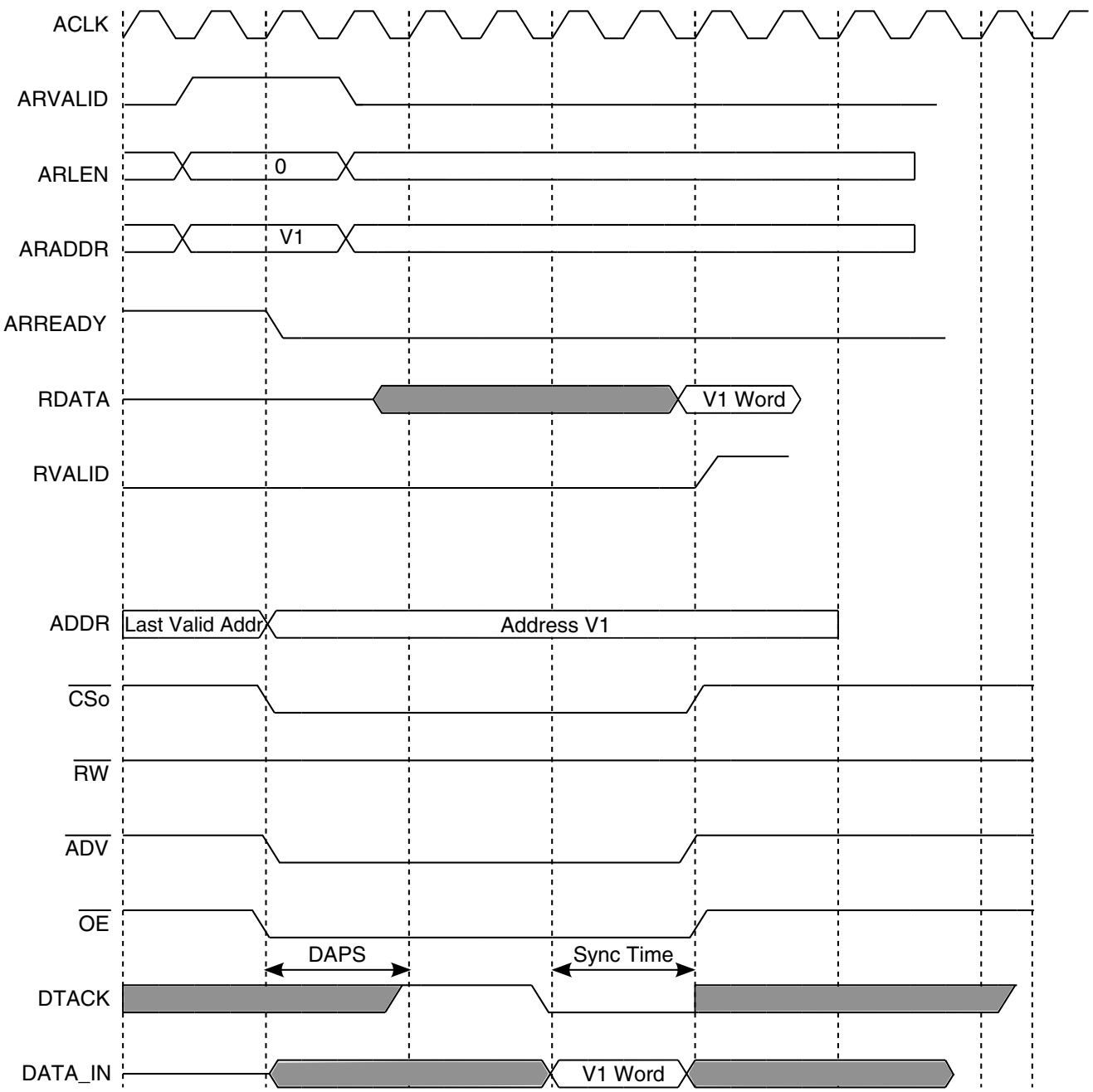


Figure 25-13. DAPS = 2

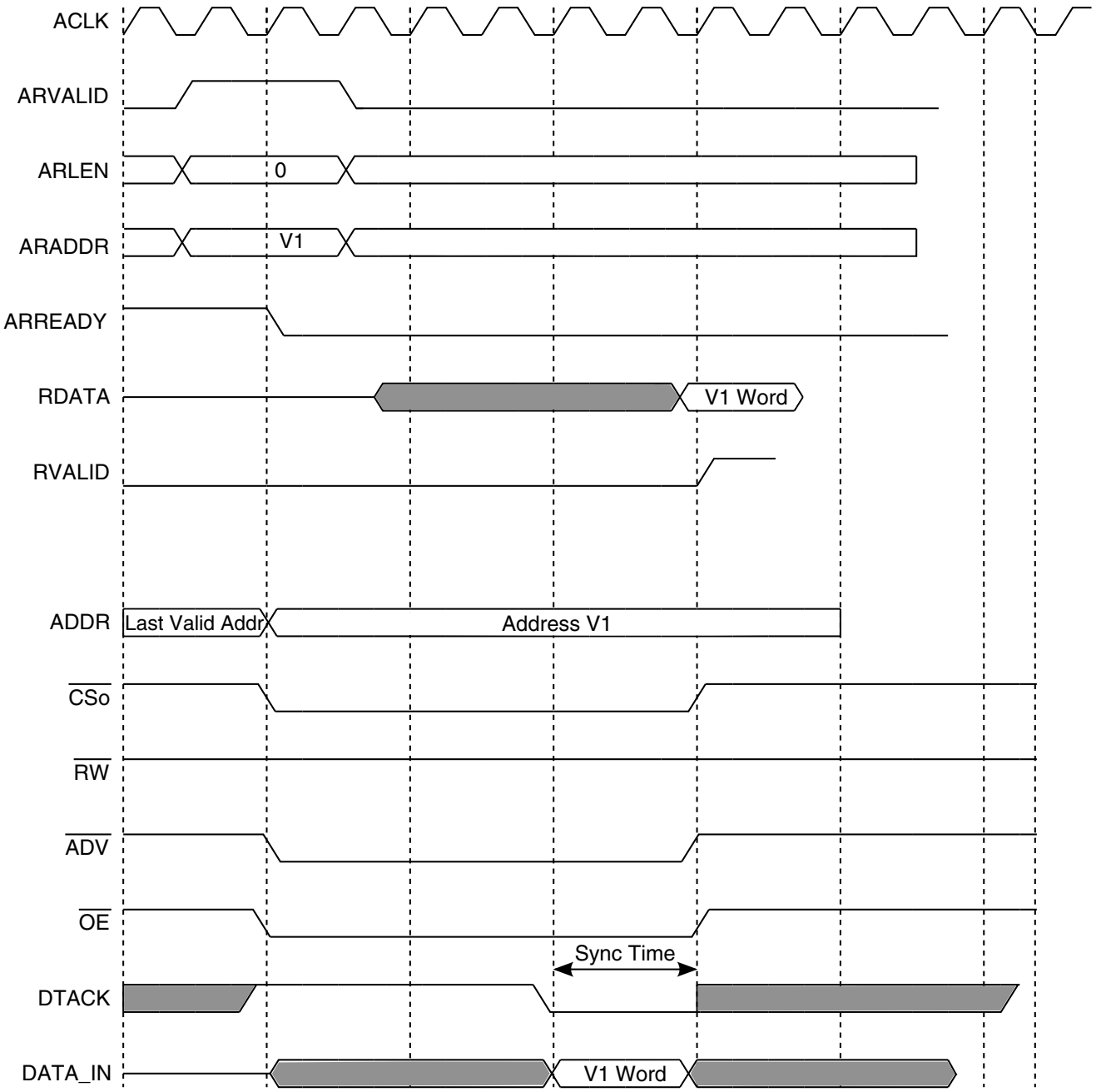


Figure 25-14. DAPS = 0

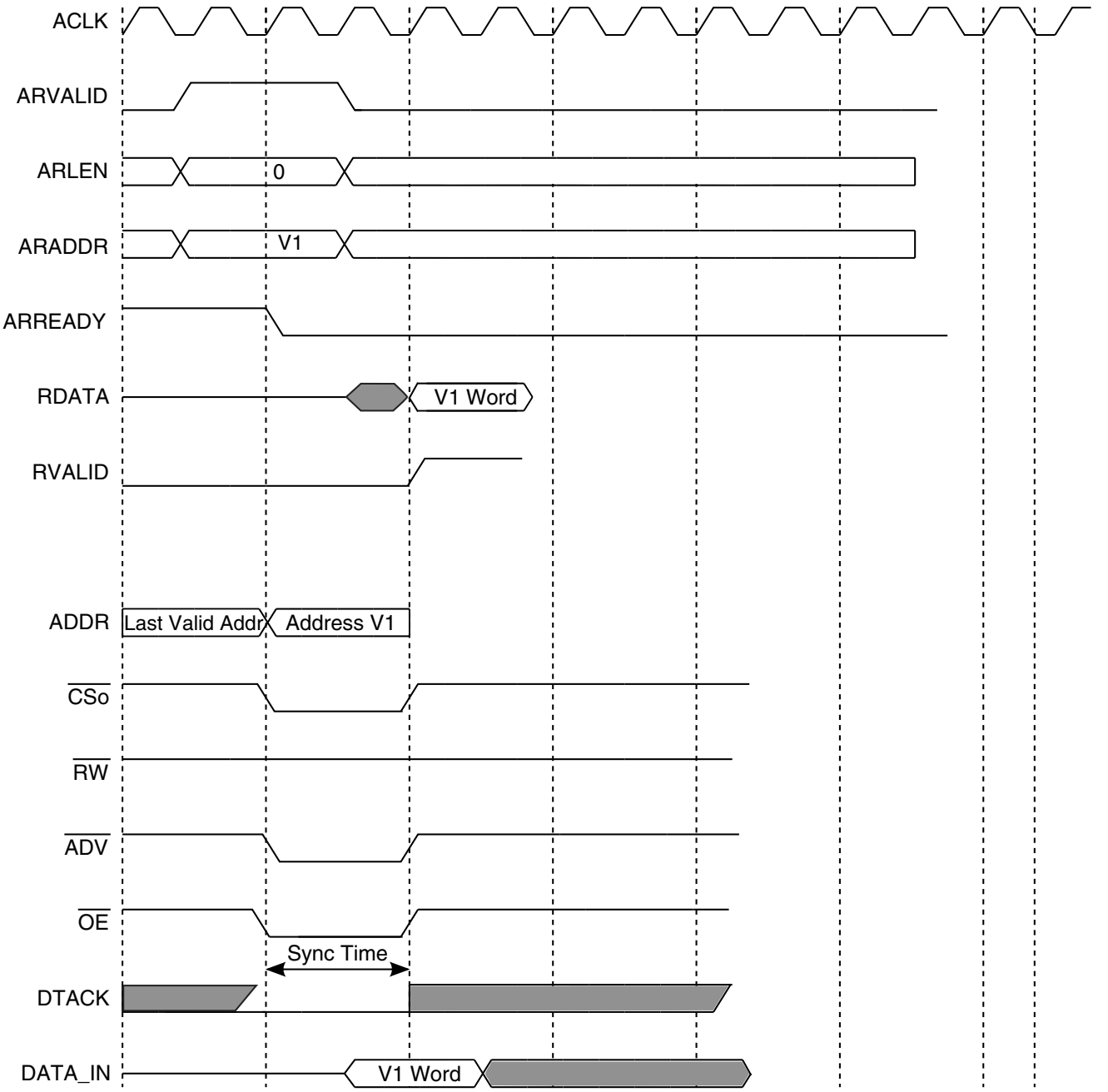


Figure 25-15. DAPS = 0

25.9.9 DTACK Mode - AXI Single Write Access

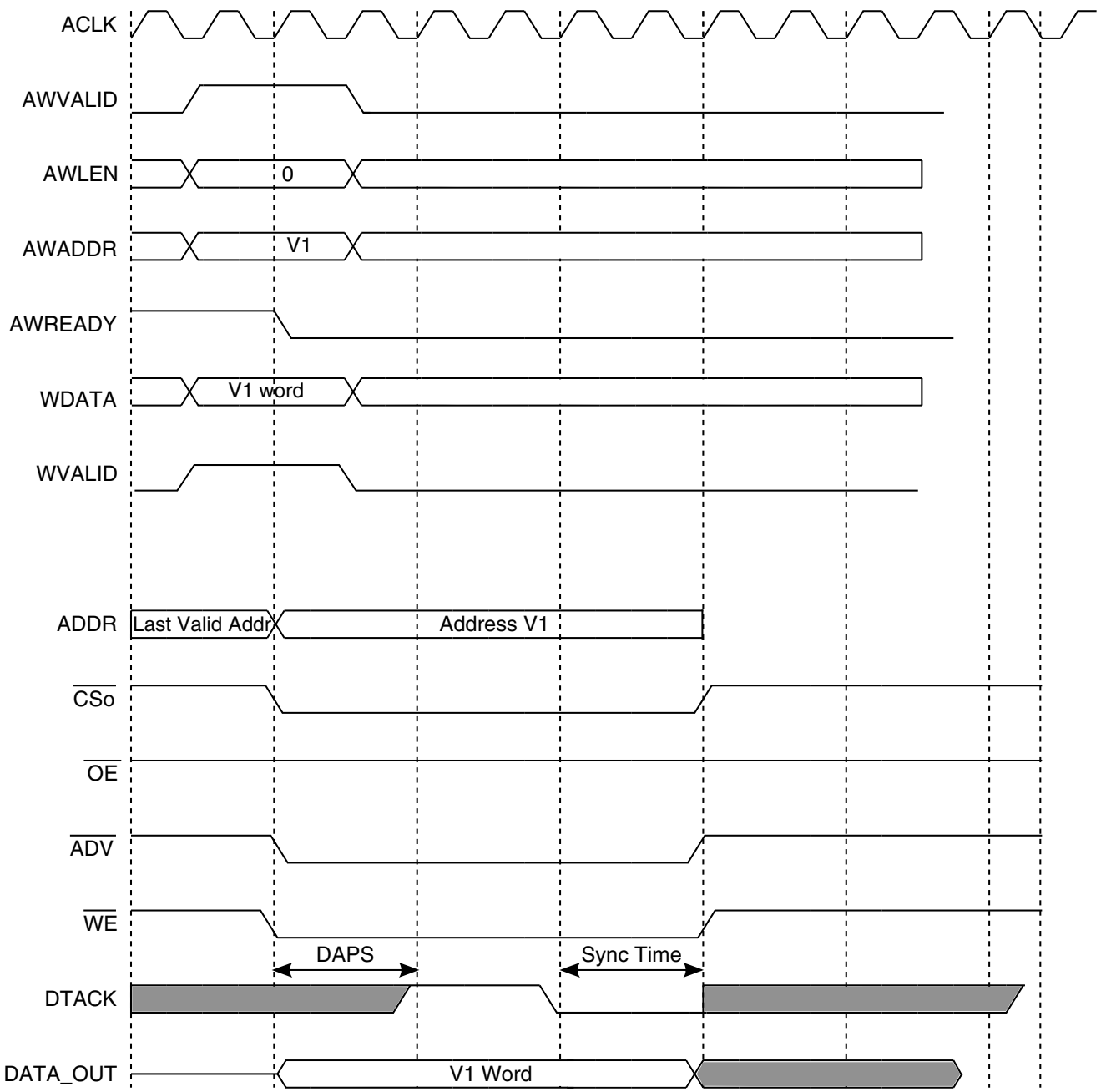


Figure 25-16. DAPS = 2

25.9.10 DTACK Mode - AXI Burst Access

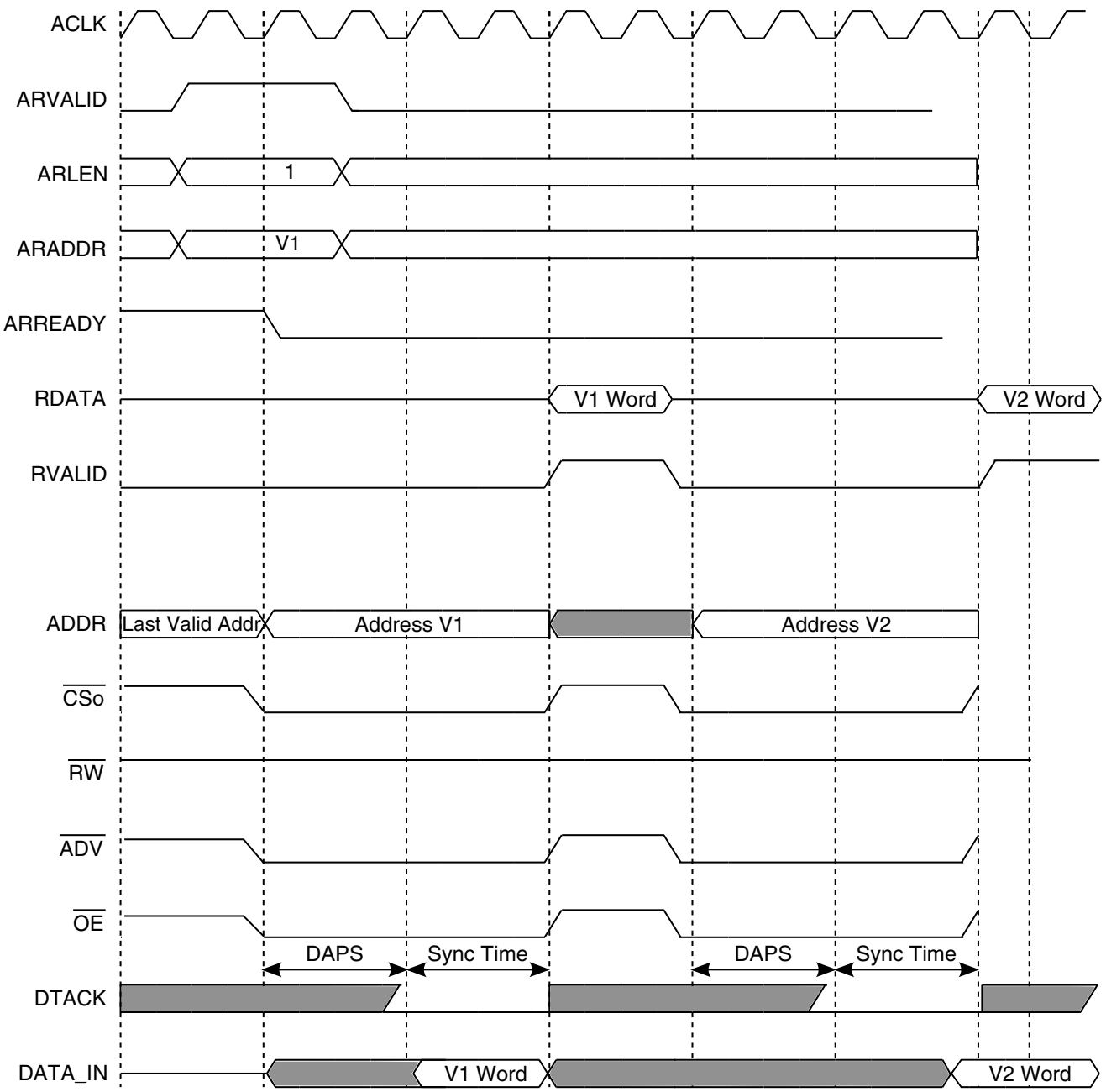


Figure 25-17. DAPS = 2 CSREC = 2

25.10 Programmable Registers

25.10.1 EIM Memory Map/Register Definition

The EIM includes 33 user-accessible 32-bit registers. The the EIM Configuration Register (EIM_WCR) contains control bits that configure the EIM for certain operation modes.

The 160 bits used to control Individual Chip Select are divided into five registers:

- Chip Select x General Configuration Register 1 (EIM_CSxGCR1)
- Chip Select x General Configuration Register 2 (EIM_CSxGCR2)
- Chip Select x Read Configuration Register 1 (EIM_CSxRCR1)
- Chip Select x Read Configuration Register 2 (EIM_CSxRCR2)
- Chip Select x Write Configuration Register (EIM_CSxWCR)

In addition there are 3 general registers: EIM_WCR, EIM_WIAR & EIM_EAR.

NOTE

- All EIM registers are sampled by IPG_CLK_S, therefore IPG_CLK_S must be active when accessing through IP bus.
- Read access from all registers (except EIM_WIAR & EIM_EAR) will generate one IPG_XFR_WAIT cycle.
- Read access from EIM_WIAR & EIM_EAR will generate six IPG_XFR_WAIT cycles.
- Write access to all registers (except EIM_EAR) will generate three IPG_XFR_WAIT cycles.
- Write access to EIM_EAR will generate six IPG_XFR_WAIT cycles.

EIM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FD_A000	Chip Select n General Configuration Register 1 (EIM_CS0GCR1)	32	R/W	See section	25.101.1/ 1116
63FD_A004	Chip Select n General Configuration Register 2 (EIM_CS0GCR2)	32	R/W	0000_1000h	25.101.2/ 1120

Table continues on the next page...

EIM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FD_A008	Chip Select n Read Configuration Register 1 (EIM_CS0RCR1)	32	R/W	See section	25.101.3/ 1122
63FD_A00C	Chip Select n Read Configuration Register 2 (EIM_CS0RCR2)	32	R/W	0000_0000h	25.101.4/ 1124
63FD_A010	Chip Select n Write Configuration Register 1 (EIM_CS0WCR1)	32	R/W	See section	25.101.5/ 1126
63FD_A014	Chip Select n Write Configuration Register 2 (EIM_CS0WCR2)	32	R/W	0000_0000h	25.101.6/ 1129
63FD_A018	Chip Select n General Configuration Register 1 (EIM_CS1GCR1)	32	R/W	See section	25.101.1/ 1116
63FD_A01C	Chip Select n General Configuration Register 2 (EIM_CS1GCR2)	32	R/W	0000_1000h	25.101.2/ 1120
63FD_A020	Chip Select n Read Configuration Register 1 (EIM_CS1RCR1)	32	R/W	See section	25.101.3/ 1122
63FD_A024	Chip Select n Read Configuration Register 2 (EIM_CS1RCR2)	32	R/W	0000_0000h	25.101.4/ 1124
63FD_A028	Chip Select n Write Configuration Register 1 (EIM_CS1WCR1)	32	R/W	See section	25.101.5/ 1126
63FD_A02C	Chip Select n Write Configuration Register 2 (EIM_CS1WCR2)	32	R/W	0000_0000h	25.101.6/ 1129
63FD_A030	Chip Select n General Configuration Register 1 (EIM_CS2GCR1)	32	R/W	See section	25.101.1/ 1116
63FD_A034	Chip Select n General Configuration Register 2 (EIM_CS2GCR2)	32	R/W	0000_1000h	25.101.2/ 1120
63FD_A038	Chip Select n Read Configuration Register 1 (EIM_CS2RCR1)	32	R/W	See section	25.101.3/ 1122
63FD_A03C	Chip Select n Read Configuration Register 2 (EIM_CS2RCR2)	32	R/W	0000_0000h	25.101.4/ 1124
63FD_A040	Chip Select n Write Configuration Register 1 (EIM_CS2WCR1)	32	R/W	See section	25.101.5/ 1126
63FD_A044	Chip Select n Write Configuration Register 2 (EIM_CS2WCR2)	32	R/W	0000_0000h	25.101.6/ 1129
63FD_A048	Chip Select n General Configuration Register 1 (EIM_CS3GCR1)	32	R/W	See section	25.101.1/ 1116
63FD_A04C	Chip Select n General Configuration Register 2 (EIM_CS3GCR2)	32	R/W	0000_1000h	25.101.2/ 1120
63FD_A050	Chip Select n Read Configuration Register 1 (EIM_CS3RCR1)	32	R/W	See section	25.101.3/ 1122
63FD_A054	Chip Select n Read Configuration Register 2 (EIM_CS3RCR2)	32	R/W	0000_0000h	25.101.4/ 1124

Table continues on the next page...

EIM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FD_A058	Chip Select n Write Configuration Register 1 (EIM_CS3WCR1)	32	R/W	See section	25.101.5/ 1126
63FD_A05C	Chip Select n Write Configuration Register 2 (EIM_CS3WCR2)	32	R/W	0000_0000h	25.101.6/ 1129
63FD_A090	EIM Configuration Register (EIM_WCR)	32	R/W	0000_0020h	25.101.7/ 1129
63FD_A094	EIM IP Access Register (EIM_WIAR)	32	R/W	0000_0010h	25.101.8/ 1131
63FD_A098	Error Address Register (EIM_EAR)	32	R/W	0000_0000h	25.101.9/ 1132

25.101.1 Chip Select n General Configuration Register 1 (EIM_CSGCR1)

Addresses: EIM_CS0GCR1 is 63FD_A000h base + 0h offset = 63FD_A000h
EIM_CS1GCR1 is 63FD_A000h base + 18h offset = 63FD_A018h
EIM_CS2GCR1 is 63FD_A000h base + 30h offset = 63FD_A030h
EIM_CS3GCR1 is 63FD_A000h base + 48h offset = 63FD_A048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PSZ				WP	GBC			AUS	CSREC			SP	DSZ			BCS	BCD			WC	BL			CREP	CRE	RFL	WFL	MUM	SRD	SWR	CSEN
Reset	0	0	0	0	0	0	0	0	0	n*	n*	n*	0	n*	n*	n*	0	0	0	0	0	0	0	0	1	0	0	0	n*	0	0	0

* Notes:
- CSREC bitfield: See field descriptions for the reset values.
- DSZ bitfield: See field descriptions for the reset values.
- MUM bitfield: See field descriptions for the reset values.

EIM_CS*n*GCR1 field descriptions

Field	Description
31–28 PSZ	Page Size. This bit field indicates memory page size in words (word is defined by the DSZ field). PSZ is used when fix latency mode is applied, WFL=1 for sync. write accesses, RFL=1 for sync. Read accesses. When working in fix latency mode WAIT signal from the external device is not being monitored, PSZ is used to determine if page boundary is reached and renewal of access is preformed. This bit field is ignored when sync. Mode is disabled or fix latency mode is not being used for write or read access separately. It can be valid for both access type, read or write, or only for one type, according to configuration. PSZ is cleared by a hardware reset. 0000 8 words page size 0001 16 words page size 0010 32 words page size

Table continues on the next page...

EIM_CS*n*GCR1 field descriptions (continued)

Field	Description
	0011 64 words page size 0100 128 words page size 0101 256 words page size 0110 512 words page size 0111 1024 (1k) words page size 1000 2048 (2k) words page size 1001 - 1111 Reserved
27 WP	Write Protect. This bit prevents writes to the address range defined by the corresponding chip select. WP is cleared by a hardware reset. 0 Writes are allowed in the memory range defined by chip. 1 Writes are prohibited. All attempts to write to an address mapped by this chip select result in a error response and no assertion of the chip select output.
26–24 GBC	Gap Between Chip Selects. This bit field, according to the settings shown below, determines the minimum time between end of access to the current chip select and start of access to different chip select. GBC is cleared by a hardware reset. Example settings: 000 minimum of 0 EIM clock cycles before next access from different chip select (async. mode only) 001 minimum of 1 EIM clock cycles before next access from different chip select 010 minimum of 2 EIM clock cycles before next access from different chip select 111 minimum of 7 EIM clock cycles before next access from different chip select
23 AUS	Address UnShifted. This bit indicates an unshifted mode for address assertion for the relevant chip select accesses. AUS bit is cleared by hardware reset. 0 Address shifted according to port size (DSZ config.) 1 Address unshifted
22–20 CSREC	CS Recovery. This bit field, according to the settings shown below, determines the minimum pulse width of CS, OE, and WE control signals before executing a new back to back access to the same chip select. CSREC is cleared by a hardware reset. **NOTE:** The reset value for EIM_CS0GCR1, CSREC[2:0] is 0b110. For EIM_CS1GCR1 - EIM_CS5GCR, the reset value is 0b000. Example settings: 000 0 EIM clock cycles minimum width of CS, OE and WE signals (read async. mode only) 001 1 EIM clock cycles minimum width of CS, OE and WE signals 010 2 EIM clock cycles minimum width of CS, OE and WE signals 111 7 EIM clock cycles minimum width of CS, OE and WE signals
19 SP	Supervisor Protect. This bit prevents accesses to the address range defined by the corresponding chip select when the access is attempted in the User mode. SP is cleared by a hardware reset. 0 User mode accesses are allowed in the memory range defined by chip select. 1 User mode accesses are prohibited. All attempts to access an address mapped by this chip select in User mode results in an error response and no assertion of the chip select output.
18–16 DSZ	Data Port Size. This bit field defines the width of an external device's data port as shown below. **NOTE:** Only async. access supported for 8 bit port.

Table continues on the next page...

EIM_CS*n*GCR1 field descriptions (continued)

Field	Description
	NOTE: The reset value for EIM_CS0GCR1, DSZ[2] = 0, DSZ[1:0] = EIM_BOOT[1:0]. For EIM_CS1GCR1 - EIM_CS5GCR1, the reset value is 0b001. 000 Reserved. 001 16 bit port resides on DATA[15:0] 010 16 bit port resides on DATA[31:16] 011 32 bit port resides on DATA[31:0] 100 8 bit port resides on DATA[7:0] 101 8 bit port resides on DATA[15:8] 110 8 bit port resides on DATA[23:16] 111 8 bit port resides on DATA[31:24]
15–14 BCS	Burst Clock Start. When SRD=1 or SWR=1,this bit field determines the number of EIM clock cycles delay from start of access before the first rising edge of BCLK is generated. When BCD=0 value of BCS=0 results in a half clock delay after the start of access. For other values of BCD a one clock delay after the start of access is applied, not an immediate assertion. BCS is cleared by a hardware reset. 00 0 EIM clock cycle additional delay 01 1 EIM clock cycle additional delay 10 2 EIM clock cycle additional delay 11 3 EIM clock cycle additional delay
13–12 BCD	Burst Clock Divisor. This bit field contains the value used to program the burst clock divisor for BCLK generation. It is used to divide the internal EIMbus frequency. BCD is cleared by a hardware reset. **NOTE:** For other then the mentioned below frequency such as 104 MHz, EIM clock (input clock) should be adjust accordingly. 00 Divide EIM clock by 1 01 Divide EIM clock by 2 10 Divide EIM clock by 3 11 Divide EIM clock by 4
11 WC	Write Continuous. The WI bit indicates that write access to the memory are always continuous accesses regardless of the BL field value. WI is cleared by hardware reset. 0 Write access burst length occurs according to BL value. 1 Write access burst length is continuous.
10–8 BL	Burst Length. The BL bit field indicates memory burst length in words (word is defined by the DSZ field) and should be properly initialized for mixed wrap/increment accesses support. Continuous BL value corresponds to continuous burst length setting of the external memory device. For fix memory burst size, type is always wrap. In case not matching wrap boundaries in both the memory (BL field) and Master access on the current address, EIM update address on the external device address bus and regenerates the access. BL is cleared by a hardware reset. When APR=1, Page Read Mode is applied, BL determine the number of words within the read page burst. BL is cleared by a hardware reset for EIM_CS0GCR1 - EIM_CS5GCR1. 000 4 words Memory wrap burst length (read page burst size when APR = 1) 001 8 words Memory wrap burst length (read page burst size when APR = 1) 010 16 words Memory wrap burst length (read page burst size when APR = 1)

Table continues on the next page...

EIM_CS*n*GCR1 field descriptions (continued)

Field	Description
	011 32 words Memory wrap burst length (read page burst size when APR = 1) 100 Continuous burst length (2 words read page burst size when APR = 1) 101 Reserved 110 Reserved 111 Reserved
7 CREP	Configuration Register Enable Polarity. This bit indicates CRE memory pin assertion state, active-low or active-high, while executing a memory register set command to the external device (PSRAM memory type). CREP is set by a hardware reset. **NOTE:** Whenever PSRAM is connected the CREP value must be correct also for accesses where CRE is disabled. For Non-PSRAM memory CREP value should be 1. 0 CRE signal is active low 1 CRE signal is active high
6 CRE	Configuration Register Enable. This bit indicates CRE memory pin state while executing a memory register set command to PSRAM external device. CRE is cleared by a hardware reset. 0 CRE signal use is disable 1 CRE signal use is enable
5 RFL	Read Fix Latency. This bit field determine if the controller is monitoring the WAIT signal from the External device connected to the chip select (handshake mode - fix or variable data latency) or if it start sampling data according to RWSC field, it only valid in synchronous mode. RFL is cleared by a hardware reset. When RFL=1 Burst access is terminated on page boundary and resume on the following page according to BL bit field configuration, because WAIT signal is not monitored from the external device. 0 the External device WAIT signal is being monitored, and it reflect the external data bus state 1 the state of the External devices is determined internally (Fix latency mode only)
4 WFL	Write Fix Latency. This bit field determine if the controller is monitoring the WAIT signal from the External device connected to the chip select (handshake mode - fix or variable data latency) or if it start data transfer according to WWSC field, it only valid in synchronous mode. WFL is cleared by a hardware reset. When WFL=1 Burst access is terminated on page boundary and resume on the following page according to BL bit field configuration, because WAIT signal is not monitored from the external device 0 the External device WAIT signal is being monitored, and it reflect the external data bus state 1 the state of the External devices is determined internally (Fix latency mode only)
3 MUM	Multiplexed Mode. This bit determines the address/data multiplexed mode for asynchronous and synchronous accesses for 8 bit, 16 bit or 32 bit devices (DSZ config. dependent). **NOTE:** The reset value for EIM_CS0GCR1[MUM] = EIM_BOOT[2]. For EIM_CS1GCR1 - EIM_CS5GCR1 the reset value is 0. 0 Multiplexed Mode disable 1 Multiplexed Mode enable
2 SRD	Synchronous Read Data. This bit field determine the read accesses mode to the External device of the chip select. The External device should be configured to the same mode as this bit implicates. SRD is cleared by a hardware reset. **NOTE:** Sync. accesses supported only for 16/32 bit port.

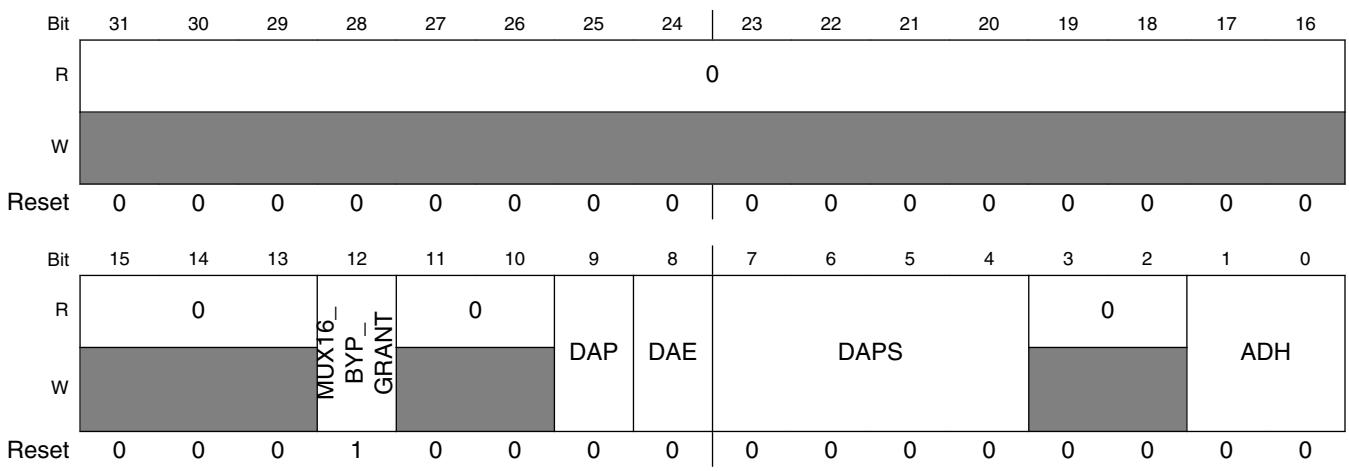
Table continues on the next page...

EIM_CS*n*GCR1 field descriptions (continued)

Field	Description
	0 read accesses are in Asynchronous mode 1 read accesses are in Synchronous mode
1 SWR	Synchronous Write Data. This bit field determine the write accesses mode to the External device of the chip select. The External device should be configured to the same mode as this bit implicates. SWR is cleared by a hardware reset. **NOTE:** Sync. accesses supported only for 16/32 bit port. 0 write accesses are in Asynchronous mode 1 write accesses are in Synchronous mode
0 CSEN	CS Enable. This bit controls the operation of the chip select pin. CSEN is set by a hardware reset for CSGCR0 to allow external boot operation. CSEN is cleared by a hardware reset to CSGCR1-CSGCR5. **NOTE:** Reset value for EIM_CS0GCR1 for CSEN is 1. For EIM_CS1GCR1-CS1GCR5 reset value is 0. 0 Chip select function is disabled; attempts to access an address mapped by this chip select results in an error respond and no assertion of the chip select output 1 Chip select is enabled, and is asserted when presented with a valid access.

25.101.2 Chip Select n General Configuration Register 2 (EIM_CSGCR2)

Addresses: EIM_CS0GCR2 is 63FD_A000h base + 4h offset = 63FD_A004h
EIM_CS1GCR2 is 63FD_A000h base + 1Ch offset = 63FD_A01Ch
EIM_CS2GCR2 is 63FD_A000h base + 34h offset = 63FD_A034h
EIM_CS3GCR2 is 63FD_A000h base + 4Ch offset = 63FD_A04Ch



EIM_CS*n*GCR2 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

EIM_CS*n*GCR2 field descriptions (continued)

Field	Description
12 MUX16_BYP_GRANT	Muxed 16 bypass grant. This bit when asserted causes EIM to bypass the grant/ack. arbitration with NFC (only for 16 bit muxed mode accesses). 0 EIM waits for grant before driving a 16 bit muxed mode access to the memory. 1 EIM ignores the grant signal and immediately drives a 16 bit muxed mode access to the memory.
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DAP	Data Acknowledge Polarity. This bit indicates DTACK memory pin assertion state, active-low or active-high, while executing an async access using DTACK signal from the external device. DAP is cleared by a hardware reset. 0 DTACK signal is active high 1 DTACK signal is active low
8 DAE	Data Acknowledge Enable. This bit indicates external device is using DTACK pin as strobe/terminator of an async. access. DTACK signal may be used only in asynchronous single read (APR=0) or write accesses. DTACK poling start point is set by DAPS bit field. polarity of DTACK is set by DAP bit field. DAE is cleared by a hardware reset. 0 DTACK signal use is disable 1 DTACK signal use is enable
7–4 DAPS	Data Acknowledge Poling Start. This bit field determine the starting point of DTACK input signal polling. DAPS is used only in asynchronous single read or write accesses. **NOTE:** Since DTACK is an async. signal the start point of DTACK signal polling is at least 3 cycles after the start of access. DAPS is cleared by a hardware reset. Example settings: 0000 3 EIM clk cycle between start of access and first $\overline{\text{DTACK}}$ check 0001 4 EIM clk cycles between start of access and first $\overline{\text{DTACK}}$ check 0010 5 EIM clk cycles between start of access and first $\overline{\text{DTACK}}$ check 0111 10 EIM clk cycles between start of access and first $\overline{\text{DTACK}}$ check 1011 14 EIM clk cycles between start of access and first DTACK check 1111 18 EIM clk cycles between start of access and first DTACK check
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 ADH	Address hold time - This bit field determine the address hold time after ADV negation when mum = 1 (muxed mode). When mum = 0 this bit has no effect. For read accesses the field determines when the pads direction will be switched. **NOTE:** Reset value for EIM_CS0GCR2 for ADH is 10. For EIM_CS1GCR2-EIM_CS5GCR2 reset value is 00. 00 0 cycle after ADV negation 01 1 cycle after ADV negation 10 2 cycle after ADV negation 11 Reserved

25.101.3 Chip Select n Read Configuration Register 1 (EIM_CSRCR1)

Addresses: EIM_CS0RCR1 is 63FD_A000h base + 8h offset = 63FD_A008h
EIM_CS1RCR1 is 63FD_A000h base + 20h offset = 63FD_A020h
EIM_CS2RCR1 is 63FD_A000h base + 38h offset = 63FD_A038h
EIM_CS3RCR1 is 63FD_A000h base + 50h offset = 63FD_A050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		RWSC						0	RADVA			RAL	RADVN			0	OEA			0	OEN			0	RCSA			0	RCSN		
W																																
Reset	0	0	n*	n*	n*	n*	n*	n*	0	0	0	0	0	0	0	0	0	n*	n*	n*	0	0	0	0	0	0	0	0	0	0	0	0

* Notes:
- RWSC bitfield: See field descriptions for the reset values.
- OEA bitfield: See field descriptions for the reset values.

EIM_CS*n*RCR1 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved
29–24 RWSC	Read Wait State Control. This bit field programs the number of wait-states, according to the settings shown below, for synchronous or asynchronous read access to the external device connected to the chip select. When SRD=1 and RFL=0, RWSC indicates the number of burst clock (BCLK) cycles from the start of an access, before the controller can start sample data.Since WAIT signal can be asserted one cycle before the first data can be sampled, the controller starts evaluating the WAIT signal state one cycle before, this is referred as handshake mode or variable latency mode. When SRD=1 and RFL=1, RWSC indicates the number of burst clock (BCLK) cycles from the start of an access, until the external device is ready for data transfer, this is referred as fix latency mode. When SRD=0, RFL bit is ignored, RWSC indicates the asynchronous access length and the number of EIM clock cycles from the start of access until the external device is ready for data transfer. RWSC is cleared by a hardware reset. **NOTE:** The reset value for EIM_CS0RCR1, RWSC[5:0] = 0b011100. For CG1RCR1 - CS1RCR5 the reset value is 0b000000. Example settings: 000000 Reserved 000001 RWSC value is 1 000010 RWSC value is 2 111101 RWSC value is 61 111110 RWSC value is 62 111111 RWSC value is 63
23 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

EIM_CS*n*RCR1 field descriptions (continued)

Field	Description
22–20 RADVA	ADV Assertion. This bit field determines when ADV signal is asserted for synchronous or asynchronous read modes according to the settings shown below. RADVA is cleared by a hardware reset. Example settings: 000 0 EIM clock cycles between beginning of access and ADV assertion 001 1 EIM clock cycles between beginning of access and ADV assertion 010 2 EIM clock cycles between beginning of access and ADV assertion 111 7 EIM clock cycles between beginning of access and ADV assertion
19 RAL	Read ADV Low. This bit field determine ADV signal negation time. When RAL=1, RADVN bit field is ignored and ADV signal will stay asserted until end of access. When RAL=0 negation of ADV signal is according to RADVN bit field configuration.
18–16 RADVN	ADV Negation. This bit field determines when ADV signal to memory is negated during read accesses. When SRD=1 (synchronous read mode), ADV negation occurs according to the following formula: (RADVN + RADVA + BCD + BCS + 1) EIM clock cycles from start of access. When asynchronous read mode is applied (SRD=0) and RAL=0 ADV negation occurs according to the following formula: (RADVN + RADVA + 1) EIM clock cycles from start of access. RADVN is cleared by a hardware reset. **NOTE:** the reset value for EIM_CS0RCR1[RADVN] = 2. For EIM_CS1RCR1 - EIM_CS5RCR1, the reset value is 0b000. **NOTE:** This field should be configured so ADV negation will occur before the end of access. For ADV negation at the same time with the end of access user should RAL bit.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 OEA	OE Assertion. This bit field determines when OE signal are asserted during read cycles (synchronous or asynchronous mode), according to the settings shown below. OEA is cleared by a hardware reset. In muxed mode OE assertion occurs (OEA + RADVN + RADVA + ADH +1) EIM clock cycles from start of access. **NOTE:** The reset value for EIM_CS0RCR1[OEA] is 0b000 if EIM_BOOT[2] = 0. If EIM_BOOT[2] is 1, the reset value for EIM_CS0RCR1 is 0b010. The reset value of this field for EIM_CS1RCR1 - EIM_CS5RCR1 is 0b000. Example settings: 000 0 EIM clock cycles between beginning of access and OE assertion 001 1 EIM clock cycles between beginning of access and OE assertion 010 2 EIM clock cycles between beginning of access and OE assertion 111 7 EIM clock cycles between beginning of access and OE assertion
11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–8 OEN	OE Negation. This bit field determines when OE signal is negated during read cycles in asynchronous single mode only (SRD=0 & APR = 0), according to the settings shown below. This bit field is ignored when SRD=1. OEN is cleared by a hardware reset. Example settings: 000 0 EIM clock cycles between end of access and OE negation 001 1 EIM clock cycles between end of access and OE negation

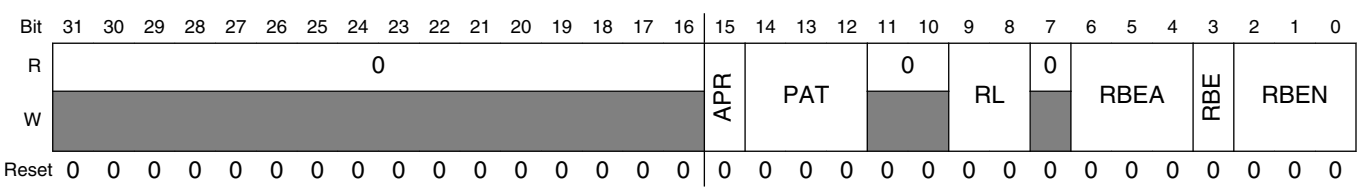
Table continues on the next page...

EIM_CS*n*RCR1 field descriptions (continued)

Field	Description
	010 2 EIM clock cycles between end of access and OE negation 111 7 EIM clock cycles between end of access and OE negation
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–4 RCSA	Read CS Assertion. This bit field determines when CS signal is asserted during read cycles (synchronous or asynchronous mode), according to the settings shown below. RCSA is cleared by a hardware reset. Example settings: 000 0 EIM clock cycles between beginning of read access and CS assertion 001 1 EIM clock cycles between beginning of read access and CS assertion 010 2 EIM clock cycles between beginning of read access and CS assertion 111 7 EIM clock cycles between beginning of read access and CS assertion
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 RCSN	Read CS Negation. This bit field determines when CS signal is negated during read cycles in asynchronous single mode only (SRD=0 & APR = 0), according to the settings shown below. This bit field is ignored when SRD=1. RCSN is cleared by a hardware reset. Example settings: 000 0 EIM clock cycles between end of read access and CS negation 001 1 EIM clock cycles between end of read access and CS negation 010 2 EIM clock cycles between end of read access and CS negation 111 7 EIM clock cycles between end of read access and CS negation

25.101.4 Chip Select n Read Configuration Register 2 (EIM_CSRCR2)

Addresses: EIM_CS0RCR2 is 63FD_A000h base + Ch offset = 63FD_A00Ch
EIM_CS1RCR2 is 63FD_A000h base + 24h offset = 63FD_A024h
EIM_CS2RCR2 is 63FD_A000h base + 3Ch offset = 63FD_A03Ch
EIM_CS3RCR2 is 63FD_A000h base + 54h offset = 63FD_A054h



EIM_CS*n*RCR2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15 APR	Asynchronous Page Read. This bit field determine the asynchronous read mode to the external device. When APR=0, the async. read access is done as single word (where word is defined by the DSZ field).

Table continues on the next page...

EIM_CS*n*RCR2 field descriptions (continued)

Field	Description
	when APR=1, the async. read access executed as page read. page size is according to BL field config., RCSN,RBEN,OEN and RADVN are being ignored. APR is cleared by a hardware reset for EIM_CS1GCR1 - EIM_CS5GCR1. **NOTE:** SRD=0 and MUM=0 must apply when APR=1
14–12 PAT	Page Access Time. This bit field is used in Asynchronous Page Read mode only (APR=1). the initial access is set by RWSC as in regular asynchronous mode. the consecutive address assertions width determine by PAT field according to the settings shown below. when APR=0 this field is ignored. PAT is cleared by a hardware reset for EIM_CS1GCR1 - EIM_CS5GCR1. 000 Address width is 2 EIM clock cycles 001 Address width is 3 EIM clock cycles 010 Address width is 4 EIM clock cycles 011 Address width is 5 EIM clock cycles 100 Address width is 6 EIM clock cycles 101 Address width is 7 EIM clock cycles 110 Address width is 8 EIM clock cycles 111 Address width is 9 EIM clock cycles
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–8 RL	Read Latency. This bit field indicates cycle latency when executing a synchronous read operation. The fields holds the feedback clock loop delay in aclk cycle units. This field is cleared by a hardware reset. 00 Feedback clock loop delay is up to 1 cycle for BCD = 0 or 1.5 cycles for BCD != 0 01 Feedback clock loop delay is up to 2 cycles for BCD = 0 or 2.5 cycles for BCD != 0 10 Feedback clock loop delay is up to 3 cycles for BCD = 0 or 3.5 cycles for BCD != 0 11 Feedback clock loop delay is up to 4 cycles for BCD = 0 or 4.5 cycles for BCD != 0
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–4 RBEA	Read BE Assertion. This bit field determines when BE signal is asserted during read cycles (synchronous or asynchronous mode), according to the settings shown below. RBEA is cleared by a hardware reset. Example settings: 000 0 EIM clock cycles between beginning of read access and BE assertion 001 1 EIM clock cycles between beginning of read access and BE assertion 010 2 EIM clock cycles between beginning of read access and BE assertion 111 7 EIM clock cycles between beginning of read access and BE assertion
3 RBE	Read BE enable. This bit field determines if BE will be asserted during read access. 0 - BE are disabled during read access. 1- BE are enable during read access according to value of RBEA & RBEN bit fields.
2–0 RBEN	Read BE Negation. This bit field determines when BE signal is negated during read cycles in asynchronous single mode only (SRD=0 & APR=0), according to the settings shown below. This bit field is ignored when SRD=1. RBEN is cleared by a hardware reset. Example settings:

Table continues on the next page...

EIM_CS*n*RCR2 field descriptions (continued)

Field	Description
	000 0 EIM clock cycles between end of read access and BE negation 001 1 EIM clock cycles between end of read access and BE negation 010 2 EIM clock cycles between end of read access and BE negation 111 7 EIM clock cycles between end of read access and BE negation

25.101.5 Chip Select n Write Configuration Register 1 (EIM_CSWCR1)

Addresses: EIM_CS0WCR1 is 63FD_A000h base + 10h offset = 63FD_A010h
EIM_CS1WCR1 is 63FD_A000h base + 28h offset = 63FD_A028h
EIM_CS2WCR1 is 63FD_A000h base + 40h offset = 63FD_A040h
EIM_CS3WCR1 is 63FD_A000h base + 58h offset = 63FD_A058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WAL	WBED	WWSC						WADVA			WADVN			WBEA		WBEN			WEA			WEN			WCSA			WCSN			
Reset	0	0	n*	n*	n*	n*	n*	n*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

* Notes:
- WWSC bitfield: See field descriptions for the reset values.

EIM_CS*n*WCR1 field descriptions

Field	Description
31 WAL	Write ADV Low. This bit field determine ADV signal negation time in write accesses. When WAL=1, WADVN bit field is ignored and ADV signal will stay asserted until end of access. When WAL=0 negation of ADV signal is according to WADVN bit field configuration.
30 WBED	Write Byte Enable Disable. When asserted this bit prevent from IPP_DO_BE_B[x] to be asserted during write accesses.This bit is cleared by hardware reset.
29–24 WWSC	Write Wait State Control. This bit field programs the number of wait-states, according to the settings shown below, for synchronous or asynchronous write access to the external device connected to the chip select. When SWR=1 and WFL=0, WWSC indicates the number of burst clock (BCLK) cycles from the start of an access, before the memory can sample the first data.Since WAIT signal can be asserted one cycle before the first data can be sampled, the controller starts evaluating the WAIT signal state one cycle before, this is referred as handshake mode or variable latency mode. When SWR=1 and WFL=1, WWSC indicates the number of burst clock (BCLK) cycles from the start of an access, until the external device is ready for data transfer, this is referred as fix latency mode. When SWR=0, WFL bit is ignored, WWSC indicates the asynchronous access length and the number of EIM clock cycles from the start of access until the external device is ready for data transfer. WWSC is cleared by a hardware reset. **NOTE:** The reset value for EIM_CS0WCR1, WWSC[5:0] = 0b011100. For EIM_CS1WCR1 - EIM_CS5WCR1, the reset value of this field is 0b000000.

Table continues on the next page...

EIM_CS*n*WCR1 field descriptions (continued)

Field	Description
	Example settings: 000000 Reserved 000001 WWSC value is 1 000010 WWSC value is 2 000011 WWSC value is 3 111111 WWSC value is 63
23–21 WADVA	ADV Assertion. This bit field determines when ADV signal is asserted for synchronous or asynchronous write modes according to the settings shown below. WADVA is cleared by a hardware reset. Example settings: 000 0 EIM clock cycles between beginning of access and ADV assertion 001 1 EIM clock cycles between beginning of access and ADV assertion 010 2 EIM clock cycles between beginning of access and ADV assertion 111 7 EIM clock cycles between beginning of access and ADV assertion
20–18 WADVN	ADV Negation. This bit field determines when ADV signal to memory is negated during write accesses. When SWR=1 (synchronous write mode), ADV negation occurs according to the following formula: (WADVN + WADVA + BCD + BCS + 1) EIM clock cycles. When asynchronous read mode is applied (SWR=0) ADV negation occurs according to the following formula: (WADVN + WADVA + 1) EIM clock cycles. **NOTE:** Reset value for EIM_CS0WCR for WADVN is 2. For EIM_CS1WCR - EIM_CS5WCR reset value is 000. **NOTE:** This field should be configured so ADV negation will occur before the end of access. For ADV negation at the same time as the end of access, S/W should set the WAL bit.
17–15 WBEA	BE Assertion. This bit field determines when BE signal is asserted during write cycles in async. mode only (SWR=0), according to the settings shown below. BEA is cleared by a hardware reset. **NOTE:** Reset value for EIM_CS0WCR for WBEA is 2. For EIM_CS1WCR - EIM_CS5WCR reset value is 000. Example settings: 000 0 EIM clock cycles between beginning of access and BE assertion 001 1 EIM clock cycles between beginning of access and BE assertion 010 2 EIM clock cycles between beginning of access and BE assertion 111 7 EIM clock cycles between beginning of access and BE assertion
14–12 WBEN	BE[3:0] Negation. This bit field determines when BE[3:0] bus signal is negated during write cycles in async. mode only (SWR=0), according to the settings shown below. This bit field is ignored when SWR=1. BEN is cleared by a hardware reset. **NOTE:** Reset value for EIM_CS0WCR for WBEN is 2. For EIM_CS1WCR - EIM_CS5WCR reset value is 000. Example settings: 000 0 EIM clock cycles between end of access and WE negation 001 1 EIM clock cycles between end of access and WE negation 010 2 EIM clock cycles between end of access and WE negation 111 7 EIM clock cycles between end of access and WE negation

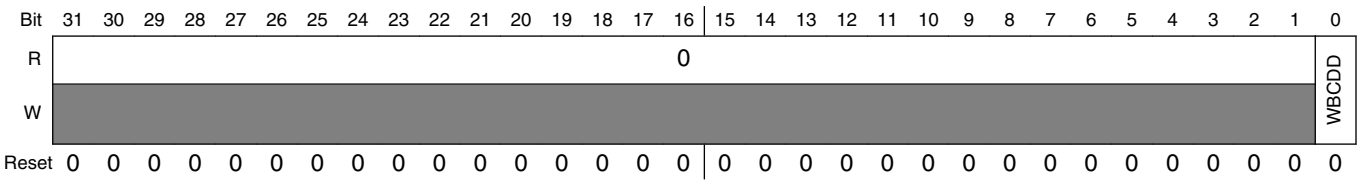
Table continues on the next page...

EIM_CS*n*WCR1 field descriptions (continued)

Field	Description
11–9 WEA	WE Assertion. This bit field determines when WE signal is asserted during write cycles (synchronous or asynchronous mode), according to the settings shown below. This bit field is ignored when executing a read access to the external device. WEA is cleared by a hardware reset. **NOTE:** Reset value for EIM_CS0WCR for WEA is 2. For EIM_CS1WCR - EIM_CS5WCR reset value is 000. Example settings: 000 0 EIM clock cycles between beginning of access and WE assertion 001 1 EIM clock cycles between beginning of access and WE assertion 010 2 EIM clock cycles between beginning of access and WE assertion 111 7 EIMclock cycles between beginning of access and WE assertion
8–6 WEN	WE Negation. This bit field determines when WE signal is negated during write cycles in asynchronous mode only (SWR=0), according to the settings shown below. This bit field is ignored when SWR=1. WEN is cleared by a hardware reset. **NOTE:** Reset value for EIM_CS0WCR for WEN is 2. For EIM_CS1WCR - EIM_CS5WCR reset value is 000. Example settings: 000 0 EIM clock cycles between beginning of access and WE assertion 001 1 EIM clock cycles between beginning of access and WE assertion 010 2 EIM clock cycles between beginning of access and WE assertion 111 7 EIM clock cycles between beginning of access and WE assertion
5–3 WCSA	Write CS Assertion. This bit field determines when CS signal is asserted during write cycles (synchronous or asynchronous mode), according to the settings shown below.this bit field is ignored when executing a read access to the external device. WCSA is cleared by a hardware reset. Example settings: 000 0 EIM clock cycles between beginning of write access and CS assertion 001 1 EIM clock cycles between beginning of write access and CS assertion 010 2 EIM clock cycles between beginning of write access and CS assertion 111 7 EIMclock cycles between beginning of write access and CS assertion
2–0 WCSN	Write CS Negation. This bit field determines when CS signal is negated during write cycles in asynchronous mode only (SWR=0), according to the settings shown below. This bit field is ignored when SWR=1. WCSN is cleared by a hardware reset. Example settings: 000 0 EIM clock cycles between end of read access and CS negation 001 1 EIM clock cycles between end of read access and CS negation 010 2 EIM clock cycles between end of read access and CS negation 111 7 EIM clock cycles between end of read access and CS negation

25.101.6 Chip Select n Write Configuration Register 2 (EIM_CSWCR2)

Addresses: EIM_CS0WCR2 is 63FD_A000h base + 14h offset = 63FD_A014h
EIM_CS1WCR2 is 63FD_A000h base + 2Ch offset = 63FD_A02Ch
EIM_CS2WCR2 is 63FD_A000h base + 44h offset = 63FD_A044h
EIM_CS3WCR2 is 63FD_A000h base + 5Ch offset = 63FD_A05Ch

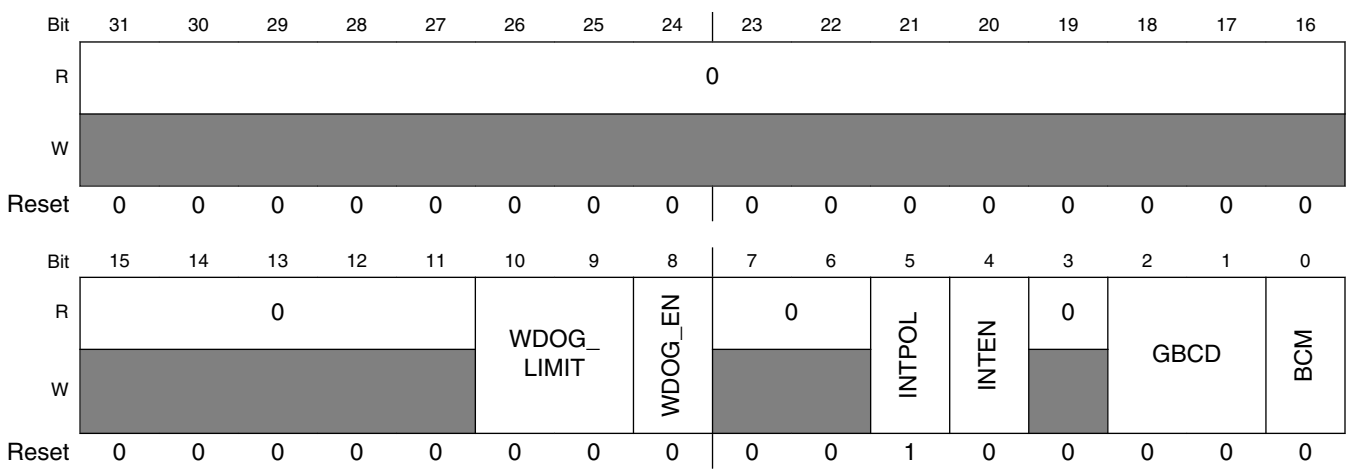


EIM_CS*n*WCR2 field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 WBCDD	Write Burst Clock Divisor Decrement. If this bit is asserted and BCD value is 0 sync. write access will be preformed as if BCD value is 1.When this bit is negated or BCD value is not 0 this bit has no affect. This bit is cleared by hardware reset.

25.101.7 EIM Configuration Register (EIM_WCR)

Address: EIM_WCR is 63FD_A000h base + 90h offset = 63FD_A090h



EIM_WCR field descriptions

Field	Description
31–11 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

EIM_WCR field descriptions (continued)

Field	Description
10–9 WDOG_LIMIT	Memory Watch Dog (WDog) cycle limit. This bit field determines the number of BCLK cycles (ACLK cycles in dtack mode) before the wdog counter terminates the access and send an error response to the master. 00 128 BCLK cycles 01 256 BCLK cycles 10 512 BCLK cycles 11 1024 BCLK cycles
8 WDOG_EN	Memory WDog enable. This bit controls the operation of the wdog counter that terminates the EIM access. 0 Memory WDog is Disabled 1 Memory WDog is Enabled
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5 INTPOL	Interrupt Polarity. This bit field determines the polarity of the external device interrupt. 0 External interrupt polarity is active low 1 External interrupt polarity is active high
4 INTEN	Interrupt Enable. When this bit is set the External signal RDY_INT as active interrupt. When interrupt occurs, INT bit at the WCR will be set and t EIM_EXT_INT signal will be asserted correspondingly. This bit is cleared by a hardware reset. 0 External interrupt Disable 1 External interrupt Enable
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–1 GBCD	General Burst Clock Divisor. When BCM bit is set, this bit field contains the value used to program the burst clock divisor for Continuos BCLK generation. The other BCD bit fields for each chip select are ignored. It is used to divide the internal AXI bus frequency. When BCM=0 GBCD bit field has no influence. GBCD is cleared by a hardware reset. 00 Divide EIM clock by 1 01 Divide EIM clock by 2 10 Divide EIM clock by 3 11 Divide EIM clock by 4
0 BCM	Burst Clock Mode. This bit selects the burst clock mode of operation. It is used for system debug mode. BCM is cleared by a hardware reset. **NOTE:** The BCLK frequency in this mode is according to GBCD bit field. **NOTE:** The BCLK phase is opposite to the EIM clock in this mode if GBCD is 0. **NOTE:** This bit should be used only in async. accesses. No sync access can be executed if this bit is set. **NOTE:** When this bit is set bcd field shouldn't be configured to 0.

Table continues on the next page...

EIM_WCR field descriptions (continued)

Field	Description
	0 The burst clock runs only when accessing a chip select range with the SWR/SRD bits set. When the burst clock is not running it remains in a logic 0 state. When the burst clock is running it is configured by the BCD and BCS bit fields in the chip select Configuration Register. 1 The burst clock runs whenever ACLK is active (independent of chip select configuration)

25.101.8 EIM IP Access Register (EIM_WIAR)

Address: EIM_WIAR is 63FD_A000h base + 94h offset = 63FD_A094h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											ACLK_EN	ERRST	INT	IPS_ACK	IPS_REQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

EIM_WIAR field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 ACLK_EN	ACLK enable. This bit gates the ACLK for the EIM except from FFs that get ipg_aclk_s. After reset ACLK is enabled. 0 ACLK is disabled 1 ACLK is enabled
3 ERRST	READY After Reset. This bit controls the initial ready/busy status for external devices on CS0 immediately after hardware reset. This is a sticky bit which is cleared once the RDY_INT signal is asserted by the external device. When ERRST = 1 the first fetch access from EIM to the external device located on CS0 will be pending until RDY_INT signal indicates that the external device is ready, then EIM will execute the access. 0 RDY_INT After Reset Disable 1 RDY_INT After Reset Enable
2 INT	Interrupt. This bit indicates interrupt assertion by an external device according to RDY_INT signal. When polling this bit, INT=0 indicates interrupt not occurred and INT=1 indicates assertion of the external device interrupt. This bit is cleared by a hardware reset.

Table continues on the next page...

EIM_WIAR field descriptions (continued)

Field	Description
1 IPS_ACK	IPS ACK. The EIM is ready for ips access. There is no active AXI access and no new AXI access is accepted till this bit is cleared. This bit is cleared by the master after it completes the ips accesses. 0 Master cannot access ips. 1 Master can access ips.
0 IPS_REQ	IPS request. The Master requests to access one of the IPS registers.Dutring such access the EIM should not preform any AXI/memory accesses. The EIM finishes the AXI accesses that already starts and asserts the IPS_ACK bit. 0 No Master requests ips access 1 Master requests ips access

25.101.9 Error Address Register (EIM_EAR)

Address: EIM_EAR is 63FD_A000h base + 98h offset = 63FD_A098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Error-ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EIM_EAR field descriptions

Field	Description
31–0 Error-ADDR	Error Address. This bit field holds the AXI address of the last access that caused error. This register is read only register.

Chapter 26
Enhanced Periodic Interrupt Timer (EPIT)

26.1 Overview

The EPIT is a 32-bit set-and-forget timer that begins counting after the EPIT is enabled by software. It is capable of providing precise interrupts at regular intervals with minimal processor intervention.

The following figure illustrates the block diagram of the EPIT.

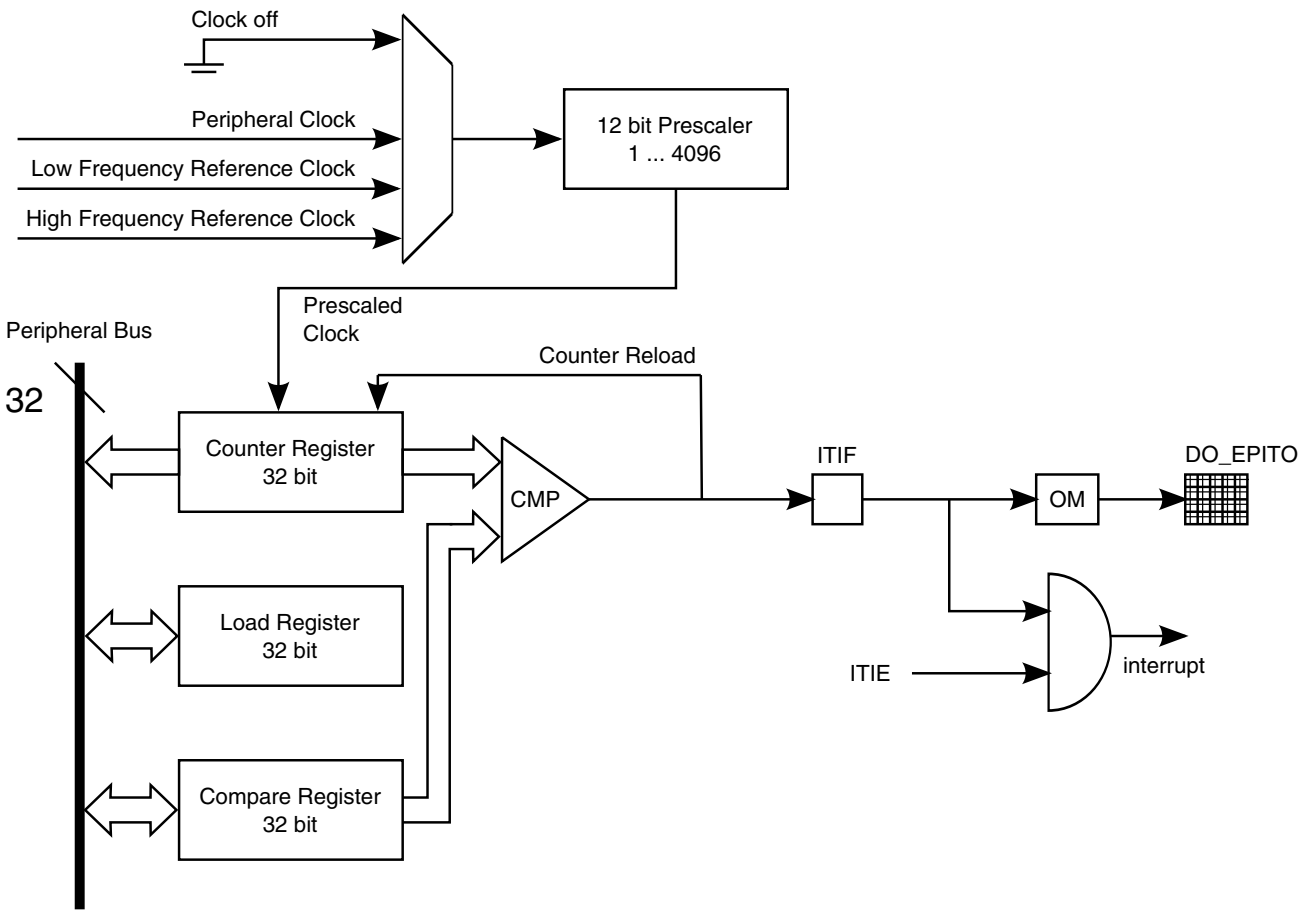


Figure 26-1. EPIT Block Diagram

26.1.1 Features

Key features of the EPIT include:

- 32-bit down counter with clock source selection
- 12-bit prescaler for division of input clock frequency
- Counter value can be programmed on the fly

- Can be programmed to be active during low-power and debug modes
- Interrupt generation when counter reaches the Compare value

26.1.2 Modes and Operations

EPIT supports the modes described in the indicated sections:

- Operating Modes
  - Set-and-Forget Mode
  - Free-Running Mode

EPIT supports the operations as described in Operations.

26.2 External Signals

The following table describes all EPIT's I/O Signals.

Table 26-1. Block I/O Signals

Name	I/O	Description	Reset State	Pull Up
DO_EPITO	O	Output pin at chip boundary for indication of occurrence of output compare event through a specified transition.	0	-

26.3 Functional Description

This section provides a complete functional description of the block.

26.3.1 Operating Modes

This section describes all functional operation modes of the block.

The EPIT can be programmed to function in set-and-forget or free-running modes.

26.3.1.1 Set-and-Forget Mode

This mode of operation is selected when the RLD bit in control register (EPIT_EPITCR) is set to a value of one.

The counter cannot be directly written from the block data bus. Instead, it gets its data from the load register (EPIT_EPITLR). Whenever the counter reaches a count of zero, the value in EPIT_EPITLR is loaded into the counter to be decremented toward zero. The counter may by directly initialized, without having to wait for the count to reach zero, when the EPIT_EPITLR is written with the IOVW bit in EPIT_EPITCR set.

26.3.1.2 Free-Running Mode

This mode of operation is selected when the RLD bit is cleared.

In this mode, the counter rolls over from 0x0000_0000 to 0xFFFF_FFFF without reloading from the modulus register and continues to count down. In this mode when writing to EPIT_EPITLR with the required initialization value after having set the IOVW bit, the counter can also be directly initialized.

26.3.2 Operations

The EPIT has a single 32-bit down counter which starts counting when the block is enabled by software.

The start value of the counter is loaded from the EPIT load register which can be written to at any time by the processor. The value in the compare register determines the time of occurrence of the interrupt.

When EPIT is disabled (EN=0), both the main counter and the prescaler counter freeze their count at current count values. ENMOD bit is a r/w bit which decides the counter value when the EPIT is enabled again by setting EN bit. If ENMOD bit is set, then the main counter is loaded with the load value (If RLD=1)/ 0xFFFF_FFFF (If RLD=0) and the prescaler counter is reset (0x000), when EPIT is enabled (EN=1). If ENMOD is programmed to 0 then both main counter and prescaler counter restart counting from their frozen values, when EPIT is enabled (EN=1). If EPIT is programmed to be disabled in a low-power mode (STOP/WAIT), then both the main counter and the prescaler counter freeze at their current count values when EPIT enters low-power mode. When EPIT exits the low-power mode, both the main counter and the prescaler counter start counting from their frozen values irrespective of the ENMOD bit.

A hardware reset resets all EPIT registers to their respective reset values. There is a software reset which has the same effect on all registers except for the EN, ENMOD, STOPEN and WAITEN bits in the control register. The state of these bits are not affected by software reset. A software reset can be asserted even when the EPIT is disabled.

26.3.3 Clocks

The clock that feeds the prescaler can be selected from among the following sources:

- **High-frequency reference clock**

  This clock is provided by the Clock Control Module (CCM). This clock remains on during low-power mode when the peripheral clock is turned off, allowing EPIT to use this clock in low-power mode. In normal mode, the CCM synchronizes this clock to ahb_clk; in low-power mode, CCM switches to an unsynchronized version.

- **Low-frequency reference clock**

  This 32 kHz reference clock is provided by the CCM. This clock remains on in low-power mode when the peripheral clock is turned off, so EPIT can use this clock during low-power mode. In normal mode, the CCM synchronizes this clock to ahb_clk; in low-power mode, CCM switches to an unsynchronized version. This clock is derived from the external 32kHz crystal.

- **Peripheral clock**

  This is the peripheral clock (PER Clock) which is provided (and optionally gated) by the CCM. This clock is typically used in normal operations. In low-power modes, if the EPIT is programmed to be disabled (via STOPEN or WAITEN), then the peripheral clock can be switched off.

The clock input source is determined by the CLKSRC field in the control register. The clock input to the prescaler can also be disabled by setting CLKSRC to 0b00. **This field value should only be changed after first disabling the EPIT by clearing the EN bit in the EPIT_EPITCR**. For other programming requirements that apply while changing clock source, refer section Change of Clock Source.

The PRESCALER field in the control register is used to select the divide ratio of the input clock that drives the main counter. The prescaler can divide the input clock by a value between 1 and 4096. A change in the value of the PRESCALER field is immediately reflected on its output clock frequency. The following figure shows the timing for a change in the prescaler value.

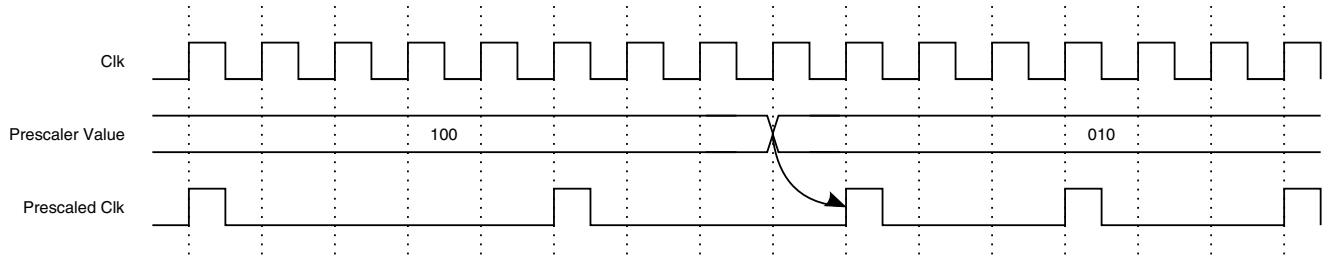


Figure 26-2. Prescaler Value Change Diagram

26.3.4 Compare Event

When the programmed value of EPIT_EPITCMPR matches the value in EPIT_EPITCNR a compare status flag is set, and an interrupt is generated if the OCIEN bit is set in the control register.

The compare output pin is set, cleared, toggled, or not affected at all depending on the setting of the output mode (OM) bits in the control register. If an interrupt is required at rollover (when the counter value reaches 0x0000_0000 and the new value is loaded) then the compare register value should be set equal to the load register value in set-and-forget mode, or equal to 0xFFFF_FFFF in free-running mode.

The following figure shows the timing for a compare event and interrupt.

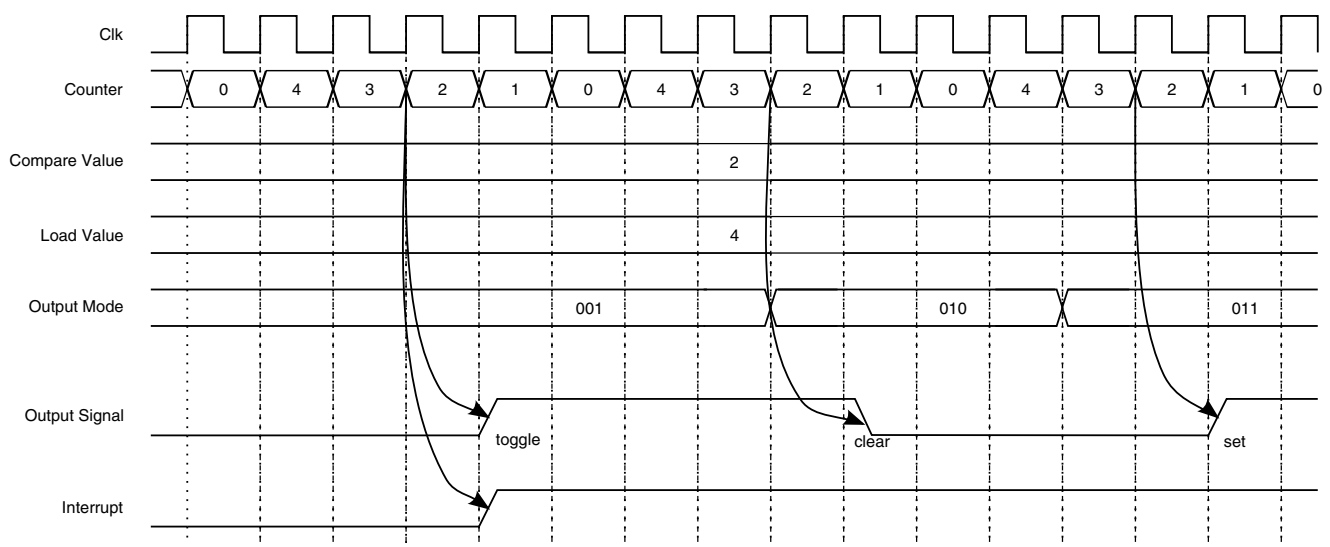


Figure 26-3. Compare Event and Interrupt Timing Diagram

26.3.4.1 Counter Value Overwrite

The EPIT counter value can be overwritten to acquire a desired value at any point of time. The procedure for this is to set the IOVW bit in the control register and then write the desired value into the load register.

This results in the load register acquiring that value and also the counter being overwritten with it. If the EPIT is running the counter resumes counting from the overwritten value.

26.3.4.2 Low-Power Mode Behavior

The EPIT timer's behavior in low-power modes depends on which clock source is being used.

If the selected clock source is available and the corresponding low-power enable bit is set, then the EPIT continues to function in the low-power mode. If the EPIT is programmed to be disabled in a low-power mode (STOP/WAIT), then main counter and the prescaler counter freeze at the current count values when the EPIT enters low-power mode. When the EPIT exits the low-power mode, both main counter and prescaler counter start counting from their frozen values irrespective of the ENMOD bit.

26.3.4.3 Debug Mode Behavior

In debug mode, the user has the option to run or halt the EPIT timers. If the DBGEN bit is reset in the EPIT Control Register, the timer is halted.

When debug mode is exited, the timer operation reverts to what it was prior to entering debug mode.

26.4 Initialization/ Application Information

26.4.1 Change of Clock Source

The CLKSRC field in EPIT_EPITCR determines the clock source. This field value should be changed only after disabling the EPIT (EN = 0).

Below is the software sequence which must be followed while changing clock source.

1. Disable the EPIT by setting EN=0 in EPIT_EPITCR.
2. Program OM=00 in the EPIT_EPITCR.

3. Disable the EPIT interrupts.
4. Program CLKSRC to desired clock source in EPIT_EPITCR.
5. Clear the EPIT status register (EPIT_EPITSR), that is, write "1" to clear (w1c).
6. Enable the EPIT interrupts.
7. Set ENMOD= 1 in the EPIT_EPITCR, to bring the EPIT Counter to defined state (EPIT_EPITLR value or 0xFFFF_FFFF).
8. Enable EPIT (EN=1) in the EPIT_EPITCR

26.5 Programmable Registers

The EPIT includes five user-accessible 32-bit registers. The following table summarizes these registers and their addresses.

Peripheral bus write access to the EPIT control register (EPITCR) and the EPIT load register (EPITLR) results in one cycle of wait state, while other valid peripheral bus accesses are with 0 wait state.

EPIT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_C000	Control register (EPIT-1_EPITCR)	32	R/W	0000_0000h	26.5.1/ 1141
53FA_C004	Status register (EPIT-1_EPITSR)	32	R/W	0000_0000h	26.5.2/ 1143
53FA_C008	Load register (EPIT-1_EPITLR)	32	R/W	FFFF_ FFFFh	26.5.3/ 1144
53FA_C00C	Compare register (EPIT-1_EPITCMPR)	32	R/W	0000_0000h	26.5.4/ 1144
53FA_C010	Counter register (EPIT-1_EPITCNR)	32	R	FFFF_ FFFFh	26.5.5/ 1145
53FB_0000	Control register (EPIT-2_EPITCR)	32	R/W	0000_0000h	26.5.1/ 1141
53FB_0004	Status register (EPIT-2_EPITSR)	32	R/W	0000_0000h	26.5.2/ 1143
53FB_0008	Load register (EPIT-2_EPITLR)	32	R/W	FFFF_ FFFFh	26.5.3/ 1144
53FB_000C	Compare register (EPIT-2_EPITCMPR)	32	R/W	0000_0000h	26.5.4/ 1144
53FB_0010	Counter register (EPIT-2_EPITCNR)	32	R	FFFF_ FFFFh	26.5.5/ 1145

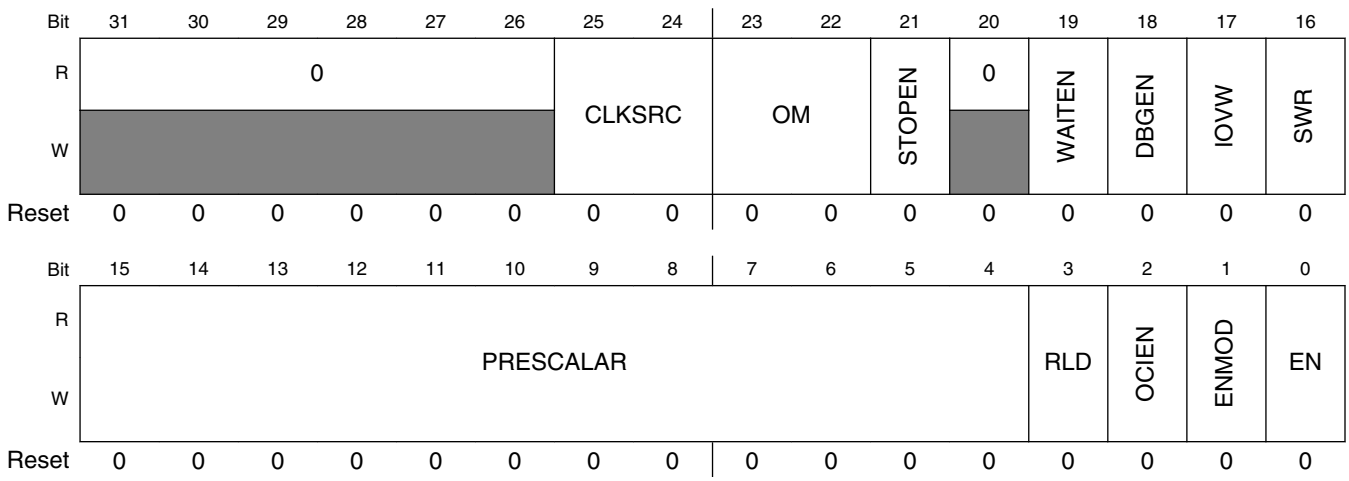
26.5.1 Control register (EPITx_EPITCR)

The EPIT control register (EPIT_EPITCR) is used to configure the operating settings of the EPIT. It contains the clock division prescaler value and also the interrupt enable bit. Additionally it contains other control bit which are outlined below.

Peripheral Bus Write access to EPIT Control Register (EPIT_EPITCR) results in one cycle of the wait state, while other valid peripheral bus accesses are with 0 wait state.

Addresses: EPIT-1_EPITCR is 53FA_C000h base + 0h offset = 53FA_C000h

EPIT-2_EPITCR is 53FB_0000h base + 0h offset = 53FB_0000h



EPITx_EPITCR field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value zero. Reserved. Writing to these bits does not affect the functionality of EPIT. These bits are always read as zero.
25–24 CLKSRC	Select clock source These bits determine which clock input is to be selected for running the counter. This field value should only be changed when the EPIT is disabled by clearing the EN bit in this register. For other programming requirements while changing clock source, refer to Change of Clock Source. 00 Clock is off 01 Peripheral clock 10 High-frequency reference clock 11 Low-frequency reference clock
23–22 OM	EPIT output mode.This bit field determines the mode of EPIT output on the output pin. 00 EPIT output is disconnected from pad 01 Toggle output pin 10 Clear output pin 11 Set output pin

Table continues on the next page...

EPITx_EPITCR field descriptions (continued)

Field	Description
21 STOPEN	EPIT stop mode enable. This read/write control bit enables the operation of the EPIT during stop mode. This bit is reset by a hardware reset and unaffected by software reset. 0 EPIT is disabled in stop mode 1 EPIT is enabled in stop mode
20 Reserved	This read-only field is reserved and always has the value zero. Reserved. Writing to these bits does not affect the functionality of EPIT. These bits are always read as zero.
19 WAITEN	This read/write control bit enables the operation of the EPIT during wait mode. This bit is reset by a hardware reset. A software reset does not affect this bit. 0 EPIT is disabled in wait mode 1 EPIT is enabled in wait mode
18 DBGEN	This bit is used to keep the EPIT functional in debug mode. When this bit is cleared, the input clock is gated off in debug mode.This bit is reset by hardware reset. A software reset does not affect this bit. 0 Inactive in debug mode 1 Active in debug mode
17 IOVW	EPIT counter overwrite enable. This bit controls the counter data when the modulus register is written. When this bit is set, all writes to the load register overwrites the counter contents and the counter starts subsequently counting down from the programmed value. 0 Write to load register does not result in counter value being overwritten. 1 Write to load register results in immediate overwriting of counter value.
16 SWR	Software reset. The EPIT is reset when this bit is set to 1. It is a self clearing bit. This bit is set when the block is in reset state and is cleared when the reset procedure is over. Setting this bit resets all the registers to their reset values, except for the EN, ENMOD, STOPEN, WAITEN and DBGEN bits in this control register 0 EPIT is out of reset 1 EPIT is undergoing reset
15–4 PRESCALAR	Counter clock prescaler value. This bit field determines the prescaler value by which the clock is divided before it goes to the counter 0x000 Divide by 1 0x001 Divide by 2 0xFFFF Divide by 4096
3 RLD	Counter reload control. This bit is cleared by hardware reset. It decides the counter functionality, whether to run in free-running mode or set-and-forget mode. 0 When the counter reaches zero it rolls over to 0xFFFF_FFFF (free-running mode) 1 When the counter reaches zero it reloads from the modulus register (set-and-forget mode)
2 OCIEN	Output compare interrupt enable. This bit enables the generation of interrupt on occurrence of compare event. 0 Compare interrupt disabled 1 Compare interrupt enabled

Table continues on the next page...

EPITx_EPITCR field descriptions (continued)

Field	Description
1 ENMOD	EPIT enable mode. When EPIT is disabled (EN=0), both main counter and prescaler counter freeze their count at current count values. ENMOD bit is a r/w bit that determines the counter value when the EPIT is enabled again by setting EN bit. If ENMOD bit is set, then main counter is loaded with the load value (If RLD=1)/ 0xFFFF_FFFF (If RLD=0) and prescaler counter is reset, when EPIT is enabled (EN=1). If ENMOD is programmed to 0 then both main counter and prescaler counter restart counting from their frozen values when EPIT is enabled (EN=1). If EPIT is programmed to be disabled in a low-power mode (STOP/WAIT/ DEBUG), then both the main counter and the prescaler counter freeze at their current count values when EPIT enters low-power mode. When EPIT exits the low-power mode, both main counter and prescaler counter start counting from their frozen values irrespective of the ENMOD bit. This bit is reset by a hardware reset. A software reset does not affect this bit. 0 Counter starts counting from the value it had when it was disabled. 1 Counter starts count from load value (RLD=1) or 0xFFFF_FFFF (If RLD=0)
0 EN	This bit enables the EPIT. EPIT counter and prescaler value when EPIT is enabled (EN =1), is dependent upon ENMOD and RLD bit as described for ENMOD bit. It is recommended that all registers be properly programmed before setting this bit. This bit is reset by a hardware reset. A software reset does not affect this bit. 0 EPIT is disabled 1 EPIT is enabled

26.5.2 Status register (EPITx_EPITSR)

The EPIT status register (EPIT_EPITSR) has a single status bit for the output compare event. The bit is a write 1 to clear bit.

Addresses: EPIT-1_EPITSR is 53FA_C000h base + 4h offset = 53FA_C004h

EPIT-2_EPITSR is 53FB_0000h base + 4h offset = 53FB_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															OCIF
W																w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPITx_EPITSR field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved. Writing to these bits does not affect the functionality of EPIT. These bits are always read as zero.

Table continues on the next page...

EPITx_EPITSR field descriptions (continued)

Field	Description
0 OCIF	Output compare interrupt flag. This bit is the interrupt flag that is set when the content of counter equals the content of the compare register (EPIT_EPITCMPR). The bit is a write 1 to clear bit. 0 Compare event hasn't occurred 1 Compare event occurred

26.5.3 Load register (EPITx_EPITLR)

The EPIT load register (EPIT_EPITLR) contains the value that is to be loaded into the counter when EPIT counter reaches zero if the RLD bit in EPIT_EPITCR is set. If the IOVW bit in the EPIT_EPITCR is set then a write to this register overwrites the value of the EPIT counter register in addition to updating this registers value. This overwrite feature is active even if the RLD bit is not set.

Addresses: EPIT-1_EPITLR is 53FA_C000h base + 8h offset = 53FA_C008h

EPIT-2_EPITLR is 53FB_0000h base + 8h offset = 53FB_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LOAD																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

EPITx_EPITLR field descriptions

Field	Description
31–0 LOAD	Load value. Value that is loaded into the counter at the start of each count cycle.

26.5.4 Compare register (EPITx_EPITCMPR)

The EPIT compare register (EPIT_EPITCMPR) holds the value that determines when a compare event is generated.

Addresses: EPIT-1_EPITCMPR is 53FA_C000h base + Ch offset = 53FA_C00Ch

EPIT-2_EPITCMPR is 53FB_0000h base + Ch offset = 53FB_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMPARE																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

EPITx_EPITCMPR field descriptions

Field	Description
31–0 COMPARE	Compare Value. When the counter value equals this bit field value a compare event is generated.

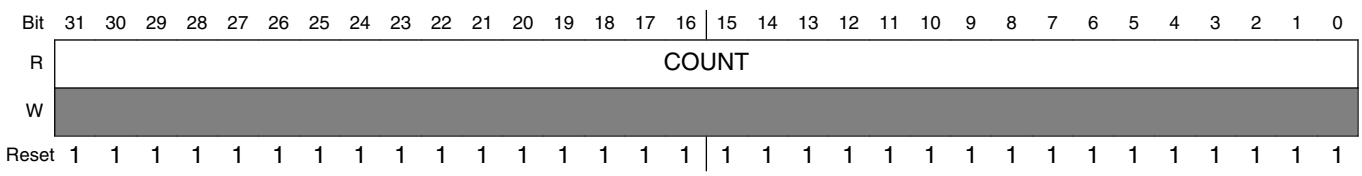
26.5.5 Counter register (EPITx_EPITCNR)

The EPIT counter register (EPIT_EPITCNR) contains the current count value and can be read at any time without disturbing the counter. This is a read-only register and any attempt to write into it generates a transfer error. But if the IOVW bit in EPIT_EPITCR is set, the value of this register can be overwritten with a write to EPIT_EPITLR. This change is reflected when this register is subsequently read.

.

Addresses: EPIT-1_EPITCNR is 53FA_C000h base + 10h offset = 53FA_C010h

EPIT-2_EPITCNR is 53FB_0000h base + 10h offset = 53FB_0010h



EPITx_EPITCNR field descriptions

Field	Description
31–0 COUNT	Counter value. This contains the current value of the counter.

Chapter 27
Enhanced Serial Audio Interface (ESAI)

27.1 Introduction

The Enhanced Serial Audio Interface (ESAI) provides a full-duplex serial port for serial communication with a variety of serial devices, including industry-standard codecs, Sony/Phillips Digital Interface (SPDIF) transceivers, and other DSPs.

The ESAI consists of independent transmitter and receiver sections, each section with its own clock generator. It is a superset of the 56300 Family ESSI peripheral and of the 56000 Family SAI peripheral.

The ESAI block diagram is shown in Figure 27-1. The ESAI is named synchronous because all serial transfers are synchronized to a clock. Additional synchronization signals are used to delineate the word frames. The normal mode of operation is used to transfer data at a periodic rate, one word per period. The network mode is similar in that it is also intended for periodic transfers; however, it supports up to 32 words (time slots) per period. This mode can be used to build time division multiplexed (TDM) networks. In contrast, the on-demand mode is intended for non-periodic transfers of data and to transfer data serially at high speed when the data becomes available.

27.1.1 Overview

The following figure shows the ESAI block diagram.

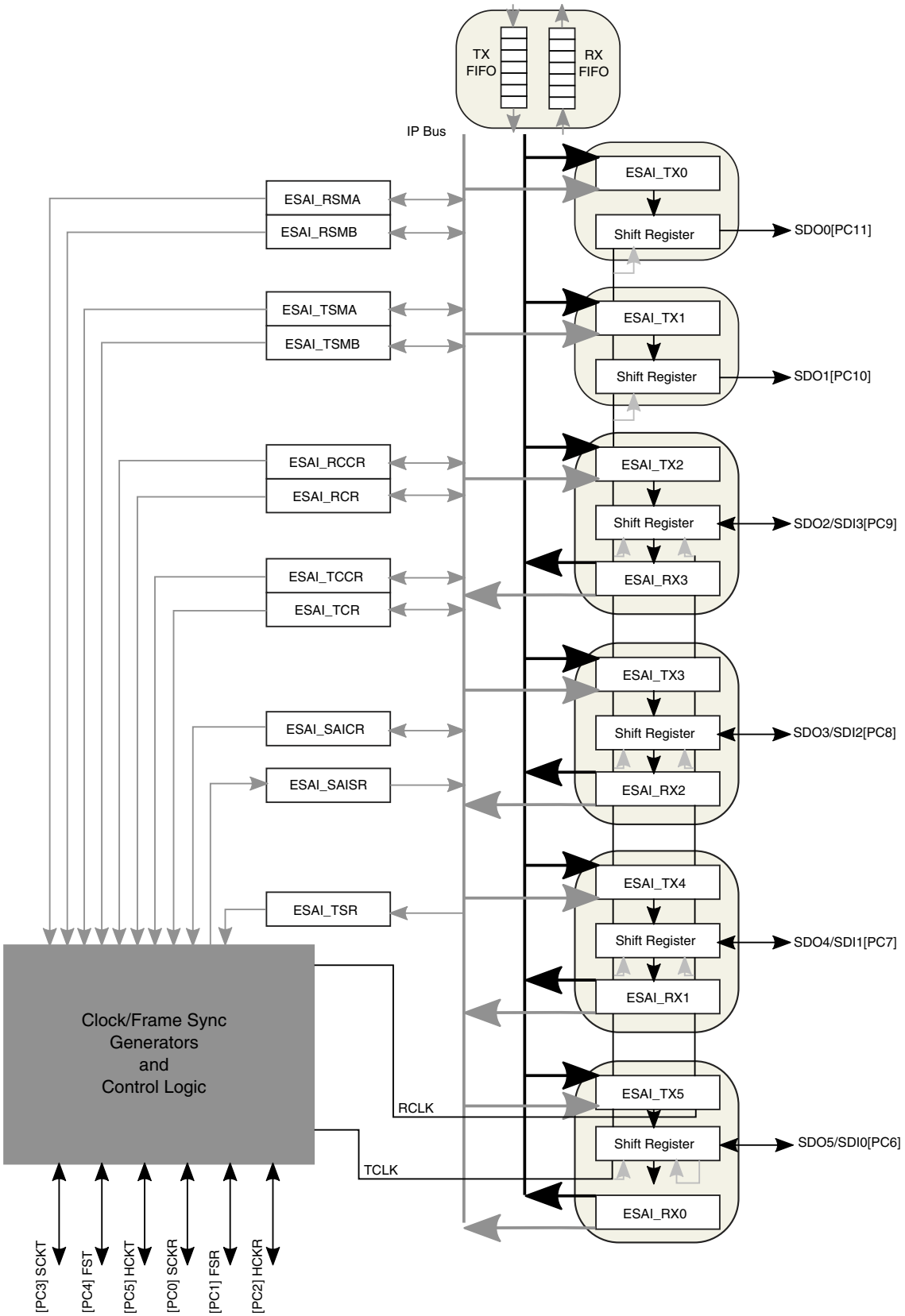


Figure 27-1. ESAI Block Diagram

27.1.2 Features

- Independent (asynchronous mode) or shared (synchronous mode) transmit and receive sections with separate or shared internal/external clocks and frame syncs, operating in Master or Slave mode.
- Up to six transmitters and four receivers with SDO2/SDI3, SDO3/SDI2, SDO4/SDI1 and SDO5/SDI0 pins shared by transmitters 2 to 5 and receivers 0 to 3. SDO0 and SDO1 pins are used by transmitters 0 and 1 only.
- Programmable data interface modes such as I2S, LSB aligned, MSB aligned
- Programmable word length (8, 12, 16, 20 or 24bits)
- Flexible selection between system clock or external oscillator as input clock source, programmable internal clock divider and frame sync generation
- AC97 support
- Time Slot Mask Registers for reduced ARM platform overhead (for both Transmit and Receive)
- 128-word Transmit FIFO shared by six transmitters
- 128-word Receive FIFO shared by four receivers

27.1.3 Modes of Operation

ESAI has three basic operating modes and many data/operation formats.

ESAI operating mode are selected by the ESAI control registers (ESAI_TCCR, ESAI_TCR, ESAI_RCCR, ESAI_RCR, and ESAI_SAICR). The main operating modes are described in the following paragraphs.

27.1.3.1 Normal/Network/On-Demand Mode Selection

Selecting between the normal mode and network mode is accomplished by clearing or setting the TMOD0-TMOD1 bits in the ESAI_TCR register for the transmitter section, as well as in the RMOD0-RMOD1 bits in the ESAI_RCR register for the receiver section.

For normal mode, the ESAI functions with one data word of I/O per frame (per enabled transmitter or receiver). The normal mode is typically used to transfer data to or from a single device.

For the network mode, 2 to 32 time slots per frame may be selected. During each frame, 0 to 32 data words of I/O may be received or transmitted. In either case, the transfers are periodic. The frame sync signal indicates the first time slot in the frame. Network mode is typically used in time division multiplexed (TDM) networks of codecs, DSPs with multiple words per frame, or multi-channel devices.

Selecting the network mode and setting the frame rate divider to zero (DC=00000) selects the on-demand mode. This special case does not generate a periodic frame sync. A frame sync pulse is generated only when data is available to transmit. The on-demand mode requires that the transmit frame sync be internal (output) and the receive frame sync be external (input). Therefore, for simplex operation, the synchronous mode could be used; however, for full-duplex operation, the asynchronous mode must be used. Data transmission that is data driven is enabled by writing data into each TX. Although the ESAI is double buffered, only one word can be written to each TX, even if the transmit shift register is empty. The receive and transmit interrupts function as usual using TDE and RDF; however, transmit underruns are impossible for on-demand transmission and are disabled.

27.1.3.2 Synchronous/Asynchronous Operating Modes

The transmit and receive sections of the ESAI may be synchronous or asynchronous, that is, the transmitter and receiver sections may use common clock and synchronization signals (synchronous operating mode), or they may have their own separate clock and sync signals (asynchronous operating mode).

The SYN bit in the ESAI_SAICR register selects synchronous or asynchronous operation. Because the ESAI is designed to operate either synchronously or asynchronously, separate receive and transmit interrupts are provided.

When SYN is cleared, the ESAI transmitter and receiver clocks and frame sync sources are independent. If SYN is set, the ESAI transmitter and receiver clocks and frame sync come from the transmitter section (either external or internal sources).

Data clock and frame sync signals can be generated internally by the ARM Core or may be obtained from external sources. If internally generated, the ESAI clock generator is used to derive high frequency clock, bit clock and frame sync signals from the ARM Core internal system clock.

27.1.3.3 Frame Sync Selection

The frame sync can be either a bit-long or word-long signal.

The transmitter frame format is defined by the TFSL bit in the ESAI_TCR register. The receiver frame format is defined by the RFSL bit in the ESAI_RCR register.

1. In the word-long frame sync format, the frame sync signal is asserted during the entire word data transfer period. This frame sync length is compatible with codecs, SPI serial peripherals, serial A/D and D/A converters, shift registers and telecommunication PCM serial I/O.
2. In the bit-long frame sync format, the frame sync signal is asserted for one bit clock immediately before the data transfer period. This frame sync length is compatible with Intel and National components, codecs and telecommunication PCM serial I/O.

The relative timing of the word length frame sync as referred to the data word is specified by the TFSR bit in the ESAI_TCR register for the transmitter section and by the RFSR bit in the ESAI_RCR register for the receive section. The word length frame sync may be generated (or expected) with the first bit of the data word, or with the last bit of the previous word. TFSR and RFSR are ignored when a bit length frame sync is selected.

Polarity of the frame sync signal may be defined as positive (asserted high) or negative (asserted low). The TFSP bit in the ESAI_TCCR register specifies the polarity of the frame sync for the transmitter section. The RFSP bit in the ESAI_RCCR register specifies the polarity of the frame sync for the receiver section.

The ESAI receiver looks for a receive frame sync leading edge (trailing edge if RFSP is set) only when the previous frame is completed. If the frame sync goes high before the frame is completed (or before the last bit of the frame is received in the case of a bit frame sync or a word length frame sync with RFSR set), the current frame sync is not recognized, and the receiver is internally disabled until the next frame sync. Frames do not have to be adjacent, that is, a new frame sync does not have to immediately follow the previous frame. Gaps of arbitrary periods can occur between frames. Enabled transmitters are tri-stated during these gaps.

When operating in the synchronous mode (SYN=1), all clocks including the frame sync are generated by the transmitter section.

27.1.3.4 Shift Direction Selection

Some data formats, such as those used by codecs, specify MSB first while other data formats, such as the AES-EBU digital audio interface, specify LSB first.

The MSB/LSB first selection is made by programming RSHFD bit in the ESAI_RCR register for the receiver section and by programming the TSHFD bit in the ESAI_TCR register for the transmitter section.

27.2 External Signal Description

Three to twelve pins are required for operation, depending on the operating mode selected and the number of transmitters and receivers enabled.

The SDO0 and SDO1 pins are used by transmitters 0 and 1 only. The SDO2/SDI3, SDO3/SDI2, SDO4/SDI1 and SDO5/SDI0 pins are shared by transmitters 2 to 5 with receivers 0 to 3. The actual mode of operation is selected under software control. All transmitters operate fully synchronized under control of the same transmitter clock signals. All receivers operate fully synchronized under control of the same receiver clock signals.

27.2.1 Serial Transmit 0 Data Pin (SDO0)

SDO0 is used for transmitting data from the ESAI_TX0 serial transmit shift register.

SDO0 is an output when data is being transmitted from the ESAI_TX0 shift register. In the on-demand mode with an internally generated bit clock, the SDO0 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO0 may be programmed as a disconnected pin (PC11) when the ESAI SDO0 function is not being used.

27.2.2 Serial Transmit 1 Data Pin (SDO1)

SDO1 is used for transmitting data from the ESAI_TX1 serial transmit shift register.

SDO1 is an output when data is being transmitted from the ESAI_TX1 shift register. In the on-demand mode with an internally generated bit clock, the SDO1 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO1 may be programmed as a disconnected pin (PC10) when the ESAI SDO1 function is not being used.

27.2.3 Serial Transmit 2/Receive 3 Data Pin (SDO2/SDI3)

SDO2/SDI3 is used as the SDO2 for transmitting data from the ESAI_TX2 serial transmit shift register when programmed as a transmitter pin, or as the SDI3 signal for receiving serial data to the ESAI_RX3 serial receive shift register when programmed as a receiver pin.

SDO2/SDI3 is an input when data is being received by the ESAI_RX3 shift register. SDO2/SDI3 is an output when data is being transmitted from the ESAI_TX2 shift register. In the on-demand mode with an internally generated bit clock, the SDO2/SDI3 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO2/SDI3 may be programmed as a disconnected pin (PC9) when the ESAI SDO2 and SDI3 functions are not being used.

27.2.4 Serial Transmit 3/Receive 2 Data Pin (SDO3/SDI2)

SDO3/SDI2 is used as the SDO3 signal for transmitting data from the ESAI_TX3 serial transmit shift register when programmed as a transmitter pin, or as the SDI2 signal for receiving serial data to the ESAI_RX2 serial receive shift register when programmed as a receiver pin.

SDO3/SDI2 is an input when data is being received by the ESAI_RX2 shift register. SDO3/SDI2 is an output when data is being transmitted from the ESAI_TX3 shift register. In the on-demand mode with an internally generated bit clock, the SDO3/SDI2 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO3/SDI2 may be programmed as a disconnected pin (PC8) when the ESAI SDO3 and SDI2 functions are not being used.

27.2.5 Serial Transmit 4/Receive 1 Data Pin (SDO4/SDI1)

SDO4/SDI1 is used as the SDO4 signal for transmitting data from the ESAI_TX4 serial transmit shift register when programmed as transmitter pin, or as the SDI1 signal for receiving serial data to the RX1 serial receive shift register when programmed as a receiver pin.

SDO4/SDI1 is an input when data is being received by the ESAI_RX1 shift register. SDO4/SDI1 is an output when data is being transmitted from the ESAI_TX4 shift register. In the on-demand mode with an internally generated bit clock, the SDO4/SDI1 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO4/SDI1 may be programmed as a disconnected pin (PC7) when the ESAI SDO4 and SDI1 functions are not being used.

27.2.6 Serial Transmit 5/Receive 0 Data Pin (SDO5/SDI0)

SDO5/SDI0 is used as the SDO5 signal for transmitting data from the ESAI_TX5 serial transmit shift register when programmed as transmitter pin, or as the SDI0 signal for receiving serial data to the ESAI_RX0 serial shift register when programmed as a receiver pin.

SDO5/SDI0 is an input when data is being received by the ESAI_RX0 shift register. SDO5/SDI0 is an output when data is being transmitted from the ESAI_TX5 shift register. In the on-demand mode with an internally generated bit clock, the SDO5/SDI0 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO5/SDI0 may be programmed as a disconnected pin (PC6) when the ESAI SDO5 and SDI0 functions are not being used.

27.2.7 Receiver Serial Clock (SCKR)

SCKR is a bidirectional pin providing the receivers serial bit clock for the ESAI interface.

The direction of this pin is determined by the RCKD bit in the ESAI_RCCR register.The SCKR operates as a clock input or output used by all the enabled receivers in the asynchronous mode (SYN=0), or as serial flag 0 pin in the synchronous mode (SYN=1).

When this pin is configured as serial flag pin, its direction is determined by the RCKD bit in the ESAI_RCCR register. When configured as the output flag OF0, this pin reflects the value of the OF0 bit in the ESAI_SAICR register, and the data in the OF0 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver

sections. When this pin is configured as the input flag IF0, the data value at the pin is stored in the IF0 bit in the ESAI_SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

SCKR may be programmed as a disconnected pin (PC0) when the ESAI SCKR function is not being used.

NOTE

Although the external ESAI serial clocks can be independent of and asynchronous to the internal 133 MHz ESAI system clock, the external ESAI serial clock frequency cannot exceed 133MHz/4 = 33.25 MHz and each external ESAI serial clock phase must exceed the minimum of 2 x 1/133MHz = 15.04ns.

For SCKR pin mode definitions, see Table 27-37.

The table below provides a list of asynchronous-mode receiver clock sources.

Table 27-1. Receiver Clock Sources (Asynchronous Mode Only)

RHCKD	RFSD	RCKD	ERI	ERO	Receiver Bit Clock Source	OUTPUTS		
0	0	0	N/A	N/A	SCKR	-	-	-
0	0	1	N/A	N/A	HCKR	-	-	SCKR
0	1	0	N/A	N/A	SCKR	-	FSR	-
0	1	1	N/A	N/A	HCKR	-	FSR	SCKR
1	0	0	0	0	SCKR	HCKR	-	-
1	0	0	0	1	SCKR	HCKR	-	-
1	0	0	1	0	SCKR	HCKR	-	-
1	0	0	1	1	SCKR	HCKR	-	-
1	0	1	0	0	Fsys[1]	HCKR	-	SCKR
1	0	1	0	1	Fsys	HCKR	-	SCKR
1	0	1	1	0	EXTAL[2]	HCKR	-	SCKR
1	0	1	1	1	EXTAL	HCKR	-	SCKR
1	1	0	0	0	SCKR	HCKR	FSR	-
1	1	0	0	1	SCKR	HCKR	FSR	-
1	1	0	1	0	SCKR	HCKR	FSR	-
1	1	0	1	1	SCKR	HCKR	FSR	-
1	1	1	0	0	Fsys	HCKR	FSR	SCKR
1	1	1	0	1	Fsys	HCKR	FSR	SCKR
1	1	1	1	0	EXTAL	HCKR	FSR	SCKR

Table continues on the next page...

Table 27-1. Receiver Clock Sources (Asynchronous Mode Only) (continued)

RHCKD	RFSD	RCKD	ERI	ERO	Receiver Bit Clock Source	OUTPUTS		
1	1	1	1	1	EXTAL	HCKR	FSR	SCKR
Fsys = 133MHz for i.MX25, i.MX53 and i.MX 6Dual/6Quad EXTAL is the on-chip clock sources other than ESAI system 133MHz clock, and it is from esai_clk_root in CCM								

27.2.8 Transmitter Serial Clock (SCKT)

SCKT is a bidirectional pin providing the transmitters serial bit clock for the ESAI interface.

The direction of this pin is determined by the TCKD bit in the ESAI_TCCR register. The SCKT is a clock input or output used by all the enabled transmitters in the asynchronous mode (SYN=0) or by all the enabled transmitters and receivers in the synchronous mode (SYN=1).

The following table provides a list of asynchronous-mode transmitter clock sources.

Table 27-2. Transmitter Clock Sources (Asynchronous Mode Only)

THCKD	TFSD	TCKD	ETI	ETO	Transmitter Bit Clock Source	OUTPUTS		
0	0	0	N/A	N/A	SCKT	-	-	-
0	0	1	N/A	N/A	HCKT	-	-	SCKT
0	1	0	N/A	N/A	SCKT	-	FST	-
0	1	1	N/A	N/A	HCKT	-	FST	SCKT
1	0	0	0	0	SCKT	HCKT	-	-
1	0	0	0	1	SCKT	HCKT	-	-
1	0	0	1	0	SCKT	HCKT	-	-
1	0	0	1	1	SCKT	HCKT	-	-
1	0	1	0	0	Fsys[1]	HCKT	-	SCKT
1	0	1	0	1	Fsys	HCKT	-	SCKT
1	0	1	1	0	EXTAL[2]	HCKT	-	SCKT
1	0	1	1	1	EXTAL	HCKT	-	SCKT
1	1	0	0	0	SCKR	HCKT	FST	-
1	1	0	0	1	SCKR	HCKT	FST	-
1	1	0	1	0	SCKR	HCKT	FST	-

Table continues on the next page...

Table 27-2. Transmitter Clock Sources (Asynchronous Mode Only) (continued)

THCKD	TFSD	TCKD	ETI	ETO	Transmitter Bit Clock Source	OUTPUTS		
1	1	0	1	1	SCKR	HCKT	FST	-
1	1	1	0	0	Fsys	HCKT	FST	SCKT
1	1	1	0	1	Fsys	HCKT	FST	SCKT
1	1	1	1	0	EXTAL	HCKT	FST	SCKT
1	1	1	1	1	EXTAL	HCKT	FST	SCKT
Fsys = 133MHz for i.MX25 and i.MX53 EXTAL is the on-chip clock sources other than ESAI system 133MHz clock, and it is from esai_clk_root in CCM								

SCKT may be programmed as a disconnected pin (PC3) when the ESAI SCKT function is not being used.

NOTE

Although the external ESAI serial clocks can be independent of and asynchronous to the internal 133 MHz ESAI system clock, the external ESAI serial clock frequency cannot exceed 133MHz/4 = 33.25 MHz and each external ESAI serial clock phase must exceed the minimum of 2 x 1/133MHz = 15.04ns.

27.2.9 Frame Sync for Receiver (FSR)

FSR is a bidirectional pin providing the receivers frame sync signal for the ESAI interface. The direction of this pin is determined by the RFSD bit in ESAI_RCR register.

In the asynchronous mode (SYN=0), the FSR pin operates as the frame sync input or output used by all the enabled receivers. In the synchronous mode (SYN=1), it operates as either the serial flag 1 pin (TEBE=0), or as the transmitter external buffer enable control (TEBE=1, RFSD=1). For FSR pin mode definitions, see Table 27-38; for receiver clock signals, see Table 27-1.

When this pin is configured as serial flag pin, its direction is determined by the RFSD bit in the ESAI_RCCR register. When configured as the output flag OF1, this pin reflects the value of the OF1 bit in the ESAI_SAICR register, and the data in the OF1 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver sections. When configured as the input flag IF1, the data value at the pin is stored in the IF1 bit in the ESAI_SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

FSR may be programmed as a disconnected pin (PC1) when the ESAI FSR function is not being used.

27.2.10 Frame Sync for Transmitter (FST)

FST is a bidirectional pin providing the frame sync for both the transmitters and receivers in the synchronous mode (SYN=1) and for the transmitters only in asynchronous mode

(SYN=0) (see Table 27-2). The direction of this pin is determined by the TFSD bit in the ESAI_TCR register. When configured as an output, this pin is the internally generated frame sync signal. When configured as an input, this pin receives an external frame sync signal for the transmitters (and the receivers in synchronous mode).

FST may be programmed as a disconnected pin (PC4) when the ESAI FST function is not being used.

27.2.11 High Frequency Clock for Transmitter (HCKT)

HCKT is a bidirectional pin providing the transmitters high frequency clock for the ESAI interface.

The direction of this pin is determined by the THCKD bit in the ESAI_TCCR register. In the asynchronous mode (SYN=0), the HCKT pin operates as the high frequency clock input or output used by all enabled transmitters. In the synchronous mode (SYN=1), it operates as the high frequency clock input or output used by all enabled transmitters and receivers. When programmed as input this pin is used as an alternative high frequency clock source to the ESAI transmitter rather than the ARM Core main clock. When programmed as output it can serve as a high frequency sample clock (to external DACs for example) or as an additional system clock (see Table 27-2).

HCKT may be programmed as a disconnected pin (PC5) when the ESAI HCKT function is not being used.

27.2.12 High Frequency Clock for Receiver (HCKR)

HCKR is a bidirectional pin providing the receivers high frequency clock for the ESAI interface.

The direction of this pin is determined by the RHCKD bit in the ESAI_RCCR register. In the asynchronous mode (SYN=0), the HCKR pin operates as the high frequency clock input or output used by all the enabled receivers. In the synchronous mode (SYN=1), it operates as the serial flag 2 pin. For HCKR pin mode definitions, see Table 27-39; for receiver clock signals, see Table 27-1.

When this pin is configured as serial flag pin, its direction is determined by the RHCKD bit in the ESAI_RCCR register. When configured as the output flag OF2, this pin reflects the value of the OF2 bit in the ESAI_SAICR register, and the data in the OF2 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver sections. When configured as the input flag IF2, the data value at the pin is stored in the IF2 bit in the ESAI_SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

HCKR may be programmed as a disconnected pin (PC2) when the ESAI HCKR function is not being used.

27.2.13 Serial I/O Flags

Three ESAI pins (FSR, SCKR and HCKR) are available as serial I/O flags when the ESAI is operating in the synchronous mode (SYN=1).

Their operation is controlled by RCKD, RFSD, TEBE bits in the ESAI_RCR, ESAI_RCCR and ESAI_SAICR registers.The output data bits (OF2, OF1 and OF0) and the input data bits (IF2, IF1 and IF0) are double buffered to/from the HCKR, FSR and SCKR pins. Double buffering the flags keeps them in sync with the TX and RX data lines.

Each flag can be separately programmed. Flag 0 (SCKR pin) direction is selected by RCKD, RCKD=1 for output and RCKD=0 for input. Flag 1 (FSR pin) is enabled when the pin is not configured as external transmitter buffer enable (TEBE=0) and its direction is selected by RFSD, RFSD=1 for output and RFSD=0 for input. Flag 2 (HCKR pin) direction is selected by RHCKD, RHCKD=1 for output and RHCKD=0 for input.

When programmed as input flags, the SCKR, FSR and HCKR logic values, respectively, are latched at the same time as the first bit of the receive data word is sampled. Because the input was latched, the signal on the input flag pin (SCKR, FSR or HCKR) can change without affecting the input flag until the first bit of the next receive data word. When the received data words are transferred to the receive data registers, the input flag latched values are then transferred to the IF0, IF1 and IF2 bits in the SAISR register, where they may be read by software.

When programmed as output flags, the SCKR, FSR and HCKR logic values are driven by the contents of the OF0, OF1 and OF2 bits in the ESAI_SAICR register respectively, and they are driven when the transmit data registers are transferred to the transmit shift registers. The value on SCKR, FSR and HCKR is stable from the time the first bit of the transmit data word is transmitted until the first bit of the next transmit data word is transmitted. Software may change the OF0-OF2 values thus controlling the SCKR, FSR and HCKR pin values for each transmitted word. The normal sequence for setting output flags when transmitting data is as follows: wait for TDE (transmitter empty) to be set; first write the flags, and then write the transmit data to the transmit registers. OF0, OF1, and OF2 are double buffered so that the flag states appear on the pins when the transmit data is transferred to the transmit shift register, that is, the flags are synchronous with the data.

27.3 Functional Description

27.3.1 ESAI After Reset

Hardware or software reset clears the port control register bits and the port direction control register bits, which configure all ESAI I/O pins as disconnected and both ESAI FIFOs are also in reset state.

The ESAI is in personal reset state while all ESAI pins are programmed as disconnected, and it is active only if at least one of the ESAI I/O pins is programmed as an ESAI pin.

27.3.2 ESAI Interrupt Requests

The ESAI can generate eight different interrupt requests

(ordered from the highest to the lowest priority):

1. **ESAI Receive Data with Exception Status**

   Occurs when the receive exception interrupt is enabled (REIE=1 in the RCR register), at least one of the enabled receive data registers is full (RDF=1) and a receiver overrun error has occurred (ROE=1 in the SAISR register). ROE is cleared by first reading the SAISR and then reading all the enabled receive data registers.

2. **ESAI Receive Even Data**

Occurs when the receive even slot data interrupt is enabled (REDIE=1), at least one of the enabled receive data registers is full (RDF=1), the data is from an even slot (REDF=1) and no exception has occurred (ROE=0 or REIE=0).

Reading all enabled receiver data registers clears RDF and REDF.

3. **ESAI Receive Data**

   Occurs when the receive interrupt is enabled (RIE=1), at least one of the enabled receive data registers is full (RDF=1), no exception has occurred (ROE=0 or REIE=0) and no even slot interrupt has occurred (REDF=0 or REDIE=0). Reading all enabled receiver data registers clears RDF.

4. **ESAI Receive Last Slot Interrupt**

   Occurs, if enabled (RLIE=1), after the last slot of the frame ended (in network mode only) regardless of the receive mask register setting. The receive last slot interrupt may be used for resetting the receive mask slot register, reconfiguring the DMA channels and reassigning data memory pointers. Using the receive last slot interrupt guarantees that the previous frame was serviced with the previous setting and the new frame is serviced with the new setting without synchronization problems. Note that the maximum receive last slot interrupt service time should not exceed N-1 ESAI bits service time (where N is the number of bits in a slot).

5. **ESAI Transmit Data with Exception Status**

   Occurs when the transmit exception interrupt is enabled (TEIE=1), at least one transmit data register of the enabled transmitters is empty (TDE=1) and a transmitter underrun error has occurred (TUE=1). TUE is cleared by first reading the SAISR and then writing to all the enabled transmit data registers, or to the TSR register.

6. **ESAI Transmit Last Slot Interrupt**

   Occurs, if enabled (TLIE=1), at the start of the last slot of the frame in network mode regardless of the transmit mask register setting. The transmit last slot interrupt may be used for resetting the transmit mask slot register, reconfiguring the DMA channels and reassigning data memory pointers. Using the transmit last slot interrupt guarantees that the previous frame was serviced with the previous setting and the new frame is serviced with the new setting without synchronization problems. Note that the maximum transmit last slot interrupt service time should not exceed N-1 ESAI bits service time (where N is the number of bits in a slot).

7. **ESAI Transmit Even Data**

Occurs when the transmit even slot data interrupt is enabled (TEDIE=1), at least one of the enabled transmit data registers is empty (TDE=1), the slot is an even slot (TEDE=1) and no exception has occurred (TUE=0 or TEIE=0). Writing to all the TX registers of the enabled transmitters or to TSR clears this interrupt request.

8. **ESAI Transmit Data**

   Occurs when the transmit interrupt is enabled (TIE=1), at least one of the enabled transmit data registers is empty (TDE=1), no exception has occurred (TUE=0 or TEIE=0) and no even slot interrupt has occurred (TEDE=0 or TEDIE=0). Writing to all the TX registers of the enabled transmitters, or to the TSR clears this interrupt request.

27.3.3 ESAI DMA Requests from the FIFOs

1. ESAI Transmit FIFO Empty - Asserts when the number of empty slots in the ESAI transmit FIFO exceeds the threshold programmed in the ESAI Transmit FIFO Configuration Register (TFCR). Automatically negates when the number of empty slots is less than the threshold programmed in the ESAI Transmit FIFO Configuration Register.
2. ESAI Receive FIFO Full - Asserts when the number of data words in the ESAI receive FIFO exceeds the threshold programmed in the ESAI Receive FIFO Configuration Register (RFCR). Automatically negates when the number of words is less than the threshold programmed in the ESAI Receive FIFO Configuration Register.

27.3.4 ESAI Transmit and Receive Shift Registers

27.3.4.1 ESAI Transmit Shift Registers

The transmit shift registers contain the data being transmitted

Data is shifted out to the serial transmit data pins by the selected (internal/external) bit clock when the associated frame sync I/O is asserted.

The number of bits shifted out before the shift registers are considered empty and may be written to again can be 8, 12, 16, 20, 24 or 32 bits (determined by the slot length control bits in the TCR register). Data is shifted out of these registers MSB first if TSHFD=0 and LSB first if TSHFD=1.

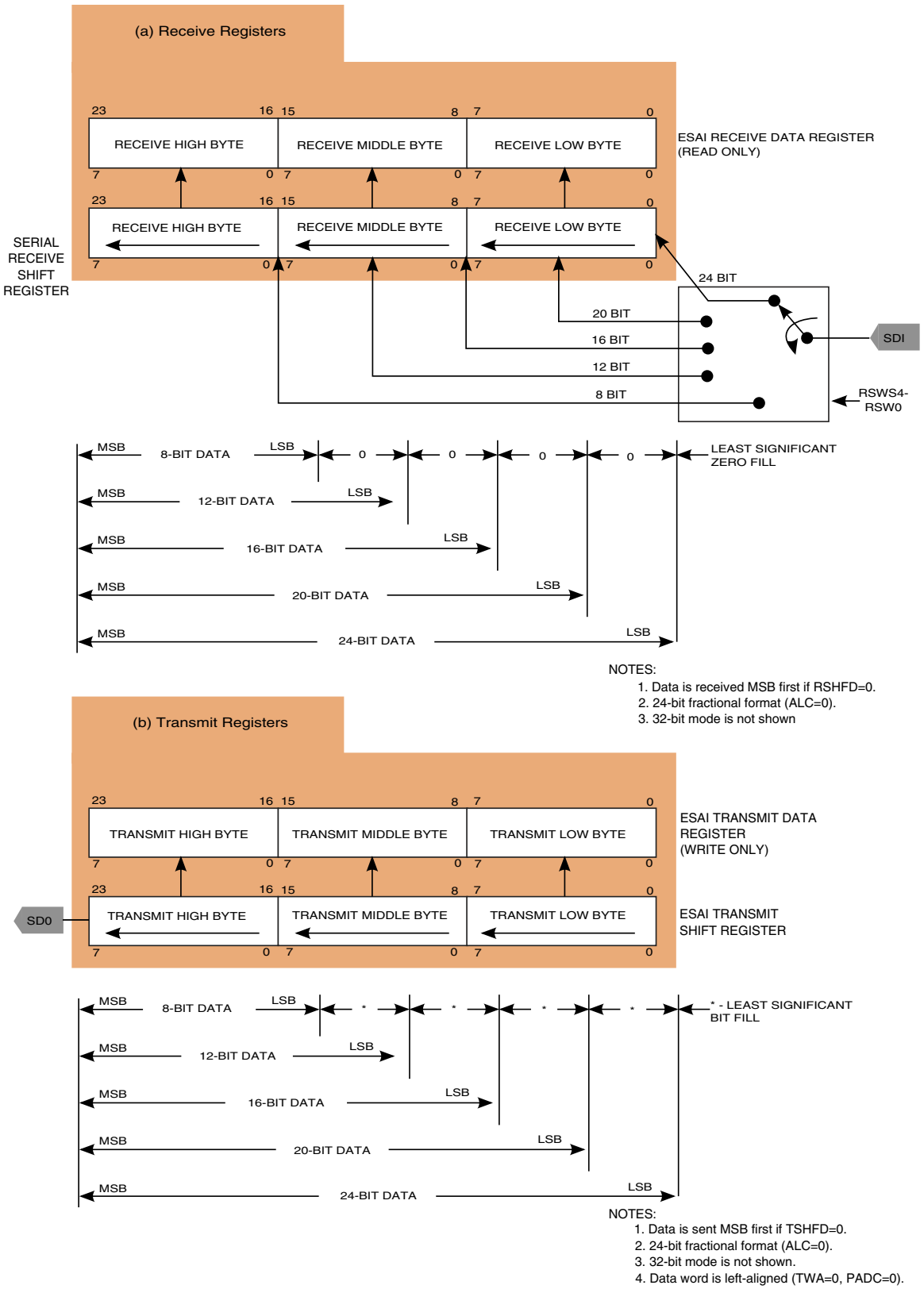


Figure 27-2. ESAI Data Path Programming Model ([R/T]SHFD=0)

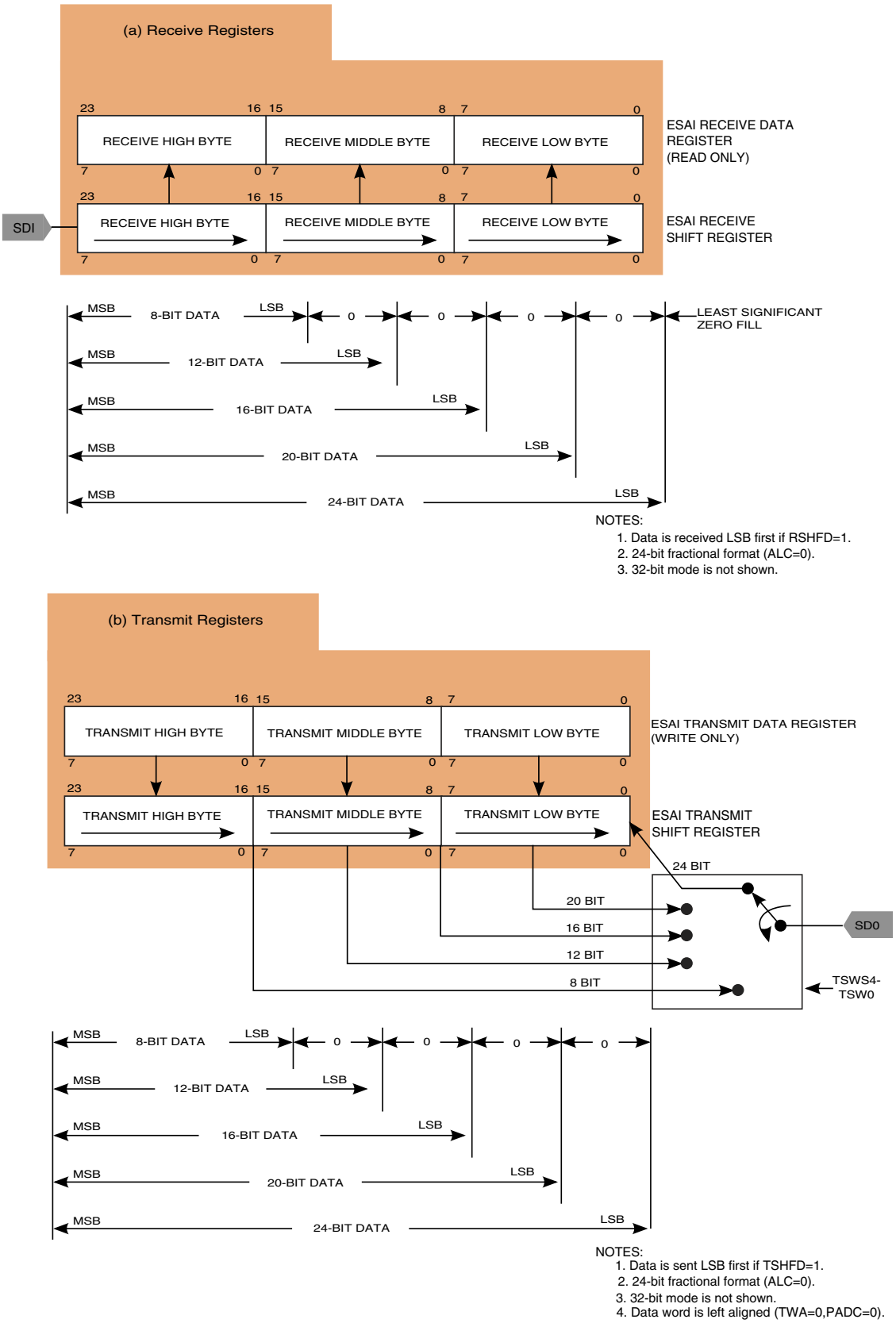


Figure 27-3. ESAI Data Path Programming Model ([R/T]SHFD=1)

27.3.4.2 ESAI Receive Shift Registers

Data is shifted in by the selected (internal/external) bit clock when the associated frame sync I/O is asserted. Data is assumed to be received MSB first if RSHFD=0 and LSB first if RSHFD=1. Data is transferred to the ESAI receive data registers after 8, 12, 16, 20, 24, or 32 serial clock cycles were counted, depending on the slot length control bits in the ESAI_RCR register.

27.4 Initialization Information

27.4.1 ESAI Initialization

The correct way to initialize the ESAI is as follows:

1. Enable the ESAI logic clock by asserting bit 0 of ESAI Control Register (ESAI_ECR[0]).
2. Hardware, software, ESAI individual reset. Note that asserting bit 1 of ESAI Control Register only reset the ESAI core logic, including configuration registers, but not the ESAI FIFOs.
3. Reset ESAI FIFOs by asserting bit 1 of ESAI_TFCR and ESAI_RFCR.
4. Program ESAI control and time slot registers.(The transmit/receive enable bits of TCR/RCR should not be set.)
5. Program ESAI FIFOs via TFCR and RFCR. (Enable Transmit/Receive FIFO, enable transmitters/receivers, transmit initialization and set Transmit FIFO/Receive FIFO watermark.)
6. Write initial words to ESAI Transmit Data Register (ESAI_ETDR), at least one word per enabled transmitter but as many as desired, if Transmit FIFO is used. If not, write initial words to all the enabled transmitters.
7. Remove ESAI personal reset by configuring ESAI_PCRC and ESAI_PRRC.
8. Enabled Transmitters/Receivers in ESAI_TCR/ESAI_RCR.

During program execution, all ESAI pins may be defined disconnected, causing the ESAI to stop serial activity and enter the individual reset state.

All status bits of the interface are set to their reset state however, the control bits are not affected. This procedure allows the programmer to reset the ESAI separately from the other internal peripherals. During individual reset, internal DMA accesses to the data registers of the ESAI are not valid and data read is undefined.

The programmer must use an individual ESAI reset when changing the ESAI control registers (except for TEIE, REIE, TLIE, RLIE, TIE, RIE, TE0-TE5, RE0-RE3) to ensure proper operation of the interface.

NOTE

If the ESAI receiver section is already operating with some of the receivers and enabling additional receivers on the fly, that is, without first putting the ESAI receiver in the personal reset state by setting their REx control bits, it will result in erroneous data being received as the first data word for the newly enabled receivers.

27.4.2 ESAI Initialization Examples

27.4.2.1 Initializing the ESAI using Personal Reset

1. Enable the ESAI logic clock by setting bit 0 of ESAI Control Register(ESAI_ECR[0]).
2. The ESAI should be in its personal reset state (ESAI_PCRC = 0x000 and ESAI_PRRC = 0x000). In the personal reset state, both the transmitter and receiver sections of the ESAI are simultaneously reset. The TPR bit in the ESAI_TCR register may be used to reset just the transmitter section. The RPR bit in the ESAI_RCR register may be used to reset just the receiver section.
3. Configure the control registers (ESAI_TCCR, ESAI_TCR, ESAI_RCCR, ESAI_RCR) and ESAI FIFOs configuration Registers (ESAI_TFCR, ESAI_RFCR) according to the operating mode, but do not enable transmitters (TE5-TE0 = 0x0) or receivers (RE3-RE0 = 0x0). It is possible to set the interrupt enable bits which are in use during the operation (no interrupt occurs).
4. Enable the ESAI by setting the ESAI_PCRC and ESAI_PRRC register bits according to pins which are in use during operation.
5. Write initial words to ESAI Transmit Data Register (ESAI_ETDR) (at least one word per enabled transmitter but as many as you want) if Transmit FIFO is used. If not, just write initial words to all the enabled transmitters which are in use during operation. This step is needed even if DMA is used to service the transmitters.
6. Enable the transmitters and receivers.
7. From now on ESAI can be serviced either by polling, interrupts, or DMA.

Operation proceeds as follows:

- For internally generated clock and frame sync, these signals are active immediately after ESAI is enabled (step 4 above).

- Data is received only when one of the receive enable (REx) bits is set and after the occurrence of frame sync signal (either internally or externally generated).
- Data is transmitted only when the transmitter enable (TEx) bit is set and after the occurrence of frame sync signal (either internally or externally generated). The transmitter outputs remain tri-stated after TEx bit is set until the frame sync occurs.

27.4.2.2 Initializing the ESAI Transmitter Section

1. It is assumed that the ESAI is operational; that is, at least one pin is defined as an ESAI pin.
2. Enable the ESAI logic clock by setting bit 0 of ESAI Control Register(ESAI_ECR[0])
3. The transmitter section should be in its individual reset state (TPR = 1) and also reset the ESAI Transmit FIFO (ESAI_TFCR[1] = 1).
4. Configure the control registers ESAI_TCCR and ESAI_TCR according to the operating mode, configure the Transmit FIFO Configuration Register (bring transmit FIFO out of reset, enable Transmit FIFO, enable transmitters, transmit initialization and set watermark). Make sure to clear the transmitter enable bits (TE0-TE5). TPR must remain set.
5. Take the transmitter section out of the individual reset state by clearing TPR.
6. Write initial words to ESAI Transmit Data Register (ESAI_ETDR) (at least one word per enabled transmitter but as many as you want) if Transmit FIFO is used. If not, just write first words to all the enabled transmitters which are in use during operation. This step is needed even if DMA is used to service the transmitters.
7. Enable the transmitters by setting their TE bits.
8. Data is transmitted only when the transmitter enable (TEx) bit is set and after the occurrence of frame sync signal (either internally or externally generated). The transmitter outputs remain tri-stated after TEx bit is set until the frame sync occurs.
9. From now on the transmitters are operating and can be serviced either by polling, interrupts, or DMA.

27.4.2.3 Initializing the ESAI Receiver Section

1. It is assumed that the ESAI is operational; that is, at least one pin is defined as an ESAI pin.
2. Enable the ESAI logic clock by setting bit 0 of ESAI Control Register (ESAI_ECR[0])
3. The receiver section should be in its individual reset state (RPR = 1) and also reset the ESAI Receive FIFO (ESAI_RFCR[1] = 1).

4. Configure the control registers ESAI_RCCR and ESAI_RCR according to the operating mode, configure the Receive FIFO Configuration Register (bring receive FIFO out of reset, enable Receive FIFO, receivers, and set watermark). Making sure to clear the receiver enable bits (RE0-RE3). RPR must remain set.
5. Take the receiver section out of the individual reset state by clearing RPR.
6. Enable the receivers by setting their RE bits.
7. From now on the receivers are operating and can be serviced either by polling, interrupts, or DMA.

27.5 Programmable Registers

ESAI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5001_8000	ESAI Transmit Data Register (ESAI_ETDR)	32	W (always reads zero)	0000_0000h	27.5.1/ 1170
5001_8004	ESAI Receive Data Register (ESAI_ERDR)	32	R	0000_0000h	27.5.2/ 1170
5001_8008	ESAI Control Register (ESAI_ECR)	32	R/W	0000_0000h	27.5.3/ 1171
5001_800C	ESAI Status Register (ESAI_ESR)	32	R	0000_0000h	27.5.4/ 1172
5001_8010	Transmit FIFO Configuration Register (ESAI_TFCR)	32	R/W	0000_0000h	27.5.5/ 1173
5001_8014	Transmit FIFO Status Register (ESAI_TFSR)	32	R	0000_0000h	27.5.6/ 1175
5001_8018	Receive FIFO Configuration Register (ESAI_RFCR)	32	R/W	0000_0000h	27.5.7/ 1176
5001_801C	Receive FIFO Status Register (ESAI_RFSR)	32	R	0000_0000h	27.5.8/ 1177
5001_8080	Transmit Data Register n (ESAI_TX0)	32	W (always reads zero)	0000_0000h	27.5.9/ 1178
5001_8084	Transmit Data Register n (ESAI_TX1)	32	W (always reads zero)	0000_0000h	27.5.9/ 1178

Table continues on the next page...

ESAI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5001_8088	Transmit Data Register n (ESAI_TX2)	32	W (always reads zero)	0000_0000h	27.5.9/ 1178
5001_808C	Transmit Data Register n (ESAI_TX3)	32	W (always reads zero)	0000_0000h	27.5.9/ 1178
5001_8090	Transmit Data Register n (ESAI_TX4)	32	W (always reads zero)	0000_0000h	27.5.9/ 1178
5001_8094	Transmit Data Register n (ESAI_TX5)	32	W (always reads zero)	0000_0000h	27.5.9/ 1178
5001_8098	ESAI Transmit Slot Register (ESAI_TSR)	32	W (always reads zero)	0000_0000h	27.5.10/ 1179
5001_80A0	Receive Data Register n (ESAI_RX0)	32	R	0000_0000h	27.5.11/ 1180
5001_80A4	Receive Data Register n (ESAI_RX1)	32	R	0000_0000h	27.5.11/ 1180
5001_80A8	Receive Data Register n (ESAI_RX2)	32	R	0000_0000h	27.5.11/ 1180
5001_80AC	Receive Data Register n (ESAI_RX3)	32	R	0000_0000h	27.5.11/ 1180
5001_80CC	Serial Audio Interface Status Register (ESAI_SAISR)	32	R	0000_0000h	27.5.12/ 1181
5001_80D0	Serial Audio Interface Control Register (ESAI_SAICR)	32	R/W	0000_0000h	27.5.13/ 1183
5001_80D4	Transmit Control Register (ESAI_TCR)	32	R/W	0000_0000h	27.5.14/ 1186
5001_80D8	Transmit Clock Control Register (ESAI_TCCR)	32	R/W	0000_0000h	27.5.15/ 1194
5001_80DC	Receive Control Register (ESAI_RCR)	32	R/W	0000_0000h	27.5.16/ 1198
5001_80E0	Receive Clock Control Register (ESAI_RCCR)	32	R/W	0000_0000h	27.5.17/ 1202
5001_80E4	Transmit Slot Mask Register A (ESAI_TSMA)	32	R/W	0000_FFFFh	27.5.18/ 1205
5001_80E8	Transmit Slot Mask Register B (ESAI_TSMB)	32	R/W	0000_FFFFh	27.5.19/ 1206

Table continues on the next page...

ESAI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5001_80EC	Receive Slot Mask Register A (ESAI_RSMA)	32	R/W	0000_FFFFh	27.5.20/ 1207
5001_80F0	Receive Slot Mask Register B (ESAI_RSMB)	32	R/W	0000_FFFFh	27.5.21/ 1208
5001_80F8	Port C Direction Register (ESAI_PRRC)	32	R/W	0000_0000h	27.5.22/ 1209
5001_80FC	Port C Control Register (ESAI_PCRC)	32	R/W	0000_0000h	27.5.23/ 1209

27.5.1 ESAI Transmit Data Register (ESAI_ETDR)

Address: ESAI_ETDR is 5001_8000h base + 0h offset = 5001_8000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W	ETDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_ETDR field descriptions

Field	Description
31–0 ETDR	ESAI Transmit Data Register. Writing to this register stores the data written into the ESAI Transmit FIFO. Writing to this register when the Transmit FIFO is full causes the data written to be lost (the existing data within the FIFO is not overwritten). When multiple ESAI transmitters are enabled, the data for each transmitter must be interleaved from lowest transmitter to highest transmitter (for example, if transmitters 0, 2 and 3 are enabled then data must be written as follows: transmitter #0, transmitter #2, transmitter #3, transmitter #0, transmitter #2, transmitter #3, transmitter #0, etc). Data within the ESAI Transmit FIFO is passed to the ESAI transmit shifter registers as defined by the Transmit Word Alignment configuration bits.

27.5.2 ESAI Receive Data Register (ESAI_ERDR)

Address: ESAI_ERDR is 5001_8000h base + 4h offset = 5001_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERDR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_ERDR field descriptions

Field	Description
31–0 ERDR	ESAI Receive Data Register. Reading this register returns the data within the ESAI Receive FIFO. Reading this register when the Receive FIFO is empty returns the last valid data word. When multiple ESAI receivers are enabled, the data for each receiver is interleaved from lowest receiver to highest receiver (for example, if receivers 0, 2 and 3 are enabled then data is returned as follows: receiver #0, receiver #2, receiver #3, receiver #0, receiver #2, receiver #3, receiver #0, etc). Data is passed from the ESAI receive shift registers to the ESAI Receive FIFO as defined by the Receiver Word Alignment configuration bits either zero or sign-extended based on the Receive Extension control bit.

27.5.3 ESAI Control Register (ESAI_ECR)

Address: ESAI_ECR is 5001_8000h base + 8h offset = 5001_8008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												ETI	ETO	ERI	ERO	0														ERST	ESAIEN
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_ECR field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value zero. Reserved
19 ETI	EXTAL Transmitter In. Mux EXTAL in place of the High Frequency Transmitter Clock input pin. HCKT can still be used to drive a divided down EXTAL or as GPIO. 0 HCKT pin has normal function. 1 EXTAL muxed into HCKT input.
18 ETO	EXTAL Transmitter Out. Drive the EXTAL input on the High Frequency Transmitter Clock pin. 0 HCKT pin has normal function. 1 EXTAL driven onto HCKT pin.
17 ERI	EXTAL Receiver In. Mux EXTAL in place of the High Frequency Receiver Clock input pin. HCKR can still be used to drive a divided down EXTAL or as GPIO. 0 HCKR pin has normal function. 1 EXTAL muxed into HCKR input.
16 ERO	EXTAL Receiver Out. Drive the EXTAL input on the High Frequency Receiver Clock pin. 0 HCKR pin has normal function. 1 EXTAL driven onto HCKR pin.
15–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1 ERST	ESAI Reset. Reset the ESAI core logic (including configuration registers) but not the ESAI FIFOs.

Table continues on the next page...

ESAI_ECR field descriptions (continued)

Field	Description
	0 ESAI not reset. 1 ESAI reset.
0 ESAIEN	ESAI Enable. Enables/disables the ESAI logic clock. Enable the ESAI before reading or writing other ESAI registers. 0 ESAI disabled. 1 ESAI enabled.

27.5.4 ESAI Status Register (ESAI_ESR)

Address: ESAI_ESR is 5001_8000h base + Ch offset = 5001_800Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					TINIT	RFF	TFE	TLS	TDE	TED	TD	RLS	RDE	RED	RD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_ESR field descriptions

Field	Description
31–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10 TINIT	Transmit Initialization. Indicates that the Transmit FIFO is writing the first word for each enabled transmitter into the Transmit Data Registers. This bit sets when the Transmit FIFO is enabled (provided Transmit Initialization is enabled) and clears after the Transmit Data Registers have been initialized. The Transmit Enable bits in the Transmit Control Register should not be set until this flag has cleared. 0 Transmitter has finished initializing the Transmit Data Registers (or Transmit FIFO is not enabled or Transmit Initialization is not enabled). 1 Transmitter has not finished initializing the Transmit Data Registers.
9 RFF	Receive FIFO Full. Indicates that the number of data words in the Receive FIFO has equaled or exceeded the Receive FIFO Watermark. This flag also drives the ESAI Receiver DMA request line. ESAI FIFO DMA requests see ESAI DMA Requests from the FIFOs. 0 Number of words in Receive FIFO less than Receive FIFO watermark. 1 Number of words in Receive FIFO is equal to or greater than Receive FIFO watermark.
8 TFE	Transmit FIFO Empty. Indicates that the number of empty slots in the Transmit FIFO has met or exceeded the Transmit FIFO Watermark. This flag also drives the ESAI Transmitter DMA request line. ESAI FIFO DMA request see ESAI DMA Requests from the FIFOs.

Table continues on the next page...

ESAI_ESR field descriptions (continued)

Field	Description
	0 Number of empty slots in Transmit FIFO less than Transmit FIFO watermark. 1 Number of empty slots in Transmit FIFO is equal to or greater than Transmit FIFO watermark.
7 TLS	Transmit Last Slot. Reading this register when TLS is set will negate the Transmit Last Slot interrupt. 0 TLS is not the highest priority active interrupt. 1 TLS is the highest priority active interrupt.
6 TDE	Transmit Data Exception. 0 TDE is not the highest priority active interrupt. 1 TDE is the highest priority active interrupt.
5 TED	Transmit Even Data. 0 TED is not the highest priority active interrupt. 1 TED is the highest priority active interrupt.
4 TD	Transmit Data. 0 TD is not the highest priority active interrupt. 1 TD is the highest priority active interrupt.
3 RLS	Receive Last Slot. Reading this register when RLS is set will negate the Receive Last Slot interrupt. 0 RLS is not the highest priority active interrupt. 1 RLS is the highest priority active interrupt.
2 RDE	Receive Data Exception. 0 RDE is not the highest priority active interrupt. 1 RDE is the highest priority active interrupt.
1 RED	Receive Even Data. 0 RED is not the highest priority active interrupt. 1 RED is the highest priority active interrupt.
0 RD	Receive Data. 0 RD is not the highest priority active interrupt. 1 RD is the highest priority active interrupt.

27.5.5 Transmit FIFO Configuration Register (ESAI_TFCR)

Address: ESAI_TFCR is 5001_8000h base + 10h offset = 5001_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												TIEN	TWA[2:0]			TFWM[7:0]								TE5	TE4	TE3	TE2	TE1	TE0	TFR	TFE
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_TFCR field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value zero. Reserved
19 TIEN	Transmitter Initialization Enable. Enables the initialization of the Transmit Data Registers when the Transmitter FIFO is enabled. 0 Transmit Data Registers are not initialized from the FIFO once the Transmit FIFO is enabled. Software must manually initialize the Transmit Data Registers separately. 1 Transmit Data Registers are initialized from the FIFO once the Transmit FIFO is enabled.
18–16 TWA[2:0]	Transmit Word Alignment. Configures the alignment of the data written into the ESAI Transmit Data Register and then passed to the relevant 24 bit Transmit shift register. 000 MSB of data is bit 31. Data bits 7-0 are ignored when passed to transmit shift register. 001 MSB of data is bit 27. Data bits 3-0 are ignored when passed to transmit shift register. 010 MSB of data is bit 23. 011 MSB of data is bit 19. Bottom 4 bits of transmit shift register are zeroed. 100 MSB of data is bit 15. Bottom 8 bits of transmit shift register are zeroed. 101 MSB of data is bit 11. Bottom 12 bits of transmit shift register are zeroed. 110 MSB of data is bit 7. Bottom 16 bits of transmit shift register are zeroed. 111 MSB of data is bit 3. Bottom 20 bits of transmit shift register are zeroed.
15–8 TFWM[7:0]	Transmit FIFO Watermark. These bits configure the threshold at which the Transmit FIFO Empty flag will set. The TFE is set when the number of empty slots in the Transmit FIFO equal or exceed the selected threshold.
7 TE5	Transmitter #5 FIFO Enable. This bit enables transmitter #5 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #5 is not using the Transmit FIFO. 1 Transmitter #5 is using the Transmit FIFO.
6 TE4	Transmitter #4 FIFO Enable. This bit enables transmitter #4 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #4 is not using the Transmit FIFO. 1 Transmitter #4 is using the Transmit FIFO.
5 TE3	Transmitter #3 FIFO Enable. This bit enables transmitter #3 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #3 is not using the Transmit FIFO. 1 Transmitter #3 is using the Transmit FIFO.
4 TE2	Transmitter #2 FIFO Enable. This bit enables transmitter #2 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #2 is not using the Transmit FIFO. 1 Transmitter #2 is using the Transmit FIFO.
3 TE1	Transmitter #1 FIFO Enable. This bit enables transmitter #1 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #1 is not using the Transmit FIFO. 1 Transmitter #1 is using the Transmit FIFO.

Table continues on the next page...

ESAI_TFCR field descriptions (continued)

Field	Description
2 TE0	Transmitter #0 FIFO Enable. This bit enables transmitter #0 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #0 is not using the Transmit FIFO. 1 Transmitter #0 is using the Transmit FIFO.
1 TFR	Transmit FIFO Reset. This bit resets the Transmit FIFO pointers. 0 Transmit FIFO not reset. 1 Transmit FIFO reset.
0 TFE	Transmit FIFO Enable. This bit enables the use of the Transmit FIFO. 0 Transmit FIFO disabled. 1 Transmit FIFO enabled.

27.5.6 Transmit FIFO Status Register (ESAI_TFSR)

Address: ESAI_TFSR is 5001_8000h base + 14h offset = 5001_8014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																	NTFO[2:0]			0	NTFI[2:0]			TFCNT[7:0]							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_TFSR field descriptions

Field	Description
31–15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 NTFO[2:0]	Next Transmitter FIFO Out. Indicates which Transmit Data Register receives the top word of the Transmit FIFO. This will usually equal the lowest enabled transmitter, unless the transmit FIFO is empty. 000 Transmitter #0 receives next word from the Transmit FIFO. 001 Transmitter #1 receives next word from the Transmit FIFO. 010 Transmitter #2 receives next word from the Transmit FIFO. 011 Transmitter #3 receives next word from the Transmit FIFO. 100 Transmitter #4 receives next word from the Transmit FIFO. 101 Transmitter #5 receives next word from the Transmit FIFO. 110 Reserved. 111 Reserved.
11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–8 NTFI[2:0]	Next Transmitter FIFO In. Indicates which transmitter receives the next word written to the FIFO. 000 Transmitter #0 receives next word written to the Transmit FIFO. 001 Transmitter #1 receives next word written to the Transmit FIFO.

Table continues on the next page...

ESAI_TFSR field descriptions (continued)

Field	Description
	010 Transmitter #2 receives next word written to the Transmit FIFO. 011 Transmitter #3 receives next word written to the Transmit FIFO. 100 Transmitter #4 receives next word written to the Transmit FIFO. 101 Transmitter #5 receives next word written to the Transmit FIFO. 110 Reserved. 111 Reserved.
7–0 TFCNT[7:0]	Transmit FIFO Counter. These bits indicate the number of data words stored in the Transmit FIFO.

27.5.7 Receive FIFO Configuration Register (ESAI_RFCR)

Address: ESAI_RFCR is 5001_8000h base + 18h offset = 5001_8018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												REXT	RWA[2:0]			RFWM[7:0]								0		RE3	RE2	RE1	RE0	RFR	RFE
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_RFCR field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value zero. Reserved
19 REXT	Receive Extension. Enables the receive data to be returned sign extended when the Receive Word Alignment is configured to return data where the MSB is not aligned with bit 31. 0 Receive data is zero extended. 1 Receive data is sign extended.
18–16 RWA[2:0]	Receive Word Alignment. Configures the alignment of the data passed from the relevant 24 bit Receive shift register and read out the ESAI Receive Data Register. 000 MSB of data is at bit 31. Data bits 7-0 are zeroed. 001 MSB of data is at bit 27. Data bits 3-0 are zeroed. 010 MSB of data is at bit 23. 011 MSB of data is at bit 19. Data bits 3-0 from receive shift register are ignored. 100 MSB of data is at bit 15. Data bits 7-0 from receive shift register are ignored. 101 MSB of data is at bit 11. Data bits 11-0 from receive shift register are ignored. 110 MSB of data is at bit 7. Data bits 15-0 from receive shift register are ignored. 111 MSB of data is at bit 3. Data bits 19-0 from receive shift register are ignored.
15–8 RFWM[7:0]	Receive FIFO Watermark. These bits configure the threshold at which the Receive FIFO Full flag will set. The RFF is set when the number of words in the Receive FIFO equal or exceed the selected threshold. It can be set to a non-zero value.
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

ESAI_RFCR field descriptions (continued)

Field	Description
5 RE3	Receiver #3 FIFO Enable. This bit enables receiver #3 to use the Receive FIFO. Do not change this bit when the Receiver FIFO is enabled. 0 Receiver #3 is not using the Receive FIFO. 1 Receiver #3 is using the Receive FIFO.
4 RE2	Receiver #2 FIFO Enable. This bit enables receiver #2 to use the Receive FIFO. Do not change this bit when the Receiver FIFO is enabled. 0 Receiver #2 is not using the Receive FIFO. 1 Receiver #2 is using the Receive FIFO.
3 RE1	Receiver #1 FIFO Enable. This bit enables receiver #1 to use the Receive FIFO. Do not change this bit when the Receiver FIFO is enabled. 0 Receiver #1 is not using the Receive FIFO. 1 Receiver #1 is using the Receive FIFO.
2 RE0	Receiver #0 FIFO Enable. This bit enables receiver #0 to use the Receive FIFO. Do not change this bit when the Receiver FIFO is enabled. 0 Receiver #0 is not using the Receive FIFO. 1 Receiver #0 is using the Receive FIFO.
1 RFR	Receive FIFO Reset. This bit resets the Receive FIFO pointers. 0 Receive FIFO not reset. 1 Receive FIFO reset.
0 RFE	Receive FIFO Enable. This bit enables the use of the Receive FIFO. 0 Receive FIFO disabled. 1 Receive FIFO enabled.

27.5.8 Receive FIFO Status Register (ESAI_RFSR)

Address: ESAI_RFSR is 5001_8000h base + 1Ch offset = 5001_801Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		NRFI [1:0]		0		NRFO [1:0]		RFCNT [7:0]							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

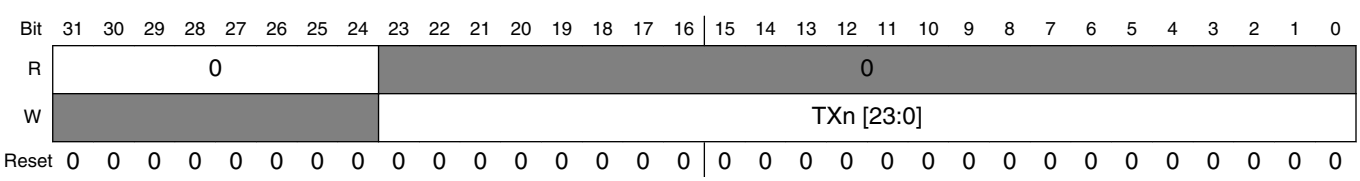
ESAI_RFSR field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13–12 NRFI [1:0]	Next Receiver FIFO In. Indicates which Receiver Data Register the Receive FIFO will load next. This will usually equal the lowest enabled receiver, unless the receive FIFO is full. 00 Receiver #0 returns next word to the Receive FIFO. 01 Receiver #1 returns next word to the Receive FIFO. 10 Receiver #2 returns next word to the Receive FIFO. 11 Receiver #3 returns next word to the Receive FIFO.
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–8 NRFO [1:0]	Next Receiver FIFO Out. Indicates which receiver returns the top word of the Receive FIFO. 00 Receiver #0 returns next word from the Receive FIFO. 01 Receiver #1 returns next word from the Receive FIFO. 10 Receiver #2 returns next word from the Receive FIFO. 11 Receiver #3 returns next word from the Receive FIFO.
7–0 RFCNT [7:0]	Receive FIFO Counter. These bits indicate the number of data words stored in the Receive FIFO.

27.5.9 Transmit Data Register n (ESAI_TX)

ESAI_TX5, ESAI_TX4, ESAI_TX3, ESAI_TX2, ESAI_TX1 and ESAI_TX0 are 32-bit write-only registers. Data to be transmitted is written into these registers and is automatically transferred to the transmit shift registers (Figure 1 and Figure 2). The data written (8, 12, 16, 20, or 24 bits) should occupy the most significant portion of the TXn according to the ALC control bit setting. The unused bits (least significant portion and the 8 most significant bits when ALC=1) of the TXn are don't care bits. The Core is interrupted whenever the TXn becomes empty if the transmit data register empty interrupt has been enabled.

Addresses: ESAI_TX0 is 5001_8000h base + 80h offset = 5001_8080h
ESAI_TX1 is 5001_8000h base + 84h offset = 5001_8084h
ESAI_TX2 is 5001_8000h base + 88h offset = 5001_8088h
ESAI_TX3 is 5001_8000h base + 8Ch offset = 5001_808Ch
ESAI_TX4 is 5001_8000h base + 90h offset = 5001_8090h
ESAI_TX5 is 5001_8000h base + 94h offset = 5001_8094h



ESAI_TX*n* field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23–0 TXn [23:0]	Stores the data to be transmitted and is automatically transferred to the transmit shift registers. See ESAI Transmit Shift Registers.

27.5.10 ESAI Transmit Slot Register (ESAI_TSR)

Address: ESAI_TSR is 5001_8000h base + 98h offset = 5001_8098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0																							
W									TSR [23:0]																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

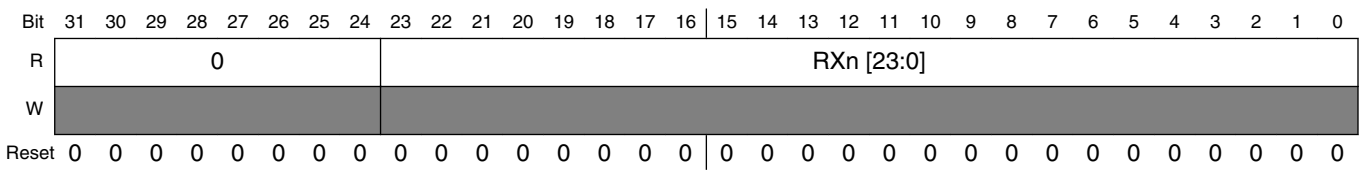
ESAI_TSR field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23–0 TSR [23:0]	The write-only Transmit Slot Register (ESAI_TSR) is effectively a null data register that is used when the data is not to be transmitted in the available transmit time slot. The transmit data pins of all the enabled transmitters are in the high-impedance state for the respective time slot where TSR has been written. The Transmitter External Buffer Enable pin (FSR pin when SYN=1, TEBE=1, RFSD=1) disables the external buffers during the slot when the ESAI_TSR register has been written.

27.5.11 Receive Data Register n (ESAI_RX)

ESAI_RX3, ESAI_RX2, ESAI_RX1, and ESAI_RX0 are 32-bit read-only registers that accept data from the receive shift registers when they become full (Figure 1 and Figure 2). The data occupies the most significant portion of the receive data registers, according to the ALC control bit setting. The unused bits (least significant portion and 8 most significant bits when ALC=1) read as zeros. The Core is interrupted whenever RXn becomes full if the associated interrupt is enabled.

Addresses: ESAI_RX0 is 5001_8000h base + A0h offset = 5001_80A0h
ESAI_RX1 is 5001_8000h base + A4h offset = 5001_80A4h
ESAI_RX2 is 5001_8000h base + A8h offset = 5001_80A8h
ESAI_RX3 is 5001_8000h base + ACh offset = 5001_80ACh



ESAI_RX*n* field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23–0 RXn [23:0]	Accept data from the receive shift registers when they become full See ESAI Receive Shift Registers

27.5.12 Serial Audio Interface Status Register (ESAI_SAISR)

The Status Register (ESAI_SAISR) is a read-only status register used by the ARM Core to read the status and serial input flags of the ESAI.

Address: ESAI_SAISR is 5001_8000h base + CCh offset = 5001_80CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														TODFE	TEDE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TDE	TUE	TFS	0		RODF	REDF	RDF	ROE	RFS	0			IF2	IF1	IF0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_SAISR field descriptions

Field	Description
31–18 Reserved	This read-only field is reserved and always has the value zero. Reserved.
17 TODFE	ESAI_SAISR Transmit Odd-Data Register Empty. When set, TODFE indicates that the enabled transmitter data registers became empty at the beginning of an odd time slot. Odd time slots are all odd-numbered slots (1, 3, 5, and so on). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. This flag is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TODFE indicates that data should be written to all the TX registers of the enabled transmitters or to the transmit slot register (ESAI_TSR). TODE is cleared when the Core writes to all the transmit data registers of the enabled transmitters, or when the Core writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TODFE is set. Hardware, software, ESAI individual reset clear TODFE.
16 TEDE	ESAI_SAISR Transmit Even-DataRegister Empty. When set, TEDE indicates that the enabled transmitter data registers became empty at the beginning of an even time slot. Even time slots are all even-numbered slots (0, 2, 4, 6, etc.). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. The zero time slot is considered even. This flag is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TEDE indicates that data should be written to all the TX registers of the enabled transmitters or to the transmit slot register (ESAI_TSR). TEDE is cleared when the Core writes to all the transmit data registers of the enabled transmitters, or when the Core writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TEDE is set. Hardware, software, ESAI individual reset clear TEDE.

Table continues on the next page...

ESAI_SAISR field descriptions (continued)

Field	Description
15 TDE	ESAI_SAISR Transmit Data Register Empty. TDE is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TDE indicates that data should be written to all the TX registers of the enabled transmitters or to the transmit slot register (ESAI_TSR). TDE is cleared when the Core writes to all the transmit data registers of the enabled transmitters, or when the Core writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TDE is set. Hardware, software, ESAI individual reset clear TDE.
14 TUE	ESAI_SAISR Transmit Underrun Error Flag. TUE is set when at least one of the enabled serial transmit shift registers is empty (no new data to be transmitted) and a transmit time slot occurs. When a transmit underrun error occurs, the previous data (which is still present in the TX registers that were not written) is retransmitted. If TEIE is set, an ESAI transmit data with exception (underrun error) interrupt request is issued when TUE is set. Hardware, software, ESAI individual reset clear TUE. TUE is also cleared by reading the ESAI_SAISR with TUE set, followed by writing to all the enabled transmit data registers or to ESAI_TSR.
13 TFS	ESAI_SAISR Transmit Frame Sync Flar. When set, TFS indicates that a transmit frame sync occurred in the current time slot. TFS is set at the start of the first time slot in the frame and cleared during all other time slots. Data written to a transmit data register during the time slot when TFS is set is transmitted (in network mode), if the transmitter is enabled, during the second time slot in the frame. TFS is useful in network mode to identify the start of a frame. TFS is cleared by hardware, software, ESAI individual reset. TFS is valid only if at least one transmitter is enabled, that is, one or more of TE0, TE1, TE2, TE3, TE4 and TE5 are set. (In normal mode, TFS always reads as a one when transmitting data because there is only one time slot per frame - the "frame sync" time slot)
12–11 Reserved	This read-only field is reserved and always has the value zero. Reserved.
10 RODF	ESAI_SAISR Receive Odd-Data Register Full. When set, RODF indicates that the received data in the receive data registers of the enabled receivers have arrived during an odd time slot when operating in the network mode. Odd time slots are all odd-numbered slots (1, 3, 5, and so on). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. RODF is set when the contents of the receive shift registers are transferred to the receive data registers. RODF is cleared when the Core reads all the enabled receive data registers or cleared by hardware, software, ESAI individual resets.
9 REDF	ESAI_SAISR Receive Even-Data Register Full. When set, REDF indicates that the received data in the receive data registers of the enabled receivers have arrived during an even time slot when operating in the network mode. Even time slots are all even-numbered slots (0, 2, 4, 6, and so on). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. The zero time slot is considered even. REDF is set when the contents of the receive shift registers are transferred to the receive data registers. REDF is cleared when the Core reads all the enabled receive data registers or cleared by hardware, software, ESAI individual resets. If REDIE is set, an ESAI receive even slot data interrupt request is issued when REDF is set.
8 RDF	ESAI_SAISR Receive Data Register Full. RDF is set when the contents of the receive shift register of an enabled receiver is transferred to the respective receive data register. RDF is cleared when the Core reads the receive data register of all enabled receivers or cleared by hardware, software, ESAI individual reset. If RIE is set, an ESAI receive data interrupt request is issued when RDF is set.
7 ROE	ESAI_SAISR Receive Overrun Error Flag. The ROE flag is set when the serial receive shift register of an enabled receiver is full and ready to transfer to its receiver data register (RXn) and the register is already full (RDF=1). If REIE is set, an ESAI receive data with exception (overrun error) interrupt request is issued when ROE is set. Hardware, software, ESAI individual reset clear ROE. ROE is also cleared by reading the SAISR with ROE set, followed by reading all the enabled receive data registers.
6 RFS	ESAI_SAISR Receive Frame Sync Flag. When set, RFS indicates that a receive frame sync occurred during reception of the words in the receiver data registers. This indicates that the data words are from

Table continues on the next page...

ESAI_SAISR field descriptions (continued)

Field	Description
	the first slot in the frame. When RFS is clear and a word is received, it indicates (only in the network mode) that the frame sync did not occur during reception of that word. RFS is cleared by hardware, software, ESAI individual reset. RFS is valid only if at least one of the receivers is enabled (REx=1). (In normal mode, RFS always reads as a one when reading data because there is only one time slot per frame - the "frame sync" time slot)
5–3 Reserved	This read-only field is reserved and always has the value zero. Reserved.
2 IF2	ESAI_SAISR Serial Input Flag 2. The IF2 bit is enabled only when the HCKR pin is defined as ESAI in the Port Control Register, SYN=1 and RHCKD=0, indicating that HCKR is an input flag and the synchronous mode is selected. Data present on the HCKR pin is latched during reception of the first received data bit after frame sync is detected. The IF2 bit is updated with this data when the receive shift registers are transferred into the receiver data registers. IF2 reads as a zero when it is not enabled. Hardware, software, ESAI individual reset clear IF2.
1 IF1	ESAI_SAISR Serial Inout Flag 1. The IF1 bit is enabled only when the FSR pin is defined as ESAI in the Port Control Register, SYN =1, RFSD=0 and TEBE=0, indicating that FSR is an input flag and the synchronous mode is selected. Data present on the FSR pin is latched during reception of the first received data bit after frame sync is detected. The IF1 bit is updated with this data when the receiver shift registers are transferred into the receiver data registers. IF1 reads as a zero when it is not enabled. Hardware, software, ESAI individual reset clear IF1.
0 IF0	ESAI_SAISR Serial Input Flag 0. The IF0 bit is enabled only when the SCKR pin is defined as ESAI in the Port Control Register, SYN=1 and RCKD=0, indicating that SCKR is an input flag and the synchronous mode is selected. Data present on the SCKR pin is latched during reception of the first received data bit after frame sync is detected. The IF0 bit is updated with this data when the receiver shift registers are transferred into the receiver data registers. IF0 reads as a zero when it is not enabled. Hardware, software, ESAI individual reset clear IF0.

27.5.13 Serial Audio Interface Control Register (ESAI_SAICR)

The read/write Common Control Register (ESAI_SAICR) contains control bits for functions that affect both the receive and transmit sections of the ESAI.

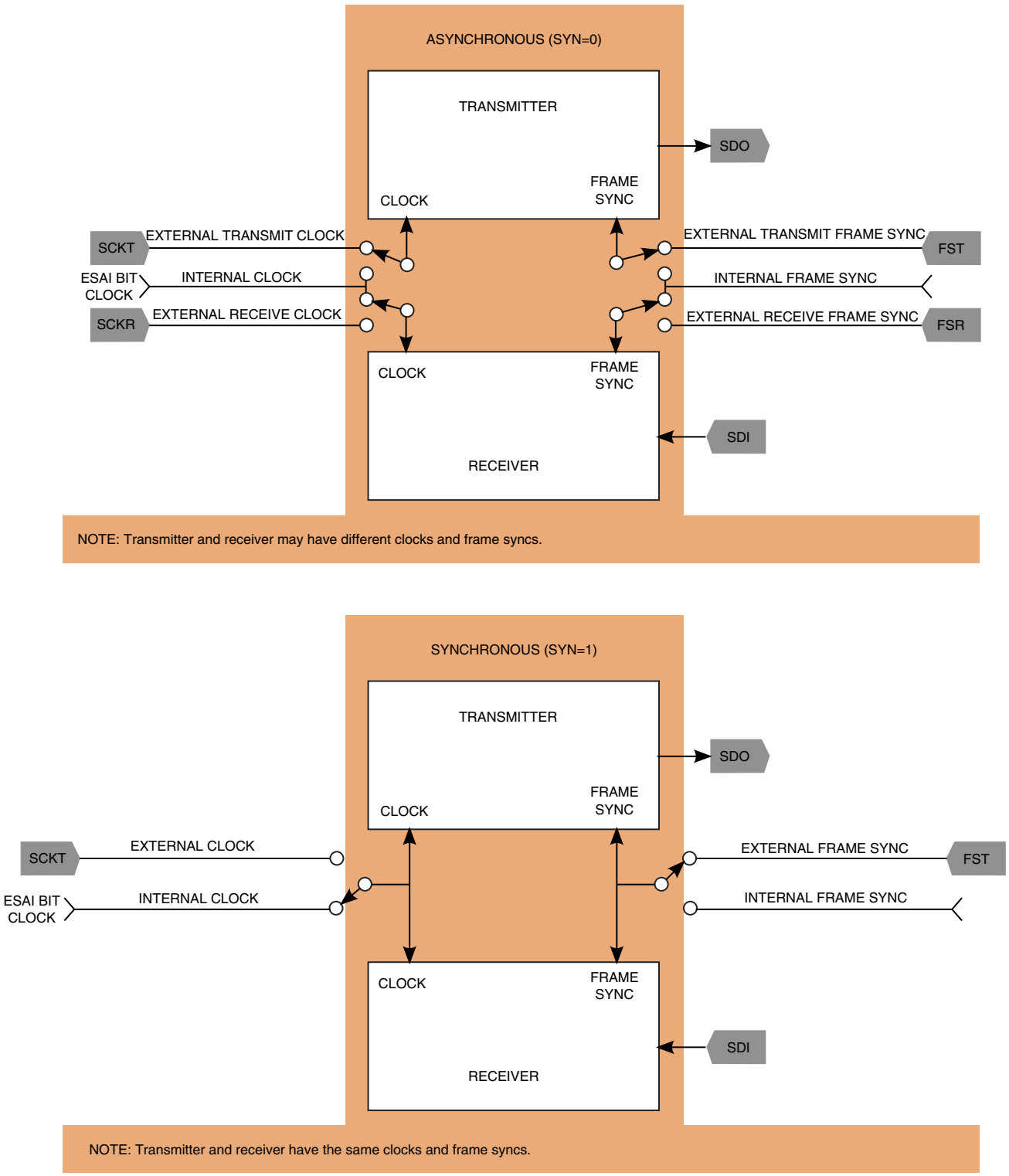
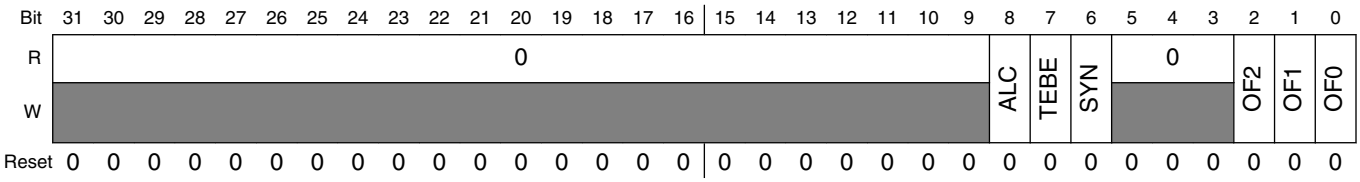


Figure 27-27. SAICR SYN Bit Operation

Address: ESAI_SAICR is 5001_8000h base + D0h offset = 5001_80D0h



ESAI_SAICR field descriptions

Field	Description
31–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 ALC	ESAI_SAICR Alignment Control. The ESAI is designed for 24-bit fractional data, thus shorter data words are left aligned to the MSB (bit 23). Some applications use 16-bit fractional data. In those cases, shorter data words may be left aligned to bit 15. The Alignment Control (ALC) bit supports these applications. If ALC is set, transmitted and received words are left aligned to bit 15 in the transmit and receive shift registers. If ALC is cleared, transmitted and received word are left aligned to bit 23 in the transmit and receive shift registers. While ALC is set, 20-bit and 24-bit words may not be used, and word length control should specify 8-, 12-, or 16-bit words; otherwise, results are unpredictable.
7 TEBE	ESAI_SAICR Transmit External Buffer Enable. The Transmitter External Buffer Enable (TEBE) bit controls the function of the FSR pin when in the synchronous mode. If the ESAI is configured for operation in the synchronous mode (SYN=1), and TEBE is set while FSR pin is configured as an output (RFSD=1), the FSR pin functions as the transmitter external buffer enable control to enable the use of an external buffers on the transmitter outputs. If TEBE is cleared, the FSR pin functions as the serial I/O flag 1. See Port C Control Register Receive Clock Control Register for a summary of the effects of TEBE on the FSR pin.
6 SYN	ESAI_SAICR Synchronous Mode Selection. The Synchronous Mode Selection (SYN) bit controls whether the receiver and transmitter sections of the ESAI operate synchronously or asynchronously with respect to each other (see Port C Control Register Serial Audio Interface Control Register). When SYN is cleared, the asynchronous mode is chosen and independent clock and frame sync signals are used for the transmit and receive sections. When SYN is set, the synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals. When in the synchronous mode (SYN=1), the transmit and receive sections use the transmitter section clock generator as the source of the clock and frame sync for both sections. Also, the receiver clock pins SCKR, FSR and HCKR now operate as I/O flags. Refer to Table 27-37, Table 27-38, and Table 27-39 for the effects of SYN on the receiver clock pins.
5–3 Reserved	This read-only field is reserved and always has the value zero. Reserved.
2 OF2	ESAI_SAICR Serial Output Flag 2. The Serial Output Flag 2 (OF2) is a data bit used to hold data to be send to the OF2 pin. When the ESAI is in the synchronous clock mode (SYN=1), the HCKR pin is configured as the ESAI flag 2. If the receiver high frequency clock direction bit (RHCKD) is set, the HCKR pin is the output flag OF2, and data present in the OF2 bit is written to the OF2 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.
1 OF1	ESAI_SAICR Serial Output Flag 1. The Serial Output Flag 1 (OF1) is a data bit used to hold data to be send to the OF1 pin. When the ESAI is in the synchronous clock mode (SYN=1), the FSR pin is configured as the ESAI flag 1. If the receiver frame sync direction bit (RFSD) is set and the TEBE bit is cleared, the FSR pin is the output flag OF1, and data present in the OF1 bit is written to the OF1 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.
0 OF0	ESAI_SAICR Serial Output Flag 0. The Serial Output Flag 0 (OF0) is a data bit used to hold data to be send to the OF0 pin. When the ESAI is in the synchronous clock mode (SYN=1), the SCKR pin is

Table continues on the next page...

ESAI_SAICR field descriptions (continued)

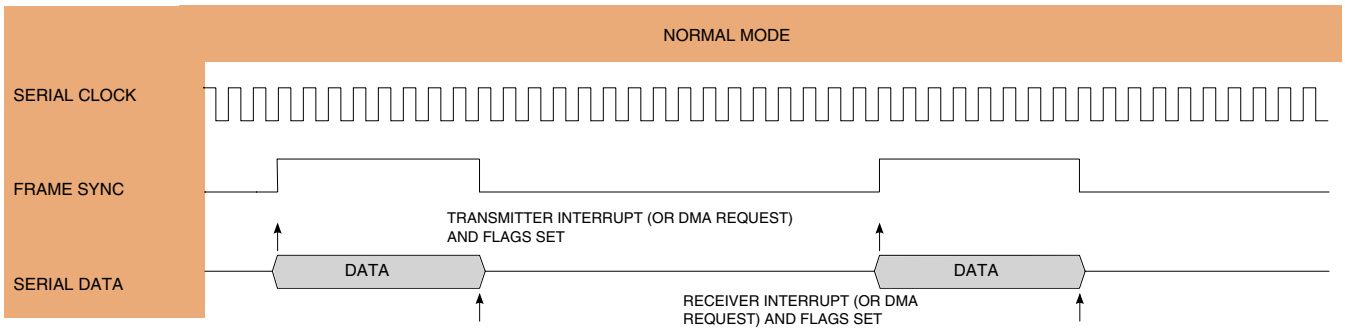
Field	Description
	configured as the ESAI flag 0. If the receiver serial clock direction bit (RCKD) is set, the SCKR pin is the output flag OF0, and data present in the OF0 bit is written to the OF0 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.

27.5.14 Transmit Control Register (ESAI_TCR)

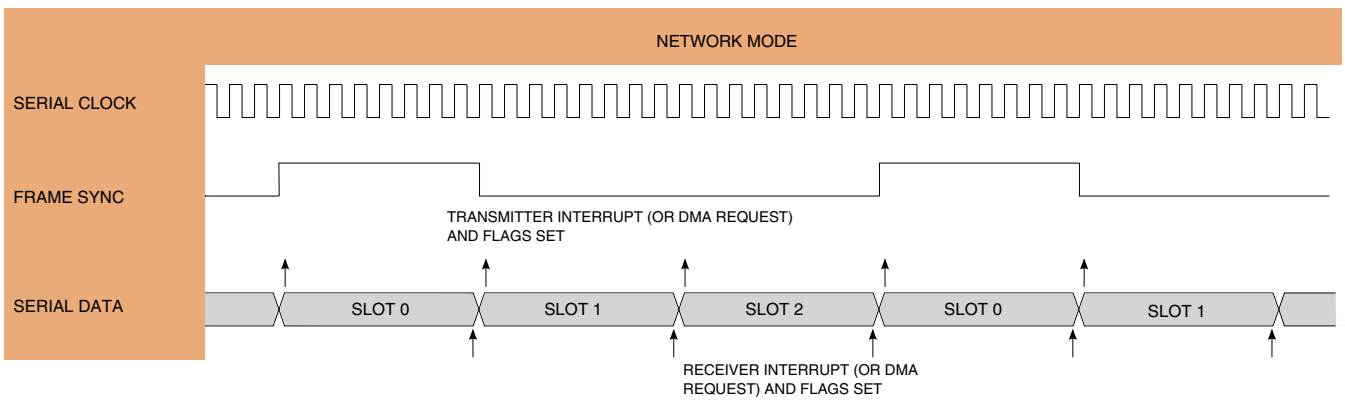
The read/write Transmit Control Register (ESAI_TCR) controls the ESAI transmitter section. Interrupt enable bits for the transmitter section are provided in this control register. Operating modes are also selected in this register.

Table 27-28. Transmit Network Mode Selection

TMOD1	TMOD0	TDC4-TDC0	Transmitter Network Mode
0	0	0x0-0x1F	Normal Mode
0	1	0x0	On-Demand Mode
0	1	0x1-0x1F	Network Mode
1	0	X	Reserved
1	1	0x0C	AC97



NOTE: Interrupts occur and data is transferred once per frame sync.



NOTE: Interrupts occur and a word may be transferred at every time slot.

Figure 27-29. Normal and Network Operation

Table 27-29. ESAI Transmit Slot and Word Length Selection

TSWS4	TSWS3	TSWS2	TSWS1	TSWS0	SLOT LENGTH	WORD LENGTH
0	0	0	0	0	8	8
0	0	1	0	0	12	8
0	0	0	0	1		12
0	1	0	0	0	16	8
0	0	1	0	1		12
0	0	0	1	0		16
0	1	1	0	0	20	8
0	1	0	0	1		12
0	0	1	1	0		16
0	0	0	1	1		20

Table continues on the next page...

Table 27-29. ESAI Transmit Slot and Word Length Selection (continued)

TSWS4	TSWS3	TSWS2	TSWS1	TSWS0	SLOT LENGTH	WORD LENGTH
1	0	0	0	0	24	8
0	1	1	0	1		12
0	1	0	1	0		16
0	0	1	1	1		20
1	1	1	1	0		24
1	1	0	0	0	32	8
1	0	1	0	1		12
1	0	0	1	0		16
0	1	1	1	1		20
1	1	1	1	1		24
0	1	0	1	1	Reserved	
0	1	1	1	0		
1	0	0	0	1		
1	0	0	1	1		
1	0	1	0	0		
1	0	1	1	0		
1	0	1	1	1		
1	1	0	0	1		
1	1	0	1	0		
1	1	0	1	1		
1	1	1	0	0		
1	1	1	0	1		

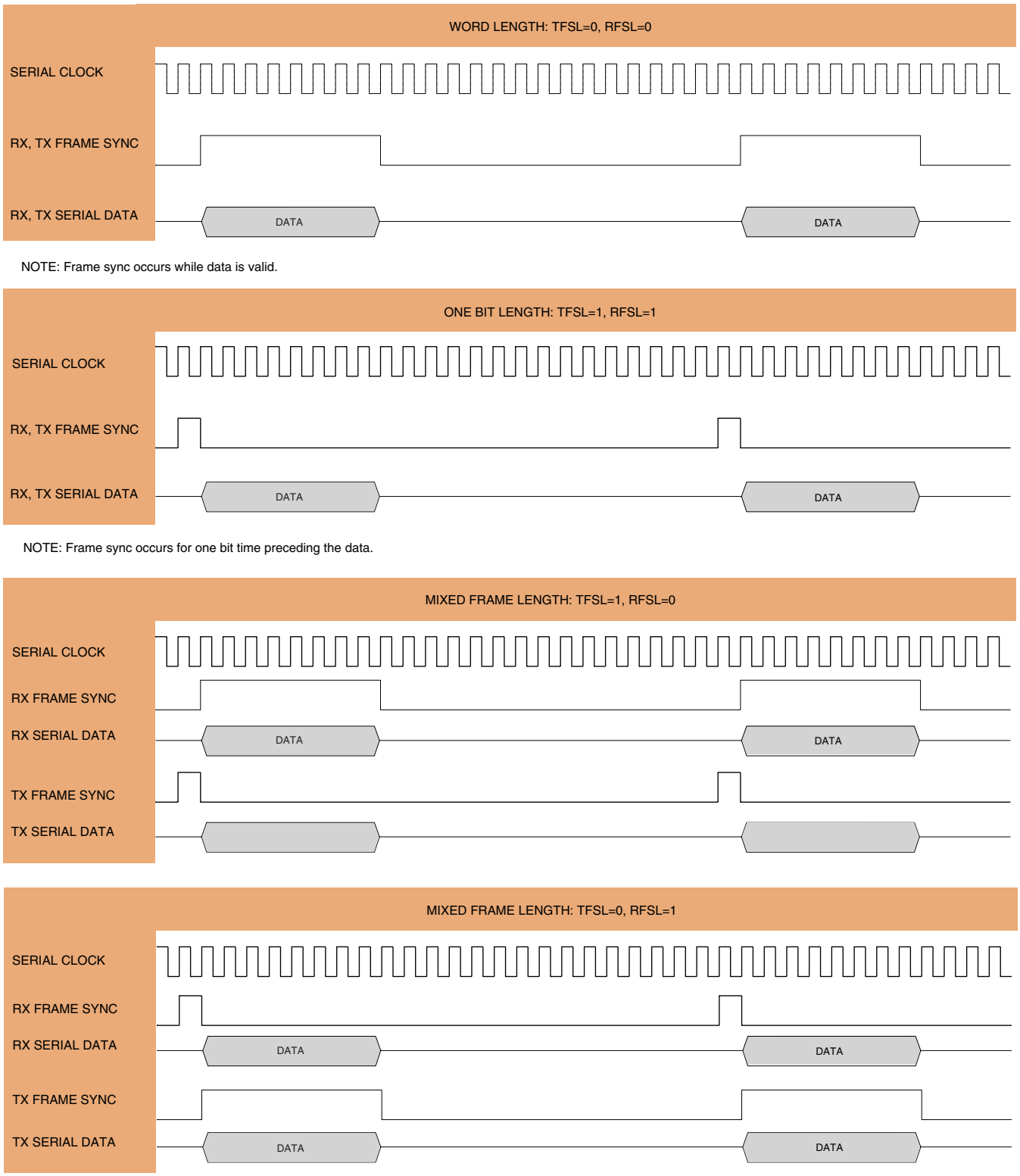


Figure 27-30. Frame Length Selection

Address: ESAI_TCR is 5001_8000h base + D4h offset = 5001_80D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								TLIE	TIE	TEDIE	TEIE	TPR	0	PADC	TFSR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TFSL	TSWS [14:10]					TMOD [9:8]		TWA	TSHFD	TE5	TE4	TE3	TE2	TE1	TE0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_TCR field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved.
23 TLIE	ESAI_TCR Transmit Last Slot Interrupt Enable. TLIE enables an interrupt at the beginning of last slot of a frame in network mode. When TLIE is set the Core is interrupted at the start of the last slot in a frame in network mode regardless of the transmit mask register setting. When TLIE is cleared the transmit last slot interrupt is disabled. TLIE is disabled when TDC[4:0]=0x00000 (on-demand mode). The use of the transmit last slot interrupt is described in ESAI Interrupt Requests.
22 TIE	ESAI_TCR Transmit Interrupt Enable. The Core is interrupted when TIE and the TDE flag in the ESAI_SAISR status register are set. When TIE is cleared, this interrupt is disabled. Writing data to all the data registers of the enabled transmitters or to ESAI_TSR clears TDE, thus clearing the interrupt. Transmit interrupts with exception have higher priority than normal transmit data interrupts, therefore if exception occurs (TUE is set) and TEIE is set, the ESAI requests an ESAI transmit data with exception interrupt from the interrupt controller.
21 TEDIE	ESAI_TCR Transmit Even Slot Data Interrupt Enable. The TEDIE control bit is used to enable the transmit even slot data interrupts. If TEDIE is set, the transmit even slot data interrupts are enabled. If TEDIE is cleared, the transmit even slot data interrupts are disabled. A transmit even slot data interrupt request is generated if TEDIE is set and the TEDE status flag in the ESAI_SAISR status register is set. Even time slots are all even-numbered time slots (0, 2, 4, etc.) when operating in network mode. The zero time slot in the frame is marked by the frame sync signal and is considered to be even. Writing data to all the data registers of the enabled transmitters or to ESAI_TSR clears the TEDE flag, thus servicing the interrupt. Transmit interrupts with exception have higher priority than transmit even slot data interrupts, therefore if exception occurs (TUE is set) and TEIE is set, the ESAI requests an ESAI transmit data with exception interrupt from the interrupt controller.
20 TEIE	ESAI_TCR Transmit Exception Interrupt Enable. When TEIE is set, the Core is interrupted when both TDE and TUE in the ESAI_SAISR status register are set. When TEIE is cleared, this interrupt is disabled. Reading the ESAI_SAISR status register followed by writing to all the data registers of the enabled transmitters clears TUE, thus clearing the pending interrupt.
19 TPR	ESAI_TCR Transmit Section Personal Reset. The TPR control bit is used to put the transmitter section of the ESAI in the personal reset state. The receiver section is not affected. When TPR is cleared, the transmitter section may operate normally. When TPR is set, the transmitter section enters the personal reset state immediately. When in the personal reset state, the status bits are reset to the same state as after hardware reset. The control bits are not affected by the personal reset state. The transmitter data

Table continues on the next page...

ESAI_TCR field descriptions (continued)

Field	Description
	pins are tri-stated while in the personal reset state; if a stable logic level is desired, the transmitter data pins should be defined as GPIO outputs, or external pull-up or pull-down resistors should be used. The transmitter clock outputs drive zeroes while in the personal reset state. Note that to leave the personal reset state by clearing TPR, the procedure described in ESAI Initialization Examples should be followed.
18 Reserved	This read-only field is reserved and always has the value zero. Reserved.
17 PADC	ESAI_TCR Transmit Zero Padding Control. When PADC is cleared, zero padding is disabled. When PADC is set, zero padding is enabled. PADC, in conjunction with the TWA control bit, determines the way that padding is done for operating modes where the word length is less than the slot length. See the TWA bit description in bit 7 for more details. Because the data word is shorter than the slot length, the data word is extended until achieving the slot length, according to the following rule: If the data word is left-aligned (TWA=0), and zero padding is disabled (PADC=0), the last data bit is repeated after the data word has been transmitted. If zero padding is enabled (PADC=1), zeroes are transmitted after the data word has been transmitted. If the data word is right-aligned (TWA=1), and zero padding is disabled (PADC=0), the first data bit is repeated before the transmission of the data word. If zero padding is enabled (PADC=1), zeroes are transmitted before the transmission of the data word.
16 TFSR	ESAI_TCR Transmit Frame Sync Relative Timing. TFSR determines the relative timing of the transmit frame sync signal as referred to the serial data lines, for a word length frame sync only (TFSL=0). When TFSR is cleared the word length frame sync occurs together with the first bit of the data word of the first slot. When TFSR is set the word length frame sync starts one serial clock cycle earlier, that is, together with the last bit of the previous data word.
15 TFSL	ESAI_TCR Transmit Frame Sync Length. The TFSL bit selects the length of frame sync to be generated or recognized. If TFSL is cleared, a word-length frame sync is selected. If TFSL is set, a 1-bit clock period frame sync is selected. See Figure 1-21 for examples of frame length selection.
14–10 TSWS [14:10]	ESAI_TCR Tx Slot and Word Length Select (TSWS4-TSWS0). The TSWS4-TSWS0 bits are used to select the length of the slot and the length of the data words being transferred through the ESAI. The word length must be equal to or shorter than the slot length. The possible combinations are shown in Table 27-29. See also the ESAI data path programming model in Figure 27-2 and Figure 27-3.
9–8 TMOD [9:8]	ESAI_TCR Transmit Network Mode Control (TMOD1-TMOD0). The TMOD1 and TMOD0 bits are used to define the network mode of ESAI transmitters, as shown in Table 27-29. In the normal mode, the frame rate divider determines the word transfer rate - one word is transferred per frame sync during the frame sync time slot, as shown in Figure 27-29. In network mode, it is possible to transfer a word for every time slot, as shown in Figure 27-29. For further details, refer to Modes of Operation In order to comply with AC-97 specifications, TSWS4-TSWS0 should be set to 00011 (20-bit slot, 20-bit word length), TFSL and TFSR should be cleared, and TDC4-TDC0 should be set to 0x0C (13 words in frame). If TMOD[1:0]=0b11 and the above recommendations are followed, the first slot and word will be 16 bits long, and the next 12 slots and words will be 20 bits long, as required by the AC97 protocol.
7 TWA	ESAI_TCR Transmit Word Alignment Control. The Transmitter Word Alignment Control (TWA) bit defines the alignment of the data word in relation to the slot. This is relevant for the cases where the word length is shorter than the slot length. If TWA is cleared, the data word is left-aligned in the slot frame during transmission. If TWA is set, the data word is right-aligned in the slot frame during transmission. Because the data word is shorter than the slot length, the data word is extended until achieving the slot length, according to the following rule: If the data word is left-aligned (TWA=0), and zero padding is disabled (PADC=0), the last data bit is repeated after the data word has been transmitted. If zero padding is enabled (PADC=1), zeroes are transmitted after the data word has been transmitted.

Table continues on the next page...

ESAI_TCR field descriptions (continued)

Field	Description
	If the data word is right-aligned (TWA=1), and zero padding is disabled (PADC=0), the first data bit is repeated before the transmission of the data word. If zero padding is enabled (PADC=1), zeroes are transmitted before the transmission of the data word.
6 TSHFD	ESAI_TCR Transmit Shift Direction. The TSHFD bit causes the transmit shift registers to shift data out MSB first when TSHFD equals zero or LSB first when TSHFD equals one (see Figure 27-2 and Figure 27-3 .
5 TE5	ESAI_TCR ESAI Transmit 5 Enable. TE5 enables the transfer of data from ESAI_TX5 to the transmit shift register #5. When TE5 is set and a frame sync is detected, the transmit #5 portion of the ESAI is enabled for that frame. When TE5 is cleared, the transmitter #5 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to ESAI_TX5 when TE5 is cleared but the data is not transferred to the transmit shift register #5. The SDO5/SDI0 pin is the data input pin for ESAI_RX0 if TE5 is cleared and RE0 in the ESAI_RCR register is set. If both RE0 and TE5 are cleared, the transmitter and receiver are disabled, and the pin is tri-stated. Both RE0 and TE5 should not be set at the same time. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE5 and setting it again disables the transmitter #5 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO5/SDI0 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE5 can be left enabled.
4 TE4	ESAI_TCR ESAI Transmit 4 Enable. TE4 enables the transfer of data from ESAI_TX4 to the transmit shift register #4. When TE4 is set and a frame sync is detected, the transmit #4 portion of the ESAI is enabled for that frame. When TE4 is cleared, the transmitter #4 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to ESAI_TX4 when TE4 is cleared but the data is not transferred to the transmit shift register #4. The SDO4/SDI1 pin is the data input pin for ESAI_RX1 if TE4 is cleared and RE1 in the RCR register is set. If both RE1 and TE4 are cleared, the transmitter and receiver are disabled, and the pin is tri-stated. Both RE1 and TE4 should not be set at the same time. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE4 and setting it again disables the transmitter #4 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO4/SDI1 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE4 can be left enabled.
3 TE3	ESAI_TCR ESAI Transmit 3 Enable. TE3 enables the transfer of data from ESAI_TX3 to the transmit shift register #3. When TE3 is set and a frame sync is detected, the transmit #3 portion of the ESAI is enabled for that frame. When TE3 is cleared, the transmitter #3 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to ESAI_TX3 when TE3 is cleared but the data is not transferred to the transmit shift register #3. The SDO3/SDI2 pin is the data input pin for ESAI_RX2 if TE3 is cleared and RE2 in the ESAI_RCR register is set. If both RE2 and TE3 are cleared, the transmitter and receiver are disabled, and the pin is tri-stated. Both RE2 and TE3 should not be set at the same time. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE3 and setting it again disables the transmitter #3 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO3/SDI2 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE3 can be left enabled.

Table continues on the next page...

ESAI_TCR field descriptions (continued)

Field	Description
2 TE2	ESAI_TCR ESAI Transmit 2 Enable. TE2 enables the transfer of data from ESAI_TX2 to the transmit shift register #2. When TE2 is set and a frame sync is detected, the transmit #2 portion of the ESAI is enabled for that frame. When TE2 is cleared, the transmitter #2 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to ESAI_TX2 when TE2 is cleared but the data is not transferred to the transmit shift register #2. The SDO2/SDI3 pin is the data input pin for ESAI_RX3 if TE2 is cleared and RE3 in the ESAI_RCR register is set. If both RE3 and TE2 are cleared, the transmitter and receiver are disabled, and the pin is tri-stated. Both RE3 and TE2 should not be set at the same time. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE2 and setting it again disables the transmitter #2 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO2/SDI3 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE2 can be left enabled.
1 TE1	ESAI_TCR ESAI Transmit 1 Enable. TE1 enables the transfer of data from TX1 to the transmit shift register #1. When TE1 is set and a frame sync is detected, the transmit #1 portion of the ESAI is enabled for that frame. When TE1 is cleared, the transmitter #1 is disabled after completing transmission of data currently in the ESAI transmit shift register. The SDO1 output is tri-stated, and any data present in TX1 is not transmitted, that is, data can be written to TX1 with TE1 cleared, but data is not transferred to the transmit shift register #1. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE1 and setting it again disables the transmitter #1 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO1 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE1 can be left enabled.
0 TE0	ESAI_TCR ESAI Transmit 0 Enable. TE0 enables the transfer of data from ESAI_TX0 to the transmit shift register #0. When TE0 is set and a frame sync is detected, the transmit #0 portion of the ESAI is enabled for that frame. When TE0 is cleared, the transmitter #0 is disabled after completing transmission of data currently in the ESAI transmit shift register. The SDO0 output is tri-stated, and any data present in ESAI_TX0 is not transmitted, that is, data can be written to ESAI_TX0 with TE0 cleared, but data is not transferred to the transmit shift register #0. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE0 and setting it again disables the transmitter #0 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO0 pin remains in the high-impedance state.The on-demand mode transmit enable sequence can be the same as the normal mode, or TE0 can be left enabled.

27.5.15 Transmit Clock Control Register (ESAI_TCCR)

The read/write Transmitter Clock Control Register (ESAI_TCCR) controls the ESAI transmitter clock generator bit and frame sync rates, the bit clock and high frequency clock sources and the directions of the HCKT, FST and SCKT signals. In the synchronous mode (SYN=1), the bit clock defined for the transmitter determines the receiver bit clock as well. ESAI_TCCR also controls the number of words per frame for the serial data. Hardware and software reset clear all the bits of the ESAI_TCCR register.

Care should be taken in asynchronous mode whenever the frame sync clock (FSR, FST) is not sourced directly from its associated bit clock (SCKR, SCKT). Proper phase relationships must be maintained between these clocks in order to guarantee proper operation of the ESAI.

NOTE

ARM Core clock is ipg_clk_esai in block ESAI which is from CCM's ahb_clk_root.

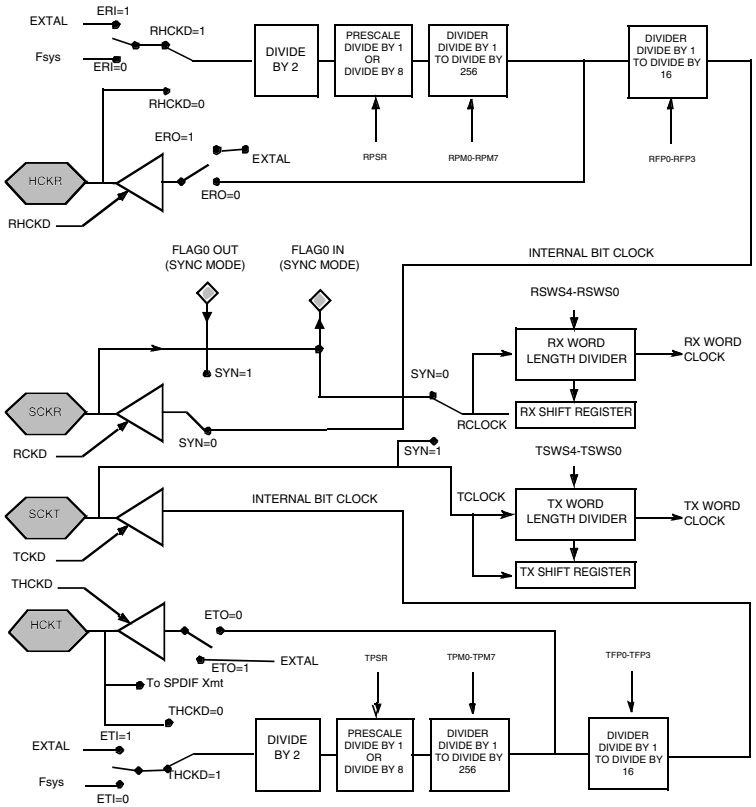


Figure 27-32. ESAI Clock Generator Functional Block Diagram

NOTE

1. ETI, ETO, ERI and ERO bit descriptions are covered in ESAI Control Register (ESAI_ECR). 2. Fsys is the ESAI system 133 MHz clock. 3. EXTAL is the on-chip clock sources other than ESAI system 133MHz clock.

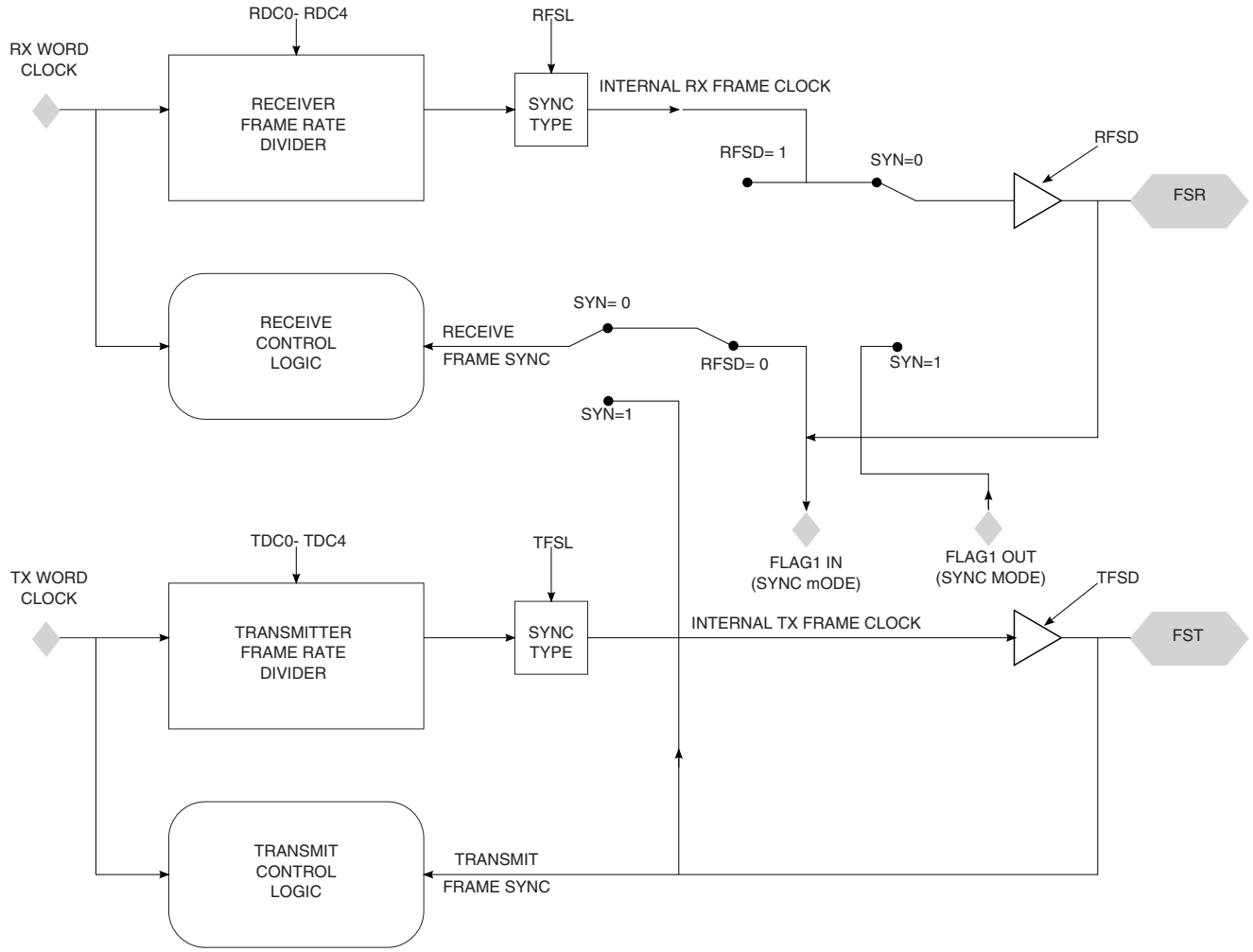
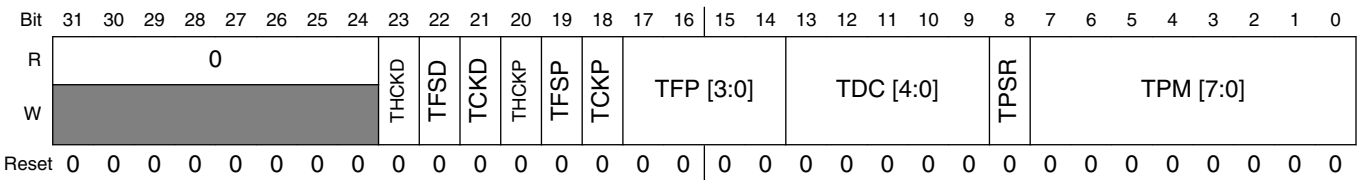


Figure 27-33. ESAI Frame Sync Generator Functional Block Diagram

Table 27-31. Transmitter High Frequency Clock Divider

TFP3-TFP0	Divide Ratio
0x0	1
0x1	2
0x2	3
0x3	4
...	...
0xF	16

Address: ESAI_TCCR is 5001_8000h base + D8h offset = 5001_80D8h



ESAI_TCCR field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved.
23 THCKD	ESAI_TCCR Transmit High Frequency Clock Direction. THCKD controls the direction of the HCKT pin. When THCKD is cleared, HCKT is an input; when THCKD is set, HCKT is an output (see Table 27-2).
22 TFSD	ESAI_TCCR Transmit Frame Sync Signal Direction. TFSD controls the direction of the FST pin. When TFSD is cleared, FST is an input; when TFSD is set, FST is an output (see Table 27-2).
21 TCKD	ESAI_TCCR Transmit Clock Source Direction. The Transmitter Clock Source Direction (TCKD) bit selects the source of the clock signal used to clock the transmit shift registers in the asynchronous mode (SYN=0) and the transmit shift registers and the receive shift registers in the synchronous mode (SYN=1). When TCKD is set, the internal clock source becomes the bit clock for the transmit shift registers and word length divider and is the output on the SCKT pin. When TCKD is cleared, the clock source is external; the internal clock generator is disconnected from the SCKT pin, and an external clock source may drive this pin (see Table 27-2).
20 THCKP	ESAI_TCCR Transmit High Frequency Clock Polarity. The Transmitter High Frequency Clock Polarity (THCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If THCKP is cleared the data and the frame sync are clocked out on the rising edge of the transmit high frequency bit clock and latched in on the falling edge of the transmit bit clock. If THCKP is set the falling edge of the transmit clock is used to clock the data out and frame sync and the rising edge of the transmit clock is used to latch the data and frame sync in.
19 TFSP	ESAI_TCCR Transmit Frame Sync Polarity. The Transmitter Frame Sync Polarity (TFSP) bit determines the polarity of the transmit frame sync signal. When TFSP is cleared, the frame sync signal polarity is positive, that is, the frame start is indicated by a high level on the frame sync pin. When TFSP is set, the frame sync signal polarity is negative, that is, the frame start is indicated by a low level on the frame sync pin.
18 TCKP	ESAI_TCCR Transmit Clock Polarity. The Transmitter Clock Polarity (TCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If TCKP is cleared the data and the frame sync are clocked out on the rising edge of the transmit bit clock and latched in on the falling edge of the transmit bit clock. If TCKP is set the falling edge of the transmit clock is used to clock the data out and frame sync and the rising edge of the transmit clock is used to latch the data and frame sync in.
17–14 TFP [3:0]	ESAI_TCCR Tx High Frequency Clock Divider. The TFP3-TFP0 bits control the divide ratio of the transmitter high frequency clock to the transmitter serial bit clock when the source of the high frequency clock and the bit clock is the internal ARM Core clock. When the HCKT input is being driven from an external high frequency clock, the TFP3-TFP0 bits specify an additional division ratio in the clock divider chain. Table 27-31 shows the specification for the divide ratio. Figure 27-32 shows the ESAI high frequency clock generator functional diagram.
13–9 TDC [4:0]	ESAI_TCCR Tx Frame Rate Divider Control. The TDC4-TDC0 bits control the divide ratio for the programmable frame rate dividers used to generate the transmitter frame clocks. In network mode, this ratio may be interpreted as the number of words per frame minus one. The divide ratio may range from 2 to 32 (TDC[4:0]=0x00001 to 0x11111) for network mode. A divide ratio of one (TDC[4:0]=0x00000) in network mode is a special case (on-demand mode).

Table continues on the next page...

ESAI_TCCR field descriptions (continued)

Field	Description
	In normal mode, this ratio determines the word transfer rate. The divide ratio may range from 1 to 32 (TDC[4:0]=0x00000 to 0x11111) for normal mode. In normal mode, a divide ratio of 1 (TDC[4:0]=0x00000) provides continuous periodic data word transfers. A bit-length frame sync (TFSL=1) must be used in this case. The ESAI frame sync generator functional diagram is shown in Figure 27-33
8 TPSR	ESAI_TCCR Transmit Prescaler Range. The TPSR bit controls a fixed divide-by-eight prescaler in series with the variable prescaler. This bit is used to extend the range of the prescaler for those cases where a slower bit clock is desired. When TPSR is set, the fixed prescaler is bypassed. When TPSR is cleared, the fixed divide-by-eight prescaler is operational (see Figure 27-32). The maximum internally generated bit clock frequency is Fsys/4; the minimum internally generated bit clock frequency is Fsys/(2 x 8 x 256 x 16)=Fsys/65536. (Do not use the combination TPSR=1, TPM7-TPM0=0x00, and TFP3-TFP0=0x0 which causes synchronization problems when using the internal ARM Core clock as source (TCKD=1 or THCKD=1))
7–0 TPM [7:0]	ESAI_TCCR Transmit Prescale Modulus Select. The TPM7-TPM0 bits specify the divide ratio of the prescale divider in the ESAI transmitter clock generator. A divide ratio from 1 to 256 (TPM[7:0]=0x00 to 0xFF) may be selected. The bit clock output is available at the transmit serial bit clock (SCKT) pin. The bit clock output is also available internally for use as the bit clock to shift the transmit and receive shift registers. The ESAI transmit clock generator functional diagram is shown in Figure 27-32.

27.5.16 Receive Control Register (ESAI_RCR)

The read/write Receive Control Register (ESAI_RCR) controls the ESAI receiver section. Interrupt enable bits for the receivers are provided in this control register. The receivers are enabled in this register (0,1,2 or 3 receivers can be enabled) if the input data pin is not used by a transmitter. Operating modes are also selected in this register.

Table 27-33. ESAI Receive Network Mode Selection

RMOD1	RMOD0	RDC4-RDC0	Receiver Network Mode
0	0	0x0-0x1F	Normal Mode
0	1	0x0	On-Demand Mode
0	1	0x1-0x1F	Network Mode
1	0	X	Reserved
1	1	0x0C	AC97

Table 27-34. ESAI Receive Slot and Word Length Selection

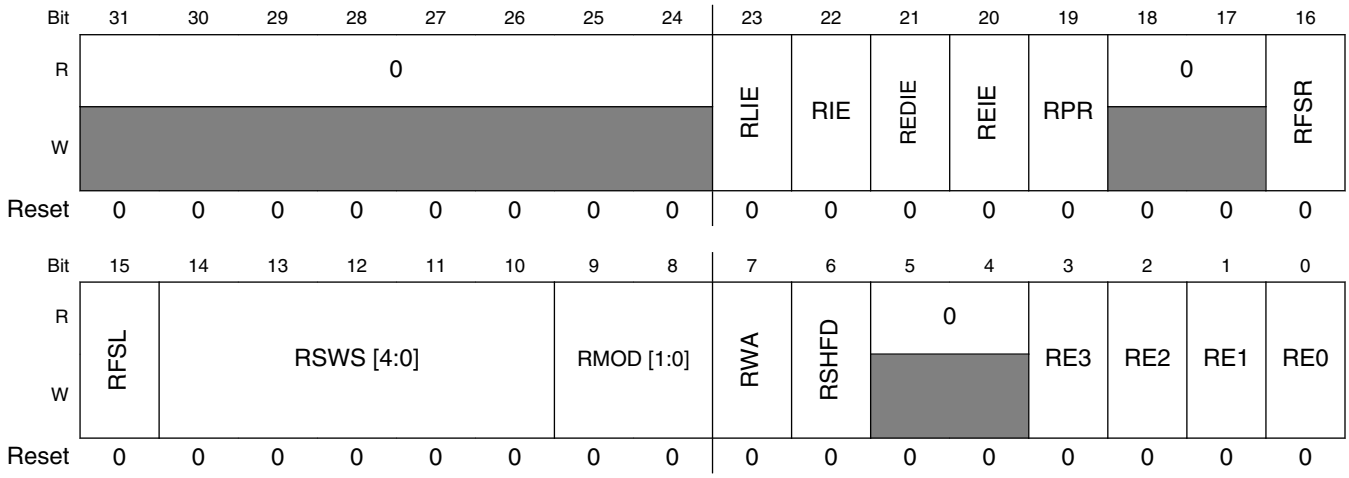
RSWS4	RSWS3	RSWS2	RSWS1	RSWS0	SLOT LENGTH	WORD LENGTH
0	0	0	0	0	8	8
0	0	1	0	0	12	8
0	0	0	0	1		12

Table continues on the next page...

Table 27-34. ESAI Receive Slot and Word Length Selection (continued)

RSWS4	RSWS3	RSWS2	RSWS1	RSWS0	SLOT LENGTH	WORD LENGTH
0	1	0	0	0	16	8
0	0	1	0	1		12
0	0	0	1	0		16
0	1	1	0	0	20	8
0	1	0	0	1		12
0	0	1	1	0		16
0	0	0	1	1		20
1	0	0	0	0	24	8
0	1	1	0	1		12
0	1	0	1	0		16
0	0	1	1	1		20
1	1	1	1	0		24
1	1	0	0	0	32	8
1	0	1	0	1		12
1	0	0	1	0		16
0	1	1	1	1		20
1	1	1	1	1		24
0	1	0	1	1	Reserved	
0	1	1	1	0		
1	0	0	0	1		
1	0	0	1	1		
1	0	1	0	0		
1	0	1	1	0		
1	0	1	1	1		
1	1	0	0	1		
1	1	0	1	0		
1	1	0	1	1		
1	1	1	0	0		
1	1	1	0	1		

Address: ESAI_RCR is 5001_8000h base + DCh offset = 5001_80DCh



ESAI_RCR field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved.
23 RLIE	ESAI_RCR Receive Last Slot Interrupt Enable. RLIE enables an interrupt after the last slot of a frame ended in network mode only. When RLIE is set the Core is interrupted after the last slot in a frame ended regardless of the receive mask register setting. When RLIE is cleared the receive last slot interrupt is disabled. Hardware and software reset clear RLIE. RLIE is disabled when RDC[4:0]=00000 (on-demand mode). The use of the receive last slot interrupt is described in ESAI Interrupt Requests.
22 RIE	ESAI_RCR Receive Interrupt Enable. The Core is interrupted when RIE and the RDF flag in the ESAI_SAISR status register are set. When RIE is cleared, this interrupt is disabled. Reading the receive data registers of the enabled receivers clears RDF, thus clearing the interrupt. Receive interrupts with exception have higher priority than normal receive data interrupts, therefore if exception occurs (ROE is set) and REIE is set, the ESAI requests an ESAI receive data with exception interrupt from the interrupt controller.
21 REDIE	ESAI_RCR Receive Even Slot Data Interrupt Enable. The REDIE control bit is used to enable the receive even slot data interrupts. If REDIE is set, the receive even slot data interrupts are enabled. If REDIE is cleared, the receive even slot data interrupts are disabled. A receive even slot data interrupt request is generated if REDIE is set and the REDF status flag in the ESAI_SAISR status register is set. Even time slots are all even-numbered time slots (0, 2, 4, etc.) when operating in network mode. The zero time slot is marked by the frame sync signal and is considered to be even. Reading all the data registers of the enabled receivers clears the REDF flag, thus servicing the interrupt. Receive interrupts with exception have higher priority than receive even slot data interrupts, therefore if exception occurs (ROE is set) and REIE is set, the ESAI requests an ESAI receive data with exception interrupt from the interrupt controller.
20 REIE	ESAI_RCR Receive Exception Interrupt Enable. When REIE is set, the Core is interrupted when both RDF and ROE in the ESAI_SAISR status register are set. When REIE is cleared, this interrupt is disabled. Reading the ESAI_SAISR status register followed by reading the enabled receivers data registers clears ROE, thus clearing the pending interrupt.
19 RPR	ESAI_RCR Receiver Section Personal Reset. The RPR control bit is used to put the receiver section of the ESAI in the personal reset state. The transmitter section is not affected. When RPR is cleared, the receiver section may operate normally. When RPR is set, the receiver section enters the personal reset state immediately. When in the personal reset state, the status bits are reset to the same state as after

Table continues on the next page...

ESAI_RCR field descriptions (continued)

Field	Description
	hardware reset.The control bits are not affected by the personal reset state.The receiver data pins are disconnected while in the personal reset state. **NOTE:** To leave the personal reset state by clearing RPR, the procedure described in ESAI Initialization Examples should be followed.
18–17 Reserved	This read-only field is reserved and always has the value zero. Reserved.
16 RFSR	ESAI_RCR Receiver Frame Sync Relative Timing. RFSR determines the relative timing of the receive frame sync signal as referred to the serial data lines, for a word length frame sync only. When RFSR is cleared the word length frame sync occurs together with the first bit of the data word of the first slot. When RFSR is set the word length frame sync starts one serial clock cycle earlier, that is, together with the last bit of the previous data word.
15 RFSL	ESAI_RCR Receiver Frame Sync Length. The RFSL bit selects the length of the receive frame sync to be generated or recognized. If RFSL is cleared, a word-length frame sync is selected. If RFSL is set, a 1-bit clock period frame sync is selected. Refer to Figure 27-30 for examples of frame length selection.
14–10 RSWS [4:0]	ESAI_RCR Receiver Slot and Word Select. The RSWS4-RSWS0 bits are used to select the length of the slot and the length of the data words being received through the ESAI. The word length must be equal to or shorter than the slot length. The possible combinations are shown in Table 27-34. See also the ESAI data path programming model in Figure 27-2 and Figure 27-3.
9–8 RMOD [1:0]	ESAI_RCR Receiver Network Mode Control. The RMOD1 and RMOD0 bits are used to define the network mode of the ESAI receivers, as shown in Table 27-33. In the normal mode, the frame rate divider determines the word transfer rate - one word is transferred per frame sync during the frame sync time slot, as shown in Figure 27-29. In network mode, it is possible to transfer a word for every time slot, as shown in Figure 27-29. For more details, see Modes of Operation. In order to comply with AC-97 specifications, RSWS4-RSWS0 should be set to 0x00011 (20-bit slot, 20-bit word); RFSL and RFSR should be cleared, and RDC4-RDC0 should be set to 0x0C (13 words in frame).
7 RWA	ESAI_RCR Receiver Word Alignment Control. The Receiver Word Alignment Control (RWA) bit defines the alignment of the data word in relation to the slot. This is relevant for the cases where the word length is shorter than the slot length. If RWA is cleared, the data word is assumed to be left-aligned in the slot frame. If RWA is set, the data word is assumed to be right-aligned in the slot frame. If the data word is shorter than the slot length, the data bits which are not in the data word field are ignored. For data word lengths of less than 24 bits, the data word is right-extended with zeroes before being stored in the receive data registers.
6 RSHFD	ESAI_RCR Receiver Shift Direction. The RSHFD bit causes the receiver shift registers to shift data in MSB first when RSHFD is cleared or LSB first when RSHFD is set (see Figure 27-2 and Figure 27-3).
5–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3 RE3	ESAI_RCR ESAI Receiver 3 Enable. When RE3 is set and TE2 is cleared, the ESAI receiver 3 is enabled and samples data at the SDO2/SDI3 pin. ESAI_TX2 and ESAI_RX3 should not be enabled at the same time (RE3=1 and TE2=1). When RE3 is cleared, receiver 3 is disabled by inhibiting data transfer into ESAI_RX3. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the ESAI_RX3 data register. If RE3 is set while some of the other receivers are already in operation, the first data word received in ESAI_RX3 will be invalid and must be discarded.
2 RE2	ESAI_RCR ESAI Receiver 2 Enable. When RE2 is set and TE3 is cleared, the ESAI receiver 2 is enabled and samples data at the SDO3/SDI2 pin. ESAI_TX3 and ESAI_RX2 should not be enabled at the same

Table continues on the next page...

ESAI_RCR field descriptions (continued)

Field	Description
	time (RE2=1 and TE3=1). When RE2 is cleared, receiver 2 is disabled by inhibiting data transfer into ESAI_RX2. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the ESAI_RX2 data register. If RE2 is set while some of the other receivers are already in operation, the first data word received in ESAI_RX2 will be invalid and must be discarded.
1 RE1	ESAI_RCR ESAI Receiver 1 Enable. When RE1 is set and TE4 is cleared, the ESAI receiver 1 is enabled and samples data at the SDO4/SDI1 pin. ESAI_TX4 and ESAI_RX1 should not be enabled at the same time (RE1=1 and TE4=1). When RE1 is cleared, receiver 1 is disabled by inhibiting data transfer into ESAI_RX1. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the ESAI_RX1 data register. If RE1 is set while some of the other receivers are already in operation, the first data word received in ESAI_RX1 will be invalid and must be discarded.
0 RE0	ESAI_RCR ESAI Receiver 0 Enable. When RE0 is set and TE5 is cleared, the ESAI receiver 0 is enabled and samples data at the SDO5/SDI0 pin. ESAI_TX5 and ESAI_RX0 should not be enabled at the same time (RE0=1 and TE5=1). When RE0 is cleared, receiver 0 is disabled by inhibiting data transfer into ESAI_RX0. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the ESAI_RX0 data register. If RE0 is set while some of the other receivers are already in operation, the first data word received in ESAI_RX0 will be invalid and must be discarded.

27.5.17 Receive Clock Control Register (ESAI_RCCR)

The read/write Receiver Clock Control Register (ESAI_RCCR) controls the ESAI receiver clock generator bit and frame sync rates, word length, and number of words per frame for the serial data. The ESAI_RCCR control bits are described in the following paragraphs.

NOTE

ARM Core clock is ipg_clk_esai in block ESAI which is from CCM's ahb_clk_root.

Table 27-36. Receiver High Frequency Clock Divider

RFP3-RFP0	Divide Ratio
0x0	1
0x1	2
0x2	3
0x3	4
...	...
0xF	16

Table 27-37. SCKR Pin Definition Table

Control Bits		SCKR PIN
SYN	RCKD	
0	0	SCKR input
0	1	SCKR output
1	0	IF0
1	1	OF0

Table 27-38. FSR Pin Definition Table

Control Bits			FSR Pin
SYN	TEBE	RFSD	
0	X	0	FSR input
0	X	1	FSR output
1	0	0	IF1
1	0	1	OF1
1	1	0	reserved
1	1	1	Transmitter Buffer Enable

Table 27-39. HCKR Pin Definition Table

Control Bits		HCKR PIN
SYN	RHCKD	
0	0	HCKR input
0	1	HCKR output
1	0	IF2
1	1	OF2

Address: ESAI_RCCR is 5001_8000h base + E0h offset = 5001_80E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RHCKD	RFSD	RCKD	RHCKP	RFSP	RCKP	RFP [3:0]		RDC [4:0]							RPSP	RPM [7:0]							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_RCCR field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved.
23 RHCKD	ESAI_RCCR Receiver High Frequency Clock Direction. The Receiver High Frequency Clock Direction (RHCKD) bit selects the source of the receiver high frequency clock when in the asynchronous mode (SYN=0) and the IF2/OF2 flag direction in the synchronous mode (SYN=1).

Table continues on the next page...

ESAI_RCCR field descriptions (continued)

Field	Description
	In the asynchronous mode, when RHCKD is set, the internal clock generator becomes the source of the receiver high frequency clock and is the output on the HCKR pin. In the asynchronous mode, when RHCKD is cleared, the receiver high frequency clock source is external; the internal clock generator is disconnected from the HCKR pin, and an external clock source may drive this pin. When RHCKD is cleared, HCKR is an input; when RHCKD is set, HCKR is an output. In the synchronous mode when RHCKD is set, the HCKR pin becomes the OF2 output flag. If RHCKD is cleared, the HCKR pin becomes the IF2 input flag. Refer to Table 27-1 and Table 27-39.
22 RFSD	ESAI_RCCR Receiver Frame Sync Signal Direction. The Receiver Frame Sync Signal Direction (RFSD) bit selects the source of the receiver frame sync signal when in the asynchronous mode (SYN=0) and the IF1/OF1/Transmitter Buffer Enable flag direction in the synchronous mode (SYN=1). In the asynchronous mode, when RFSD is set, the internal clock generator becomes the source of the receiver frame sync and is the output on the FSR pin. In the asynchronous mode, when RFSD is cleared, the receiver frame sync source is external; the internal clock generator is disconnected from the FSR pin, and an external clock source may drive this pin. In the synchronous mode when RFSD is set, the FSR pin becomes the OF1 output flag or the Transmitter Buffer Enable, according to the TEBE control bit. If RFSD is cleared, the FSR pin becomes the IF1 input flag. Refer to Table 27-1 and Table 27-38 .
21 RCKD	ESAI_RCCR Receiver Clock Source Direction. The Receiver Clock Source Direction (RCKD) bit selects the source of the clock signal used to clock the receive shift register in the asynchronous mode (SYN=0) and the IF0/OF0 flag direction in the synchronous mode (SYN=1). In the asynchronous mode, when RCKD is set, the internal clock source becomes the bit clock for the receive shift registers and word length divider and is the output on the SCKR pin. In the asynchronous mode when RCKD is cleared, the clock source is external; the internal clock generator is disconnected from the SCKR pin, and an external clock source may drive this pin. In the synchronous mode when RCKD is set, the SCKR pin becomes the OF0 output flag. If RCKD is cleared, the SCKR pin becomes the IF0 input flag. Refer to Table 27-1 and Table 27-37.
20 RHCKP	ESAI_RCCR Receiver High Frequency Clock Polarity. The Receiver High Frequency Clock Polarity (RHCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If RHCKP is cleared the data and the frame sync are clocked out on the rising edge of the receive high frequency bit clock and the frame sync is latched in on the falling edge of the receive bit clock. If RHCKP is set the falling edge of the receive clock is used to clock the data and frame sync out and the rising edge of the receive clock is used to latch the frame sync in.
19 RFSP	ESAI_RCCR Receiver Frame Sync Polarity. The Receiver Frame Sync Polarity (RFSP) determines the polarity of the receive frame sync signal. When RFSP is cleared the frame sync signal polarity is positive, that is, the frame start is indicated by a high level on the frame sync pin. When RFSP is set the frame sync signal polarity is negative, that is, the frame start is indicated by a low level on the frame sync pin.
18 RCKP	The Receiver Clock Polarity (RCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If RCKP is cleared the data and the frame sync are clocked out on the rising edge of the receive bit clock and the frame sync is latched in on the falling edge of the receive bit clock. If RCKP is set the falling edge of the receive clock is used to clock the data and frame sync out and the rising edge of the receive clock is used to latch the frame sync in.
17–14 RFP [3:0]	ESAI_RCCR Rx High Frequency Clock Divider. The RFP3-RFP0 bits control the divide ratio of the receiver high frequency clock to the receiver serial bit clock when the source of the receiver high frequency clock and the bit clock is the internal Arm Core clock. When the HCKR input is being driven from an external high frequency clock, the RFP3-RFP0 bits specify an additional division ration in the clock divider chain. Table 27-36 provides the specification of the divide ratio. Figure 27-32 shows the ESAI high frequency generator functional diagram.

Table continues on the next page...

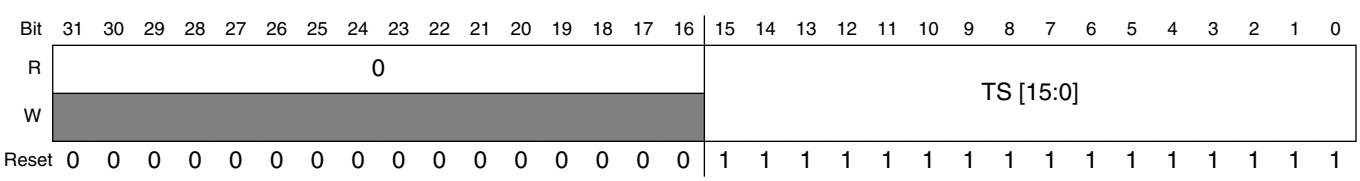
ESAI_RCCR field descriptions (continued)

Field	Description
13–9 RDC [4:0]	ESAI_RCCR Rx Frame Rate Divider Control. The RDC4-RDC0 bits control the divide ratio for the programmable frame rate dividers used to generate the receiver frame clocks. In network mode, this ratio may be interpreted as the number of words per frame minus one. The divide ratio may range from 2 to 32 (RDC[4:0]=0x00001 to 0x11111) for network mode. A divide ratio of one (RDC[4:0]=0x00000) in network mode is a special case (on-demand mode). In normal mode, this ratio determines the word transfer rate. The divide ratio may range from 1 to 32 (RDC[4:0]=0x00000 to 0x11111) for normal mode. In normal mode, a divide ratio of one (RDC[4:0]=0x00000) provides continuous periodic data word transfers. A bit-length frame sync (RFSL=1) must be used in this case. The ESAI frame sync generator functional diagram is shown in Figure 27-33 .
8 RPSP	ESAI_RCCR Receiver Prescaler Range. The RPSR controls a fixed divide-by-eight prescaler in series with the variable prescaler. This bit is used to extend the range of the prescaler for those cases where a slower bit clock is desired. When RPSR is set, the fixed prescaler is bypassed. When RPSR is cleared, the fixed divide-by-eight prescaler is operational (see Figure 27-32). The maximum internally generated bit clock frequency is Fsys/4, the minimum internally generated bit clock frequency is Fsys/(2 x 8 x 256 x 16)=Fsys/65536. (Do not use the combination RPSR=1 and RPM7-RPM0 =0x00, which causes synchronization problems when using the internal Core clock as source (RHCKD=1 or RCKD=1))
7–0 RPM [7:0]	ESAI_RCCR Receiver Prescale Modulus Select. The RPM7-RPM0 bits specify the divide ratio of the prescale divider in the ESAI receiver clock generator. A divide ratio from 1 to 256 (RPM[7:0]=0x00 to 0xFF) may be selected. The bit clock output is available at the receiver serial bit clock (SCKR) pin. The bit clock output is also available internally for use as the bit clock to shift the receive shift registers. The ESAI receive clock generator functional diagram is shown in Figure 27-32.

27.5.18 Transmit Slot Mask Register A (ESAI_TSMA)

The Transmit Slot Mask Register A together with Transmit Slot Mask Register B (ESAI_TSMA and ESAI_TSMB) are two read/write registers used by the transmitters in network mode to determine for each slot whether to transmit a data word and generate a transmitter empty condition (TDE=1), or to tri-state the transmitter data pins. Fields ESAI_TSMA [TS[15:0]] and ESAI_TSMB [TS[31:16]] are concatenated to form the 32-bit field TS[31:0]. Bit number n in TS[31:0] is the enable/disable control bit for transmission in slot number n.

Address: ESAI_TSMA is 5001_8000h base + E4h offset = 5001_80E4h



ESAI_TSMA field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–0 TS [15:0]	When bit number N in ESAI_TSMA is cleared, all the transmit data pins of the enabled transmitters are tri-stated during transmit time slot number N. The data is still transferred from the transmit data registers to the transmit shift registers but neither the TDE nor the TUE flags are set. This means that during a disabled slot, no transmitter empty interrupt is generated. The Core is interrupted only for enabled slots. Data that is written to the transmit data registers when servicing this request is transmitted in the next enabled transmit time slot. When bit number N in ESAI_TSMA register is set, the transmit sequence is as usual: data is transferred from the TX registers to the shift registers and transmitted during slot number N, and the TDE flag is set. Using the slot mask in ESAI_TSMA does not conflict with using TSR. Even if a slot is enabled in ESAI_TSMA, the user may choose to write to TSR instead of writing to the transmit data registers TXn. This causes all the transmit data pins of the enabled transmitters to be tri-stated during the next slot. Data written to the ESAI_TSMA affects the next frame transmission. The frame being transmitted is not affected by this data and would comply to the last ESAI_TSMA setting. Data read from ESAI_TSMA returns the last written data. After hardware or software reset, the ESAI_TSMA register is preset to 0x0000FFFF, which means that all 16 possible slots are enabled for data transmission. When operating in normal mode, bit 0 of the ESAI_TSMA register must be set, otherwise no output is generated.

27.5.19 Transmit Slot Mask Register B (ESAI_TSMB)

The Transmit Slot Mask Register B together with Transmit Slot Mask Register A (ESAI_TSMA and ESAI_TSMB) are two read/write registers used by the transmitters in network mode to determine for each slot whether to transmit a data word and generate a transmitter empty condition (TDE=1), or to tri-state the transmitter data pins. Fields ESAI_TSMA [TS[15:0]] and ESAI_TSMB [TS[31:16]] are concatenated to form the 32-bit field TS[31:0]. Bit number n in TS[31:0] is the enable/disable control bit for transmission in slot number n.

Address: ESAI_TSMB is 5001_8000h base + E8h offset = 5001_80E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TS [31:16]															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ESAI_TSMB field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

ESAI_TSMB field descriptions (continued)

Field	Description
15–0 TS [31:16]	When bit number N in ESAI_TSMB is cleared, all the transmit data pins of the enabled transmitters are tri-stated during transmit time slot number N. The data is still transferred from the transmit data registers to the transmit shift registers but neither the TDE nor the TUE flags are set. This means that during a disabled slot, no transmitter empty interrupt is generated. The Core is interrupted only for enabled slots. Data that is written to the transmit data registers when servicing this request is transmitted in the next enabled transmit time slot. When bit number N in ESAI_TSMB register is set, the transmit sequence is as usual: data is transferred from the TX registers to the shift registers and transmitted during slot number N, and the TDE flag is set. Using the slot mask in ESAI_TSMB does not conflict with using TSR. Even if a slot is enabled in TSMB, the user may chose to write to TSR instead of writing to the transmit data registers TXn. This causes all the transmit data pins of the enabled transmitters to be tri-stated during the next slot. Data written to the ESAI_TSMB affects the next frame transmission. The frame being transmitted is not affected by this data and would comply to the last ESAI_TSMB setting. Data read from ESAI_TSMB returns the last written data. After hardware or software reset, the ESAI_TSMB register is preset to 0x0000FFFF, which means that all 16 possible slots are enabled for data transmission.

27.5.20 Receive Slot Mask Register A (ESAI_RSMA)

The Receive Slot Mask Register A together with Receive Slot Mask Register B (ESAI_RSMA and ESAI_RSMB) are two read/write registers used by the receiver in network mode to determine for each slot whether to receive a data word and generate a receiver full condition (RDF=1), or to ignore the received data. Fields ESAI_RSMA [RS[15:0]] and ESAI_RSMB [RS31:16]] are concatenated to form the 32-bit field RS[31:0]. Bit number n in RS[31:0] is an enable/disable control bit for receiving data in slot number n.

Address: ESAI_RSMA is 5001_8000h base + ECh offset = 5001_80ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																RS [15:0]															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ESAI_RSMA field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–0 RS [15:0]	When bit number N in the ESAI_RSMA register is cleared, the data from the enabled receivers input pins are shifted into their receive shift registers during slot number N. The data is not transferred from the receive shift registers to the receive data registers, and neither the RDF nor the ROE flag is set. This means that during a disabled slot, no receiver full interrupt is generated. The Core is interrupted only for enabled slots.

Table continues on the next page...

ESAI_RSMA field descriptions (continued)

Field	Description
	When bit number N in the ESAI_RSMA is set, the receive sequence is as usual: data which is shifted into the enabled receivers shift registers is transferred to the receive data registers and the RDF flag is set. Data written to the ESAI_RSMA affects the next received frame. The frame being received is not affected by this data and would comply to the last ESAI_RSMA setting. Data read from ESAI_RSMA returns the last written data. After hardware or software reset, the ESAI_RSMA register is preset to 0x0000FFFF, which means that all 16 possible slots are enabled for data reception. When operating in normal mode, bit 0 of the ESAI_RSMA register must be set to one, otherwise no input is received.

27.5.21 Receive Slot Mask Register B (ESAI_RSMB)

The Receive Slot Mask Register B together with Receive Slot Mask Register A (ESAI_RSMA and ESAI_RSMB) are two read/write registers used by the receiver in network mode to determine for each slot whether to receive a data word and generate a receiver full condition (RDF=1), or to ignore the received data. Fields ESAI_RSMA [RS[15:0]] and ESAI_RSMB [RS31:16]] are concatenated to form the 32-bit field RS[31:0]. Bit number n in RS[31:0] is an enable/disable control bit for receiving data in slot number n.

Address: ESAI_RSMB is 5001_8000h base + F0h offset = 5001_80F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																RS [31:16]															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ESAI_RSMB field descriptions

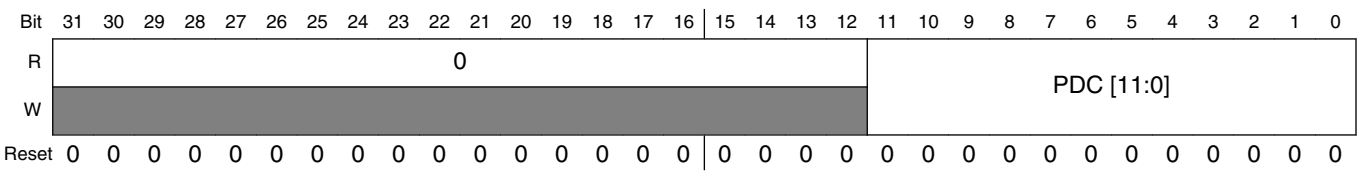
Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–0 RS [31:16]	When bit number N in the ESAI_RSMB register is cleared, the data from the enabled receivers input pins are shifted into their receive shift registers during slot number N. The data is not transferred from the receive shift registers to the receive data registers, and neither the RDF nor the ROE flag is set. This means that during a disabled slot, no receiver full interrupt is generated. The Core is interrupted only for enabled slots. When bit number N in the ESAI_RSMB is set, the receive sequence is as usual: data which is shifted into the enabled receivers shift registers is transferred to the receive data registers and the RDF flag is set. Data written to the ESAI_RSMB affects the next received frame. The frame being received is not affected by this data and would comply to the last ESAI_RSMB setting. Data read from ESAI_RSMB returns the last written data. After hardware or software reset, the ESAI_RSMB register is preset to 0x0000FFFF, which means that all 16 possible slots are enabled for data reception.

27.5.22 Port C Direction Register (ESAI_PRRC)

There are two registers to control the ESAI personal reset status: Port C Direction Register (ESAI_PRRC) and Port C Control Register (ESAI_PCRC).

The read/write 32-bit Port C Direction Register (ESAI_PRRC) in conjunction with the Port C Control Register (ESAI_PCRC) controls the functionality of the ESAI personal reset state. Table 27-46 provides the port pin configurations. Hardware and software reset clear all ESAI_PRRC bits.

Address: ESAI_PRRC is 5001_8000h base + F8h offset = 5001_80F8h



ESAI_PRRC field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11–0 PDC [11:0]	See Table 27-46.

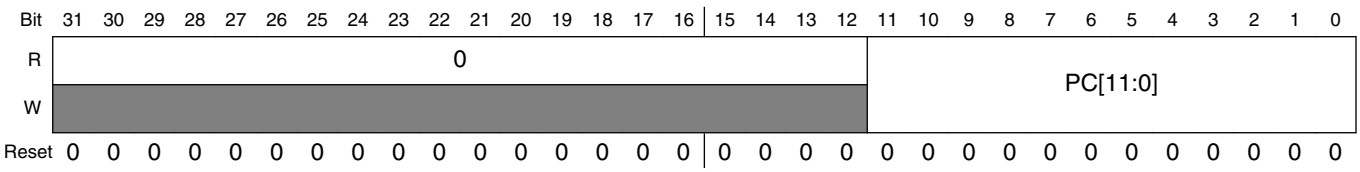
27.5.23 Port C Control Register (ESAI_PCRC)

The read/write 32-bit Port C Control Register (ESAI_PCRC) in conjunction with the Port C Direction Register (ESAI_PRRC) controls the functionality of the ESAI personal reset state. Each of the PC(11:0) bits controls the functionality of the corresponding port pin. Table 27-46 provides the port pin configurations. Hardware and software reset clear all ESAI_PCRC bits.

Table 27-46. PCRC and PRRC Bits Functionality

PDC[i]	PC[i]	Port Pin[i] Function
0	0	Disconnected
1	1	ESAI

Address: ESAI_PCRC is 5001_8000h base + FCh offset = 5001_80FCh



ESAI_PCRC field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11–0 PC[11:0]	See Table 27-46.

Chapter 28
Enhanced SDRAM Controller (ESDCTL)

28.1 Introduction

Figure 28-1 shows the block diagram of the ESDCTL.

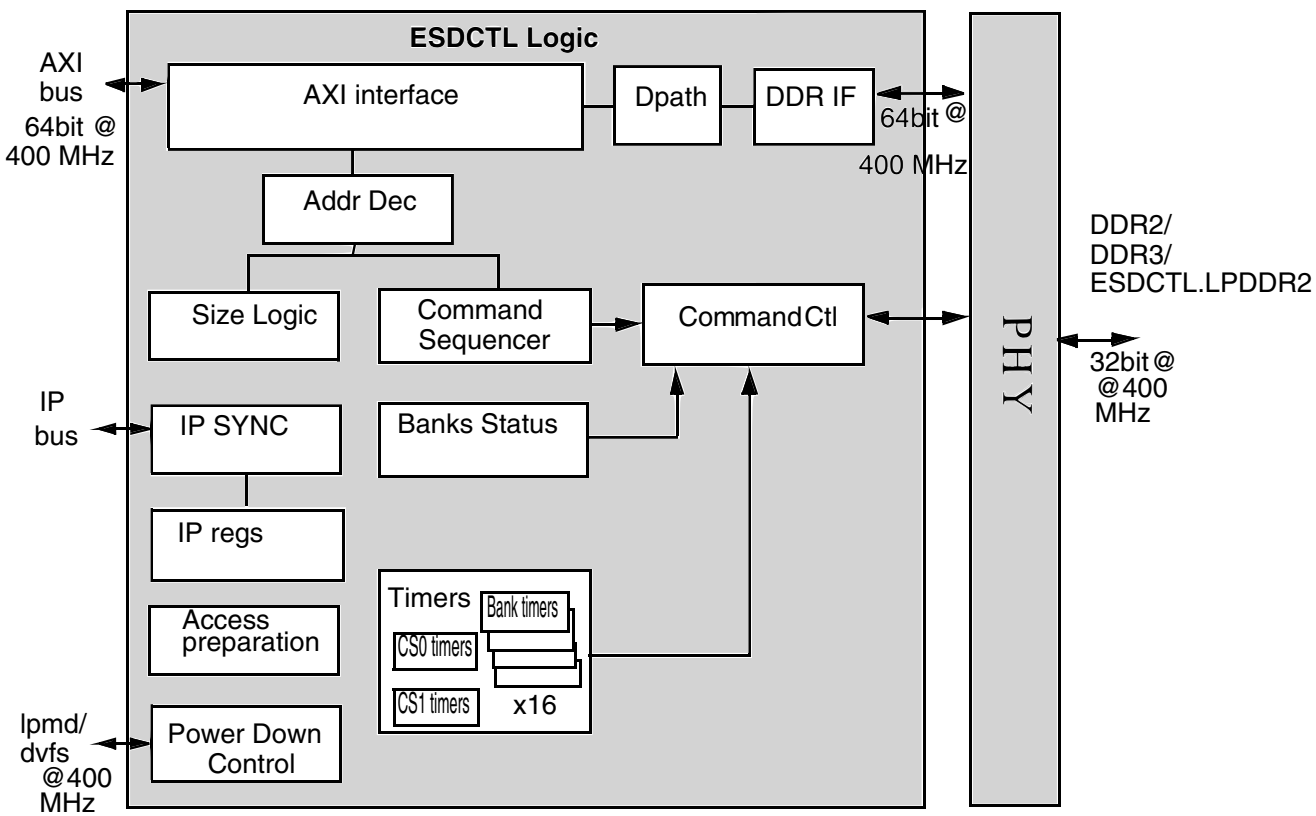


Figure 28-1. ESDCTL Block Diagram

28.2 Overview

The ESDCTL is a configurable high performance and optimized SDRAM controller that supports DDR2, DDR3 and LPDDR2-S memories.

The ESDCTL is composed from the logic part and PHY part. The logic is responsible for the command preparation and data path while the PHY is responsible for the timing adjustment using a special calibration mechanism to ensure data capture margin at clock rate of up to 400MHz.

28.3 Features

28.3.1 ESDCTL Logic Features

- 64bits AXI I/F operating at 400MHz.
- Configurable 2GB memory space (1GB per chip select or 2GB in chip select 0).
- 4/8 banks memory devices.
- Column size between 9 to 11 bits for DDR2/DDR3 and 8 to 12 bits for LPDDR2.
- Row size between 11 to 16 bits.
- x16 or x32 bits memory data width.
- Burst length 4 or 8 for DDR2 devices (best performance for BL4).
- Burst length 4 or 8 for LPDDR2 devices (best performance for BL4).
- Burst length 8 for DDR3 devices.
- DDR2 allowed frequency 125 MHz to 400MHz.
- DDR3 allowed frequency 303 MHz to 400 MHz.
- LPDDR2 allowed frequency 10 MHz to 400 MHz.
- Power down DDR2/DDR3/LPDDR2 memories when inactive (per chip select).
- Several different ODT scheme support (For DDR2/DDR3 only).
- MIF3 - future accesses preparation mechanism (Increase hit rate).
- Bank Interleaving mode.
- Configurable refresh scheme, either by a 32KHz clock or by a special counter.
- Configurable timing parameters.
- Power saving / Frequency change
  - External DVFS / LPMD request.
  - Power down counter per rank after configurable amount of idle cycles.
  - Precharge counter per rank after configurable idle cycles.
- Debug Features

- 32bit Read command counter with reset option (used for bus utilization measurement).
- 32bit Write command counter with reset option (used for bus utilization measurement).
- Latency hiding disable.

28.3.2 PHY Features

- ZQ HW and SW calibration.
- Read data calibration. Adjustment of read DQS with read data byte.
- Read DQS gating calibration. Adjustment of DQS gate with read preamble window.
- Write data calibration. Adjustment of write DQS with write data byte.
- Write leveling calibration. Adjustment of write DQS with SDCLK.
- Read fine tuning. Adjustment of up to 7 delay line units for each read data bit.
- Write fine tuning. Adjustment of up to 3 delay line units for each read data bit.

28.4 AXI Restrictions

- One AXI access is handled at a time (One ID system).
- Out of order accesses are not supported. That means that data interleaving of two different IDs is not allowed in write cycles.
- The controller will not provide data interleaving in the AXI read data bus.
- Exclusive AXI accesses are not supported.
- Atomic/Locked access are not supported.
- All accesses are treated as non-bufferable and non-cachable accesses. Bufferable and cachable accesses should be handeled by the ESDCTL bus master.
- Bursts of up to 8 double-words (64 bits), is supported, for incr and wrap burst types. AWSIZE/ARSIZE are 2 bits wide. AWLEN/ARLEN are 3 bits wide.
- ARRAEDY and AWREADY are all defaulted to high in idle state. Accepting a read or a write will lower the other ready as well until the controller is ready for another address for latency hiding. If both a read and a write are valid at the same cycle, the read is serviced first, then the write, and only after the write is done the readies will go back up.
- The controller BRESP and RRESP signals are returning OKAY by default (No error handling).

28.5 Functional Description

28.5.1 Address Decoding

ESDCTL supports two chip selects, each of which can be used to reach a maximum of 1 GB (a total of 2 GB for both chip selects).

The incoming addresses arrive as 32-bit through the AXI bus, then the ESDCTL Control Register (ESDCTL_ESDCTL) decodes this 32-bit into the following:

- chip select
- bank number
- row number
- column number

For this matter a set of parameters are required:

1. 4 or 8 banks in device - ESDCTL Timing Miscellaneous Register (ESDCTL_ESDMISC) register field: DDR_4_BANK.
2. Memory port size 16 or 32 - ESDCTL Control Register (ESDCTL_ESDCTL) register field: DSIZ.
3. Bank interleaved on or off - ESDCTL Timing Miscellaneous Register (ESDCTL_ESDMISC) register field: BI.
4. Column size - ESDCTL Control Register (ESDCTL_ESDCTL) register field: COL.
5. Row size - ESDCTL Control Register (ESDCTL_ESDCTL) register field: ROW.

In the table below, decoding with no bank interleaving is shown for 16 and 32-bit SDRAM (assuming: bank size = 3, row size = 15, column size = 10).

Table 28-1. Decoding with no bank interleaving for 16 and 32 bit SDRAM

ARM Platform ADDRESS	16-bit SDRAM	32-bit SDRAM
A29	-	BANK[2]
A28	BANK[2]	BANK[1]
A27	BANK[1]	BANK[0]
A26	BANK[0]	ROW[14]
A25	ROW[14]	ROW[13]
A24	ROW[13]	ROW[12]
A23	ROW[12]	ROW[11]
A22	ROW[11]	ROW[10]
A21	ROW[10]	ROW[9]
A20	ROW[9]	ROW[8]
A19	ROW[8]	ROW[7]

Table continues on the next page...

Table 28-1. Decoding with no bank interleaving for 16 and 32 bit SDRAM (continued)

ARM Platform ADDRESS	16-bit SDRAM	32-bit SDRAM
A18	ROW[7]	ROW[6]
A17	ROW[6]	ROW[5]
A16	ROW[5]	ROW[4]
A15	ROW[4]	ROW[3]
A14	ROW[3]	ROW[2]
A13	ROW[2]	ROW[1]
A12	ROW[1]	ROW[0]
A11	ROW[0]	COL[9]
A10	COL[9]	COL[8]
A9	COL[8]	COL[7]
A8	COL[7]	COL[6]
A7	COL[6]	COL[5]
A6	COL[5]	COL[4]
A5	COL[4]	COL[3]
A4	COL[3]	COL[2]
A3	COL[2]	COL[1]
A2	COL[1]	COL[0]
A1	COL[0]	-
A0	-	-

Now displaying the same table for bank interleaving on:

(Assuming: bank size = 3, row size = 15, column size = 10)

Table 28-2. Decoding with bank interleaving for 16 and 32 bit SDRAM

ARM Platform ADDRESS	16-bit SDRAM	32-bit SDRAM
A29	-	ROW[14]
A28	ROW[14]	ROW[13]
A27	ROW[13]	ROW[12]
A26	ROW[12]	ROW[11]
A25	ROW[11]	ROW[10]
A24	ROW[10]	ROW[9]
A23	ROW[9]	ROW[8]
A22	ROW[8]	ROW[7]
A21	ROW[7]	ROW[6]
A20	ROW[6]	ROW[5]

Table continues on the next page...

Table 28-2. Decoding with bank interleaving for 16 and 32 bit SDRAM (continued)

ARM Platform ADDRESS	16-bit SDRAM	32-bit SDRAM
A19	ROW[5]	ROW[4]
A18	ROW[4]	ROW[3]
A17	ROW[3]	ROW[2]
A16	ROW[2]	ROW[1]
A15	ROW[1]	ROW[0]
A14	ROW[0]	BANK[2]
A13	BANK[2]	BANK[1]
A12	BANK[1]	BANK[0]
A11	BANK[0]	COL[9]
A10	COL[9]	COL[8]
A9	COL[8]	COL[7]
A8	COL[7]	COL[6]
A7	COL[6]	COL[5]
A6	COL[5]	COL[4]
A5	COL[4]	COL[3]
A4	COL[3]	COL[2]
A3	COL[2]	COL[1]
A2	COL[1]	COL[0]
A1	COL[0]	-
A0	-	-

Selection of Chip select is decoded from the upper 4 bits of the AXI address.

Per the value of 2 vias: 1. sdram_base_cs0_via 2.sdram_base_cs1_via.

Chip select 0 is selected if:

sdram_base_cs0_via <= axi_address[28:31] <= sdram_base_cs0_via +3

Chip select 1is selected if:

sdram_base_cs1_via <= axi_address[28:31] <= sdram_base_cs1_via +3

NOTE

In case of an access to an uninitialized or disconnected chip select, the controller might behave unexpectedly.

28.5.2 Address Mirroring

When enabling this feature, the address bits A3, A4, A5, A6, A7, A8, B0, B1 behave differently according to the referenced chip select.

The mirroring is done in according to the referenced chip select, as shown in table 1-24.

Table 28-3. Address Mirroring

Controller pin	Chip select 0 pin	Chip select 1 pin
A3	A3	A4
A4	A4	A3
A5	A5	A6
A6	A6	A5
A7	A7	A8
A8	A8	A7
B0	B0	B1
B1	B1	B0

28.5.3 MIF3 - Optimization Strategy

The target of this optimization is to increase the data throughput in a certain period of time.

Maximum throughput is achieved when AXI access are back to back hits; in that case the controller issues only RD/WR commands and does not have to issue row preparation commands ACTIVE/ PRECHARGE. The controller does this by looking on the future accesses and, if required, it prepares their bank/row. When the access turn is reached, it is a hit access, and no time is wasted on PRECHARGE/ACTIVE commands.

The controller has 3 levels of pending accesses:

1. Access in first stage of pipeline
2. Valid access on AXI bus either read channel or write channel
3. Valid access on special bus from arbitration, which is chosen by the arbitration as the next miss access in its buffers

The controller can enable/disable this sources by configuration (ESDCTL_ESDMISC register field MIF3_MODE).

28.5.4 Auto Refresh Behavior

The ESDCTL supports various automatic refresh options which can be configured via ESDCTL_ESDREF register.

The periodic auto refresh can be triggered by the following clocks:

- 32KHz clock
- 64KHz clock
- Fast clock which is similar to the frequency which the controller is operated (equal to sdclk frequency).

This flexibility allows the system to choose a desired refresh scheme that will not delay the AXI accesses in each refresh cycle.

The following is an example that specifies 4 options to configure the ESDCTL in order to meet a refresh rate of 3.9us (tREFI):

Option number	Description	REFR	REF_SEL	REF_CNT	DDR hang time
1	Issue 8 refresh commands every 31,250 ns	0x7 (8 refreshes)	0x2 (64KHz)	not needed	tRFC * 8
2	Issue 4 refresh commands every 15,625ns	0x3 (4 refreshes)	0x1(32KHz)	not needed	tRFC * 4
3	Issue 2 refresh commands every 7800ns	0x1(2 refreshes)	0x3 (fast counter)	7800/2.5 = 3120 (0xC30)	tRFC * 2
4	Issue 1 refresh command every 3900 ns	0x0 (1 refresh)	0x3 (fast counter)	3900/2.5 = 1560(0x618)	tRFC

As shown in the table above all options meet the memory requirement of a refresh command every 3.9 us, the difference is how often a refresh cycle occurs and what is the DDR hang time for each refresh cycle.

28.5.5 Initialization Information

The controller gets out of reset while it is disabled, meaning that no clock is driven to the memory and all I/F signals to the memory are at non-active values.

The following are the required steps to activate the controller properly:

1. Set bit 15 on ESDCTL_ESDSCR register - configuration request (because the controller is disabled there is no need to poll configuration acknowledge bit).

2. Configure registers ESDCTL_ESDCFG0, ESDCTL_ESDCFG1, ESDCTL_ESDCFG2 and ESDCTL_ESDOTC- timing parameters.
3. Configure register ESDCTL_ESDMISC with miscellaneous parameters.
4. Configure ESDCTL_ESDOR with requires out of reset delays.
5. Configure ESDCTL with memory parameters and enable the working chip selects. At this point the controller start counting the required time according to configuration. DDR2 and DDR3 has different initialization as described on DDR2 SDRAM Specification JESD79-2E (April 2008) and on DDR3 SDRAM Specification JESD79-3C (April 2008).
6. Issue the required commands for memory initialization (REF,LMR,PRE) by using ESDCTL_ESDSCR register.
7. DDR2 and DDR3 has different command sequence as described on DDR2 SDRAM Specification JESD79-2E (April 2008) and on DDR3 SDRAM Specification JESD79-3C (April 2008).
8. At this point the memory model is ready to be used, but there are still few registers to set.
9. Configure ESDCTL_ESDPDC register - used to enable power saving modes of memory.
10. Enable ODT / ZQ functions - registers can be found in PHY spec.
11. Configure ESDCTL_ESDREF - to enable periodic refreshes.
12. Configure PHY delay parameters (using hardware or software calibrations)
13. Clear bit 15 on ESDCTL_ESDSCR register - configuration request.

Now controller is ready to accept AXI requests.

28.5.6 LPMD/DVFS requests

The controller has LPMD req/ack and DVFS req/ack interface that is used for entering the DDR into self-refresh mode which allow changing/stopping the clock to the memory. At this mode the memory does not need periodic refresh commands.

The controller handles LPMD/DVFS in the same manner.

Upon the assertion of LPMD/DVFS request the controller do the following:

1. Blocks new AXI access from entering the controller.
2. Completing all the acknowledged AXI accesses.
3. If banks are not idle issue precharge all commands when timing allow.
4. Lowers clock enable signal while giving refresh command to the memory, this is done after tRP/tRPA pass from the precharge all command (if issued).
5. Stop clock to the memory after tCKSRE from self refresh entry.

6. Assert LPMD/DVFS ack, which means the clock to the contoller can be shut off or changed.

Upon the deassertion of LPMD/DVFS request the following is done:

1. LPMD/DVFS acknowledge is deasserted.
2. Memory clock turned on again.
3. After tCKSRX pass from clock renewal the CKE signal is asserted again.
4. After tXS pass from CKE assertion a refresh command is given to the memory.
5. If ZQ calibration is disabled skip to step 7.
6. tRFC is counted from the refresh command and a long ZQ command is asserted.
7. tZQoper idle cycles are counted after the ZQ command.
8. tDLLK should pass from the CKE assertion, then the controller return to normal operation.

28.5.7 Writing to ESDCTL configuration registers

In order to modify ESDCTL's internal configuration registers few steps are required as following:

1. Issue configuration request - set bit 15 on ESDCTL_ESDSCR register.
2. Poll configuration acknowledge until high - bit 14 on ESDCTL_ESDSCR register.
3. At this point the controller is in configuration mode and will not treat R/W requests from M4IF.
4. Write to ESDCTL configuration registers.
5. Clear bit 15 on the ESDCTL_ESDSCR register - this will bring the controller back to normal mode.

NOTE

When the controller is in LPMD or DVFS mode, writing to internal configuration registers is prohibited.

NOTE

Controller can enter these modes (LPMD or DVFS) as a result of:

- CCM - Manual DVFS / LPMD.
- M4IF - Manual DVFS / LPMD.
- M4IF - Automatic power saving.

28.5.8 Warm reset

The controller supports warm reset. In case of warm reset all the registers in controller will receive the reset except those that are essential for returning to normal operation without repeating the initialization sequence or losing data stored on the memory.

In order to use warm reset the following should be done:

1. The controller should be brought to self refresh mode. This can be done by either LPMD or DVFS requests.
2. After receiving an acknowledge the warm reset signal can be asserted.
3. While the warm reset signal is asserted it is safe to deassert the rst_b signal.
4. After rst_b is asserted the warm reset signal can be deasserted.
5. LPMD/ DVFS can now be exited by deasserting the LPMD/DVFS request signal.

28.5.9 Software reset

The controller allows the user to issue software reset to FF in controller except FF that are essential for returning to normal operation without repeating the initialization sequence or losing data stored in the memory. Issuing software reset is done as follows:

1. Assert configuration request, bit 15 in ESDCTL_ESDSCR register.
2. Wait for configuration acknowledge assertion, bit 14 in ESDCTL_ESDSCR register.
3. Issue soft reset, bit 1in ESDCTL_ESDMISC register.
4. Deassert configuration request, bit 15 in ESDCTL_ESDSCR register.
5. Normal operation can be resumed.

28.5.10 Power Saving modes

Various DDR power saving modes are supported by the controller as following:

NOTE

Those modes may dramatically decrease the power consumption of DDR memories. At default those modes are disabled.

1. Self refresh entry to the entire DDR device (for both chip selects 0 and 1) can be activated as follows:
   - HW handshaking with the clock module in the system
   - SW handshaking by setting the field MCR0[FLPMD] in the M4IF block.
   - Automatic entry by configuring MCR1[FPST] field in the M4IF block. Define the amount number of idle cycles before entering self refresh automatically.

2. Automatic active/precharge power down entry to a specific chip select (it is possible to enter certain CS to low power consumption while the second chip select is activated). Can be activated by configuring the ESDCTL_ESDPDC register as following:
   - PWDT_0/PWDT_1 - define the number of idle cycles before entering power down, can be different value per chip select.
   - SLOW_PD - In case of DDR2 memory is configured to use slow active power down then this bit should be set as well. In case of DDR3 memory is configured to use slow precharge power down then this bit should be set as well.
   - BOTH_CS_PS - The controller can either set each chip select independently to power down, according to its idle state, or set both chip selects to power down only if both in idle state for the configured period.
   - Few paramters must be configured in addition:
     - Timing parameters at ESDCTL_ESDCFG0[tXP and tXPDLL].
     - ODT timing at ESDCTL_ESDOTC[tAOFPD, tAONPD, tANPD and tAXPD]
3. Automatic precharge of all DDR banks to a specific chip select . Can be activated be configuring ESDCTL_ESDPDC fields: PRCT_0 and PRCT_1. Each field defines a value loaded to a different chip select.

28.5.11 Burst Length options

When connecting DDR3 devices then only burst length 8 can be used.

When connecting DDR2 devices then either burst length 4 or 8 can be used.

Using burst length 4 in DDR2 is recommended as it produces higher throughput.

As mentioned above DDR3 can work only in burst length 8 mode, in this mode read/write accesses to the memory are always align to 8 words in the width of the connected memory (either x16 bit or x32 bit).

In case of AXI INCREMENT accesses that are not aligned the irrelevant data strobes are masked in write and ignored in read. In case of AXI WRAP accesses, even if the access is not aligned the arbitration switches the data so there are no wasted cycles, this method reduce latency in read/write AXI WRAP accesses.

28.6 ZQ calibration

The i.MX ZQ calibration is done in parallel to the DRAM ZQ calibration.There are 2 types of DRAM ZQ calibration: short & long.

The memory long calibration is done during power up sequence, when existing self refresh mode or when exiting slow precharge power down (DLL lock can be done in parallel). Short calibration is done according to a configurable timer defined by ZQ_HW_PER.

28.6.1 PHY ZQ SW calibration sequence

The ZQ calibration can also be done in SW. However because SW ZQ calibration is much slower than HW calibration, it should be used mainly for debugging.

SW should configure the ZQ calibration parameters (Pull-up or Pull-down and their value) and assert the ZQ_SW_FOR bit.Then SW should wait till ZQ_SW_FOR is de-asserted and use ZQ_SW_RES status bit in order to calculate the next ZQ calibration parameters.

28.6.2 Memory ZQ calibration sequence

Before the controller can issue a ZQCL/ZQCS command to the memory, it should precharge all memory banks and wait for tRP. A single ZQ command can be issued to all devices as long as the devices do not share the same ZQ resistor.

When the controller issues the ZQ command, it should also drive A10 (long or short command) and CS (0, 1 or both).

The controller must keep the memory lines quiet (except for CK) for the ZQ calibration time as defined in the Jedec (512 cycles for ZQCL after reset, 256 for other ZQCL and 64 for ZQCS).

28.7 Delay line

The delay line adds a configurable delay of up to 1/2 cycle. The delay is instantiated 8 times (4 for read and 4 for write).

28.7.1 Delay line Calibration

By default the delay in configured to generate 1/4 cycle of delay. However in some systems this may not be the optimal value and an additional calibration should be done.

NOTE

Delay line default configuration value must be a valid value (so data can be write & read in this case) though it might not be the optimal value. The Delay line calibration should be done after dqs gating & write level calibrations.

28.7.1.1 Read Delay Line Calibration

1. If MPR mode is used SW enables the MPR_CMP_BIT in PDCMPR and issues precharge all command to the device followed by MRS command. If pre-defined mode is used the SW issues precharge all command to the device followed by bank 0 ACT command and writes the predefined data to bank 0 address 0 of CS0 and then writes the same pre defined data to the PDCMPR and disables he MPR_CMP_BIT in PDCMPR.
2. SW asserts HW_RD_DL_EN bit.
3. PHY counts for 16 cycles (Tmod+4) (only if MPR mode is used) and asserts the read command request to the controller for one cycle.
4. PHY samples the dl abs default value drive on the delay line inputs.
5. PHY enables the measure signal and wait for 16 cycles.
6. Controller drives read command and data is sampled in the rd FIFO 16/32 cycles (according to HW_RD_DL_CMP_CYC) after read request was acknowledged.
7. PHY compares the data to the pre-defined value if data is incorrect PHY sets the current abs value +1 to be delay line lower boundary.
8. PHY resets the rd FIFO (to the inv. pre-defined value) and its pointers.
9. PHY decrements the rd delay line abs value by 1 and enables the measure signal.
10. PHY asserts the dqs gating read command to the controller for one cycle and data is sampled in the rd FIFO after 16/32 cycles (according to HW_RD_DL_CMP_CYC) after read request was acknowledged.
11. If read data is incorrect, PHY sets the current abs value + 1 to be delay line lower boundary and skip to step 13 if delay line phy_rd_dl_unit_num is 0 PHY sets this value to be delay line lower and skip to step 12 boundary else it repeats steps 7-8.
12. PHY drives the rd delay to abs bypass mode and if compared data is correct, then PHY sets the delay line lower boundary to 0.
13. PHY drives the rd delay line abs value back to it default value.
14. PHY increments the rd delay line abs value by 1 and enables the measure signal.
15. PHY asserts the read command to the controller for one cycle and data is sampled in the rd FIFO 16/32 cycles after read request was acknowledged.
16. If read data is incorrect PHY sets this value - 1 to be delay line upper boundary else it repeats steps 11-12.

17. Upon completion of step 13 PHY drives the values of lower and upper boundaries (HW_RD_DL_LOW & HW_RD_DL_UP) and the average value of those field is written to RD_DL_ABS_OFFSET_x field.
18. PHY enables the measure signal.
19. PHY asserts the hw_rd_dl_finish signal.

28.7.1.2 Write Delay line Calibration

1. SW asserts HW_WR_DL_EN bit
2. Controller drives write command. Wr Dqs should be driven by the controller as in regular write command while data is driven according to the ESDCTL_PDCMPR1/2.
3. Controller drives read command and data is sampled in the rd FIFO16/32 cycles (according to HW_RD_DL_CMP_CYC) after read request was acknowledged.
4. PHY compares the data to the pre-defined value. Every odd (first, third...) access' data is compared to PDV1 while every even (second, fourth...) access' data is compared to PDV2. If read data is incorrect, HW_WR_DL_ERR is asserted and hw_wr_dl_finish is asserted.
5. PHY resets the rd FIFO (to the inv. pre-defined value) and its pointers.
6. PHY decrements the wr delay line abs value by 1 and enables the measure signal of the write delay line.
7. PHY asserts the write command enable to the controller for one cycle and data is written to the DRAM.
8. Controller drives read command and data is sampled in the rd FIFO after 16/32 cycles after read request was acknowledged.
9. If read data is incorrect PHY sets this value + 1 to be delay line lower boundary if delay line qtr_cyc_del is 0 PHY sets this value to be delay line lower boundary else it repeats steps 9-10.
10. PHY drives the wr delay line abs value back to it default value.
11. PHY increments the wr delay line abs value by 1 and enables the measure signal of the write delay line.
12. PHY asserts the write command enable to the controller for one cycle and data is written to the DRAM.
13. PHY asserts the read command enable to the controller for one cycle and data is sampled in the rd FIFO after 16/32 cycles after read request was acknowledged.
14. If read data is incorrect, PHY set this value - 1 to be delay line upper boundary else it repeats steps 13-15.
15. Upon completion of step 16, PHY drives the values of lower and upper boundaries (HW_WR_DL_LOW & HW_WR_DL_UP) and the average value of those fields is written to WR_DL_ABS_OFFSET_x field
16. PHY enables the measure signal.
17. PHY asserts the hw_wr_dl_finish signal.

28.8 Write leveling

The write leveling includes a physical delay of up to 2.875 cycles. The delay is chosen according to a register that holds the configurable value of the delay.

Write leveling can be calibrated by HW or SW. The HW WL calibration points out to the optimal write dqs location inside the cycle but not between cycles. Therefor after completion of the WL HW calibration user should read the HW_WLx_RES and update the HW_WLx_CYC field to the correct value.

Each DQS has a it own delay. In DDR3 the configurable value for each DQS can be different. The write leveling of all DQS is done in parallel.

28.8.1 SW write leveling

Below is the SW write leveling sequence.

1. The SW issues a MRS command to enter write leveling mode.
2. SW asserts WL_SW_CNT_EN & WL_SW_EN bits.
3. PHY enters sw write leveling mode, counts 25 cycles and than asserts the dqs obe.
4. The PHY counts 15 cycles.
5. PHY asserts the dqs_en for one cycle.
6. PHY waits for 16 cycles before sampling prime DQ for each byte.
7. PHY drives the prime DQ's value on the write leveling bits and asserts sw_write_level_finish.
8. Controller sample prime DQ's to write_level_dq status bits.
9. SW read WL_SWx_RES bits.
10. SW clears the WL_SW_CNT_EN bit and de-asserts the WL_SW_EN bit.
11. SW should repeat 2-10(there is no need to assert WL_SW_CNT_EN) till it detects the correct wl value for each byte.
12. SW updates WL_SWx_VAL fields with the new correct values.
13. SW issues a MRS command to exit write leveling mode.

28.8.2 HW write leveling

Below is the HW write leveling sequence.

1. Same as steps 1-3 in SW write leveling (in step 2 SW asserts HW_WL_EN bit and there is no need to assert WL_SW_CNT bit).

2. PHY drives wr_level_delay to 0, asserts the dqs_en for one cycle and samples the DQ prime
3. PHY increments the wr_level_delay by 4, asserts the dqs_en for one cycle and waits for 16 cycles before sampling the DQ prime.
4. PHY decrements the wr_level_delay by 3 and generate measure signal.
5. PHY asserts the dqs_en for one cycle and waits for 16 cycles before sampling the DQ prime.
6. PHY increments the wr_level_delay by 4, asserts the dqs_en for one cycle and waits for 16/32 cycles before sampling the DQ prime.
7. PHY repeats steps 4-6 till all 8 wr_level_delay are calculated.
8. PHY checks the 8 bit prime DQ results for 0 to 1 transition. If 1 transition is found the wr_level_delay is the value of the first 1. If number of transitions isn't 1 HW_WLx_RES is set to 1000 for error indication.
9. Upon completion of this process the PHY asserts the hw_write_level_finish while HW_WLx_RES are valid.
10. PHY enables the measure signal.
11. SW issues a MRS command to exit write leveling mode.

28.9 Write fine tuning

Write fine tuning is an additional circuit that enables the option to fine tune the timing of the dq/dm bits (relative to dqs) by up to +/-100 ps. This is done by reducing the delay between the wl_dqs by 100 ps and adding a configurable delay of up to 200 ps (6 delay units of around 30-35 ps each) for each DQ/DM output. The delay can be configured independently for each DQ/DM. The calibration of this mechanism can be done only by writing & reading data from the memory.

28.10 Read fine tuning

Read fine tuning is an additional circuit that enables the option to fine tune the timing of the coming dq bits (relative to coming dqs) by up to +/-100 ps. This is done by reducing the delay between the incoming rd_dqs by 100 ps and adding a configurable delay of up to 200 ps (6 delay units of around 30-35 ps each) for each DQ input. The delay can be configured independently for each DQ. The calibration of this mechanism can be done only by writing & reading data from the memory.

28.11 DQS Gating

The assertion of the dqs gating is done according to the dqs gating delay, the negation of the dqs gating is done after the negedge of fourth dqs if there is no back to back access.

The dqs gating includes a delay of up to 7.875 cycles (The delay is chosen according to a 6 bit register that holds the configurable value of the delay.

Each DQS has a it own delay. In DDR3 the configurable value for each DQS can be different. The DQS gating training can be done for all DQS is done in parallel.

28.11.1 SW DQS gating training

1. SW writes data to a location that is going to be read in the next steps (the wrutten data should be the same data as in the pre-defined register)or by putting the device in MPR pre-defined mode (SW enables the MPR_CMP_BIT in PDCMPR and send precharge all command to the device followed by MRS command).
2. Generate a burst read access to a pre-defined data location/MPR.
3. Compare the data read from DDR to the pre-defined data to check if data is valid.
4. Reset the read data FIFO pointers and data. (writing 1 to the RST_RD_FIFO bit) and then clear the RST_RD_FIFO bit.
5. Repeat steps 3-4 till the data valid window is found. DQS gating should be configured to the middle of this window.
6. Exit MPR mode (if needed).

28.11.2 HW DQS Gating

Below is the HW DQS gating sequence.

1. If MPR mode is used, SW enables the MPR_CMP_BIT in PDCMPR and issues precharge all command to the device followed by MRS command. If pre-defined mode is used the SW issues precharge all command to the device followed by bank 0 ACT command and then writes data to bank 0 address 0 of CS0 and then writes the same pre defined data to the ESDCTL_PDCMPR1 and disables he MPR_CMP_BIT in ESDCTL_PDCMPR2.
2. SW asserts HW_DG_EN bit.
3. PHY counts for 12 cycles (Tmod) (only if MPR mode is used) and asserts the read command request to the controller for one cycle.

4. Controller drives read command and data is sampled in the rd FIFO 16/32 cycles (according to HW_RD_DL_CMP_CYC) after read request was acknowledged.
5. PHY compares the data to the pre-defined value.
6. PHY resets the rd fifo (to the inv. pre-defined value) and it's pointers.
7. PHY increments the dqs_gating_value by 4 and asserts the dqs gating read command to the controller for one cycle
8. If read data is incorrect PHY repeat step 7 till if finds the correct data. (This value is the used as the lower boundary for stages 10). If no correct data is found till max. DG value DG_HW_ERR is asserted.
9. When the PHY finds correct data it repeats steps 7 till if finds the incorrect data.This value is the used as the upper boundary for stages 13).If no incorrect data is found till lower boundary + 16 DG value DG_HW_ERR is asserted.
10. PHY drives the dqs_gating_value to lower_boundary - 3 and asserts the measure signal.
11. PHY asserts the dqs gating read command to the controller for one cycle.
12. PHY samples the data in rd fifo 16/32 after read request was acknowledged. It data is correct it writes 1 to the dqs_gating result register bit 0 else it write 0 to this bit.
13. If dqs_gating_value + 4 < upper boundary PHY increments dqs_gating_value by 4 and go to step 11 else go to step 14.
14. Phy repeats steps 11-13 for lower boundary -2, lower boundary -1 and lower boundary.
15. Upon completion of step 14 PHY has dqs gating register is ready. HW_DGx_RES is calculated according to the average value of the correct data window.
16. PHY enables the measure signal.
17. Upon completion of the training the PHY asserts hw_dqs_gating_finish (to clear the HW_DG_EN bit) while HW_DGx_RES fields are valid
18. SW should Exit MPR mode.

28.12 Programmable Registers

28.12.1 ESDCTL Memory Map/Register Definition

This section contains the description of the ESDCTL registers. Reset values are according to Micron DDR2-400 device (mt47h128m8_25). When assuming that four x8 bit devices of 1Gbits each are connected to each chip select.

ESDCTL memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FD_9000	ESDCTL Control Register (ESDCTL_ESDCTL)	32	R/W	0311_0000h	28.121.1/ 1233
63FD_9004	ESDCTL Power Down Control Register (ESDCTL_ESDPDC)	32	R/W	0003_0012h	28.121.2/ 1234
63FD_9008	ESDCTL ODT Timing Control Register (ESDCTL_ESDOTC)	32	R/W	1227_2000h	28.121.3/ 1237
63FD_900C	ESDCTL Logic Timing Configuration Register 0 (ESDCTL_ESDCFG0)	32	R/W	3236_22D3h	28.121.4/ 1239
63FD_9010	ESDCTL Timing Configuration Register 1 (ESDCTL_ESDCFG1)	32	R/W	B6B1_8A23h	28.121.5/ 1240
63FD_9014	ESDCTL Timing Configuration Register 2 (ESDCTL_ESDCFG2)	32	R/W	00C7_0092h	28.121.6/ 1243
63FD_9018	ESDCTL Timing Miscellaneous Register (ESDCTL_ESDMISC)	32	R/W	0000_9610h	28.121.7/ 1244
63FD_901C	ESDCTL Special Command Register (ESDCTL_ESDSCR)	32	R/W	0000_0000h	28.121.8/ 1247
63FD_9020	ESDCTL Refresh Control Register (ESDCTL_ESDREF)	32	R/W	0000_C000h	28.121.9/ 1249
63FD_9024	ESDCTL Logic Write Command Counter - Debug (ESDCTL_ESDWCC)	32	R/W	0000_0000h	28.121.10/ 1252
63FD_9028	ESDCTL Read Command Counter - Debug (ESDCTL_ESDRCC)	32	R/W	0000_0000h	28.121.11/ 1253
63FD_902C	ESDCTL Read/Write Command Delay (ESDCTL_ESDRWD)	32	R/W	0F9F_26D2h	28.121.12/ 1253
63FD_9030	ESDCTL Out of Reset Delays (ESDCTL_ESDOR)	32	R/W	009F_0E0Eh	28.121.13/ 1255
63FD_9034	ESDCTL MRR DATA Register (ESDCTL_ESDMRR)	32	R	0000_0000h	28.121.14/ 1257
63FD_9038	ESDCTL Timing Configuration Register 3 (ESDCTL_ESDCFG3_LP)	32	R/W	0000_0000h	28.121.15/ 1257
63FD_903C	ESDCTL MR4 Derating Register (ESDCTL_ESDMR4)	32	R/W	0000_0000h	28.121.16/ 1258
63FD_9040	PHY ZQ HW Control Register (ESDCTL_ZQHWCTRL)	32	R/W	A138_0000h	28.121.17/ 1260
63FD_9044	PHY ZQ SW Control Register (ESDCTL_ZQSWCTRL)	32	R/W	0000_0000h	28.121.18/ 1262
63FD_9048	PHY Write Leveling General Control Register (ESDCTL_WLGCR)	32	R/W	0000_0000h	28.121.19/ 1263
63FD_904C	PHY Write Leveling Delay Control Register 0 (ESDCTL_WLDECTRL0)	32	R/W	0000_0000h	28.121.20/ 1265

Table continues on the next page...

ESDCTL memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FD_9050	PHY Write Leveling Delay Control Register 1 (ESDCTL_WLDECTRL1)	32	R/W	0000_0000h	28.121.21/ 1267
63FD_9054	PHY Write Leveling Delay Line Status Register (ESDCTL_WLDLST)	32	R	0000_0000h	28.121.22/ 1268
63FD_9058	PHY ODT Control Register (ESDCTL_ODTCTRL)	32	R/W	0000_0000h	28.121.23/ 1269
63FD_905C	PHY Read DQ Byte0 Delay Register (ESDCTL_RDDQBY0DL)	32	R/W	0000_0000h	28.121.24/ 1271
63FD_9060	PHY Read DQ Byte1 Delay Register (ESDCTL_RDDQBY1DL)	32	R/W	0000_0000h	28.121.25/ 1274
63FD_9064	PHY Read DQ Byte2 Delay Register (ESDCTL_RDDQBY2DL)	32	R/W	0000_0000h	28.121.26/ 1277
63FD_9068	PHY Read DQ Byte3 Delay Register (ESDCTL_RDDQBY3DL)	32	R/W	0000_0000h	28.121.27/ 1279
63FD_906C	PHY Write DQ Byte0 Delay Register (ESDCTL_WRDQBY0DL)	32	R/W	0000_0000h	28.121.28/ 1282
63FD_9070	PHY Write DQ Byte1 Delay Register (ESDCTL_WRDQBY1DL)	32	R/W	0000_0000h	28.121.29/ 1284
63FD_9074	PHY Write DQ Byte2 Delay Register (ESDCTL_WRDQBY2DL)	32	R/W	0000_0000h	28.121.30/ 1285
63FD_9078	PHY Write DQ Byte3 Delay Register (ESDCTL_WRDQBY3DL)	32	R/W	0000_0000h	28.121.31/ 1287
63FD_907C	PHY DQS Gating Control Register0 (ESDCTL_DGCTRL0)	32	R/W	0000_0000h	28.121.32/ 1289
63FD_9080	PHY DQS Gating Control Register1 (ESDCTL_DGCTRL1)	32	R/W	0000_0000h	28.121.33/ 1291
63FD_9084	PHY DQS Gating Delay Line Status Register (ESDCTL_DGDLST)	32	R	0000_0000h	28.121.34/ 1292
63FD_9088	PHY Read Delay Lines Configuration Register (ESDCTL_RDDLCTL)	32	R/W	0000_0000h	28.121.35/ 1293
63FD_908C	PHY Read Delay Lines Status Register (ESDCTL_RDDLST)	32	R	0000_0000h	28.121.36/ 1294
63FD_9090	PHY Write Delay Lines Configuration Register (ESDCTL_WRDLCTL)	32	R/W	0000_0000h	28.121.37/ 1295
63FD_9094	PHY Write Delay Lines Status Register (ESDCTL_WRDLST)	32	R	0000_0000h	28.121.38/ 1297
63FD_9098	PHY SDCLK Control Register (ESDCTL_SDCTRL)	32	R/W	0000_0000h	28.121.39/ 1298
63FD_909C	ZQ LPDDR2 HW Control Register (ESDCTL_ZQLP2CTL)	32	R/W	638F_018Fh	28.121.40/ 1298

Table continues on the next page...

ESDCTL memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FD_90A0	PHY RD DL HW Calibration Control Register (ESDCTL_RDDLHWCTL)	32	R/W	0000_0000h	28.121.41/ 1300
63FD_90A4	PHY WR DL HW Calibration Control Register (ESDCTL_WRDLHWCTL)	32	R/W	0000_0000h	28.121.42/ 1301
63FD_90A8	PHY RD DL HW Calibration Status Register 0 (ESDCTL_RDDLHWST0)	32	R	0000_0000h	28.121.43/ 1302
63FD_90AC	PHY RD DL HW Calibration Status Register 1 (ESDCTL_RDDLHWST1)	32	R	0000_0000h	28.121.44/ 1303
63FD_90B0	PHY WR DL HW Calibration Status Register 0 (ESDCTL_WRDLHWST0)	32	R	0000_0000h	28.121.45/ 1304
63FD_90B4	PHY WR DL HW Calibration Status Register 1 (ESDCTL_WRDLHWST1)	32	R	0000_0000h	28.121.46/ 1305
63FD_90B8	PHY Write Leveling HW Error Register (ESDCTL_WLHWERR)	32	R	0000_0000h	28.121.47/ 1306
63FD_90BC	PHY DQS Gating HW Status Register 0 (ESDCTL_DGHWST0)	32	R	0000_0000h	28.121.48/ 1306
63FD_90C0	PHY DQS Gating HW Status Register 1 (ESDCTL_DGHWST1)	32	R	0000_0000h	28.121.49/ 1307
63FD_90C4	PHY DQS Gating HW Status Register 2 (ESDCTL_DGHWST2)	32	R	0000_0000h	28.121.50/ 1307
63FD_90C8	PHY DQS Gating HW Status Register 3 (ESDCTL_DGHWST3)	32	R	0000_0000h	28.121.51/ 1308
63FD_90CC	PHY Pre-defined Compare Register 1 (ESDCTL_PDCMPR1)	32	R/W	0000_0000h	28.121.52/ 1309
63FD_90D0	PHY Pre-defined Compare Register 2 (ESDCTL_PDCMPR2)	32	R/W	0000_0000h	28.121.53/ 1309
63FD_90D4	PHY SW Dummy Access Register (ESDCTL_SWDAR)	32	R/W	0000_0000h	28.121.54/ 1311
63FD_90D8	PHY SW Dummy Read Data Register n (ESDCTL_SWDRDR0)	32	R	FFFF_ FFFFh	28.121.55/ 1312
63FD_90DC	PHY SW Dummy Read Data Register n (ESDCTL_SWDRDR1)	32	R	FFFF_ FFFFh	28.121.55/ 1312
63FD_90E0	PHY SW Dummy Read Data Register n (ESDCTL_SWDRDR2)	32	R	FFFF_ FFFFh	28.121.55/ 1312
63FD_90E4	PHY SW Dummy Read Data Register n (ESDCTL_SWDRDR3)	32	R	FFFF_ FFFFh	28.121.55/ 1312
63FD_90E8	PHY SW Dummy Read Data Register n (ESDCTL_SWDRDR4)	32	R	FFFF_ FFFFh	28.121.55/ 1312
63FD_90EC	PHY SW Dummy Read Data Register n (ESDCTL_SWDRDR5)	32	R	FFFF_ FFFFh	28.121.55/ 1312

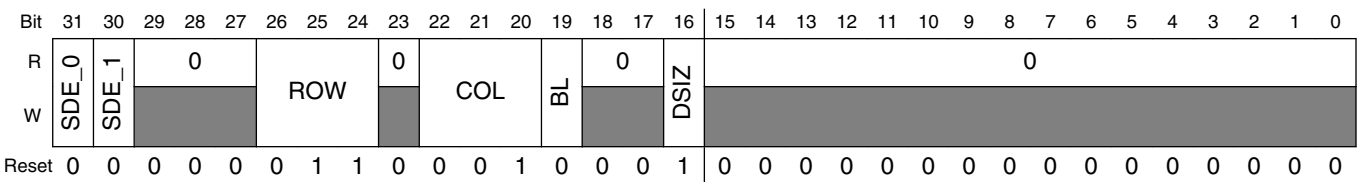
Table continues on the next page...

ESDCTL memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FD_90F0	PHY SW Dummy Read Data Register n (ESDCTL_SWDRDR6)	32	R	FFFF_ FFFFh	28.121.55/ 1312
63FD_90F4	PHY SW Dummy Read Data Register n (ESDCTL_SWDRDR7)	32	R	FFFF_ FFFFh	28.121.55/ 1312
63FD_90F8	PHY Measure Unit Register (ESDCTL_MUR)	32	R/W	0000_0000h	28.121.56/ 1313
63FD_90FC	Write CA Delay Line controller (ESDCTL_WRCADL)	32	R/W	0000_0000h	28.121.57/ 1314

28.121.1 ESDCTL Control Register (ESDCTL_ESDCTL)

Address: ESDCTL_ESDCTL is 63FD_9000h base + 0h offset = 63FD_9000h



ESDCTL_ESDCTL field descriptions

Field	Description
31 SDE_0	This should be set if a memory is connected to chip select 0. If SDE0 and SDE1 are both 0, no clock will be issued to the memory. Writing 1 to SDE0 or SDE1 will initiate power up delays as JEDEC defines. Power up delays are a function of the configured memory type (DDR2/DDR3/LPDDR2). 0 Disabled 1 Enabled
30 SDE_1	This should be set if a memory is connected to chip select 1. If SDE0 and SDE1 are both 0, no clock will be issued to the memory. Writing 1 to SDE0 or SDE1 will initiate power up delays as JEDEC defines. Power up delays are a function of the configured memory type (DDR2/DDR3/LPDDR2). 0 Disabled 1 Enabled
29–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–24 ROW	Row Address Width. This control field specifies the number of row addresses used by the memory array. It will affect the way an incoming address will be decoded. 000 11 bits Row 001 12 bits Row

Table continues on the next page...

ESDCTL_ESDCTL field descriptions (continued)

Field	Description
	010 13 bits Row 011 14 bits Row 100 15 bits Row 101 16 bits Row 110 through 111 are reserved
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–20 COL	Column Address Width. The COL control field is used to specify the number of column addresses in the memory array. It will determine how an incoming address will be decoded. 0x0 9 bits column 0x1 10 bits column 0x2 11 bits column 0x3 8 bits column 0x4 12 bits column 0x5-0xF Reserved
19 BL	Burst Length. DDR2 ,LPDDR2 : Controller supports either burst length 8 or 4 (Optimized for BL4). DDR3: Controller supports only burst length 8. 0 Burst Length 4is used. 1 Burst Length 8 is used.
18–17 Reserved	This read-only field is reserved and always has the value zero. Reserved
16 DSIZ	SDRAM Memory Data Width. This field defines the width of the SDRAM memory and its alignment on the external data bus. 16-bit ports aligned to the low half word. Data qualifier mask control outputs must match the selected data bus alignment. Memories aligned to D[15:0] use DQM0 and DQM1. 0 16-bit memory width aligned to D[15:0] 1 32-bit memory width
15–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

28.121.2 ESDCTL Power Down Control Register (ESDCTL_ESDPDC)

Table 28-8. PRCT field encoding

PRCT[2:0]	Precharge Timer
000	Disabled (Bit field reset value)
001	2 clocks
010	4 clocks
011	8 clocks

Table continues on the next page...

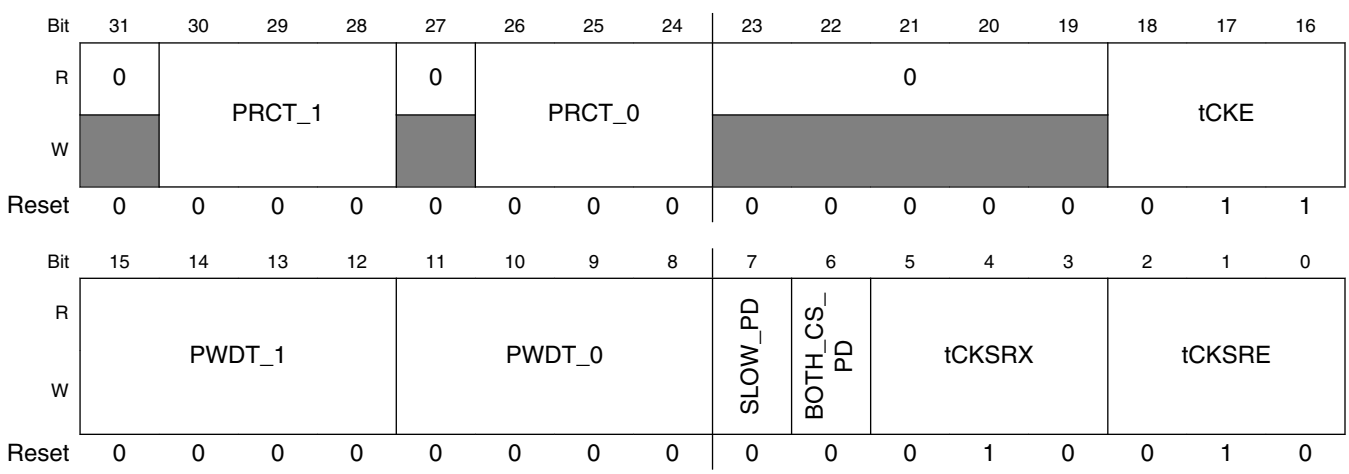
Table 28-8. PRCT field encoding (continued)

PRCT[2:0]	Precharge Timer
100	16 clocks
101	32 clocks
110	64 clocks
111	128 clocks

Table 28-9. PWDT field encoding

PWDT[3:0]	Power Down Time-out
0000	Disabled (bit field reset value)
0001	16 cycles
0010	32 cycles
0011	64 cycles
0100	128 cycles
0101	256 cycles
0110	512 cycles
0111	1024 cycles
1000	2048 cycles
1001	4096 cycles
1010	8196 cycles
1011	16384 cycles
1100	32768 cycles
1101-1111	Reserved

Address: ESDCTL_ESDPDC is 63FD_9000h base + 4h offset = 63FD_9004h



ESDCTL_ESDPDC field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–28 PRCT_1	Precharge Timer - Chip Select 1. This chip select will be precharged after certain amount of cycles from the last access. The amount of cycles is determined according to Write CA Delay Line controllerESDCTL Power Down Control Register above.
27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–24 PRCT_0	Precharge Timer - Chip Select 0. This chip select will be precharged after certain amount of cycles from the last access. The amount of cycles is determined according to Write CA Delay Line controllerESDCTL Power Down Control Register above.
23–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18–16 tCKE	CKE minimum pulse width. 0x0 1 cycle 0x1 2 cycles 0x6 7 cycles 0x7 8 cycles
15–12 PWDT_1	Power Down Timer - Chip Select 1. This field determines whether the memory on chip select 1 will be placed in a Power Down condition after a selectable delay from the last access. The amount of cycles is determined according to Write CA Delay Line controllerESDCTL Power Down Control Register above.
11–8 PWDT_0	Power Down Timer - Chip Select 0. This field determines whether the memory on chip select 0 will be placed in a Power Down condition after a selectable delay from the last access. The amount of cycles is determined according to Write CA Delay Line controllerESDCTL Power Down Control Register above.
7 SLOW_PD	DDR3: Slow precharge power-down. DDR2: Slow active power-down. LPDDR2: Not relevant. **NOTE:** Memory should be configured the same. 0 Fast mode. 1 Slow mode.
6 BOTH_CS_PD	When both chip selects are using power down timer, will enter power down only if both chip select 0 and chip select 1 required idle time was reached. 0 Each chip select can enter power down independently according to its configuration. 1 Chip selects can enter power down only if both chip selects idle time was reached.
5–3 tCKSRX	Valid clock requirement before self-refresh exit. 0x0 0 cycle 0x1 1 cycles 0x6 6 cycles 0x7 7 cycles

Table continues on the next page...

ESDCTL_ESDPDC field descriptions (continued)

Field	Description
2–0 tCKSRE	Valid clock requirement after self-refresh entry. 0x0 0cycle 0x1 1 cycles 0x6 6cycles 0x7 7cycles

28.121.3 ESDCTL ODT Timing Control Register (ESDCTL_ESDOTC)

Address: ESDCTL_ESDOTC is 63FD_9000h base + 8h offset = 63FD_9008h

ESDCTL_ESDOTC field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved
29–27 tAOFPD	DDR2: ODT turn-off (power down mode) max value. DDR3: Asynchronous RTT turn-off delay(power down with DLL frozen). LPDDR2: Not relevant. 0x0 1 cycle 0x1 2 cycles 0x6 7 cycles 0x7 8 cycles
26–24 tAONPD	DDR2: ODT turn-on (power down mode) max value. DDR3: Asynchronous RTT turn-on delay(power down with DLL frozen). LPDDR2: Not relevant. 0x0 1 cycle 0x1 2 cycles 0x6 7 cycles 0x7 8 cycles
23–20 tANPD	DDR2: ODT to power down entry latency. DDR3: ODT to power down entry latency (Should be set to tCWL-1). LPDDR2: Not relevant. 0x0 1 clock 0x1 2 clocks

Table continues on the next page...

ESDCTL_ESDOTC field descriptions (continued)

Field	Description
	0x2 3 clocks 0xE 15 clocks 0xF 16 clocks
19–16 tAXPD	DDR2: ODT power down exit latency. DDR3: ODT power down exit latency (Should be set to tCWL-1). LPDDR2: Not relevant. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0xE 15 clocks 0xF 16 clocks
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 tODTLon	ODT turn on latency. DDR2: Named 'tAOND' LPDDR2: Not relevant. 0x0 0x1 Reserved 0x2 2 cycles 0x3 3 cycles 0x4 4 cycles 0x5 5 cycles 0x6 6 cycles 0x7 Reserved
11–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8–4 tODT_idle_off	Idle period before turning memory ODT off. LPDDR2: Not relevant. 0x0 0 cycle (turned off at the earliest possible time) 0x1 1 cycle 0x2 2 cycles 0x1E 30 cycles 0x1F 31 cycles
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

28.121.4 ESDCTL Logic Timing Configuration Register 0 (ESDCTL_ESDCFG0)

Address: ESDCTL_ESDCFG0 is 63FD_9000h base + Ch offset = 63FD_900Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	tRFC								tXS								tXP			tXPDLL				tFAW						tCL			
Reset	0	0	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0	0	1	0	0	0	1	0	1	1	0	1	0	0	1	1

ESDCTL_ESDCFG0 field descriptions

Field	Description
31–24 tRFC	REF command to ACT or REF command time. Please refer to DDR2 SDRAM Specification JESD79-2E (April 2008) and to DDR3 SDRAM Specification JESD79-3C (April 2008) for a detailed description of this parameter. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0xFE 255 clocks 0xFF 256 clocks
23–16 tXS	Exit self refresh to non READ command. (Named tXSNR in DDR2 devices) LPDDR2: Named tXSR, self-refresh exit to next valid command delay. Please refer to DDR2 SDRAM Specification JESD79-2E (April 2008) and to DDR3 SDRAM Specification JESD79-3C (April 2008) for a detailed description of this parameter. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0xFE 255 clocks 0xFF 256 clocks
15–13 tXP	DDR2: Exit precharge power down to any command. DDR3:Exit power down with DLL-on to any valid command;Exit power down with DLL-frozen to commands not requiring a locked DLL. LPDDR2: Exit power-down to next valid command delay. Please refer to DDR2 SDRAM Specification JESD79-2E (April 2008) and to DDR3 SDRAM Specification JESD79-3C (April 2008) for a detailed description of this parameter. 0x0 1 cycle 0x1 2 cycles 0x6 7 cycles 0x7 8 cycles
12–9 tXPDLL	DDR2: Exit active power down to read commands(tXARD/tXARDS depending to memory configuration). DDR3: Exit precharge power down with DLL frozen to commands requiring DLL. LPDDR2: Not relevant.

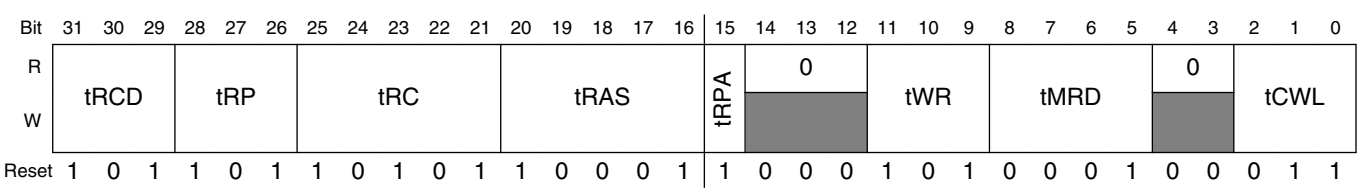
Table continues on the next page...

ESDCTL_ESDCFG0 field descriptions (continued)

Field	Description
	Please refer to DDR2 SDRAM Specification JESD79-2E (April 2008) and to DDR3 SDRAM Specification JESD79-3C (April 2008) for a detailed description of this parameter. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0xE 15 clocks 0xF 16 clocks
8–4 tFAW	Four Active Window (all banks). Please refer to DDR2 SDRAM Specification JESD79-2E (April 2008) and to DDR3 SDRAM Specification JESD79-3C (April 2008) for a detailed description of this parameter. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0x1E 31 clocks 0x1F 32 clocks
3–0 tCL	CAS Read Latency. DDR3: referred as CL. LPDDR2/ DDR2: referred as RL. Please refer to DDR2 SDRAM Specification JESD79-2E (April 2008) and to DDR3 SDRAM Specification JESD79-3C (April 2008) for a detailed description of this parameter. 0x0 3 cycles 0x1 4 cycles 0x2 5 cycles 0x3 6 cycles 0x4 7 cycles 0x5 8 cycles 0x6 9 cycles 0x7 10 cycles 0x8 11 cycles 0x9-0xF Reserved

28.121.5 ESDCTL Timing Configuration Register 1 (ESDCTL_ESDCFG1)

Address: ESDCTL_ESDCFG1 is 63FD_9000h base + 10h offset = 63FD_9010h



ESDCTL_ESDCFG1 field descriptions

Field	Description
31–29 tRCD	ACT command to internal read or write delay time (same bank). (This field is valid only for DDR2/DDR3 memories) LPDDR2: This field is not relevant, dedicated field exist tRCD_LP. Please refer to DDR2 SDRAM Specification JESD79-2E (April 2008) and to DDR3 SDRAM Specification JESD79-3C (April 2008) for a detailed description of this parameter. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0x3 4 clocks 0x4 5 clocks 0x5 6 clocks 0x6 7 clocks 0x7 8 clocks
28–26 tRP	PRE command period (same bank). (This field is valid only for DDR2/DDR3 memories) LPDDR2: This field is not relevant, dedicated field exist tRPpb_LP . Please refer to DDR2 SDRAM Specification JESD79-2E (April 2008) and to DDR3 SDRAM Specification JESD79-3C (April 2008) for a detailed description of this parameter. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0x3 4 clocks 0x4 5 clocks 0x5 6 clocks 0x6 7 clocks 0x7 8 clocks
25–21 tRC	ACT to ACT or REF command period (same bank). (This field is valid only for DDR2/DDR3 memories) ESDCTL.LPDDR2: This field is not relevant, dedicated field exist tRC_LP. Please refer to DDR2 SDRAM Specification JESD79-2E (April 2008) and to DDR3 SDRAM Specification JESD79-3C (April 2008) for a detailed description of this parameter. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0x1E 31 clocks 0x1F 32 clocks
20–16 tRAS	ACT to PRE command period (same bank). Please refer to DDR2 SDRAM Specification JESD79-2E (April 2008) and to DDR3 SDRAM Specification JESD79-3C (April 2008) for a detailed description of this parameter. 0x0 1 clock 0x1 2 clocks

Table continues on the next page...

ESDCTL_ESDCFG1 field descriptions (continued)

Field	Description
	0x2 3 clocks 0x1E 31 clocks 0x1F Reserved
15 tRPA	Precharge-all command period. (This field is valid only for DDR2/DDR3 memories) LPDDR2: This field is not relevant, dedicated field exist tRPab_LP. Please refer to DDR2 SDRAM Specification JESD79-2E (April 2008) and to DDR3 SDRAM Specification JESD79-3C (April 2008) for a detailed description of this parameter. 0 Will be equal to: tRP. 1 Will be equal to: tRP+1.
14–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11–9 tWR	WRITE recovery time (same bank). Please refer to DDR2 SDRAM Specification JESD79-2E (April 2008) and to DDR3 SDRAM Specification JESD79-3C (April 2008) for a detailed description of this parameter. 0x0 1cycle 0x1 2cycles 0x2 3cycles 0x3 4cycles 0x4 5cycles 0x5 6cycles 0x6 7cycles 0x7 8 cycles
8–5 tMRD	Mode Register Set command cycle (all banks). DDR3: Set to max (tMRD,tMOD). DDR2: Set to tMRD. LPDDR2: Set to max(tMRR,tMRW) Please refer to DDR2 SDRAM Specification JESD79-2E (April 2008) and to DDR3 SDRAM Specification JESD79-3C (April 2008) for a detailed description of this parameter. 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0xE 15 clocks 0xF 16 clocks
4–3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 tCWL	CAS Write Latency. DDR2: Refers as WL and equals RL-1. DDR3: Refers as tCWL. LPDDR2: referred as WL. 0x0 2cycles (DDR2/DDR3) , 1 cycle (LPDDR2)

Table continues on the next page...

ESDCTL_ESDCFG1 field descriptions (continued)

Field	Description
	0x1 3cycles (DDR2/DDR3) , 2 cycles (LPDDR2) 0x2 4cycles (DDR2/DDR3) , 3 cycles (LPDDR2) 0x3 5cycles (DDR2/DDR3) , 4 cycles (LPDDR2) 0x4 6cycles (DDR2/DDR3) , 5 cycles (LPDDR2) 0x5 7cycles (DDR2/DDR3) , 6 cycles (LPDDR2) 0x6 8cycles (DDR2/DDR3) , 7 cycles (LPDDR2) 0x7 Reserved

28.121.6 ESDCTL Timing Configuration Register 2 (ESDCTL_ESDCFG2)

Address: ESDCTL_ESDCFG2 is 63FD_9000h base + 14h offset = 63FD_9014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							tDLLK									0							tRTP			tWTR			tRRD		
W																																
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0

ESDCTL_ESDCFG2 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved
24–16 tDLLK	DDR3: DLL locking time. DDR2: Named tXSRD, exit self refresh to read command. LPDDR2: Not relevant. Please refer to DDR2 SDRAM Specification JESD79-2E (April 2008) and to DDR3 SDRAM Specification JESD79-3C (April 2008) for a detailed description of this parameter. 0x0 1 cycle. 0x1 2 cycles. 0x2 3 cycles. 0xC7 200 cycles (JEDEC value for DDR2). 0x1FE 511 cycles. 0x1FF 512 cycles (JEDEC value for DDR3).
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8–6 tRTP	Internal READ command to PRECHARGE command delay (same bank). Please refer to DDR2 SDRAM Specification JESD79-2E (April 2008) and to DDR3 SDRAM Specification JESD79-3C (April 2008) for a detailed description of this parameter. 0x0 1cycle 0x1 2cycles

Table continues on the next page...

ESDCTL_ESDCFG2 field descriptions (continued)

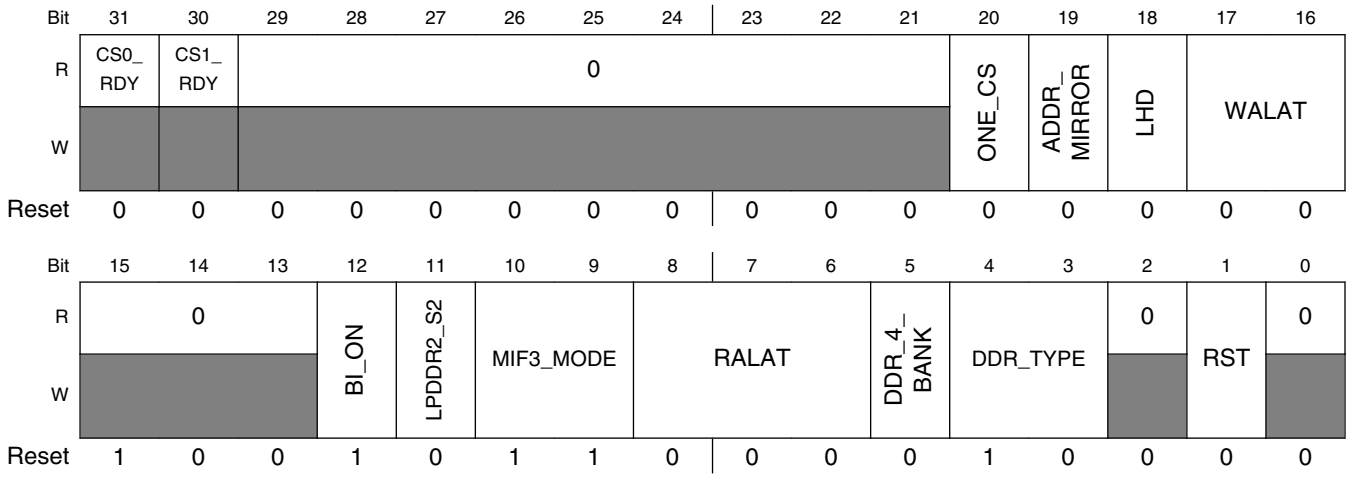
Field	Description
	0x2 3cycles 0x3 4cycles 0x4 5cycles 0x5 6cycles 0x6 7cycles 0x7 8 cycles
5–3 tWTR	Internal WRITE to READ command delay (same bank). Please refer to DDR2 SDRAM Specification JESD79-2E (April 2008) and to DDR3 SDRAM Specification JESD79-3C (April 2008) for a detailed description of this parameter. 0x0 1cycle 0x1 2cycles 0x2 3cycles 0x3 4cycles 0x4 5cycles 0x5 6cycles 0x6 7cycles 0x7 8 cycles
2–0 tRRD	ACTIVE to ACTIVE command period (all banks). Please refer to DDR2 SDRAM Specification JESD79-2E (April 2008) and to DDR3 SDRAM Specification JESD79-3C (April 2008) for a detailed description of this parameter. 0x0 1cycle 0x1 2cycles 0x2 3cycles 0x3 4cycles 0x4 5cycles 0x5 6cycles 0x6 7cycles 0x7 Reserved

28.121.7 ESDCTL Timing Miscellaneous Register (ESDCTL_ESDMISC)

This register contains the controlling various memory and control settings for the ESDCTL.

.

Address: ESDCTL_ESDMISC is 63FD_9000h base + 18h offset = 63FD_9018h



ESDCTL_ESDMISC field descriptions

Field	Description
31 CS0_RDY	External status device on CS0. This is a read-only status bit, that indicates whether the external memory is in wake-up period. 0 Device in wake-up period. 1 Device is ready for initialization.
30 CS1_RDY	External status device on CS1. This is a read-only status bit, that indicates whether the external memory is in wake-up period. 0 Device in wake-up period. 1 Device is ready for initialization.
29–21 Reserved	This read-only field is reserved and always has the value zero. Reserved
20 ONE_CS	One chip select. This feature enable to use all 2GB of memory space in chip select 0 instead of using 1GB for every chip select. 0x0 Two chip select are used, 1 GB per chip select. 0x1 Chip select 0 is used for entire 2GB memory space.
19 ADDR_MIRROR	Address mirroring. Note: This feature is not supported for LPDDR2 memories. But only for DDR2 or DDR2 memories. See table 1-24 for more details. 0 Address mirroring disabled. 1 Address mirroring enabled.
18 LHD	Latency hiding disable. This is a debug feature. When used the controller handles one read/write at a time. Meaning, the address axi channel waits for the data channel to finish before accepting new access. 0 Latency hiding on. 1 Latency hiding disable.

Table continues on the next page...

ESDCTL_ESDMISC field descriptions (continued)

Field	Description
17–16 WALAT	Write Additional latency. This latency can result from using Write Leveling and causes DQS to arrive later to some memory devices. 0x0 No additional latency required. 0x1 1 cycle additional delay 0x2 2 cycles additional delay 0x3 3 cycles additional delay
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 BI_ON	Bank Interleaving On: 0 Banks are not interleaved, and address is decoded as bank-row-column 1 Banks are interleaved, and address is decoded as row-bank-column
11 LPDDR2_S2	LPDDR2 S2 device type indication. In case LPDDR2 device is used (DDR_TYPE = 0x1), this bit indicates whether S2 or S4 device is used. 0x0 LPDDR2-S4 device is used. 0x1 LPDDR2-S2 device is used.
10–9 MIF3_MODE	MIF3 working mode: 00 Disable MIF3. 01 Enable treatment for: Valid access on first pipe line stage. 10 Enable treatment for: Valid access on first pipe line stage. Valid access on axi bus. 11 Enable treatment for: Valid access on first pipe line stage. Valid access on axi bus. Predicted next miss access from M4IF.
8–6 RALAT	Read Additional Latency, which determines when the contoller retrieves the data from the ESDCTL internal FIFO. Using this field to compensate on board/chip delays. **NOTE:** On LPDDR2 memories 2 extra cycles are waited internally in order to compensate on tDQSCK delay. 0x0 The data is retrieved from the FIFO at the earliest step possible. 0x1 1 cycle additional latency. 0x2 2 cycles additional latency. 0x3 3 cycles additional latency. 0x4 4 cycles additional latency. 0x5 5 cycles additional latency. 0x6 6 cycles additional latency. 0x7 7 cycles additional latency.
5 DDR_4_BANK	DDR device with 4 Banks: 0 8 banks device is being used. 1 4 banks device is being used
4–3 DDR_TYPE	DDR TYPE: 0x0 DDR3 device is used. 0x1 LPDDR2 device is used.

Table continues on the next page...

ESDCTL_ESDMISC field descriptions (continued)

Field	Description
	0x2 DDR2 device is used. (Default) 0x3 Reserved.
2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1 RST	Software Reset. **NOTE:** This bit once asserted gets deasserted automatically. 0 Do nothing. 1 Assert reset to the controller.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

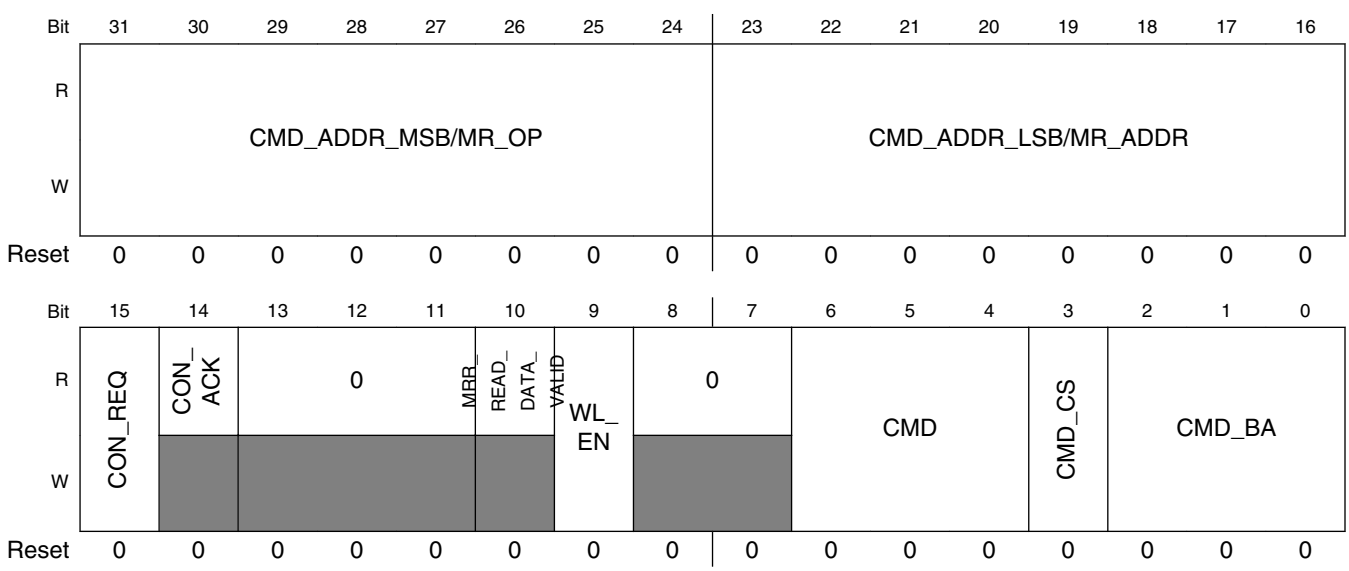
28.121.8 ESDCTL Special Command Register (ESDCTL_ESDSCR)

This register is used to issue special commands on the external device bus (such as load mode register, manual self refresh, manual precharge etc.). Every write to this register is interpreted as a command, and a read from this register will show you the last commands executed.

Every write to this register will result in one special command, and the IP bus will assert ips_xfr_wait as long as the special command is being carried out, thus introducing wait states for the next IP write

.

Address: ESDCTL_ESDSCR is 63FD_9000h base + 1Ch offset = 63FD_901Ch



ESDCTL_ESDSCR field descriptions

Field	Description
31–24 CMD_ADDR_ MSB/MR_OP	DDR2/DDR3: CMD_ADDR_MSB Address 8 LSB bits that match the issued command. ESDCTL.LPDDR2: MRW OPERAND MRR 8 bit operand
23–16 CMD_ADDR_ LSB/MR_ADDR	DDR2/DDR3: CMD_ADDR_MSB Address 8 MSB bits that match the issued command. ESDCTL.LPDDR2: MRR/MRW ADDRESS MRR/MRW 8 bit address
15 CON_REQ	Configuration request. When this bit is set it tells the controller the user would like to configure the IP registers, so in order to prevent a clash with the AXI bus, the controller holds the aready and awready signals at low, and will not accept LPMD request. This bit is high out of reset, meaning the controller is waiting on configuration and initialization of external memory before accepting any AXI accesses. Note that this bit should be held high during subsequent accesses to this register if AXI readies in low state are to be maintained. 1 AXI address readies are being held low 0 AXI address readies are operating normally
14 CON_ACK	Configuration acknowledge. This read only bit indicates that the controller is idle and no AXI accesses are pending execution. It is recommended that the master will wait until both CON_REQ and CON_ACK are 1 before performing any writes on the IP bus. 1 AXI channels state machines are all idle 0 At least one of the AXI state machines is not idle
13–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10 MRR_READ_ DATA_VALID	MRR READ DATA VALID - read only 0 Cleared upon the assertion of MRR command 1 Set after MRR data is valid and stored at ESDCTL_ESDMRR register.
9 WL_EN	Write Level Enable. This bit controls the DQS pads direction. It should be set high when sending write leveling entry command. And should be set low when sending write leveling exit command. 0 Exit write leveling mode or stay in normal mode. 1 Write leveling entry command was sent.
8–7 Reserved	This read-only field is reserved and always has the value zero. Reserved.
6–4 CMD	Command. This field contains the command to be executed: This field is automatically cleared after the command is asserted. 0x0 Normal operation

Table continues on the next page...

ESDCTL_ESDSCR field descriptions (continued)

Field	Description
	0x1 Precharge all, command is sent independently of bank status (set correct CMD_CS). Is issued even if banks are closed. Mainly used for init sequence purpose. 0x2 Auto-Refresh Command (set correct CMD_CS). 0x3 Load Mode Register Command (DDR2/DDR3, set correct CMD_CS, CMD_BA, CMD_ADDR_LSB, CMD_ADDR_MSB), MRW Command (ESDCTL.LPDDR2, set correct CMD_CS, MR_OP, MR_ADDR) 0x4 ZQ calibration (DDR2/DDR3, set correct CMD_CS, {CMD_ADDR_MSB,CMD_ADDR_LSB} = 0x400 or 0x0) 0x5 Precharge all, only if banks open (set correct CMD_CS). 0x6 MRR command (ESDCTL.LPDDR2, set correct CMD_CS, MR_ADDR) 0x7 Reserved
3 CMD_CS	Chip Select. This field determines which chip select the command is directed at. 0 to Chip-select 0 1 to Chip-select 1
2–0 CMD_BA	Bank Address. This field determines which bank within the chip select the command is directed at.

28.121.9 ESDCTL Refresh Control Register (ESDCTL_ESDREF)

This register defines how to refresh the connected memory device. It set how often begins a refresh cycle and how many refreshes are given in every cycle.

Table 28-17. Refresh rate example for REF_SEL = 0

REFR[2:0]	Number of refresh commands every 64KHz	Average periodic refresh rate (tREFI)	System Refresh period
0x0	1	15.6 µs	tRFC
0x1	2	7.8 µs	2*tRFC
0x3	4	3.9µs	4*tRFC
0x7	8	1.95 µs	8*tRFC

Table 28-18. Refresh rate example for REF_SEL = 1

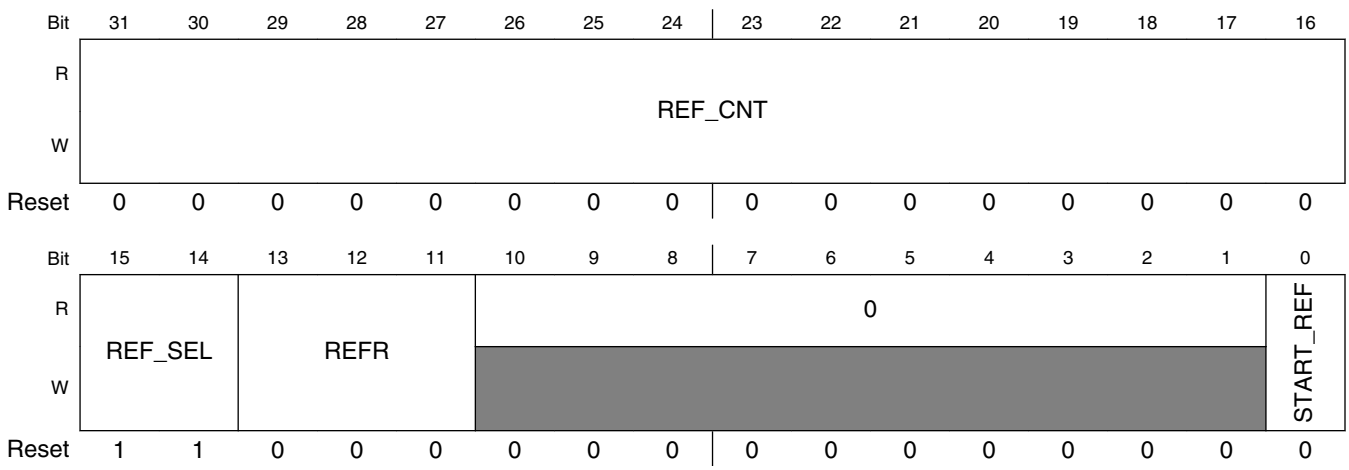
REFR[2:0]	Number of refresh commands every 32KHz	Average periodic refresh rate (tREFI)	System Refresh period
0x1	2	15.6 µs	2*tRFC
0x3	4	7.8 µs	4*tRFC
0x7	8	3.9µs	8*tRFC

Table 28-19. Refresh rate example for REF_SEL = 2@ 400MHz

REFR[2:0]	Number of refresh commands every refresh cycle	REF_CNT	Average periodic refresh rate (tREFI)	System Refresh period
0x0	1	0x618	3.9 µs	tRFC
0x1	2	0xC30	3.9 µs	2*tRFC
0x2	3	0x1248	3.9µs	3*tRFC
0x3	4	0x1860	3.9 µs	4*tRFC

- Other refresh configurations are also allowed, the configuration values in the tables above are only examples for reaching desired average periodic refresh rate.
- If keeping the required average periodic refresh rate (tREFI), all the rows are refreshed every refresh window. It can be seen in memories specifications that regardless of number of rows in the device the tREFI stays the same across this devices. The reason for that is that the memory device issues additional refresh commands for every refresh it receive, this comes into consideration in the tRFC parameter which gets bigger as the density increase.

Address: ESDCTL_ESDREF is 63FD_9000h base + 20h offset = 63FD_9020h



ESDCTL_ESDREF field descriptions

Field	Description
31–16 REF_CNT	Refresh Counter. If REF_SEL equals '2' a refresh cycle begins every amount of cycles defined in this field. (Counts in fast clock cycles) 0x0 Reserved. 0x1 1 cycle. 0xFFFE 65534 cycles. 0xFFFF 65535 cycles.

Table continues on the next page...

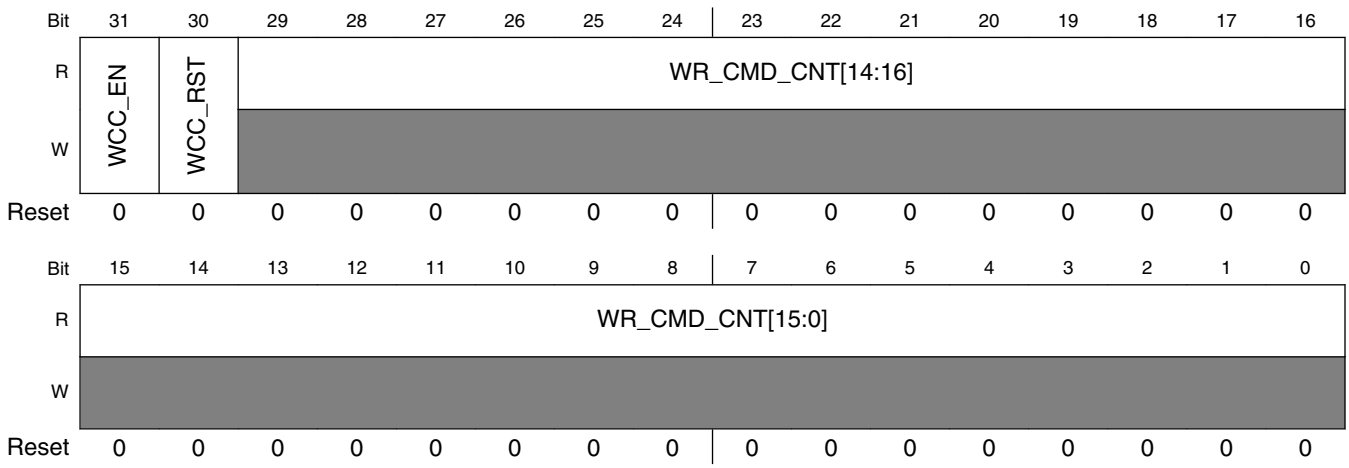
ESDCTL_ESDREF field descriptions (continued)

Field	Description
15–14 REF_SEL	Refresh Selector. This bit choose what will trigger each refresh cycle: 0 Refresh cycles are triggered in frequency of 64KHz. 1 Refresh cycles are triggered in frequency of 32KHz. 2 Refresh cycles are triggered every amount of cycles that is defined in REF_CNT field in this register. 3 No refresh cycles are triggered.
13–11 REFR	Refresh Rate. This field determine how many refresh commands are issued every refresh cycle. After every refresh command a time period of tRFC is kept until next valid command. 0x0 1 refresh 0x1 2 refreshes 0x2 3 refreshes 0x3 4 refreshes 0x4 5 refreshes 0x5 6 refreshes 0x6 7 refreshes 0x7 8 refreshes
10–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 START_REF	Start Refresh cycle. When set to '1' it will start a refresh cycle according to number of refreshes set in 'REFR' field. This bit returns to zero automatically. are0 Do nothing. 1 Start a refresh cycle.

28.121.10 ESDCTL Logic Write Command Counter - Debug (ESDCTL_ESDWCC)

This debug register is used for counting write burst commands that were issued to the memory.

Address: ESDCTL_ESDWCC is 63FD_9000h base + 24h offset = 63FD_9024h



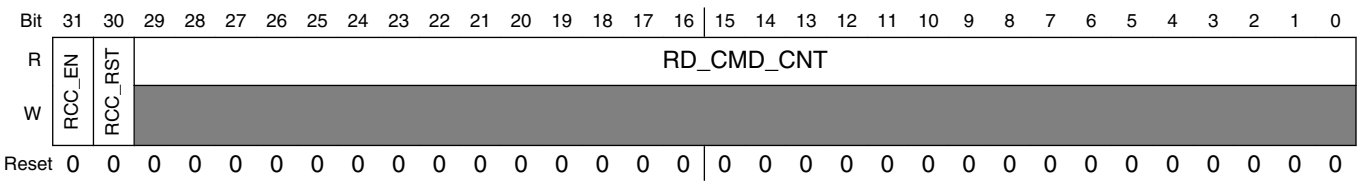
ESDCTL_ESDWCC field descriptions

Field	Description
31 WCC_EN	Write Command Counter Enable. 0 The counter is disabled. 1 The counter is enabled.
30 WCC_RST	Write Command Counter Reset. Note: Once asserted this bit deasserts automatically. 0 Do nothing. 1 The counter is set to zero.
29–0 WR_CMD_CNT	Write Command Counter value. This read-only field will display the number of write burst accesses that were sent to the memory. This is used mainly for bus utilization calculation.

28.121.11 ESDCTL Read Command Counter - Debug (ESDCTL_ESDRCC)

This debug register is used for counting write burst commands that were issued to the memory.

Address: ESDCTL_ESDRCC is 63FD_9000h base + 28h offset = 63FD_9028h



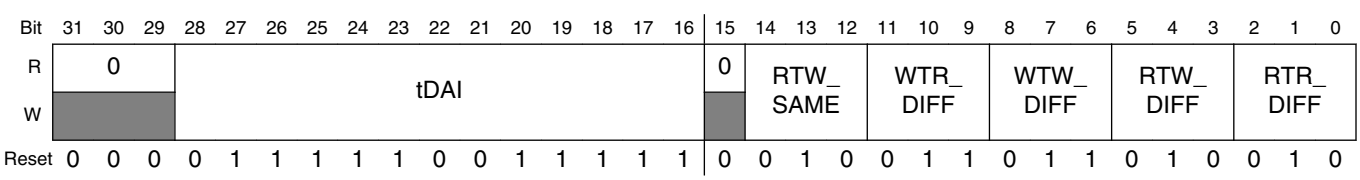
ESDCTL_ESDRCC field descriptions

Field	Description
31 RCC_EN	Read Command Counter Enable. 0 The counter is disabled. 1 The counter is enabled.
30 RCC_RST	Read Command Counter Reset. Note: Once asserted this bit deasserts automatically. 0 DO nothing. 1 The counter is set to zero.
29–0 RD_CMD_CNT	Read Command Counter value. This read-only field will display the number of read burst accesses that were sent to the memory. This is used mainly for bus utilization calculation.

28.121.12 ESDCTL Read/Write Command Delay (ESDCTL_ESDRWD)

This register affects the delay between back to back RD / WR accesses according to the details of each field. The register reset values are set to the minimum required value. The default values are set for achieving the best results and thus not recommended to be changed.

Address: ESDCTL_ESDRWD is 63FD_9000h base + 2Ch offset = 63FD_902Ch



ESDCTL_ESDRWD field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28–16 tDAI	Device auto initialization period.(maximum) Field relevant only to LPDDR2. 0x0 1 cycle 0xF9F 4000 cycles (Default, JEDEC value for LPDDR2, gives 10us at 400MHz clock). 0x1FFF 8192 cycles
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 RTW_SAME	Controls the cycles delay between Read to Write commands in same chip select. Delay is: BL/2 + RTW_SAME + (tCL-tCWL) + RALAT This field determines the value of RTW_SAME. 0x0 0 cycle 0x1 1 cycle 0x2 2 cycles (Default) 0x3 3 cycles 0x4 4 cycles 0x5 5 cycles 0x6 6 cycles 0x7 7 cycles
11–9 WTR_DIFF	Controls the cycles delay between Write to Read commands in different chip select. Delay is: BL/2 + WTR_DIFF - (tCL-tCWL) + WALAT This field determines the value of WTR_DIFF. 0x0 0 cycle 0x1 1 cycle 0x2 2 cycles 0x3 3 cycles (Default) 0x4 4 cycles 0x5 5 cycles 0x6 6 cycles 0x7 7 cycles
8–6 WTW_DIFF	Controls the cycles delay between Write to Write commands in different chip select. Delay is: BL/2 + WTW_DIFF This field determines the value of WTW_DIFF. 0x0 0 cycle 0x1 1 cycle 0x2 2 cycles 0x3 3 cycles (Default) 0x4 4 cycles 0x5 5 cycles

Table continues on the next page...

ESDCTL_ESDRWD field descriptions (continued)

Field	Description
	0x6 6 cycles 0x7 7 cycles
5–3 RTW_DIFF	Controls the cycles delay between Read to Write commands in different chip select. Delay is: BL/2 + RTW_DIFF + (tCL - tCWL) + RALAT This field determines the value of RTW_DIFF. 0x0 0 cycle 0x1 1 cycle 0x2 2 cycles (Default) 0x3 3 cycles 0x4 4 cycles 0x5 5 cycles 0x6 6 cycles 0x7 7 cycles
2–0 RTR_DIFF	Controls the cycles delay between Read to Read commands in different chip select. Delay is: BL/2 + RTR_DIFF This field determines the value of RTR_DIFF. 0x0 0 cycle 0x1 1 cycle 0x2 2 cycles (Default) 0x3 3 cycles 0x4 4 cycles 0x5 5 cycles 0x6 6 cycles 0x7 7 cycles

28.121.13 ESDCTL Out of Reset Delays (ESDCTL_ESDOR)

This register defines some delays that must be kept when controller exit from reset.

Address: ESDCTL_ESDOR is 63FD_9000h base + 30h offset = 63FD_9030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								tXPR								0		SDE_to_RST						0		RST_to_CKE					
W																																
Reset	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0

ESDCTL_ESDOR field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

ESDCTL_ESDOR field descriptions (continued)

Field	Description
23–16 tXPR	DDR2/DDR3: CKE HIGH to a valid command. LPDDR2: Not relevant. DDR2: This value equals a fixed 400 ns by JEDEC. DDR3: As defined in timing parameter table. 0x0 Reserved 0x1 2 cycles 0x2 3 cycles 0xFE 255 cycles 0xFF 256 cycles
15–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13–8 SDE_to_RST	DDR3: Time from SDE enable until DDR reset# is high. DDR2 /LPDDR2 : Not relevant. Note: Each cycle in this field is 15.625 us. 0x0 Reserved 0x1 Reserved 0x2 1 cycles 0x3 2 cycles 0xE 13 cycles (Jedec value for DDR3) - total of 200 us 0x3E 61 cycles 0x3F 62 cycles
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5–0 RST_to_CKE	DDR3: Time from SDE enable to CKE rise. In case that DDR reset# is low, will wait until it's high and then wait this period until rising CKE. (JEDEC value is 500 us) DDR2: Time from SDE enable to CKE rise. (JEDEC value is 200 us) LPDDR2: Idle time after first CKE assertion. (JEDEC value is 200 us) **NOTE:** Each cycle in this field is 15.625 us. 0x0 Reserved 0x1 Reserved 0x2 1 cycles 0x3 2 cycles 0xE 13 cycles (Jedec value for DDR2 /LPDDR2) - total of 200 us 0x21 32 cycles (Jedec value for DDR3) - total of 500 us 0x3E 61 cycles 0x3F 62 cycles

28.121.14 ESDCTL MRR DATA Register (ESDCTL_ESDMRR)

This register contains data which was collected after issuing MRR command. (LPDDR2 only)

Address: ESDCTL_ESDMRR is 63FD_9000h base + 34h offset = 63FD_9034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MRR_READ_DATA3								MRR_READ_DATA2								MRR_READ_DATA1								MRR_READ_DATA0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_ESDMRR field descriptions

Field	Description
31–24 MRR_READ_DATA3	MRR DATA that arrived on DQ[31:24]
23–16 MRR_READ_DATA2	MRR DATA that arrived on DQ[23:16]
15–8 MRR_READ_DATA1	MRR DATA that arrived on DQ[15:8]
7–0 MRR_READ_DATA0	MRR DATA that arrived on DQ[7:0]

28.121.15 ESDCTL Timing Configuration Register 3 (ESDCTL_ESDCFG3_LP)

Address: ESDCTL_ESDCFG3_LP is 63FD_9000h base + 38h offset = 63FD_9038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										RC_LP						0				tRCD_LP				tRPpb_LP				tRPab_LP			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_ESDCFG3_LP field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value zero. Reserved
21–16 RC_LP	ACT to ACT or REF command period (same bank). (This field is valid only for LPDDR2 memories)

Table continues on the next page...

ESDCTL_ESDCFG3_LP field descriptions (continued)

Field	Description
	0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0x3E 63 clocks 0x3F Reserved
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11–8 tRCD_LP	ACT command to internal read or write delay time (same bank). (This field is valid only for LPDDR2 memories) 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0xE 15 clocks 0xF Reserved
7–4 tRPpb_LP	PRECHARGE (per bank) command period (same bank). (This field is valid only for LPDDR2 memories) 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0xE 15 clocks 0xF Reserved
3–0 tRPab_LP	PRECHARGE (all banks) command period. (This field is valid only for LPDDR2 memories) 0x0 1 clock 0x1 2 clocks 0x2 3 clocks 0xE 15 clocks 0xF Reserved

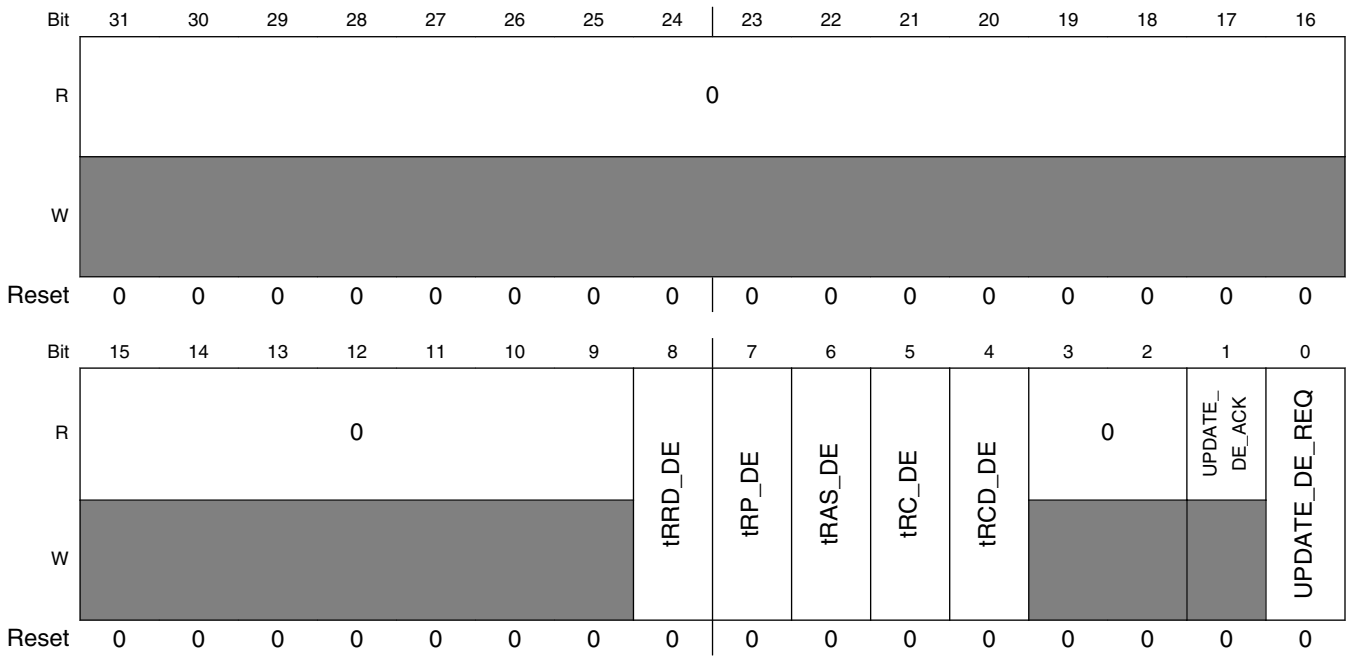
28.121.16 ESDCTL MR4 Derating Register (ESDCTL_ESDMR4)

This register is relevant only to LPDDR2 memories. It is used to dynamically change certain values depending on MR4 read result, which is based on memory temperature sensor result.

NOTE

The temperature sensor is located in the external LPDDR2 memory if supported, it is NOT part of the i.MX53 ESDCTL interface. This Temperature sensor is not the same as the one located within SATA module.

Address: ESDCTL_ESDMR4 is 63FD_9000h base + 3Ch offset = 63FD_903Ch



ESDCTL_ESDMR4 field descriptions

Field	Description
31–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 tRRD_DE	tRRD derating value. 0x0 Original tRRD is used. 0x1 tRRD is derated in 1 cycle.
7 tRP_DE	tRP derating value. 0x0 Original tRP is used. 0x1 tRP is derated in 1 cycle.
6 tRAS_DE	tRAS derating value. 0x0 Original tRAS is used. 0x1 tRAS is derated in 1 cycle.
5 tRC_DE	tRC derating value. 0x0 Original tRC is used. 0x1 tRC is derated in 1 cycle.
4 tRCD_DE	tRCD derating value. 0x0 Original tRCD is used. 0x1 tRCD is derated in 1 cycle.
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved

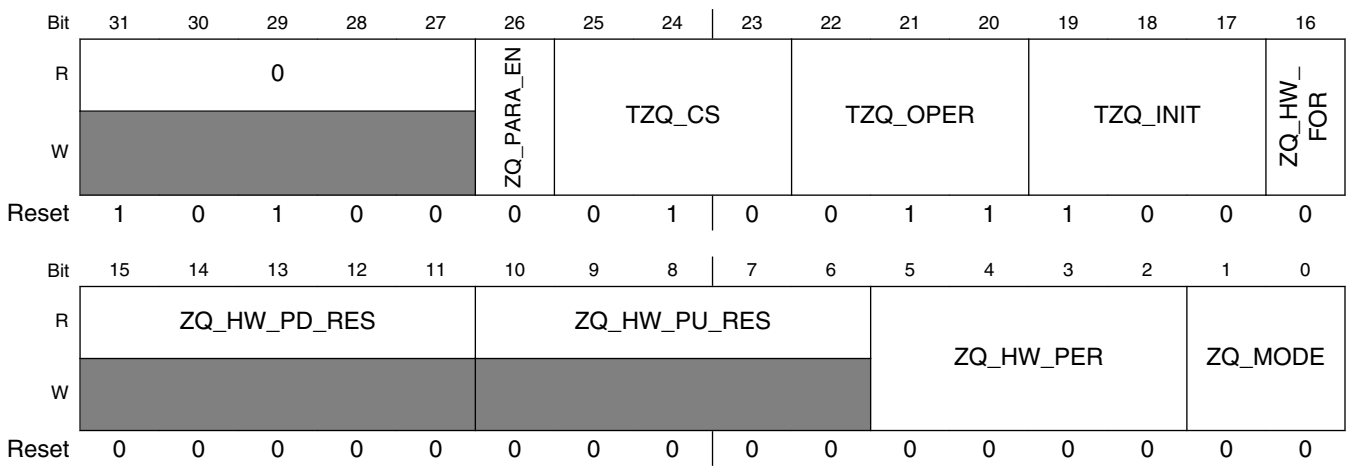
Table continues on the next page...

ESDCTL_ESDMR4 field descriptions (continued)

Field	Description
1 UPDATE_DE_ACK	Update Derated Values Acknowledge. This read only bit is arecleared upon UPDATE_DE_REQ assertion and isare set after the new values are taken.
0 UPDATE_DE_REQ	Update Derated Values Request. This read modify write field is automatically cleared after the request is issued. 0x0 Do nothing. 0x1 Request to update the following values: tRRD, tRCD, tRP, tRC, tRAS and refresh related fields(ESDCTL_ESDREF register): REF_CNT, REF_SEL, REFR

28.121.17 PHY ZQ HW Control Register (ESDCTL_ZQHWCTRL)

Address: ESDCTL_ZQHWCTRL is 63FD_9000h base + 40h offset = 63FD_9040h



ESDCTL_ZQHWCTRL field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26 ZQ_PARA_EN	Device ZQ calibration parallel enable. 0- Device ZQ calibration is done in serial (CS0 first and then CS1). 1- ZQ calibration of both CS is done in parallel
25–23 TZQ_CS	Device ZQ CS time. This field holds the number of cycles that are required by the device to preform ZQ short calibration. **NOTE:** This field should not be update during ZQ calibration. 000 Reserved 001 Reserved 010 128 cycles - Default

Table continues on the next page...

ESDCTL_ZQHWCTRL field descriptions (continued)

Field	Description
	011 256 cycles 100 512 cycles 101 1024 cycles 110-111 Reserved
22–20 TZQ_OPER	Device ZQ oper time. This field holds the number of cycles that are required by the device to preform ZQ oper calibration. **NOTE:** This field should not be update during ZQ calibration. 000 Reserved 001 Reserved 010 128 cycles 011 256 cycles - Default (JEDEC value for DDR3) 100 512 cycles 101 1024 cycles 110-111 Reserved
19–17 TZQ_INIT	Device ZQ init time. This field holds the number of cycles that are required by the device to preform ZQ init calibration. **NOTE:** This field should not be update during ZQ calibration. 000 Reserved 001 Reserved 010 128 cycles 011 256 cycles 100 512 cycles - Default (JEDEC value for DDR3) 101 1024 cycles 110-111 Reserved
16 ZQ_HW_FOR	Force ZQ HW calibration. This bit when asserted force ZQ HW calibration. SW should make sure all access to DDR is finished before asserting this bit. HW should negated this bit upon completion of the ZQ HW calibration.Upon negation of this bit the ZQ HW calibration result is valid
15–11 ZQ_HW_PD_ RES	ZQ HW calibration pull-down result. This field holds the pull-down resistor value calculated by ZQ HW calibration. 00000 Min. resistance. 11111 Max. resistance.
10–6 ZQ_HW_PU_ RES	ZQ HW calibration pull-up result. This field holds the pull-up resistor value calculated by ZQ HW calibration. 00000 Min. resistance. 11111 Max. resistance.
5–2 ZQ_HW_PER	ZQ HW periodic calibration time. This field determines how often ZQ HW calibration time is preformed in regular operation mode. This field is ignored if ZQ_HW_PER_EN is disabled. 0000 ZQ calibration performed every 1 ms sec. 0001 ZQ calibration performed every 2 ms sec.

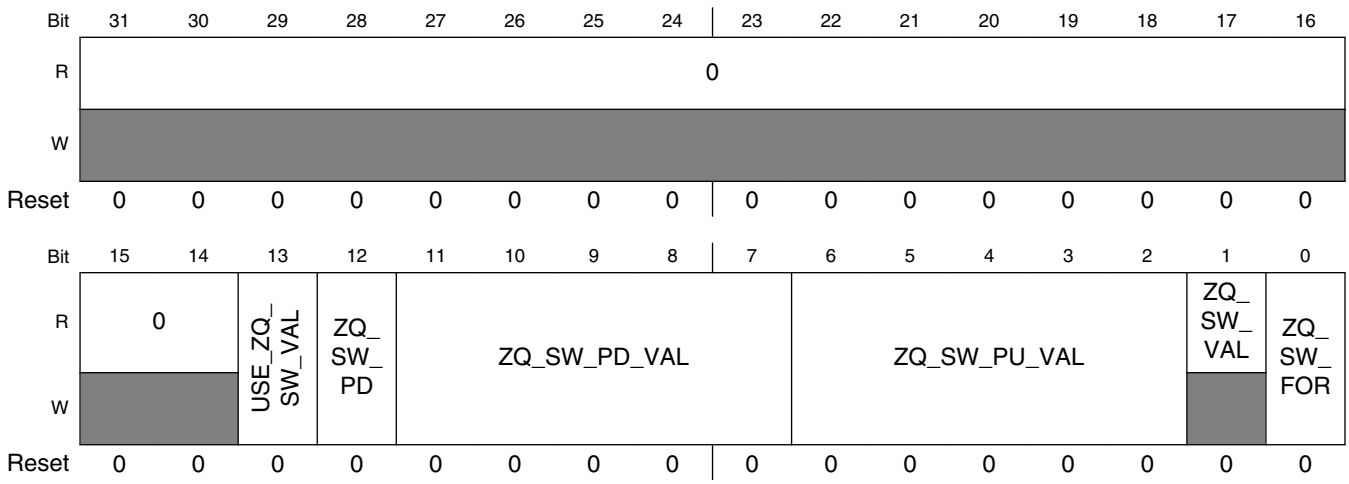
Table continues on the next page...

ESDCTL_ZQHWCTRL field descriptions (continued)

Field	Description
	0010 ZQ calibration performed every 4ms sec. 1111 ZQ calibration performed every 32 sec.
1–0 ZQ_MODE	ZQ calibration mode: 0x0 No ZQ calibration is issued. (Default) 0x1 ZQ calibration is issued to i.MX ZQ pad and external device.(Only when exiting self refresh) 0x2 ZQ calibration is issued to external device only. (Both periodic and when exiting self refresh) 0x3 ZQ calibration is issued to i.MX ZQ pad and external device. (Both periodic and when exiting self refresh)

28.121.18 PHY ZQ SW Control Register (ESDCTL_ZQSWCTRL)

Address: ESDCTL_ZQSWCTRL is 63FD_9000h base + 44h offset = 63FD_9044h



ESDCTL_ZQSWCTRL field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 USE_ZQ_SW_VAL	Use SW ZQ configured value for I/O pads resistor controls. This bit if asserted causes to DDR PHY to drive SW ZQ configured value on the resistor controls of the I/O pads. By default this bit is cleared and PHY drives the HW ZQ status bits on the resistor controls of the I/O pads. **NOTE:** This bit should not be updated during ZQ calibration. 0 -Fields ZQ_HW_PD_VAL & ZQ_HW_PU_VAL are used for driving I/O pads resistor controls. 1- Fields ZQ_SW_PD_VAL & ZQ_SW_PU_VAL are used for driving I/O pads resistor controls.
12 ZQ_SW_PD	ZQ software PU/PD calibration. This bit determines the calibration stage (PU or PD). 0 PU resistor calibration 1 PD resistor calibration

Table continues on the next page...

ESDCTL_ZQSWCTRL field descriptions (continued)

Field	Description
11–7 ZQ_SW_PD_VAL	ZQ software pull-down resistence.This field determines the value of the PD resistor during SW ZQ calibration. 00000 Max. resistance. 11111 Min. resistance.
6–2 ZQ_SW_PU_VAL	ZQ software pull-up resistence.This field determines the value of the PU resistor during SW ZQ calibration. 00000 Max. resistance. 11111 Min. resistance.
1 ZQ_SW_VAL	ZQ software calibration result. This bit reflects the ZQ calibration voltage comparator value. 0 Current ZQ calibration voltage is less than VDD/2. 1 Current ZQ calibration voltage is more than VDD/2
0 ZQ_SW_FOR	Force ZQ SW calibration. This bit when asserted force ZQ SW calibration. HW should negated this bit upon completion of the ZQ SW calibration.Upon negation of this bit the ZQ SW calibration result is valid

28.121.19 PHY Write Leveling General Control Register (ESDCTL_WLGCR)

Address: ESDCTL_WLGCR is 63FD_9000h base + 48h offset = 63FD_9048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				WL_HW_ERR3	WL_HW_ERR2	WL_HW_ERR1	WL_HW_ERR0	WL_SW_RES3	WL_SW_RES2	WL_SW_RES1	WL_SW_RES0	0	SW_WL_CNT_EN	SW_WL_EN	HW_WL_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_WLGCR field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11 WL_HW_ERR3	Byte3 WL HW calibration error. This bit is asserted when an error was found on byte3 during wl hw calibration. This bit is valid only upon completion of the wl HW calibration. (hw_wl_en bit is de-asserted)

Table continues on the next page...

ESDCTL_WLGCR field descriptions (continued)

Field	Description
	0 No error was found on byte3 during wl HW calibration. 1 An error was found on byte3 during wl HW calibration.
10 WL_HW_ERR2	Byte2 WL HW calibration error. This bit is asserted when an error was found on byte2 during wl hw calibration. This bit is valid only upon completion of the wl HW calibration. (hw_wl_en bit is de-asserted) 0 No error was found on byte2 during wl HW calibration. 1 An error was found on byte2 during wl HW calibration.
9 WL_HW_ERR1	Byte1 WL HW calibration error. This bit is asserted when an error was found on byte1 during wl hw calibration. This bit is valid only upon completion of the wl HW calibration. (hw_wl_en bit is de-asserted) 0 No error was found on byte1 during wl HW calibration. 1 An error was found on byte1 during wl HW calibration.
8 WL_HW_ERR0	Byte0 WL HW calibration error. This bit is asserted when an error was found on byte0 during wl hw calibration. This bit is valid only upon completion of the wl HW calibration. (hw_wl_en bit is de-asserted) 0 No error was found on byte0 during wl HW calibration. 1 An error was found on byte0 during wl HW calibration.
7 WL_SW_RES3	Byte3 WL software result. This bit reflects the DQ24 value of the SW WL. 0 DQS3 sampled low CK during SW WL. 1 DQS3 sampled high CK during SW WL.
6 WL_SW_RES2	Byte2 WL software result. This bit reflects the DQ16 value of the SW WL. 0 DQS2 sampled low CK during SW WL. 1 DQS2 sampled high CK during SW WL.
5 WL_SW_RES1	Byte1 WL software result. This bit reflects the DQ8 value of the SW WL. 0 DQS1 sampled low CK during SW WL. 1 DQS1 sampled high CK during SW WL.
4 WL_SW_RES0	Byte0 WL software result. This bit reflects the DQ0 value of the SW WL. 0 DQS0 sampled low CK during SW WL. 1 DQS0 sampled high CK during SW WL.
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2 SW_WL_CNT_EN	SW WL count down enable.This bit when asserted delay the WL DQS by 25+15 cycles.This bit should be asserted before the issue of the first SW WL request after issuing the write leveling MRS command 0 PHY doesn't count 25+15 cycles before issuing WL DQS. 1 PHY counts 25+15 cycles before issuing WL DQS.
1 SW_WL_EN	Enable WL SW update. This bit when asserted causes the PHY to preform WL SW update. HW negate this bit upon completion of the SW WL .Negation of this bit also points that the WL SW calibration result is valid

Table continues on the next page...

ESDCTL_WLGCR field descriptions (continued)

Field	Description
	NOTE: If this bit and the WL_SW_CNT_EN are enabled the PHY counts 25 + 15 cycles before issuing the SW WL dqs.
0 HW_WL_EN	Enable WL HW update. This bit when asserted causes the PHY to preform WL HW update. HW negate this bit upon completion of the WL HW update. Negation of this bit also points that the WL HW calibration result is valid before issuing the first dqs the PHY counts 25 + 15 cycles.

28.121.20 PHY Write Leveling Delay Control Register 0 (ESDCTL_WLDECTRL0)

Address: ESDCTL_WLDECTRL0 is 63FD_9000h base + 4Ch offset = 63FD_904Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					WL_CYC_DEL1		WL_HC_DEL1	0	WR_DL_ABS_OFFSET1						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					WL_CYC_DEL0		WL_HC_DEL0	0	WR_DL_ABS_OFFSET0						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_WLDECTRL0 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 WL_CYC_DEL1	Write level cycle delay. This field holds the number of delay cycles to be added for byte1 write-leveling value This delay is added to the delay that is generated by the write leveling delay line 1. 0 No delay is added. 1 1 cycle delay is added. 2 2 cycles delay is added. 3 Reserved.
24 WL_HC_DEL1	Write level half cycle delay. This field holds the number of half delay cycles to be added for byte1 write-leveling value This delay is added to the delay that is generated by the write leveling delay line 1. 0 No delay is added. 1 Half cycle delay is added.

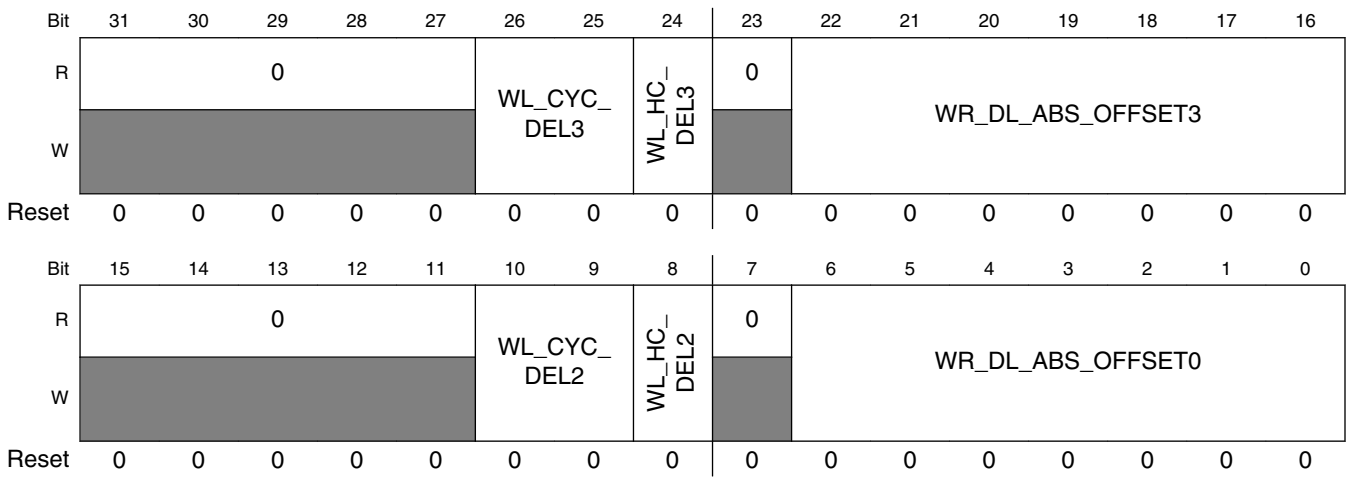
Table continues on the next page...

ESDCTL_WLDECTRL0 field descriptions (continued)

Field	Description
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–16 WR_DL_ABS_OFFSET1	Absolute delay offset. This field decides the specific delay line absolute delay in fraction of a cycle terms. The fraction is process and frequency independent.The delay for the delay line would be (WR_DL_ABS_OFFSET1 / 256) * fast_clk. This field can also bit updated by HW. Upon completion of the wl hw calibration this field gets the value of the wl HW calibration result. Note that not all changes will have effect on the actual delay. If the requested change is smaller than the delay line resolution (1 delay unit, which varies between 20 pSec in best case to 50 in worst case), then no change will occur.
15–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–9 WL_CYC_DEL0	Write level cycle delay. This field holds the number of delay cycles to be added for byte0 write-leveling value This delay is added to the delay that is generated by the write leveling delay line 0. 0 No delay is added. 1 1 cycle delay is added. 2 2 cycles delay is added. 3 Reserved.
8 WL_HC_DEL0	Write level half cycle delay. This field holds the number of half delay cycles to be added for byte0 write-leveling value This delay is added to the delay that is generated by the write leveling delay line 0. 0 No delay is added. 1 Half cycle delay is added.
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–0 WR_DL_ABS_OFFSET0	Absolute delay offset. This field decides the specific delay line absolute delay in fraction of a cycle terms. The fraction is process and frequency independent.The delay for the delay line would be (WR_DL_ABS_OFFSET0 / 256) * fast_clk. This field can also bit updated by HW. Upon completion of the wl hw calibration this field gets the value of the wl HW calibration result. Note that not all changes will have effect on the actual delay. If the requested change is smaller than the delay line resolution (1 delay unit, which varies between 20 pSec in best case to 50 in worst case), then no change will occur.

28.121.21 PHY Write Leveling Delay Control Register 1 (ESDCTL_WLDECTRL1)

Address: ESDCTL_WLDECTRL1 is 63FD_9000h base + 50h offset = 63FD_9050h



ESDCTL_WLDECTRL1 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 WL_CYC_DEL3	Write level cycle delay. This field holds the number of delay cycles to be added for byte3 write-leveling value This delay is added to the delay that is generated by the write leveling delay line 3. 0 No delay is added. 1 1 cycle delay is added. 2 2 cycles delay is added. 3 Reserved.
24 WL_HC_DEL3	Write level half cycle delay. This field holds the number of half delay cycles to be added for byte3 write-leveling value This delay is added to the delay that is generated by the write leveling delay line 3. 0 No delay is added. 1 Half cycle delay is added.
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–16 WR_DL_ABS_OFFSET3	Absolute delay offset. This field decides the specific delay line absolute delay in fraction of a cycle terms. The fraction is process and frequency independent.The delay for the delay line would be (WR_DL_ABS_OFFSET3/ 256) * fast_clk. This field can also bit updated by HW. Upon completion of the wl hw calibration this field gets the value of the wl HW calibration result.

Table continues on the next page...

ESDCTL_WLDECTRL1 field descriptions (continued)

Field	Description
	Note that not all changes will have effect on the actual delay. If the requested change is smaller than the delay line resolution (1 delay unit, which varies between 20 pSec in best case to 50 in worst case), then no change will occur.
15–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–9 WL_CYC_DEL2	Write level cycle delay. This field holds the number of delay cycles to be added for byte2 write-leveling value This delay is added to the delay that is generated by the write leveling delay line 2. 0 No delay is added. 1 1 cycle delay is added. 2 2 cycles delay is added. 3 Reserved.
8 WL_HC_DEL2	Write level half cycle delay. This field holds the number of half delay cycles to be added for byte2 write-leveling value This delay is added to the delay that is generated by the write leveling delay line 2. 0 No delay is added. 1 Half cycle delay is added.
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–0 WR_DL_ABS_OFFSET0	Absolute delay offset. This field decides the specific delay line absolute delay in fraction of a cycle terms. The fraction is process and frequency independent.The delay for the delay line would be (WR_DL_ABS_OFFSET0 / 256) * fast_clk. This field can also bit updated by HW. Upon completion of the wl hw calibration this field gets the value of the wl HW calibration result. Note that not all changes will have effect on the actual delay. If the requested change is smaller than the delay line resolution (1 delay unit, which varies between 20 pSec in best case to 50 in worst case), then no change will occur.

28.121.22 PHY Write Leveling Delay Line Status Register (ESDCTL_WLDLST)

This register holds the status of the 4 write leveling delay lines.

Address: ESDCTL_WLDLST is 63FD_9000h base + 54h offset = 63FD_9054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	WL_DL_UNIT_NUM3							0	WL_DL_UNIT_NUM2							0	WL_DL_UNIT_NUM1							0	WL_DL_UNIT_NUM0						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_WLDLST field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–24 WL_DL_UNIT_NUM3	This field shows the number of delay units that is actually used by write leveling delay line 3.
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–16 WL_DL_UNIT_NUM2	This field shows the number of delay units that is actually used by write leveling delay line 2.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–8 WL_DL_UNIT_NUM1	This field shows the number of delay units that is actually used by write leveling delay line 1.
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–0 WL_DL_UNIT_NUM0	This field shows the number of delay units that is actually used by write leveling delay line 0.

28.121.23 PHY ODT Control Register (ESDCTL_ODTCTRL)

Table 28-34. ODT Values

	ODTx_INT_Res Value 000	ODTx_INT_Res Value 001	ODTx_INT_Res Value 010	ODTx_INT_Res Value 011	ODTx_INT_Res Value 100	ODTx_INT_Res Value 101	ODTx_INT_Res Value 110	ODTx_INT_Res Value 111	Unit
DDR2 mode (1.8V, ddr_sel=00, ZQ calibration res = 300 Ohm)									
Resistance Value	-	150	75	50	37.5	30	25	21	Ohm
DDR2 mode (1.8V, ddr_sel=01, ZQ calibration res = 180 Ohm)									
Resistance Value	-	90	45	30	22.5	18	15	13	Ohm
DDR2 mode (1.8V, ddr_sel=10, ZQ calibration res = 200 Ohm)									
Resistance Value	-	100	50	33	25	20	16.5	14	Ohm
DDR2 mode (1.8V, ddr_sel=11, ZQ calibration res = 140 Ohm)									
Resistance Value	-	70	35	23	17.5	14	11.5	10	Ohm
DDR3 mode (1.5V, ddr_sel=00, ZQ calibration res = 240 Ohm)									

Table continues on the next page...

Table 28-34. ODT Values (continued)

	ODTx_INT_Res Value 000	ODTx_INT_Res Value 001	ODTx_INT_Res Value 010	ODTx_INT_Res Value 011	ODTx_INT_Res Value 100	ODTx_INT_Res Value 101	ODTx_INT_Res Value 110	ODTx_INT_Res Value 111	Unit
Resistance Value	-	120	60	40	30	24	20	17	Ohm
DDR3 mode (1.5V, ddr_sel=01, ZQ calibration res = 160 Ohm)									
Resistance Value	-	80	40	26.5	20	16	13.5	11.5	Ohm

Address: ESDCTL_ODTCTRL is 63FD_9000h base + 58h offset = 63FD_9058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													ODT3_INT_RES		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	ODT2_INT_RES			0	ODT1_INT_RES			0	ODT0_INT_RES			ODT_RD_ACT_EN	ODT_RD_PAS_EN	ODT_WR_ACT_EN	ODT_WR_PAS_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_ODTCTRL field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18–16 ODT3_INT_RES	On chip ODT byte3 resistor - This field determines the resistance of the on chip ODT byte3 resistor during read accesses. 000 ODT is Disabled. For the rest of the values, see Table 28-34 .
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 ODT2_INT_RES	On chip ODT byte2 resistor - This field determines the resistance of the on chip ODT byte2 resistor during read accesses. 000 ODT is Disabled. For the rest of the values, see Table 28-34 .
11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–8 ODT1_INT_RES	On chip ODT byte1 resistor - This field determines the resistance of the on chip ODT byte1 resistor during read accesses.

Table continues on the next page...

ESDCTL_ODTCTRL field descriptions (continued)

Field	Description
	000 ODT is Disabled. For the rest of the values, see Table 28-34 .
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–4 ODT0_INT_RES	On chip ODT byte0 resistor - This field determines the resistance of the on chip ODT byte0 resistor during read accesses. 000 ODT is Disabled. For the rest of the values, see Table 28-34 .
3 ODT_RD_ACT_EN	Active read CS ODT enable. The bit determines if ODT pin of the active CS is areasserted during read accesses. 0 Active CS ODT pin is disabled during read access. 1 Active CS ODT pin is enabled during read access.
2 ODT_RD_PAS_EN	Inactive read CS ODT enable. The bit determines if ODT pin of the inactive CS is asserted during read accesses to other CS. 0 Inactive CS ODT pin is disabled during read accesses to other CS. 1 Inactive CS ODT pin is enabled during read accesses to other CS.
1 ODT_WR_ACT_EN	Active write CS ODT enable. The bit determines if ODT pin of the active CS is asserted during write accesses. 0 Active CS ODT pin is disabled during write access. 1 Active CS ODT pin is enabled during write access.
0 ODT_WR_PAS_EN	Inactive write CS ODT enable. The bit determines if ODT pin of the inactive CS is asserted during write accesses to other CS. 0 Inactive CS ODT pin is disabled during write accesses to other CS. 1 Inactive CS ODT pin is enabled during write accesses to other CS.

28.121.24 PHY Read DQ Byte0 Delay Register (ESDCTL_RDDQBY0DL)

Address: ESDCTL_RDDQBY0DL is 63FD_9000h base + 5Ch offset = 63FD_905Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	rd_dq7_del			0	rd_dq6_del			0	rd_dq5_del			0	rd_dq4_del		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	rd_dq3_del			0	rd_dq2_del			0	rd_dq1_del			0	rd_dq0_del		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_RDDQBY0DL field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–28 rd_dq7_del	Read dqs0 to dq7 delay fine tuning.This field holds the del. value that is added/reduced from dq7 relative to dqs0. 000 No change in dq7 delay 001 Add dq7 delay of 1 delay unit 010 Add dq7 delay of 2 delay units 011 Add dq7 delay of 3 delay units 100 Add dq7 delay of 4 delay units 101 Add dq7 delay of 5 delay units 110 Add dq7 delay of 6 delay units 111 Add dq7 delay of 7 delay units
27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–24 rd_dq6_del	Read dqs0 to dq6 delay fine tuning.This field holds the del. value that is added/reduced from dq6 relative to dqs0. 000 No change in dq6 delay 001 Add dq6 delay of 1 delay unit 010 Add dq6 delay of 2 delay units 011 Add dq6 delay of 3 delay units 100 Add dq6 delay of 4 delay units 101 Add dq6 delay of 5 delay units 110 Add dq6 delay of 6 delay units 111 Add dq6 delay of 7 delay units
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–20 rd_dq5_del	Read dqs0 to dq5 delay fine tuning.This field holds the del. value that is added/reduced from dq5 relative to dqs0. 000 No change in dq5 delay 001 Add dq5 delay of 1 delay unit 010 Add dq5 delay of 2 delay units 011 Add dq5 delay of 3 delay units 100 Add dq5 delay of 4 delay units 101 Add dq5 delay of 5 delay units 110 Add dq5 delay of 6 delay units 111 Add dq5 delay of 7 delay units
19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18–16 rd_dq4_del	Read dqs0 to dq4 delay fine tuning.This field holds the del. value that is added/reduced from dq4 relative to dqs0. 000 No change in dq4 delay 001 Add dq4 delay of 1 delay unit 010 Add dq4 delay of 2 delay units

Table continues on the next page...

ESDCTL_RDDQBY0DL field descriptions (continued)

Field	Description
	011 Add dq4 delay of 3 delay units 100 Add dq4 delay of 4 delay units 101 Add dq4 delay of 5 delay units 110 Add dq4 delay of 6 delay units 111 Add dq4 delay of 7 delay units
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 rd_dq3_del	Read dqs0 to dq3 delay fine tuning.This field holds the del. value that is added/reduced from dq3 relative to dqs0. 000 No change in dq3 delay 001 Add dq3 delay of 1 delay unit 010 Add dq3 delay of 2 delay units 011 Add dq3 delay of 3 delay units 100 Add dq3 delay of 4 delay units 101 Add dq3 delay of 5 delay units 110 Add dq3 delay of 6 delay units 111 Add dq3 delay of 7 delay units
11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–8 rd_dq2_del	Read dqs0 to dq2 delay fine tuning.This field holds the del. value that is added/reduced from dq2 relative to dqs0. 000 No change in dq2 delay 001 Add dq2 delay of 1 delay unit 010 Add dq2 delay of 2 delay units 011 Add dq2 delay of 3 delay units 100 Add dq2 delay of 4 delay units 101 Add dq2 delay of 5 delay units 110 Add dq2 delay of 6 delay units 111 Add dq2 delay of 7 delay units
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–4 rd_dq1_del	Read dqs0 to dq1 delay fine tuning.This field holds the del. value that is added/reduced from dq1 relative to dqs0. 000 No change in dq1 delay 001 Add dq1 delay of 1 delay unit 010 Add dq1 delay of 2 delay units 011 Add dq1 delay of 3 delay units 100 Add dq1 delay of 4 delay units 101 Add dq1 delay of 5 delay units 110 Add dq1 delay of 6 delay units 111 Add dq1 delay of 7 delay units
3 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

ESDCTL_RDDQBY0DL field descriptions (continued)

Field	Description
2–0 rd_dq0_del	Read dqs0 to dq0 delay fine tuning.This field holds the del. value that is added/reduced from dq0 relative to dqs0. 000 No change in dq0 delay 001 Add dq0 delay of 1 delay unit 010 Add dq0 delay of 2 delay units 011 Add dq0 delay of 3 delay units 100 Add dq0 delay of 4 delay units 101 Add dq0 delay of 5 delay units 110 Add dq0 delay of 6 delay units 111 Add dq0 delay of 7 delay units

28.121.25 PHY Read DQ Byte1 Delay Register (ESDCTL_RDDQBY1DL)

Address: ESDCTL_RDDQBY1DL is 63FD_9000h base + 60h offset = 63FD_9060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	rd_dq15_del			0	rd_dq14_del			0	rd_dq13_del			0	rd_dq12_del		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	rd_dq11_del			0	rd_dq10_del			0	rd_dq9_del			0	rd_dq8_del		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_RDDQBY1DL field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–28 rd_dq15_del	Read dqs1 to dq15 delay fine tuning.This field holds the del. value that is added/reduced from dq15 relative to dqs1. 000 No change in dq15 delay 001 Add dq15 delay of 1 delay unit 010 Add dq15 delay of 2 delay units 011 Add dq15 delay of 3 delay units 100 Add dq15 delay of 4 delay units 101 Add dq15 delay of 5 delay units 110 Add dq15 delay of 6 delay units 111 Add dq15 delay of 7 delay units

Table continues on the next page...

ESDCTL_RDDQBY1DL field descriptions (continued)

Field	Description
27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–24 rd_dq14_del	Read dqs1 to dq14 delay fine tuning.This field holds the del. value that is added/reduced from dq14 relative to dqs1. 000 No change in dq14 delay 001 Add dq14 delay of 1 delay unit 010 Add dq14 delay of 2 delay units 011 Add dq14 delay of 3 delay units 100 Add dq14 delay of 4 delay units 101 Add dq14 delay of 5 delay units 110 Add dq14 delay of 6 delay units 111 Add dq14 delay of 7 delay units
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–20 rd_dq13_del	Read dqs1 to dq13 delay fine tuning.This field holds the del. value that is added/reduced from dq13 relative to dqs1. 000 No change in dq13 delay 001 Add dq13 delay of 1 delay unit 010 Add dq13 delay of 2 delay units 011 Add dq13 delay of 3 delay units 100 Add dq13 delay of 4 delay units 101 Add dq13 delay of 5 delay units 110 Add dq13 delay of 6 delay units 111 Add dq13 delay of 7 delay units
19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18–16 rd_dq12_del	Read dqs1 to dq12 delay fine tuning.This field holds the del. value that is added/reduced from dq12 relative to dqs1. 000 No change in dq12 delay 001 Add dq12 delay of 1 delay unit 010 Add dq12 delay of 2 delay units 011 Add dq12 delay of 3 delay units 100 Add dq12 delay of 4 delay units 101 Add dq12 delay of 5 delay units 110 Add dq12 delay of 6 delay units 111 Add dq12 delay of 7 delay units
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 rd_dq11_del	Read dqs1 to dq11 delay fine tuning.This field holds the del. value that is added/reduced from dq11 relative to dqs1. 000 No change in dq11 delay 001 Add dq11 delay of 1 delay unit 010 Add dq11 delay of 2 delay units

Table continues on the next page...

ESDCTL_RDDQBY1DL field descriptions (continued)

Field	Description
	011 Add dq11 delay of 3 delay units 100 Add dq11 delay of 4 delay units 101 Add dq11 delay of 5 delay units 110 Add dq11 delay of 6 delay units 111 Add dq11 delay of 7 delay units
11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–8 rd_dq10_del	Read dqs1 to dq10 delay fine tuning.This field holds the del. value that is added/reduced from dq10 relative to dqs1. 000 No change in dq10 delay 001 Add dq10 delay of 1 delay unit 010 Add dq10 delay of 2 delay units 011 Add dq10 delay of 3 delay units 100 Add dq10 delay of 4 delay units 101 Add dq10 delay of 5 delay units 110 Add dq10 delay of 6 delay units 111 Add dq10 delay of 7 delay units
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–4 rd_dq9_del	Read dqs1 to dq9 delay fine tuning.This field holds the del. value that is added/reduced from dq9 relative to dqs1. 000 No change in dq9 delay 001 Add dq9 delay of 1 delay unit 010 Add dq9 delay of 2 delay units 011 Add dq9 delay of 3 delay units 100 Add dq9 delay of 4 delay units 101 Add dq9 delay of 5 delay units 110 Add dq9 delay of 6 delay units 111 Add dq9 delay of 7 delay units
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 rd_dq8_del	Read dqs1 to dq8 delay fine tuning.This field holds the del. value that is added/reduced from dq8 relative to dqs1. 000 No change in dq8 delay 001 Add dq8 delay of 1 delay unit 010 Add dq8 delay of 2 delay units 011 Add dq8 delay of 3 delay units 100 Add dq8 delay of 4 delay units 101 Add dq8 delay of 5 delay units 110 Add dq8 delay of 6 delay units 111 Add dq8 delay of 7 delay units

28.121.26 PHY Read DQ Byte2 Delay Register (ESDCTL_RDDQBY2DL)

Address: ESDCTL_RDDQBY2DL is 63FD_9000h base + 64h offset = 63FD_9064h

Bit	31	30 29 28	27	26 25 24	23	22 21 20	19	18 17 16	15	14 13 12	11	10 9 8	7	6 5 4	3	2 1 0
R	0	rd_dq23_del	0	rd_dq22_del	0	rd_dq21_del	0	rd_dq20_del	0	rd_dq19_del	0	rd_dq18_del	0	rd_dq17_del	0	rd_dq16_del
W																
Reset	0	0 0 0	0	0 0 0	0	0 0 0	0	0 0 0	0	0 0 0	0	0 0 0	0	0 0 0	0	0 0 0

ESDCTL_RDDQBY2DL field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–28 rd_dq23_del	Read dqs2 to dq23 delay fine tuning.This field holds the del. value that is added/reduced from dq23 relative to dqs2. 000 No change in dq23 delay 001 Add dq23 delay of 1 delay unit 010 Add dq23 delay of 2 delay units 011 Add dq23 delay of 3 delay units 100 Add dq23 delay of 4 delay units 101 Add dq23 delay of 5 delay units 110 Add dq23 delay of 6 delay units 111 Add dq23 delay of 7 delay units
27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–24 rd_dq22_del	Read dqs2 to dq22 delay fine tuning.This field holds the del. value that is added/reduced from dq22 relative to dqs2. 000 No change in dq22 delay 001 Add dq22 delay of 1 delay unit 010 Add dq22 delay of 2 delay units 011 Add dq22 delay of 3 delay units 100 Add dq22 delay of 4 delay units 101 Add dq22 delay of 5 delay units 110 Add dq22 delay of 6 delay units 111 Add dq22 delay of 7 delay units
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–20 rd_dq21_del	Read dqs2 to dq21 delay fine tuning.This field holds the del. value that is added/reduced from dq21 relative to dqs2. 000 No change in dq21 delay 001 Add dq21 delay of 1 delay unit 010 Add dq21 delay of 2 delay units 011 Add dq21 delay of 3 delay units

Table continues on the next page...

ESDCTL_RDDQBY2DL field descriptions (continued)

Field	Description
	100 Add dq21 delay of 4 delay units 101 Add dq21 delay of 5 delay units 110 Add dq21 delay of 6 delay units 111 Add dq21 delay of 7 delay units
19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18–16 rd_dq20_del	Read dqs2 to dq20 delay fine tuning.This field holds the del. value that is added/reduced from dq20 relative to dqs2. 000 No change in dq20 delay 001 Add dq20 delay of 1 delay unit 010 Add dq20 delay of 2 delay units 011 Add dq20 delay of 3 delay units 100 Add dq20 delay of 4 delay units 101 Add dq20 delay of 5 delay units 110 Add dq20 delay of 6 delay units 111 Add dq20 delay of 7 delay units
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 rd_dq19_del	Read dqs2 to dq19 delay fine tuning.This field holds the del. value that is added/reduced from dq19 relative to dqs2. 000 No change in dq19 delay 001 Add dq19 delay of 1 delay unit 010 Add dq19 delay of 2 delay units 011 Add dq19 delay of 3 delay units 100 Add dq19 delay of 4 delay units 101 Add dq19 delay of 5 delay units 110 Add dq19 delay of 6 delay units 111 Add dq19 delay of 7 delay units
11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–8 rd_dq18_del	Read dqs2 to dq18 delay fine tuning.This field holds the del. value that is added/reduced from dq18 relative to dqs2. 000 No change in dq18 delay 001 Add dq18 delay of 1 delay unit 010 Add dq18 delay of 2 delay units 011 Add dq18 delay of 3 delay units 100 Add dq18 delay of 4 delay units 101 Add dq18 delay of 5 delay units 110 Add dq18 delay of 6 delay units 111 Add dq18 delay of 7 delay units
7 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

ESDCTL_RDDQBY2DL field descriptions (continued)

Field	Description
6–4 rd_dq17_del	Read dqs2 to dq17 delay fine tuning.This field holds the del. value that is added/reduced from dq17 relative to dqs2. 000 No change in dq17 delay 001 Add dq17 delay of 1 delay unit 010 Add dq17 delay of 2 delay units 011 Add dq17 delay of 3 delay units 100 Add dq17 delay of 4 delay units 101 Add dq17 delay of 5 delay units 110 Add dq17 delay of 6 delay units 111 Add dq17 delay of 7 delay units
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 rd_dq16_del	Read dqs2 to dq16 delay fine tuning.This field holds the del. value that is added/reduced from dq16 relative to dqs2. 000 No change in dq16 delay 001 Add dq16 delay of 1 delay unit 010 Add dq16 delay of 2 delay units 011 Add dq16 delay of 3 delay units 100 Add dq16 delay of 4 delay units 101 Add dq16 delay of 5 delay units 110 Add dq16 delay of 6 delay units 111 Add dq16 delay of 7 delay units

28.121.27 PHY Read DQ Byte3 Delay Register (ESDCTL_RDDQBY3DL)

Address: ESDCTL_RDDQBY3DL is 63FD_9000h base + 68h offset = 63FD_9068h

Bit	31	30 29 28	27	26 25 24	23	22 21 20	19	18 17 16	15	14 13 12	11	10 9 8	7	6 5 4	3	2 1 0
R	0	rd_dq31_del	0	rd_dq30_del	0	rd_dq29_del	0	rd_dq28_del	0	rd_dq27_del	0	rd_dq26_del	0	rd_dq25_del	0	rd_dq24_del
W																
Reset	0	0 0 0	0	0 0 0	0	0 0 0	0	0 0 0	0	0 0 0	0	0 0 0	0	0 0 0	0	0 0 0

ESDCTL_RDDQBY3DL field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–28 rd_dq31_del	Read dqs3 to dq31 delay fine tuning.This field holds the del. value that is added/reduced from dq31 relative to dqs3. 000 No change in dq31 delay 001 Add dq31 delay of 1 delay unit

Table continues on the next page...

ESDCTL_RDDQBY3DL field descriptions (continued)

Field	Description
	010 Add dq31 delay of 2 delay units 011 Add dq31 delay of 3 delay units 100 Add dq31 delay of 4 delay units 101 Add dq31 delay of 5 delay units 110 Add dq31 delay of 6 delay units 111 Add dq31 delay of 7 delay units
27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–24 rd_dq30_del	Read dqs3 to dq30 delay fine tuning.This field holds the del. value that is added/reduced from dq30 relative to dqs3. 000 No change in dq30 delay 001 Add dq30 delay of 1 delay unit 010 Add dq30 delay of 2 delay units 011 Add dq30 delay of 3 delay units 100 Add dq30 delay of 4 delay units 101 Add dq30 delay of 5 delay units 110 Add dq30 delay of 6 delay units 111 Add dq30 delay of 7 delay units
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–20 rd_dq29_del	Read dqs3 to dq29 delay fine tuning.This field holds the del. value that is added/reduced from dq29 relative to dqs3. 000 No change in dq29 delay 001 Add dq29 delay of 1 delay unit 010 Add dq29 delay of 2 delay units 011 Add dq29 delay of 3 delay units 100 Add dq29 delay of 4 delay units 101 Add dq29 delay of 5 delay units 110 Add dq29 delay of 6 delay units 111 Add dq29 delay of 7 delay units
19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18–16 rd_dq28_del	Read dqs3 to dq28 delay fine tuning.This field holds the del. value that is added/reduced from dq28 relative to dqs3. 000 No change in dq28 delay 001 Add dq28 delay of 1 delay unit 010 Add dq28 delay of 2 delay units 011 Add dq28 delay of 3 delay units 100 Add dq28 delay of 4 delay units 101 Add dq28 delay of 5 delay units 110 Add dq28 delay of 6 delay units 111 Add dq28 delay of 7 delay units

Table continues on the next page...

ESDCTL_RDDQBY3DL field descriptions (continued)

Field	Description
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 rd_dq27_del	Read dqs3 to dq27 delay fine tuning.This field holds the del. value that is added/reduced from dq27 relative to dqs3. 000 No change in dq27 delay 001 Add dq27 delay of 1 delay unit 010 Add dq27 delay of 2 delay units 011 Add dq27 delay of 3 delay units 100 Add dq27 delay of 4 delay units 101 Add dq27 delay of 5 delay units 110 Add dq27 delay of 6 delay units 111 Add dq27 delay of 7 delay units
11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–8 rd_dq26_del	Read dqs3 to dq26 delay fine tuning.This field holds the del. value that is added/reduced from dq26 relative to dqs3. 000 No change in dq26 delay 001 Add dq26 delay of 1 delay unit 010 Add dq26 delay of 2 delay units 011 Add dq26 delay of 3 delay units 100 Add dq26 delay of 4 delay units 101 Add dq26 delay of 5 delay units 110 Add dq26 delay of 6 delay units 111 Add dq26 delay of 7 delay units
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–4 rd_dq25_del	Read dqs3 to dq25 delay fine tuning.This field holds the del. value that is added/reduced from dq25 relative to dqs3. 000 No change in dq25 delay 001 Add dq25 delay of 1 delay unit 010 Add dq25 delay of 2 delay units 011 Add dq25 delay of 3 delay units 100 Add dq25 delay of 4 delay units 101 Add dq25 delay of 5 delay units 110 Add dq25 delay of 6 delay units 111 Add dq25 delay of 7 delay units
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 rd_dq24_del	Read dqs3 to dq24 delay fine tuning.This field holds the del. value that is added/reduced from dq24 relative to dqs3. 000 No change in dq24 delay 001 Add dq24 delay of 1 delay unit 010 Add dq24 delay of 2 delay units

Table continues on the next page...

ESDCTL_RDDQBY3DL field descriptions (continued)

Field	Description
	011 Add dq24 delay of 3 delay units 100 Add dq24 delay of 4 delay units 101 Add dq24 delay of 5 delay units 110 Add dq24 delay of 6 delay units 111 Add dq24 delay of 7 delay units

28.121.28 PHY Write DQ Byte0 Delay Register (ESDCTL_WRDQBY0DL)

Address: ESDCTL_WRDQBY0DL is 63FD_9000h base + 6Ch offset = 63FD_906Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	wr_dm0_del		wr_dq7_del		0		wr_dq6_del		0		wr_dq5_del		0		wr_dq4_del		0		wr_dq3_del		0		wr_dq2_del		0		wr_dq1_del		0		wr_dq0_del	
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_WRDQBY0DL field descriptions

Field	Description
31–30 wr_dm0_del	Write dm0 delay fine tuning.This field holds the del. value that is added to dm0 relative to dqs0. 00 No change in dm0 delay 01 Add dm0 delay of 1 delay units. 10 Add dm0 delay of 2 delay units. 11 Add dm0 delay of 3 delay units
29–28 wr_dq7_del	Write dq7 delay fine tuning.This field holds the del. value that is added to dq7 relative to dqs0. 00 No change in dq7 delay 01 Add dq7 delay of 1 delay units. 10 Add dq7 delay of 2 delay units. 11 Add dq7 delay of 3 delay units
27–26 Reserved	This read-only field is reserved and always has the value zero. Reserved
25–24 wr_dq6_del	Write dq6 delay fine tuning.This field holds the del. value that is added to dq6 relative to dqs0. 00 No change in dq6 delay 01 Add dq6 delay of 1 delay units. 10 Add dq6 delay of 2 delay units. 11 Add dq6 delay of 3 delay units
23–22 Reserved	This read-only field is reserved and always has the value zero. Reserved
21–20 wr_dq5_del	Write dq5 delay fine tuning.This field holds the del. value that is added to dq5 relative to dqs0.

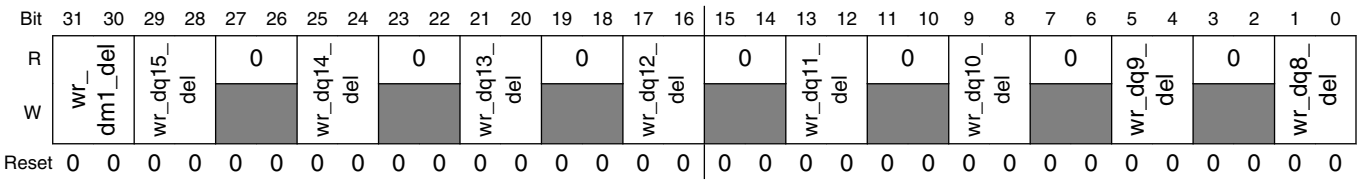
Table continues on the next page...

ESDCTL_WRDQBY0DL field descriptions (continued)

Field	Description
	00 No change in dq5 delay 01 Add dq5 delay of 1 delay units. 10 Add dq5 delay of 2 delay units. 11 Add dq5 delay of 3 delay units
19–18 Reserved	This read-only field is reserved and always has the value zero. Reserved
17–16 wr_dq4_del	Write dq4 delay fine tuning.This field holds the del. value that is added to dq4 relative to dqs0. 00 No change in dq4 delay 01 Add dq4 delay of 1 delay units. 10 Add dq4 delay of 2 delay units. 11 Add dq4 delay of 3 delay units
15–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13–12 wr_dq3_del	Write dq3 delay fine tuning.This field holds the del. value that is added to dq3 relative to dqs0. 00 No change in dq3 delay 01 Add dq3 delay of 1 delay units. 10 Add dq3 delay of 2 delay units. 11 Add dq3 delay of 3 delay units
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–8 wr_dq2_del	Write dq2 delay fine tuning.This field holds the del. value that is added to dq2 relative to dqs0. 00 No change in dq2 delay 01 Add dq2 delay of 1 delay units. 10 Add dq2 delay of 2 delay units. 11 Add dq2 delay of 3 delay units
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5–4 wr_dq1_del	Write dq1 delay fine tuning.This field holds the del. value that is added to dq1 relative to dqs0. 00 No change in dq1 delay 01 Add dq1 delay of 1 delay units. 10 Add dq1 delay of 2 delay units. 11 Add dq1 delay of 3 delay units
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 wr_dq0_del	Write dq0 delay fine tuning.This field holds the del. value that is added to dq0 relative to dqs0. 00 No change in dq0 delay 01 Add dq0 delay of 1 delay units. 10 Add dq0 delay of 2 delay units. 11 Add dq0 delay of 3 delay units

28.121.29 PHY Write DQ Byte1 Delay Register (ESDCTL_WRDQBY1DL)

Address: ESDCTL_WRDQBY1DL is 63FD_9000h base + 70h offset = 63FD_9070h



ESDCTL_WRDQBY1DL field descriptions

Field	Description
31–30 wr_dm1_del	Write dm1 delay fine tuning.This field holds the del. value that is added to dm1 relative to dqs1. 00 No change in dm1 delay 01 Add dm1 delay of 1 delay units. 10 Add dm1 delay of 2 delay units. 11 Add dm1 delay of 3 delay units
29–28 wr_dq15_del	Write dq15 delay fine tuning.This field holds the del. value that is added to dq15 relative to dqs1. 00 No change in dq15 delay 01 Add dq15 delay of 1 delay units. 10 Add dq15 delay of 2 delay units. 11 Add dq15 delay of 3 delay units
27–26 Reserved	This read-only field is reserved and always has the value zero. Reserved
25–24 wr_dq14_del	Write dq14 delay fine tuning.This field holds the del. value that is added to dq14 relative to dqs1. 00 No change in dq14 delay 01 Add dq14 delay of 1 delay units. 10 Add dq14 delay of 2 delay units. 11 Add dq14 delay of 3 delay units
23–22 Reserved	This read-only field is reserved and always has the value zero. Reserved
21–20 wr_dq13_del	Write dq13 delay fine tuning.This field holds the del. value that is added to dq13 relative to dqs1. 00 No change in dq13 delay 01 Add dq13 delay of 1 delay units. 10 Add dq13 delay of 2 delay units. 11 Add dq13 delay of 3 delay units
19–18 Reserved	This read-only field is reserved and always has the value zero. Reserved
17–16 wr_dq12_del	Write dq12 delay fine tuning.This field holds the del. value that is added to dq12 relative to dqs1. 00 No change in dq12 delay 01 Add dq12 delay of 1 delay units.

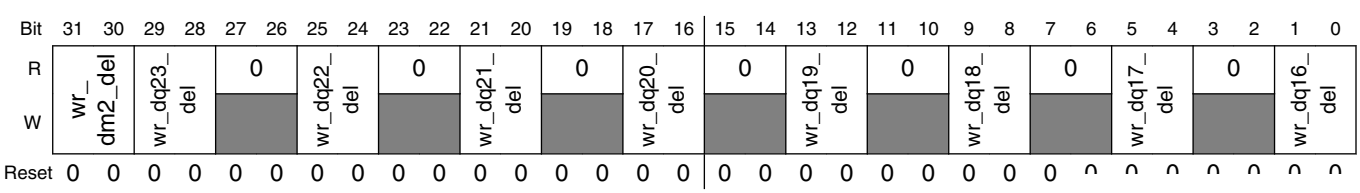
Table continues on the next page...

ESDCTL_WRDQBY1DL field descriptions (continued)

Field	Description
	10 Add dq12 delay of 2 delay units. 11 Add dq12 delay of 3 delay units
15–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13–12 wr_dq11_del	Write dq11 delay fine tuning.This field holds the del. value that is added to dq11 relative to dqs1. 00 No change in dq11 delay 01 Add dq11 delay of 1 delay units. 10 Add dq11 delay of 2 delay units. 11 Add dq11 delay of 3 delay units
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–8 wr_dq10_del	Write dq10 delay fine tuning.This field holds the del. value that is added to dq10 relative to dqs1. 00 No change in dq10 delay 01 Add dq10 delay of 1 delay units. 10 Add dq10 delay of 2 delay units. 11 Add dq10 delay of 3 delay units
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5–4 wr_dq9_del	Write dq9 delay fine tuning.This field holds the del. value that is added to dq9 relative to dqs1. 00 No change in dq9 delay 01 Add dq9 delay of 1 delay units. 10 Add dq9 delay of 2 delay units. 11 Add dq9 delay of 3 delay units
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 wr_dq8_del	Write dq8 delay fine tuning.This field holds the del. value that is added to dq8 relative to dqs1. 00 No change in dq8 delay 01 Add dq8 delay of 1 delay units. 10 Add dq8 delay of 2 delay units. 11 Add dq8 delay of 3 delay units

28.121.30 PHY Write DQ Byte2 Delay Register (ESDCTL_WRDQBY2DL)

Address: ESDCTL_WRDQBY2DL is 63FD_9000h base + 74h offset = 63FD_9074h



ESDCTL_WRDQBY2DL field descriptions

Field	Description
31–30 wr_dm2_del	Write dm2 delay fine tuning.This field holds the del. value that is added to dm2 relative to dqs2. 00 No change in dm2 delay 01 Add dm2 delay of 1 delay units. 10 Add dm2 delay of 2 delay units. 11 Add dm2 delay of 3 delay units
29–28 wr_dq23_del	Write dq23 delay fine tuning.This field holds the del. value that is added to dq23 relative to dqs2. 00 No change in dq23 delay 01 Add dq23 delay of 1 delay units. 10 Add dq23 delay of 2 delay units. 11 Add dq23 delay of 3 delay units
27–26 Reserved	This read-only field is reserved and always has the value zero. Reserved
25–24 wr_dq22_del	Write dq22 delay fine tuning.This field holds the del. value that is added to dq22 relative to dqs2. 00 No change in dq22 delay 01 Add dq22 delay of 1 delay units. 10 Add dq22 delay of 2 delay units. 11 Add dq22 delay of 3 delay units
23–22 Reserved	This read-only field is reserved and always has the value zero. Reserved
21–20 wr_dq21_del	Write dq21 delay fine tuning.This field holds the del. value that is added to dq21 relative to dqs2. 00 No change in dq21 delay 01 Add dq21 delay of 1 delay units. 10 Add dq21 delay of 2 delay units. 11 Add dq21 delay of 3 delay units
19–18 Reserved	This read-only field is reserved and always has the value zero. Reserved
17–16 wr_dq20_del	Write dq20 delay fine tuning.This field holds the del. value that is added to dq20 relative to dqs2. 00 No change in dq20 delay 01 Add dq20 delay of 1 delay units. 10 Add dq20 delay of 2 delay units. 11 Add dq20 delay of 3 delay units
15–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13–12 wr_dq19_del	Write dq19 delay fine tuning.This field holds the del. value that is added to dq19 relative to dqs2. 00 No change in dq19 delay 01 Add dq19 delay of 1 delay units. 10 Add dq19 delay of 2 delay units. 11 Add dq19 delay of 3 delay units
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

ESDCTL_WRDQBY2DL field descriptions (continued)

Field	Description
9–8 wr_dq18_del	Write dq18 delay fine tuning.This field holds the del. value that is added to dq18 relative to dqs2. 00 No change in dq18 delay 01 Add dq18 delay of 1 delay units. 10 Add dq18 delay of 2 delay units. 11 Add dq18 delay of 3 delay units
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5–4 wr_dq17_del	Write dq17 delay fine tuning.This field holds the del. value that is added to dq17 relative to dqs2. 00 No change in dq17 delay 01 Add dq17 delay of 1 delay units. 10 Add dq17 delay of 2 delay units. 11 Add dq17 delay of 3 delay units
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 wr_dq16_del	Write dq16 delay fine tuning.This field holds the del. value that is added to dq16 relative to dqs2. 00 No change in dq16 delay 01 Add dq16 delay of 1 delay units. 10 Add dq16 delay of 2 delay units. 11 Add dq16 delay of 3 delay units

28.121.31 PHY Write DQ Byte3 Delay Register (ESDCTL_WRDQBY3DL)

Address: ESDCTL_WRDQBY3DL is 63FD_9000h base + 78h offset = 63FD_9078h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	wr_dm3_del		wr_dq31_del		0		wr_dq30_del		0		wr_dq29_del		0		wr_dq28_del		0		wr_dq27_del		0		wr_dq26_del		0		wr_dq25_del		0		wr_dq24_del	
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_WRDQBY3DL field descriptions

Field	Description
31–30 wr_dm3_del	Write dm3 delay fine tuning.This field holds the del. value that is added to dm3 relative to dqs3. 00 No change in dm3 delay 01 Add dm3 delay of 1 delay units. 10 Add dm3 delay of 2 delay units. 11 Add dm3 delay of 3 delay units
29–28 wr_dq31_del	Write dq31 delay fine tuning.This field holds the del. value that is added to dq31 relative to dqs3.

Table continues on the next page...

ESDCTL_WRDQBY3DL field descriptions (continued)

Field	Description
	00 No change in dq31 delay 01 Add dq31 delay of 1 delay units. 10 Add dq31 delay of 2 delay units. 11 Add dq31 delay of 3 delay units
27–26 Reserved	This read-only field is reserved and always has the value zero. Reserved
25–24 wr_dq30_del	Write dq30 delay fine tuning.This field holds the del. value that is added to dq30 relative to dqs3. 00 No change in dq30 delay 01 Add dq30 delay of 1 delay units. 10 Add dq30 delay of 2 delay units. 11 Add dq30 delay of 3 delay units
23–22 Reserved	This read-only field is reserved and always has the value zero. Reserved
21–20 wr_dq29_del	Write dq29 delay fine tuning.This field holds the del. value that is added to dq29 relative to dqs3. 00 No change in dq29 delay 01 Add dq29 delay of 1 delay units. 10 Add dq29 delay of 2 delay units. 11 Add dq29 delay of 3 delay units
19–18 Reserved	This read-only field is reserved and always has the value zero. Reserved
17–16 wr_dq28_del	Write dq28 delay fine tuning.This field holds the del. value that is added to dq28 relative to dqs3. 00 No change in dq28 delay 01 Add dq28 delay of 1 delay units. 10 Add dq28 delay of 2 delay units. 11 Add dq28 delay of 3 delay units
15–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13–12 wr_dq27_del	Write dq27 delay fine tuning.This field holds the del. value that is added to dq27 relative to dqs3. 00 No change in dq27 delay 01 Add dq27 delay of 1 delay units. 10 Add dq27 delay of 2 delay units. 11 Add dq27 delay of 3 delay units
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–8 wr_dq26_del	Write dq26 delay fine tuning.This field holds the del. value that is added to dq26 relative to dqs3. 00 No change in dq26 delay 01 Add dq26 delay of 1 delay units. 10 Add dq26 delay of 2 delay units. 11 Add dq26 delay of 3 delay units
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved

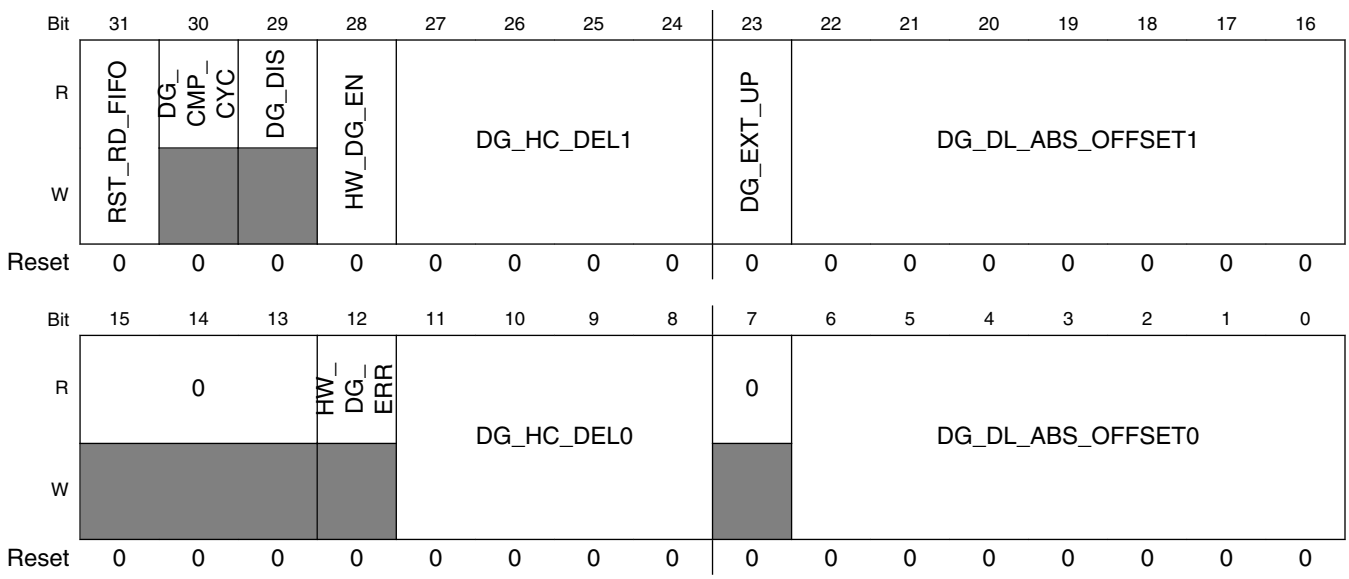
Table continues on the next page...

ESDCTL_WRDQBY3DL field descriptions (continued)

Field	Description
5–4 wr_dq25_del	Write dq25 delay fine tuning.This field holds the del. value that is added to dq25 relative to dqs3. 00 No change in dq25 delay 01 Add dq25 delay of 1 delay units. 10 Add dq25 delay of 2 delay units. 11 Add dq25 delay of 3 delay units
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 wr_dq24_del	Write dq24 delay fine tuning.This field holds the del. value that is added to dq24 relative to dqs3. 00 No change in dq24 delay 01 Add dq24 delay of 1 delay units. 10 Add dq24 delay of 2 delay units. 11 Add dq24 delay of 3 delay units

28.121.32 PHY DQS Gating Control Register0 (ESDCTL_DGCTRL0)

Address: ESDCTL_DGCTRL0 is 63FD_9000h base + 7Ch offset = 63FD_907Ch



ESDCTL_DGCTRL0 field descriptions

Field	Description
31 RST_RD_FIFO	Reset Data FIFO & pointers. This bit if asserted causes to DDR PHY reset the data and the pointers of the read data FIFO. This bit is self cleared after the FIFO reset is done.
30 DG_CMP_CYC	DG sample cycle. This bit when asserted causes the PHY to compare the data 32 cycles after the PHY sent the read command enable pulse else it compares the data after 16 cycles.

Table continues on the next page...

ESDCTL_DGCTRL0 field descriptions (continued)

Field	Description
29 DG_DIS	Dqs gating disable. This bit when asserted disable to dqs gating circuit and external dqs is driven directly into phy. This bit can be asserted only when pull down is used in the dqs pads.
28 HW_DG_EN	Enable DG HW update. This bit when asserted causes the PHY to preform DG HW update. HW negates this bit upon completion of the DG HW update. Note: Before issuing the first read command PHY counts 12 cycles.
27–24 DG_HC_DEL1	Dqs1 gating half cycles delay. This field hold the sw delay in half cycles of the dqs1 gating.This delay is added to the delay that is generated by the dqs gating delay line 1. This field can also bit written by HW. Upon completion of the dg hw calibration this field gets the value of the 4 MSB of ((HW_DG_LOW1 + HW_DG_UP1) /2). 0000 0 cycles delay. 0001 Half cycle delay. 0010 1 cycle delay 1101 6.5 cycles delay 1110 Reserved 1111 Reserved
23 DG_EXT_UP	DG extend upper boundary. By default the upper boundary of dqs gating hw calibration is set according to first fail after at least one pass. If this bit is asserted upper boundary is set according to the last pass.
22–16 DG_DL_ABS_ OFFSET1	Absolute delay offset. This field decides the specific delay line absolute delay in fraction of a cycle terms. The fraction is process and frequency independent. The delay for the delay line would be (DG_DL_ABS_OFFSET1 / 256)* fast_clk. This field can also bit written by HW. Upon completion of the dg hw calibration this field gets the value of the 7 LSB of ((HW_DG_LOW1 + HW_DG_UP1) /2). Note that not all changes will have effect on the actual delay. If the requested change is smaller than the delay line resolution (1 delay unit, which varies between 20 pSec in best case to 50 in worst case), then no change will occur.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 HW_DG_ERR	HW DG error. This bit valid is asserted when an error was found during the DQS HW calibration sequence. Error can occur when no valid value was found during hw calibration. This bit is valid only after HW_DG_EN is de-asserted. 0 No error was found during the DQS gating HW calibration sequence. 1 An error was found during the DQS gating HW calibration sequence.
11–8 DG_HC_DEL0	Dqs0 gating half cycles delay. This field hold the sw delay in half cycles of the dqs0 gating.This delay is added to the delay that is generated by the dqs gating delay line 0. This field can also bit written by HW. Upon completion of the dg hw calibration this field gets the value of the 4 MSB of ((HW_DG_LOW0 + HW_DG_UP0) /2). 0000 0 cycles delay. 0001 Half cycle delay. 0010 1 cycle delay 1101 6.5 cycles delay 1110 Reserved 1111 Reserved

Table continues on the next page...

ESDCTL_DGCTRL0 field descriptions (continued)

Field	Description
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–0 DG_DL_ABS_OFFSET0	Absolute delay offset. This field decides the specific delay line absolute delay in fraction of a cycle terms. The fraction is process and frequency independent. The delay for the delay line would be (DG_DL_ABS_OFFSET0 / 256)* fast_clk. This field can also bit written by HW. Upon completion of the dg hw calibration this field gets the value of the 7 LSB of ((HW_DG_LOW0 + HW_DG_UP0) /2). Note that not all changes will have effect on the actual delay. If the requested change is smaller than the delay line resolution (1 delay unit, which varies between 20 pSec in best case to 50 in worst case), then no change will occur.

28.121.33 PHY DQS Gating Control Register1 (ESDCTL_DGCTRL1)

Address: ESDCTL_DGCTRL1 is 63FD_9000h base + 80h offset = 63FD_9080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DG_HC_DEL3				0	DG_DL_ABS_OFFSET3							0				DG_HC_DEL2				0	DG_DL_ABS_OFFSET2						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_DGCTRL1 field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved
27–24 DG_HC_DEL3	Dqs3 gating half cycles delay. This field hold the sw delay in half cycles of the dqs3 gating.This delay is added to the delay that is generated by the dqs gating delay line 3. This field can also bit written by HW. Upon completion of the dg hw calibration this field gets the value of the 4 MSB of ((HW_DG_LOW3 + HW_DG_UP3) /2). 0000 0 cycles delay. 0001 Half cycle delay. 0010 1 cycle delay 1101 6.5 cycles delay 1110 Reserved 1111 Reserved
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–16 DG_DL_ABS_OFFSET3	Absolute delay offset. This field decides the specific delay line absolute delay in fraction of a cycle terms. The fraction is process and frequency independent. The delay for the delay line would be (DG_DL_ABS_OFFSET3 / 256)* fast_clk. This field can also bit written by HW. Upon completion of the dg hw calibration this field gets the value of the 7 LSB of ((HW_DG_LOW3 + HW_DG_UP3) /2).

Table continues on the next page...

ESDCTL_DGCTRL1 field descriptions (continued)

Field	Description
	Note that not all changes will have effect on the actual delay. If the requested change is smaller than the delay line resolution (1 delay unit, which varies between 20 pSec in best case to 50 in worst case), then no change will occur.
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11–8 DG_HC_DEL2	Dqs2 gating half cycles delay. This field hold the sw delay in half cycles of the dqs2 gating.This delay is added to the delay that is generated by the dqs gating delay line 2. This field can also bit written by HW. Upon completion of the dg hw calibration this field gets the value of the 4 MSB of ((HW_DG_LOW2 + HW_DG_UP2) /2). 0000 0 cycles delay. 0001 Half cycle delay. 0010 1 cycle delay 1101 6.5 cycles delay 1110 Reserved 1111 Reserved
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–0 DG_DL_ABS_OFFSET2	Absolute delay offset. This field decides the specific delay line absolute delay in fraction of a cycle terms. The fraction is process and frequency independent. The delay for the delay line would be (DG_DL_ABS_OFFSET2 / 256)* fast_clk. This field can also bit written by HW. Upon completion of the dg hw calibration this field gets the value of the 7 LSB of ((HW_DG_LOW2 + HW_DG_UP2) /2). Note that not all changes will have effect on the actual delay. If the requested change is smaller than the delay line resolution (1 delay unit, which varies between 20 pSec in best case to 50 in worst case), then no change will occur.

28.121.34 PHY DQS Gating Delay Line Status Register (ESDCTL_DGDLST)

This register holds the status of the 4 dqs gating delay lines.

Address: ESDCTL_DGDLST is 63FD_9000h base + 84h offset = 63FD_9084h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	DG_DL_UNIT_NUM3							0	DG_DL_UNIT_NUM2							0	DG_DL_UNIT_NUM1							0	DG_DL_UNIT_NUM0						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_DGDLST field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

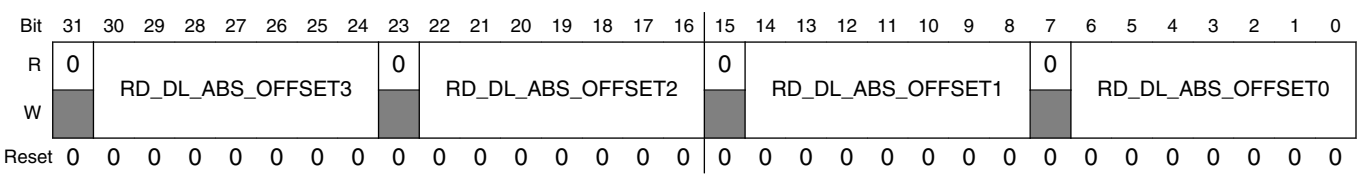
ESDCTL_DGDLST field descriptions (continued)

Field	Description
30–24 DG_DL_UNIT_NUM3	This field shows the number of delay units that is actually used by dqs gating delay line 3.
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–16 DG_DL_UNIT_NUM2	This field shows the number of delay units that is actually used by dqs gating delay line 2.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–8 DG_DL_UNIT_NUM1	This field shows the number of delay units that is actually used by dqs gating delay line 1.
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–0 DG_DL_UNIT_NUM0	This field shows the number of delay units that is actually used by dqs gating delay line 0.

28.121.35 PHY Read Delay Lines Configuration Register (ESDCTL_RDDLCTL)

This register controls read delay lines functionality, that is, DQS[x] delay used during read cycles. The delay line compensates for process variations, and produces a constant delay regardless of the process, temperature and voltage.

Address: ESDCTL_RDDLCTL is 63FD_9000h base + 88h offset = 63FD_9088h



ESDCTL_RDDLCTL field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–24 RD_DL_ABS_OFFSET3	Absolute delay offset. This field decides the specific delay line absolute delay in fraction of a cycle terms. The fraction is process and frequency independent. The delay for the delay line would be (RD_DL_ABS_OFFSET3 / 256) * fast_clk. So for the default value of 64 we get a quarter cycle delay. This field can also bit written by HW. Upon completion of the read delay line hw calibration this field gets the value of (HW_RD_DL_LOW3 + HW_RD_DL_UP3) /2

Table continues on the next page...

ESDCTL_RDDLCTL field descriptions (continued)

Field	Description
	Note that not all changes will have effect on the actual delay. If the requested change is smaller than the delay line resolution (1 delay unit, which varies between 20 pSec in best case to 50 in worst case), then no change will occur.
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–16 RD_DL_ABS_OFFSET2	Absolute delay offset. This field decides the specific delay line absolute delay in fraction of a cycle terms. The fraction is process and frequency independent. The delay for the delay line would be (RD_DL_ABS_OFFSET2 / 256) * fast_clk. So for the default value of 64 we get a quarter cycle delay. This field can also bit written by HW. Upon completion of the read delay line hw calibration this field gets the value of (HW_RD_DL_LOW2 + HW_RD_DL_UP2) /2 Note that not all changes will have effect on the actual delay. If the requested change is smaller than the delay line resolution (1 delay unit, which varies between 20 pSec in best case to 50 in worst case), then no change will occur.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–8 RD_DL_ABS_OFFSET1	Absolute delay offset. This field decides the specific delay line absolute delay in fraction of a cycle terms. The fraction is process and frequency independent. The delay for the delay line would be (RD_DL_ABS_OFFSET1 / 256) * fast_clk. So for the default value of 64 we get a quarter cycle delay. This field can also bit written by HW. Upon completion of the read delay line hw calibration this field gets the value of (HW_RD_DL_LOW1 + HW_RD_DL_UP1) /2 Note that not all changes will have effect on the actual delay. If the requested change is smaller than the delay line resolution (1 delay unit, which varies between 20 pSec in best case to 50 in worst case), then no change will occur.
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–0 RD_DL_ABS_OFFSET0	Absolute delay offset. This field decides the specific delay line absolute delay in fraction of a cycle terms. The fraction is process and frequency independent. The delay for the delay line would be (RD_DL_ABS_OFFSET0 / 256) * fast_clk. So for the default value of 64 we get a quarter cycle delay. This field can also bit written by HW. Upon completion of the read delay line hw calibration this field gets the value of (HW_RD_DL_LOW0 + HW_RD_DL_UP0) /2 Note that not all changes will have effect on the actual delay. If the requested change is smaller than the delay line resolution (1 delay unit, which varies between 20 pSec in best case to 50 in worst case), then no change will occur.

28.121.36 PHY Read Delay Lines Status Register (ESDCTL_RDDLST)

This register holds the status of the 4 read delay lines.

Address: ESDCTL_RDDLST is 63FD_9000h base + 8Ch offset = 63FD_908Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	RD_DL_UNIT_NUM3							0	RD_DL_UNIT_NUM2							0	RD_DL_UNIT_NUM1							0	RD_DL_UNIT_NUM0						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

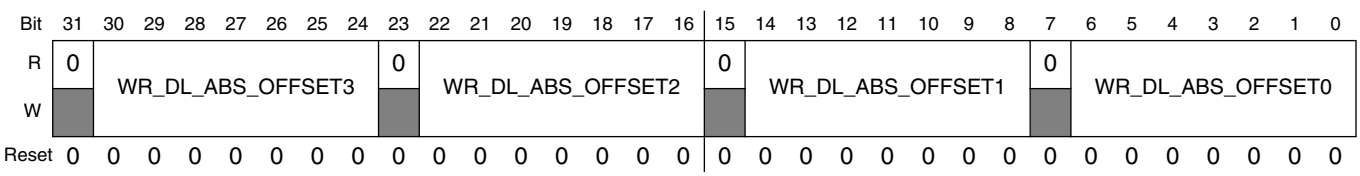
ESDCTL_RDDLST field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–24 RD_DL_UNIT_NUM3	This field shows the number of delay units that is actually used by read delay line 3.
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–16 RD_DL_UNIT_NUM2	This field shows the number of delay units that is actually used by read delay line 2.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–8 RD_DL_UNIT_NUM1	This field shows the number of delay units that is actually used by read delay line 1.
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–0 RD_DL_UNIT_NUM0	This field shows the number of delay units that is actually used by read delay line 0.

28.121.37 PHY Write Delay Lines Configuration Register (ESDCTL_WRDLCTL)

This register controls write delay lines functionality, that is, DQS[x] delay used during write cycles. The delay line compensates for process variations, and produces a constant delay regardless of the process, temperature and voltage.

Address: ESDCTL_WRDLCTL is 63FD_9000h base + 90h offset = 63FD_9090h



ESDCTL_WRDLCTL field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–24 WR_DL_ABS_OFFSET3	Absolute delay offset. This field decides the specific delay line absolute delay in fraction of a cycle terms. The fraction is process and frequency independent. The delay for the delay line would be (WR_DL_ABS_OFFSET3 / 256) * fast_clk. So for the default value of 64 we get a quarter cycle delay.

Table continues on the next page...

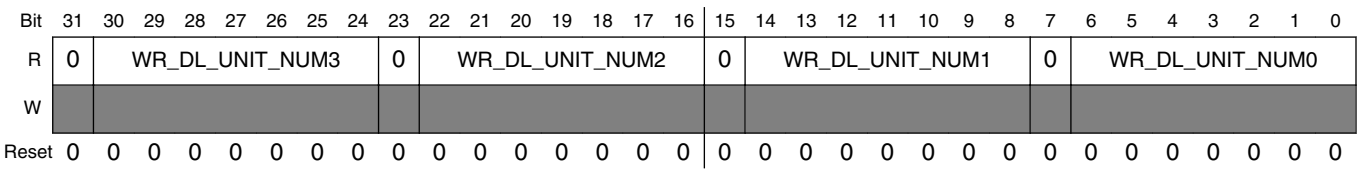
ESDCTL_WRDLCTL field descriptions (continued)

Field	Description
	This field can also bit written by HW. Upon completion of the read delay line hw calibration this field gets the value of (HW_WR_DL_LOW3 + HW_WR_DL_UP3) /2 Note that not all changes will have effect on the actual delay. If the requested change is smaller than the delay line resolution (1 delay unit, which varies between 20 pSec in best case to 50 in worst case), then no change will occur.
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–16 WR_DL_ABS_OFFSET2	Absolute delay offset. This field decides the specific delay line absolute delay in fraction of a cycle terms. The fraction is process and frequency independent. The delay for the delay line would be (WR_DL_ABS_OFFSET2 / 256) * fast_clk. So for the default value of 64 we get a quarter cycle delay. This field can also bit written by HW. Upon completion of the read delay line hw calibration this field gets the value of (HW_WR_DL_LOW2 + HW_WR_DL_UP2) /2 Note that not all changes will have effect on the actual delay. If the requested change is smaller than the delay line resolution (1 delay unit, which varies between 20 pSec in best case to 50 in worst case), then no change will occur.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–8 WR_DL_ABS_OFFSET1	Absolute delay offset. This field decides the specific delay line absolute delay in fraction of a cycle terms. The fraction is process and frequency independent. The delay for the delay line would be (WR_DL_ABS_OFFSET1 / 256) * fast_clk. So for the default value of 64 we get a quarter cycle delay. This field can also bit written by HW. Upon completion of the read delay line hw calibration this field gets the value of (HW_WR_DL_LOW1 + HW_WR_DL_UP1) /2 Note that not all changes will have effect on the actual delay. If the requested change is smaller than the delay line resolution (1 delay unit, which varies between 20 pSec in best case to 50 in worst case), then no change will occur.
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–0 WR_DL_ABS_OFFSET0	Absolute delay offset. This field decides the specific delay line absolute delay in fraction of a cycle terms. The fraction is process and frequency independent. The delay for the delay line would be (WR_DL_ABS_OFFSET0 / 256) * fast_clk. So for the default value of 64 we get a quarter cycle delay. This field can also bit written by HW. Upon completion of the read delay line hw calibration this field gets the value of (HW_WR_DL_LOW0 + HW_WR_DL_UP0) /2 Note that not all changes will have effect on the actual delay. If the requested change is smaller than the delay line resolution (1 delay unit, which varies between 20 pSec in best case to 50 in worst case), then no change will occur.

28.121.38 PHY Write Delay Lines Status Register (ESDCTL_WRDLST)

This register holds the status of the 4 write delay line.

Address: ESDCTL_WRDLST is 63FD_9000h base + 94h offset = 63FD_9094h



ESDCTL_WRDLST field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–24 WR_DL_UNIT_NUM3	This field shows the number of delay units that is actually used by write delay line 3.
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–16 WR_DL_UNIT_NUM2	This field shows the number of delay units that is actually used by write delay line 2.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–8 WR_DL_UNIT_NUM1	This field shows the number of delay units that is actually used by write delay line 1.
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–0 WR_DL_UNIT_NUM0	This field shows the number of delay units that is actually used by write delay line 0.

28.121.39 PHY SDCLK Control Register (ESDCTL_SDCTRL)

This register controls the program able delay on SDCLK. SDCLK delay can be up to 1 cycle.The delay is a sum of two delays. A half cycle delay (programmable by bit 7) and the delay line configuration.

Address: ESDCTL_SDCTRL is 63FD_9000h base + 98h offset = 63FD_9098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				SDCLK1_del		SDclk0_del		0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_SDCTRL field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11–10 SDCLK1_del	SDCLK1 delay fine tuning.This field holds the del. value that is added to sdclk1. 00 No change in sdclk1 delay 01 Add sdclk1 delay of 1 delay units. 10 Add sdclk1 delay of 2 delay units. 11 Add sdclk1 delay of 3 delay units
9–8 SDclk0_del	SDCLK0 delay fine tuning.This field holds the del. value that is added to sdclk0. 00 No change in sdclk0 delay 01 Add sdclk0 delay of 1 delay units. 10 Add sdclk0 delay of 2 delay units. 11 Add sdclk0 delay of 3 delay units
7–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

28.121.40 ZQ LPDDR2 HW Control Register (ESDCTL_ZQLP2CTL)

Address: ESDCTL_ZQLP2CTL is 63FD_9000h base + 9Ch offset = 63FD_909Ch

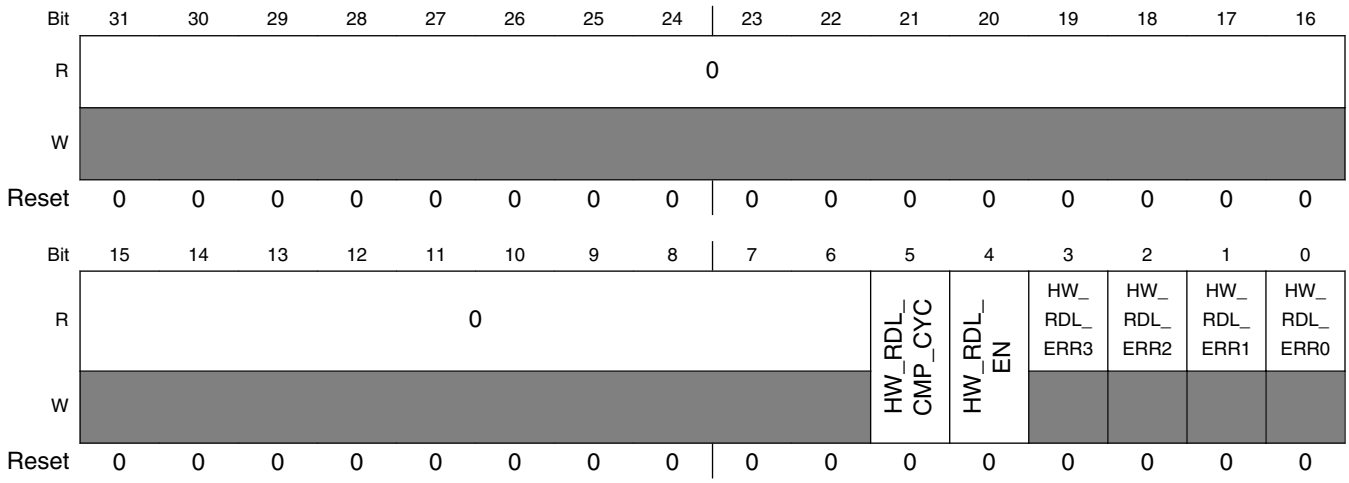
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	ZQ_LP2_HW_ZQCS							ZQ_LP2_HW_ZQCL								0								ZQ_LP2_HW_ZQINIT							
W																																
Reset	0	1	1	0	0	0	1	1	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0	1	1							

ESDCTL_ZQLP2CTL field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–24 ZQ_LP2_HW_ZQCS	This register defines the period in cycles that it takes the memory device to perform a Short ZQ calibration. This is the period of time that the controller has to wait after sending a long ZQ calibration before sending other commands. This delay will also be used if ZQ reset is sent. 0x0 0x63 Reserved 0x63 100 cycles (Default) 0x64 101 cycles 0x7E 127 cycles 0x7F 128 cycles
23–16 ZQ_LP2_HW_ZQCL	This register defines the period in cycles that it takes the memory device to perform a long ZQ calibration. This is the period of time that the controller has to wait after sending a Short ZQ calibration before sending other commands. 0x0 0x44 Reserved 0x45 70 cycles 0x46 71 cycles 0x8F 144 cycles (Default, JEDEC value, tZQCL, for LPDDR2, 360ns @ clock frequency 400 MHz) 0xFE 255 cycles 0xFF 256 cycles
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8–0 ZQ_LP2_HW_ZQINIT	This register defines the period in cycles that it takes the memory device to perform a Init ZQ calibration. This is the period of time that the controller has to wait after sending a init ZQ calibration before sending other commands. 0x0 0x44 Reserved 0x45 70 cycles 0x46 71 cycles 0x18F 400 cycles (Default, JEDEC value, tZQINIT, for LPDDR2, 1us @ clock frequency 400 MHz) 0x1FE 511 cycles 0x1FF 512 cycles

28.121.41 PHY RD DL HW Calibration Control Register (ESDCTL_RDDLHWCTL)

Address: ESDCTL_RDDLHWCTL is 63FD_9000h base + A0h offset = 63FD_90A0h



ESDCTL_RDDLHWCTL field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5 HW_RDL_CMP_CYC	HW RD DL sample cycle. This bit when asserted causes the PHY to compare the data 32 cycles after the PHY sent the read command enable pulse else it compares the data after 16 cycles.
4 HW_RDL_EN	Enable HW RD DL calibration. This bit when asserted causes the PHY to preform HW RD DL calibration. HW should negate this bit upon completion of the RD DL HW calibration.Negation of this bit also points that the RD DL HW calibration result is valid **NOTE:** Before issuing the first read command PHY counts 12 cycles.
3 HW_RDL_ERR3	HW RD DL3 error. This bit valid is asserted when an error was found during the HW calibration sequence of read delay line 3. The exact type of error can be found in ESDCTL_RDDLHWST1 register. This bit is valid only after HW_RD_DL_EN is de-asserted. 0 No error was found in read delay line 3 during the HW RD DL calibration sequence of read delay line 3. 1 An error was found in read delay line 3 during the HW RD DL calibration sequence of read delay line 3.
2 HW_RDL_ERR2	HW RD DL2 error. This bit valid is asserted when an error was found during the HW calibration sequence of read delay line 2. The exact type of error can be found in ESDCTL_RDDLHWST1 register. This bit is valid only after HW_RD_DL_EN is de-asserted. 0 No error was found in read delay line 2 during the HW RD DL calibration sequence of read delay line 2. 1 An error was found in read delay line 2 during the HW RD DL calibration sequence of read delay line 2.

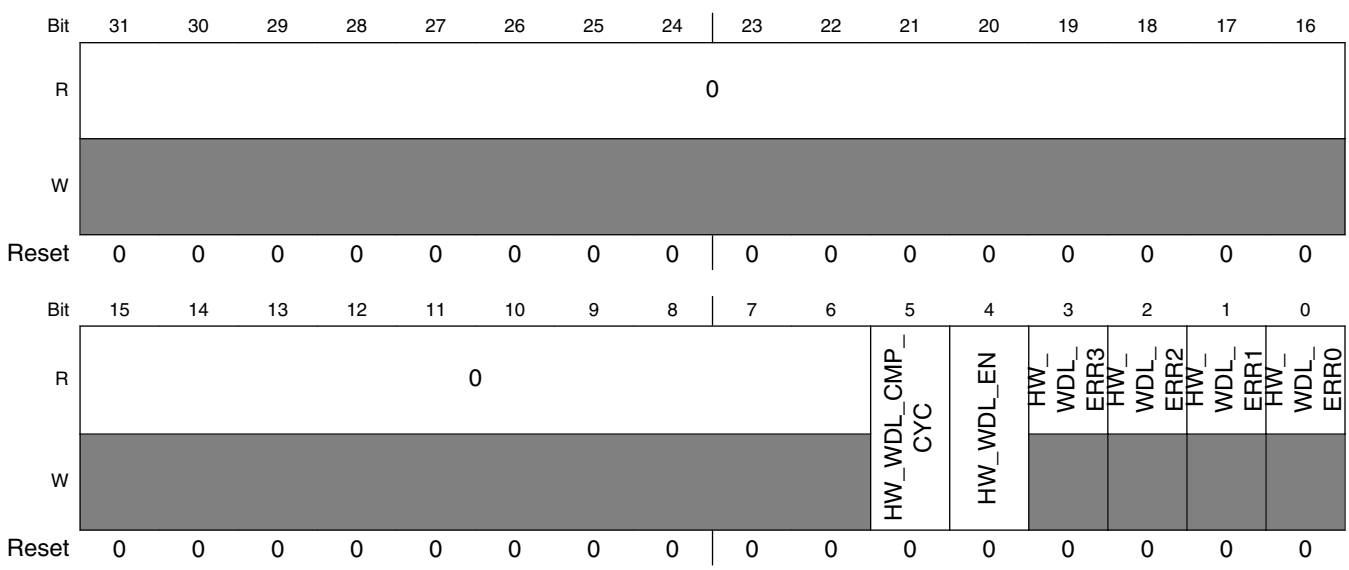
Table continues on the next page...

ESDCTL_RDDLHWCTL field descriptions (continued)

Field	Description
1 HW_RDL_ERR1	HW RD DL1 error. This bit valid is asserted when an error was found during the HW calibration sequence of read delay line 1. The exact type of error can be found in ESDCTL_RDDLHWST0 register. This bit is valid only after HW_RD_DL_EN is de-asserted. 0 No error was found in read delay line 1 during the HW RD DL calibration sequence of read delay line 1. 1 An error was found in read delay line 1 during the HW RD DL calibration sequence of read delay line 1.
0 HW_RDL_ERR0	HW RD DL0 error. This bit valid is asserted when an error was found during the HW calibration sequence of read delay line 0. The exact type of error can be found in ESDCTL_RDDLHWST0 register. This bit is valid only after HW_RD_DL_EN is de-asserted. 0 No error was found in read delay line 0 during the HW RD DL calibration sequence of read delay line 0. 1 An error was found in read delay line 0 during the HW RD DL calibration sequence of read delay line 0.

28.121.42 PHY WR DL HW Calibration Control Register (ESDCTL_WRDLHWCTL)

Address: ESDCTL_WRDLHWCTL is 63FD_9000h base + A4h offset = 63FD_90A4h



ESDCTL_WRDLHWCTL field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

ESDCTL_WRDLHWCTL field descriptions (continued)

Field	Description
5 HW_WDL_ CMP_CYC	WR DL sample cycle. This bit when asserted causes the PHY to compare the data 32 cycles after the PHY sent the read command enable pulse else it compares the data after 16 cycles.
4 HW_WDL_EN	Enable HW WR DL calibration. This bit when asserted causes the PHY to preform HW WR DL calibration. HW should negate this bit upon completion of the WR DL HW calibration.Negation of this bit also points that the WR DL HW calibration result is valid **NOTE:** Before issuing the first read command PHY counts 12 cycles.
3 HW_WDL_ERR3	HW WR DL3 error. This bit valid is asserted when an error was found during the HW calibration sequence of write delay line 3. The exact type of error can be found in HWWDLRES1 register. This bit is valid only after HW_WR_DL_EN is de-asserted. 0 No error was found during the HW calibration sequence of write delay line 3. 1 An error was found during the HW WR DL calibration sequence of write delay line 3.
2 HW_WDL_ERR2	HW WR DL2 error. This bit valid is asserted when an error was found during the HW calibration sequence of write delay line 3. The exact type of error can be found in HWWDLRES1 register. This bit is valid only after HW_WR_DL_EN is de-asserted. 0 No error was found during the HW calibration sequence of write delay line 2. 1 An error was found during the HW WR DL calibration sequence of write delay line 2.
1 HW_WDL_ERR1	HW WR DL1 error. This bit valid is asserted when an error was found during the HW calibration sequence of write delay line 1. The exact type of error can be found in HWWDLRES0 register. This bit is valid only after HW_WR_DL_EN is de-asserted. 0 No error was found during the HW calibration sequence of write delay line 1. 1 An error was found during the HW WR DL calibration sequence of write delay line 1.
0 HW_WDL_ERR0	HW WR DL0 error. This bit valid is asserted when an error was found during the HW calibration sequence of write delay line 0. The exact type of error can be found in HWWDLRES0 register. This bit is valid only after HW_WR_DL_EN is de-asserted. 0 No error was found during the HW calibration sequence of write delay line 0. 1 An error was found during the HW WR DL calibration sequence of write delay line 0.

28.121.43 PHY RD DL HW Calibration Status Register 0 (ESDCTL_RDDLHWST0)

Address: ESDCTL_RDDLHWST0 is 63FD_9000h base + A8h offset = 63FD_90A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	HW_RD_DL_UP1							0	HW_RD_DL_LOW1							0	HW_RD_DL_UP0							0	HW_RD_DL_LOW0						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_RDDLHWST0 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–24 HW_RD_DL_ UP1	HW RD DL1 upper boundary. This field holds the upper boundary of RD DL1 calibration sequence.
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–16 HW_RD_DL_ LOW1	HW RD DL1 lower boundary. This field holds the lower boundary of RD DL1 calibration sequence.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–8 HW_RD_DL_ UP0	HW RD DL0 upper boundary. This field holds the upper boundary of RD DL0 calibration sequence.
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–0 HW_RD_DL_ LOW0	HW RD DL0 lower boundary. This field holds the lower boundary of RD DL0 calibration sequence.

28.121.44 PHY RD DL HW Calibration Status Register 1 (ESDCTL_RDDLHWST1)

Address: ESDCTL_RDDLHWST1 is 63FD_9000h base + ACh offset = 63FD_90ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	HW_RD_DL_UP3							0	HW_RD_DL_LOW3							0	HW_RD_DL_UP2							0	HW_RD_DL_LOW2						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_RDDLHWST1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–24 HW_RD_DL_ UP3	HW RD DL3 upper boundary. This field holds the upper boundary of RD DL3 calibration sequence.
23 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

ESDCTL_RDDLHWST1 field descriptions (continued)

Field	Description
22–16 HW_RD_DL_ LOW3	HW RD DL3 lower boundary. This field holds the lower boundary of RD DL3 calibration sequence.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–8 HW_RD_DL_ UP2	HW RD DL2 upper boundary. This field holds the upper boundary of RD DL2 calibration sequence.
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–0 HW_RD_DL_ LOW2	HW RD DL2 lower boundary. This field holds the lower boundary of RD DL2 calibration sequence.

28.121.45 PHY WR DL HW Calibration Status Register 0 (ESDCTL_WRDLHWST0)

Address: ESDCTL_WRDLHWST0 is 63FD_9000h base + B0h offset = 63FD_90B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	HW_WR_DL_UP1							0	HW_WR_DL_LOW1							0	HW_WR_DL_UP0							0	HW_WR_DL_LOW0						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_WRDLHWST0 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–24 HW_WR_DL_ UP1	HW WR DL1 upper boundary. This field holds the upper boundary of WR DL1 calibration sequence.
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–16 HW_WR_DL_ LOW1	HW WR DL1 lower boundary. This field holds the lower boundary of WR DL1 calibration sequence.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–8 HW_WR_DL_ UP0	HW WR DL0 upper boundary. This field holds the upper boundary of WR DL0 calibration sequence.

Table continues on the next page...

ESDCTL_WRDLHWST0 field descriptions (continued)

Field	Description
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–0 HW_WR_DL_ LOW0	HW WR DL0 lower boundary. This field holds the lower boundary of WR DL0 calibration sequence.

28.121.46 PHY WR DL HW Calibration Status Register 1 (ESDCTL_WRDLHWST1)

Address: ESDCTL_WRDLHWST1 is 63FD_9000h base + B4h offset = 63FD_90B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	HW_WR_DL_UP3							0	HW_WR_DL_LOW3							0	HW_WR_DL_UP2							0	HW_WR_DL_LOW2						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_WRDLHWST1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–24 HW_WR_DL_ UP3	HW WR DL3 upper boundary. This field holds the upper boundary of WR DL3 calibration sequence.
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–16 HW_WR_DL_ LOW3	HW WR DL3 lower boundary. This field holds the lower boundary of WR DL3 calibration sequence.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–8 HW_WR_DL_ UP2	HW WR DL2 upper boundary. This field holds the upper boundary of WR DL2 calibration sequence.
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–0 HW_WR_DL_ LOW2	HW WR DL2 lower boundary. This field holds the lower boundary of WR DL2 calibration sequence.

28.121.47 PHY Write Leveling HW Error Register (ESDCTL_WLHWERR)

Address: ESDCTL_WLHWERR is 63FD_9000h base + B8h offset = 63FD_90B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HW_WL3_DQ								HW_WL2_DQ								HW_WL1_DQ								HW_WL0_DQ							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_WLHWERR field descriptions

Field	Description
31–24 HW_WL3_DQ	HW_WL3_DQ. This fields holds the DQ results for all the 8 configurable hw wl delays. bit 0 holds the wl result of 0 delay bit 1 holds the wl result of 1/8 cycle delay till bit 7 which holds the wl result of 7/8 cycle delay
23–16 HW_WL2_DQ	HW_WL2_DQ - This fields holds the DQ results for all the 8 configurable hw wl delays. bit 0 holds the wl result of 0 delay bit 1 holds the wl result of 1/8 cycle delay till bit 7 which holds the wl result of 7/8 cycle delay
15–8 HW_WL1_DQ	HW_WL1_DQ - This fields holds the DQ results for all the 8 configurable hw wl delays. bit 0 holds the wl result of 0 delay bit 1 holds the wl result of 1/8 cycle delay till bit 7 which holds the wl result of 7/8 cycle delay
7–0 HW_WL0_DQ	HW_WL0_DQ - This fields holds the DQ results for all the 8 configurable hw wl delays. bit 0 holds the wl result of 0 delay bit 1 holds the wl result of 1/8 cycle delay till bit 7 which holds the wl result of 7/8 cycle delay

28.121.48 PHY DQS Gating HW Status Register 0 (ESDCTL_DGHWST0)

Address: ESDCTL_DGHWST0 is 63FD_9000h base + BCh offset = 63FD_90BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					HW_DG_UP0											0					HW_DG_LOW0										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_DGHWST0 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–16 HW_DG_UP0	HW DG0 upper boundary. This field holds the upper boundary of DG0 calibration sequence. The actual delay in cycles is HW_DG_UP0/256.

Table continues on the next page...

ESDCTL_DGHWST0 field descriptions (continued)

Field	Description
15–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–0 HW_DG_LOW0	HW DG0 lower boundary. This field holds the lower boundary of DG0 calibration sequence. The actual delay in cycles is HW_DG_LOW0/256.

28.121.49 PHY DQS Gating HW Status Register 1 (ESDCTL_DGHWST1)

Address: ESDCTL_DGHWST1 is 63FD_9000h base + C0h offset = 63FD_90C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					HW_DG_UP1											0					HW_DG_LOW1										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_DGHWST1 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–16 HW_DG_UP1	HW DG1 upper boundary. This field holds the upper boundary of DG1 calibration sequence. The actual delay in cycles is HW_DG_UP1/256.
15–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–0 HW_DG_LOW1	HW DG1 lower boundary. This field holds the lower boundary of DG1 calibration sequence. The actual delay in cycles is HW_DG_LOW1/256.

28.121.50 PHY DQS Gating HW Status Register 2 (ESDCTL_DGHWST2)

Address: ESDCTL_DGHWST2 is 63FD_9000h base + C4h offset = 63FD_90C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					HW_DG_UP2											0					HW_DG_LOW2										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDCTL_DGHWST2 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–16 HW_DG_UP2	HW DG2 upper boundary. This field holds the upper boundary of DG2 calibration sequence. The actual delay in cycles is HW_DG_UP2/256.
15–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–0 HW_DG_LOW2	HW DG2 lower boundary. This field holds the lower boundary of DG2 calibration sequence. The actual delay in cycles is HW_DG_LOW2/256.

28.121.51 PHY DQS Gating HW Status Register 3 (ESDCTL_DGHWST3)

Address: ESDCTL_DGHWST3 is 63FD_9000h base + C8h offset = 63FD_90C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					HW_DG_UP3											0					HW_DG_LOW3										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

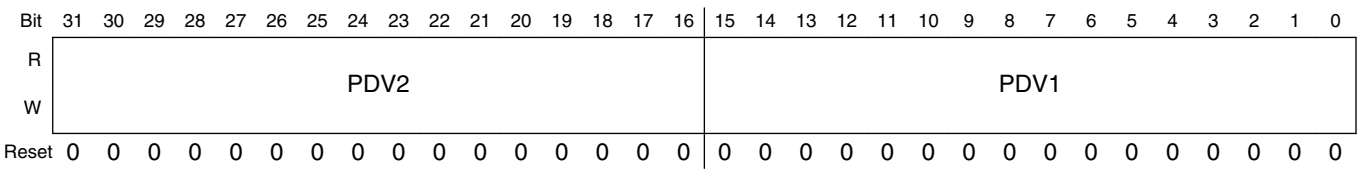
ESDCTL_DGHWST3 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–16 HW_DG_UP3	HW DG3 upper boundary. This field holds the upper boundary of DG3 calibration sequence. The actual delay in cycles is HW_DG_UP3/256.
15–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–0 HW_DG_LOW3	HW DG3 lower boundary. This field holds the lower boundary of DG3 calibration sequence. The actual delay in cycles is HW_DG_LOW3/256.

28.121.52 PHY Pre-defined Compare Register 1 (ESDCTL_PDCMPR1)

This set of two registers (ESDCTL_PDCMPR1 & ESDCTL_PDCMPR2) controls the pre-defined compare value.The compare value can be the MPR value (as defined in the ddr3 jedec) or can be programmed by the PDV1 & PDV2 fields.

Address: ESDCTL_PDCMPR1 is 63FD_9000h base + CCh offset = 63FD_90CCh

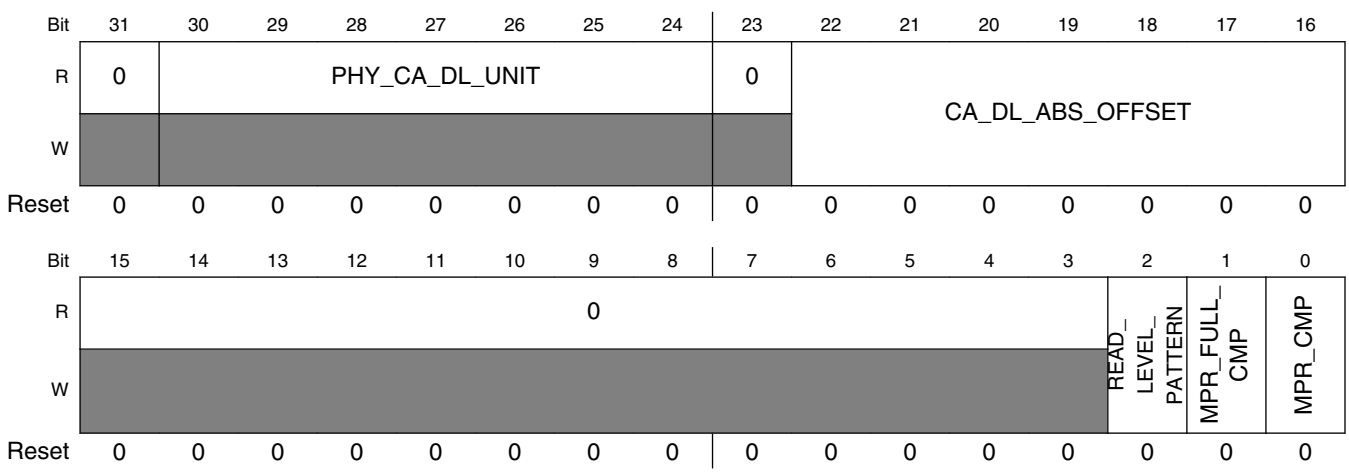


ESDCTL_PDCMPR1 field descriptions

Field	Description
31–16 PDV2	Pre defined compare value2. This field holds data to be compared to the data coming from the ddr device (every second access) during hw write delay line calibration (if MPR_CMP is disabled). Note: The inverted value of this field is used as the reset value for the read fifo when this value is used for comparison
15–0 PDV1	Pre defined compare value1. This field holds that data to be compared to the data coming from the ddr device in all cases except from the one which uses PDV2 (if MPR_CMP is disabled). Note: The inverted value of this field is used as the reset value for the read fifo when this value is used for comparison

28.121.53 PHY Pre-defined Compare Register 2 (ESDCTL_PDCMPR2)

Address: ESDCTL_PDCMPR2 is 63FD_9000h base + D0h offset = 63FD_90D0h

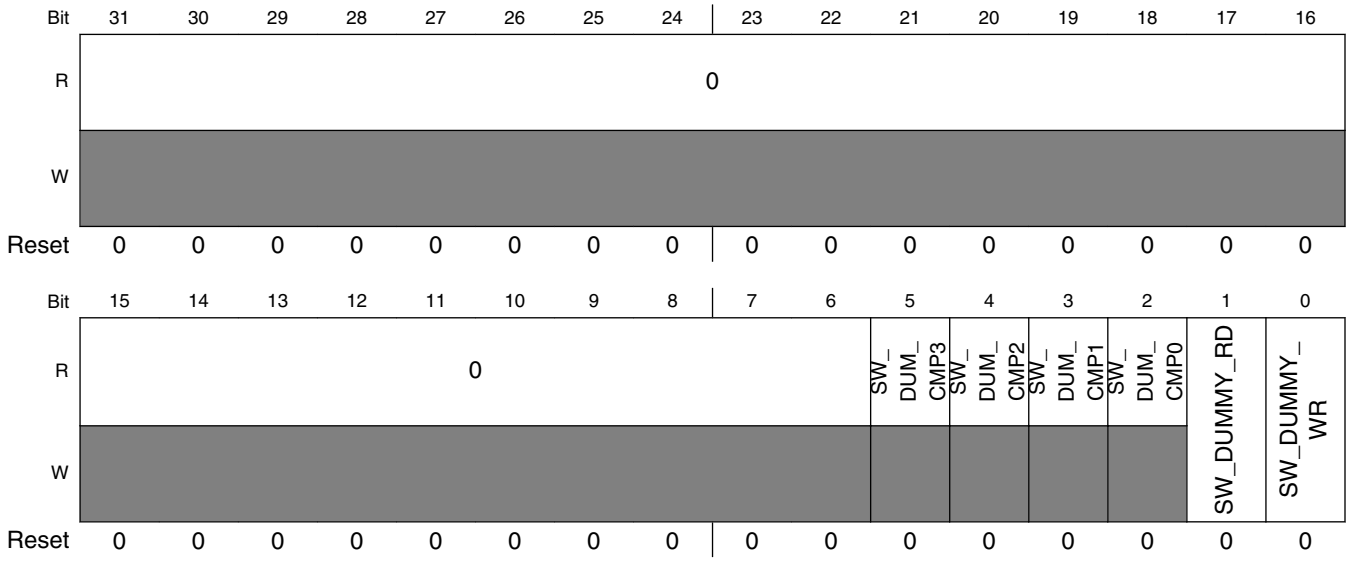


ESDCTL_PDCMPR2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–24 PHY_CA_DL_UNIT	This field shows the number of delay units that is actually used by phy CA delay unit
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–16 CA_DL_ABS_OFFSET	Absolute delay offset. This field decides the specific delay line absolute delay in fraction of a cycle terms. The fraction is process and frequency independent.The delay for the delay line would be (CA_DL_ABS_OFFSET/ 256) * fast_clk.
15–3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2 READ_LEVEL_PATTERN	in LPDDR2 MPR command is done through MRR command. And there are 2 patterns. Phy will send controller the configured pattern by signal: "read_level_pattern" 0 pattern 1010 1 pattern 0011
1 MPR_FULL_CMP	MPR full compare enable.This bit when asserted (and when MPR_CMP is asserted) causes all hw calibration & sw dummy processes to compare all bit of the data read from the ddr device to the MPR pre-defined value. When this bit is de-asserted only LSB of each byte is compared.
0 MPR_CMP	MPR compare enable. This bit when asserted causes all hw calibration & sw dummy processes to compare the data read from the ddr device to the MPR pre-defined value. When this bit is disabled data is compared to the data of the pre defined compare value field Note: In case of LPDDR2 this bit is used as MRR enable and when asserted causes all hw calibration & sw dummy processes to compare the data read from the ddr device to the MRR pre-defined value, the Pattern type is selected according to READ_LEVEL_PATTERN bit.

28.121.54 PHY SW Dummy Access Register (ESDCTL_SWDAR)

Address: ESDCTL_SWDAR is 63FD_9000h base + D4h offset = 63FD_90D4h



ESDCTL_SWDAR field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5 SW_DUM_CMP3	SW dummy read byte3 compare results. This bit reflects the outcome of the dummy read.This bit is valid only after SW_DUMMY_RD is de-asserted. 0 Dummy read fail 1 Dummy read pass
4 SW_DUM_CMP2	SW dummy read byte2 compare results. This bit reflects the outcome of the dummy read.This bit is valid only after SW_DUMMY_RD is de-asserted. 0 Dummy read fail 1 Dummy read pass
3 SW_DUM_CMP1	SW dummy read byte1 compare results. This bit reflects the outcome of the dummy read.This bit is valid only after SW_DUMMY_RD is de-asserted. 0 Dummy read fail 1 Dummy read pass
2 SW_DUM_CMP0	SW dummy read byte0 compare results. This bit reflects the outcome of the dummy read.This bit is valid only after SW_DUMMY_RD is de-asserted. 0 Dummy read fail 1 Dummy read pass
1 SW_DUMMY_RD	SW dummy read. This bit when asserted generate a dummy read sequence. In this succulence there is no axi read access instead data is read from memory and compared to the PDCMPR or MPR default value.Upon completion of the access this bit is de-asserted and the data & compare results are valid.

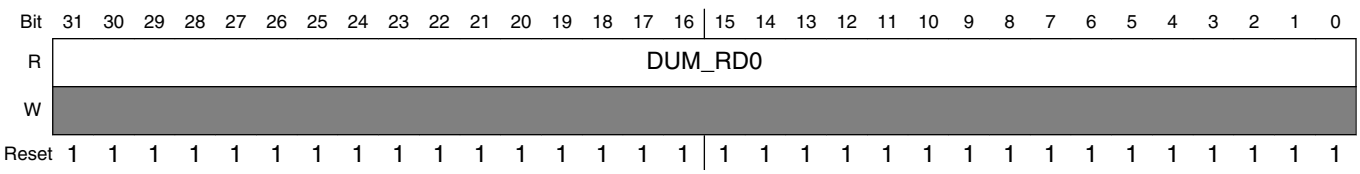
Table continues on the next page...

ESDCTL_SWDAR field descriptions (continued)

Field	Description
0 SW_DUMMY_WR	SW dummy write. This bit when asserted generate a dummy write sequence. In this seqeunce there is no axi write access instead data is written from PDCMPR. The bit is de-asserted upon completion of the access.

28.121.55 PHY SW Dummy Read Data Register n (ESDCTL_SWDRDR*n*)

Addresses: ESDCTL_SWDRDR0 is 63FD_9000h base + D8h offset = 63FD_90D8h
ESDCTL_SWDRDR1 is 63FD_9000h base + DCh offset = 63FD_90DCh
ESDCTL_SWDRDR2 is 63FD_9000h base + E0h offset = 63FD_90E0h
ESDCTL_SWDRDR3 is 63FD_9000h base + E4h offset = 63FD_90E4h
ESDCTL_SWDRDR4 is 63FD_9000h base + E8h offset = 63FD_90E8h
ESDCTL_SWDRDR5 is 63FD_9000h base + ECh offset = 63FD_90ECh
ESDCTL_SWDRDR6 is 63FD_9000h base + F0h offset = 63FD_90F0h
ESDCTL_SWDRDR7 is 63FD_9000h base + F4h offset = 63FD_90F4h

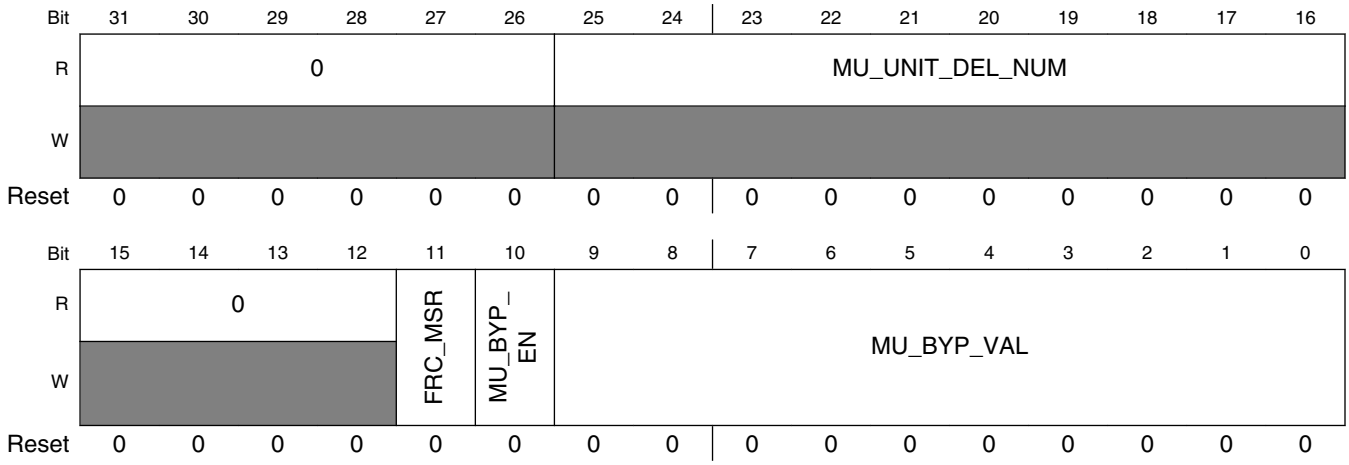


ESDCTL_SWDRDR*n* field descriptions

Field	Description
31–0 DUM_RD0	Dummy read data n. This field holds the nth data read from memory during sw dummy read access.

28.121.56 PHY Measure Unit Register (ESDCTL_MUR)

Address: ESDCTL_MUR is 63FD_9000h base + F8h offset = 63FD_90F8h

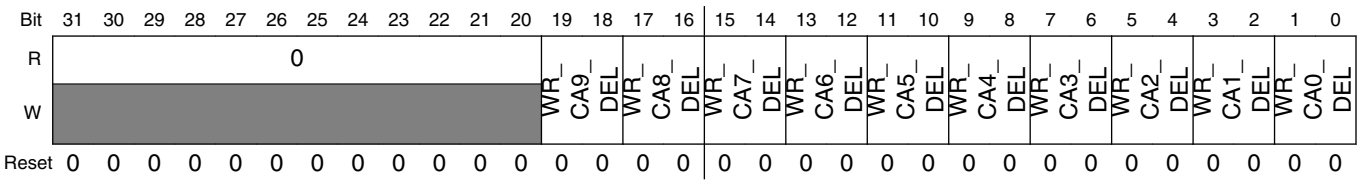


ESDCTL_MUR field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value zero. Reserved
25–16 MU_UNIT_DEL_NUM	Measure unit measured number of unit delay per cycle. This field holds the number of unit delays per cycle measure by the measure unit.
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11 FRC_MSR	Force measurement on delay line.Writing '1' to this bit asserts the measure signal to the delay line and measure unit. Upon completion of the measurement the measure unit & delay line return to functional more.This bit is self cleared. Note: • User should make sure that there is no active accesses to/from DDR before asserting this bit. • This bit is used only for DDR2 / DDR3 (i.e DDR_TYPE = 0x0 or 0x2). In case of LPDDR2 (i.e DDR_TYPE= 0x1) this bit should not be asserted 0 Not forcing delay line to make measurement. 1 Delay line is forced to measure.This bit is cleared upon completion of the measurement process.
10 MU_BYP_EN	Measure unit bypass enable. Selects the num of unit delay per cycle between measure unit measured value and programmable value (MU_BYPASS_VAL). 0 The measured number of delay units is used by the delay lines. 1 MU_BYPASS_VAL value is used by the delay lines.
9–0 MU_BYP_VAL	This field is the measure unit bypass value. The bypass value is used only if MU_BYPASS_EN is set. The bypass value is holds the number of unit delay per cycle.

28.121.57 Write CA Delay Line controller (ESDCTL_WRCADL)

Address: ESDCTL_WRCADL is 63FD_9000h base + FCh offset = 63FD_90FCh



ESDCTL_WRCADL field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value zero. Reserved
19–18 WR_CA9_DEL	ca9 delay fine tuning.This field holds the del. value that is added to ca9 relative to ca selector delay line. 00 No change in ca9 delay 01 Add ca9 delay of 1 delay units. 10 Add ca9 delay of 2 delay units. 11 Add ca9 delay of 3 delay units
17–16 WR_CA8_DEL	ca8 delay fine tuning.This field holds the del. value that is added to ca8 relative to ca selector delay line. 00 No change in ca8 delay 01 Add ca8 delay of 1 delay units. 10 Add ca8 delay of 2 delay units. 11 Add ca8 delay of 3 delay units
15–14 WR_CA7_DEL	ca7 delay fine tuning.This field holds the del. value that is added to ca7 relative to ca selector delay line. 00 No change in ca7 delay 01 Add ca7 delay of 1 delay units. 10 Add ca7 delay of 2 delay units. 11 Add ca7 delay of 3 delay units
13–12 WR_CA6_DEL	ca6 delay fine tuning.This field holds the del. value that is added to ca6 relative to ca selector delay line. 00 No change in ca6 delay 01 Add ca6 delay of 1 delay units. 10 Add ca6 delay of 2 delay units. 11 Add ca6 delay of 3 delay units
11–10 WR_CA5_DEL	ca0 delay fine tuning.This field holds the del. value that is added to ca5 relative to ca selector delay line. 00 No change in ca5 delay 01 Add ca5 delay of 1 delay units. 10 Add ca5 delay of 2 delay units. 11 Add ca5 delay of 3 delay units
9–8 WR_CA4_DEL	ca4 delay fine tuning.This field holds the del. value that is added to ca4 relative to ca selector delay line. 00 No change in ca4 delay

Table continues on the next page...

ESDCTL_WRCADL field descriptions (continued)

Field	Description
	01 Add ca4 delay of 1 delay units. 10 Add ca4 delay of 2 delay units. 11 Add ca4 delay of 3 delay units
7–6 WR_CA3_DEL	ca3 delay fine tuning.This field holds the del. value that is added to ca3 relative to ca selector delay line. 00 No change in ca3 delay 01 Add ca3 delay of 1 delay units. 10 Add ca3 delay of 2 delay units. 11 Add ca3 delay of 3 delay units
5–4 WR_CA2_DEL	ca2 delay fine tuning.This field holds the del. value that is added to ca2 relative to ca selector delay line. 00 No change in ca2 delay 01 Add ca2 delay of 1 delay units. 10 Add ca2 delay of 2 delay units. 11 Add ca2 delay of 3 delay units
3–2 WR_CA1_DEL	ca1 delay fine tuning.This field holds the del. value that is added to ca1 relative to ca selector delay line. 00 No change in ca1 delay 01 Add ca1 delay of 1 delay units. 10 Add ca1 delay of 2 delay units. 11 Add ca1 delay of 3 delay units
1–0 WR_CA0_DEL	ca0 delay fine tuning.This field holds the del. value that is added to ca0 relative to ca selector delay line. 00 No change in ca0 delay 01 Add ca0 delay of 1 delay units. 10 Add ca0 delay of 2 delay units. 11 Add ca0 delay of 3 delay units

Chapter 29
Enhanced Secured Digital Host Controller (eSDHCv2)

29.1 Overview

The Enhanced Secured Digital Host Controller Version 2 (ESDHCv2, referred to as ESDHC hereafter) provides the interface between the host system and the SD/SDIO/MMC cards, as depicted in Figure 29-1. The ESDHC acts as a bridge, passing host bus transactions to the SD/SDIO/MMC cards by sending commands and performing data accesses to/from the cards. It handles the SD/SDIO/MMC protocols at the transmission level.

Two versions of the ESDHC block are utilized in the i.MX53 SoC: Version 2 and Version 3 (ESDHCV2 and ESDHCV3). There are three instances of the block ESDHCV2, and one instance of the block ESDHCV3. The text will use the designation ESDHC when referring to features that are common to both.

The types of cards supported by the ESDHC are described briefly as follows:

The Multi Media Card (MMC) is a universal low cost data storage and communication media that is designed to cover a wide area of applications including mobile video and gaming. Old MMC cards are based on a 7-pin serial bus with a single data pin, while the new high speed MMC communication is based on an advanced 11-pin serial bus designed to operate in the low voltage range.

The Secure Digital Card (SD) is an evolution of the old MMC technology. It is specifically designed to meet the security, capacity, performance, and environment requirements inherent in newly emerging audio and video consumer electronic devices. The physical form factor, pin assignment and data transfer protocol are forward compatible with the old MMC (with some additions).

Under the SD protocol, it can be categorized into Memory card, I/O card and Combo card, which has both memory and I/O functions. The memory card invokes a copyright protection mechanism that complies with the security of the SDMI standard. The I/O

card, which is also known as SDIO card, provides high-speed data I/O with low power consumption for mobile electronic devices. For the sake of simplicity, Figure 29-1 does not show cards with reduced size or mini cards.

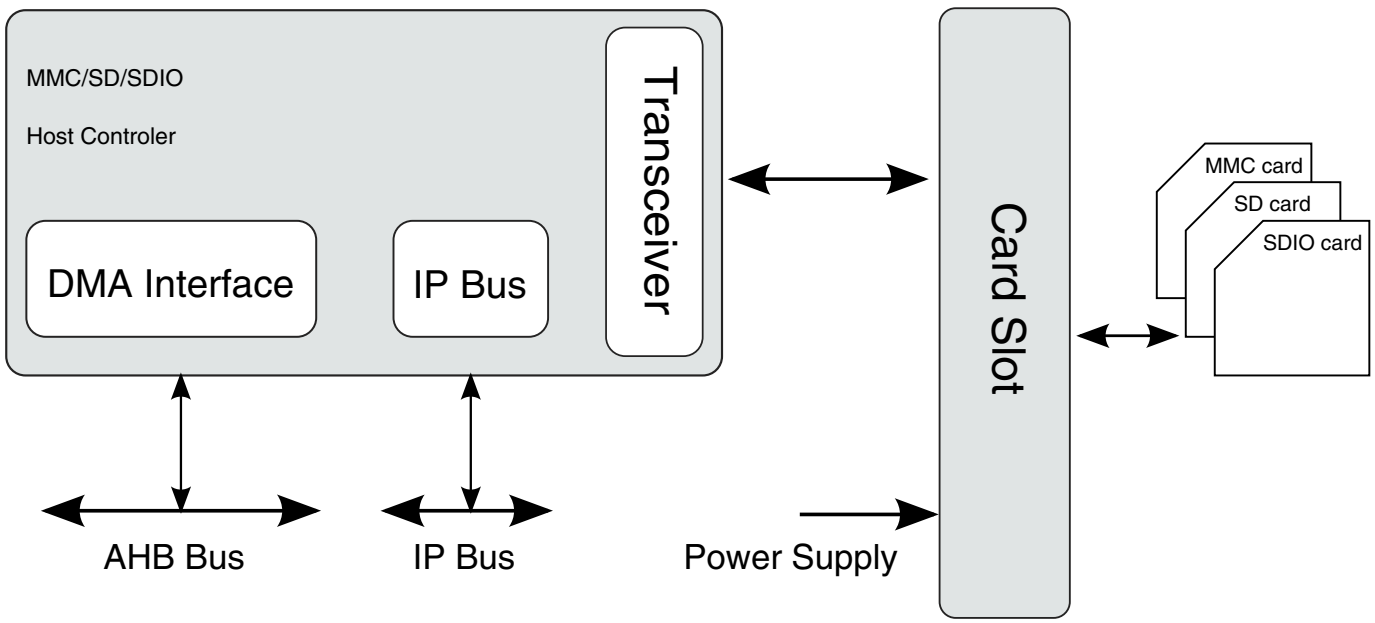


Figure 29-1. System Connection of the ESDHC

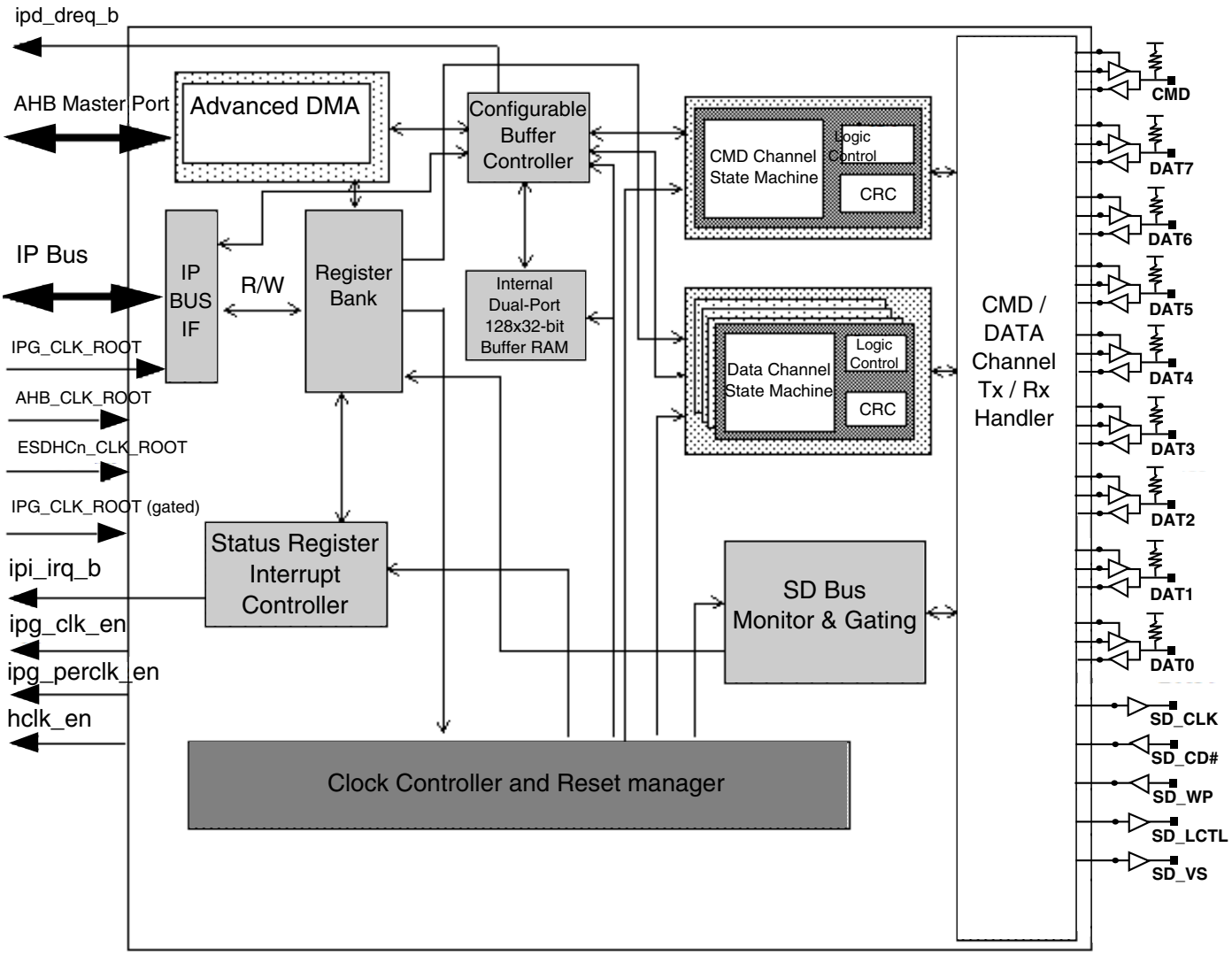


Figure 29-2. enhanced Secure Digital Host Controller Block Diagram

29.1.1 Features

The features of the ESDHC include the following:

- Conforms to the SD Host Controller Standard Specification version 2.0 including Test Event register support
- Compatible with the MMC System Specification version 4.2/4.3
- Compatible with the SD Memory Card Specification version 2.0 and supports the High Capacity SD Memory Card
- Compatible with the SDIO Card Specification version 2.0
- Designed to work with SD Memory, miniSD Memory, SDIO, miniSDIO, SD Combo, MMC, MMC plus, and MMC RS cards

- Card bus clock frequency up to 52 MHz
- Supports 1-bit / 4-bit SD and SDIO modes, 1-bit / 4-bit / 8-bit MMC modes devices
  - Up to 200 Mbps of data transfer for SD/SDIO cards using 4 parallel data lines
  - Up to 416 Mbps of data transfer for MMC cards using 8 parallel data lines in SDR (Single Data Rate) mode
- Supports Single Block, Multi Block read and write
- Supports block sizes of 1 ~ 4096 bytes
- Supports the write protection switch for write operations
- Supports both synchronous and asynchronous abort
- Supports pause during the data transfer at block gap
- Supports SDIO Read Wait and Suspend Resume operations
- Supports Auto CMD12 for multi-block transfer
- Host can initiate non-data transfer command while data transfer is in progress
- Allows cards to interrupt the host in 1-bit and 4-bit SDIO modes, also supports interrupt period
- Embodies a fully configurable 128x32-bit FIFO for read/write data
- Supports internal and external DMA capabilities
- Supports Advanced DMA to perform linked memory access

29.1.2 Modes and Operations

29.1.2.1 Data transfer Modes

The ESDHC can select the following modes for data transfer:

- SD 1-bit
- SD 4-bit
- MMC 1-bit
- MMC 4-bit
- MMC 8-bit
- Identification Mode (up to 400 kHz)
- MMC full speed mode (up to 20 MHz)
- MMC high speed mode (up to 52 MHz)
- SD/SDIO full speed mode (up to 25 MHz)
- SD/SDIO high speed mode (up to 50 MHz)

29.2 External Signals

29.2.1 Signals Overview

The ESDHC has 15 associate I/O signals.

- The SD_CLK is an internally generated clock used to drive the MMC, SD, SDIO cards.
- The CMD I/O is used to send commands and receive responses to/from the card. Eight data lines (DAT7~DAT0) are used to perform data transfers between the ESDHC and the card.
- The SD_CD# and SD_WP are card detection and write protection signals directly routed from the socket. A low on SD_CD# means that a card is inserted and a high on SD_WP means that the write protect switch is active.
- SD_OD is an output signal generated in SoC level outside ESDHC and is used to select the external open drain resistor.
- SD_LCTL is an output signal used to drive an external LED to indicate that the SD interface is busy.

SD_CD#, SD_WP, SD_OD, SD_LCTL are all optional for system implementation. If the ESDHC is desired to support a 4-bit data transfer, DAT7~DAT4 can also be optional and tied to high.

29.2.2 Ports Table

See Table 29-1 for the signal properties of the I/Os.

Table 29-1. Properties of I/O Signals

Name	Port	Function	Reset State	Pull up
SD_CLK	O	Clock for MMC/SD/SDIO card	0	N/A
SD_CMD	I/O	CMD line connect to card	1	Pull up
SD_DAT7	I/O	DAT7 line in 8-bit mode Not used in other modes	1	Pull up
SD_DAT6	I/O	DAT6 line in 8-bit mode Not used in other modes	1	Pull up
SD_DAT5	I/O	DAT5 line in 8-bit mode Not used in other modes	1	Pull up
SD_DAT4	I/O	DAT4 line in 8-bit mode Not used in other modes	1	Pull up

Table continues on the next page...

Table 29-1. Properties of I/O Signals (continued)

Name	Port	Function	Reset State	Pull up
SD_DAT3	I/O	DAT3 line in 4/8-bit mode or configured as card detection pin May be configured as card detection pin in 1-bit mode	0	Should be pull-up/pull-down configurable as this port may be used for card detection
SD_DAT2	I/O	DAT2 line or Read Wait in 4-bit mode Read Wait in 1-bit mode	1	Pull up
SD_DAT1	I/O	DAT1 line in 4/8-bit mode Also used to detect interrupt in 1/4-bit mode	1	Pull up
SD_DAT0	I/O	DAT0 line in all modes Also used to detect busy state	1	Pull up
SD_CD#	I	Card detection pin If not used tie high	N/A	N/A
SD_WP	I	Card write protect detect If not used tie low	N/A	N/A
SD_OD	O	Open drain select (not generated within the ESDHC). Optional output	N/A	N/A
SD_LCTL	O	LED control used to drive an external LED Active high Fully controlled by the driver Optional output	0	N/A

29.3 Functional Description

The following sections provide a brief functional description of the major system blocks, including the Data Buffer, DMA AHB interface, register bank as well as IP Bus interface, dual-port memory wrapper, data/command controller, clock & reset manager and clock generator.

29.3.1 Data Buffer

The ESDHC uses one configurable data buffer, so that data can be transferred between the system bus (IP Bus or AHB Bus) and the SD card, with an optimized manner to maximize throughput between the two clock domains (that is, the IP peripheral clock, and the master clock). See the table below for illustration of the buffer scheme. The buffer is used as temporary storage for data being transferred between the host system and the

card. The watermark levels for read and write are both configurable, and can be any number from 1 to 128 words. The burst lengths for read and write are also configurable, and can be any number from 1 to 31 words.

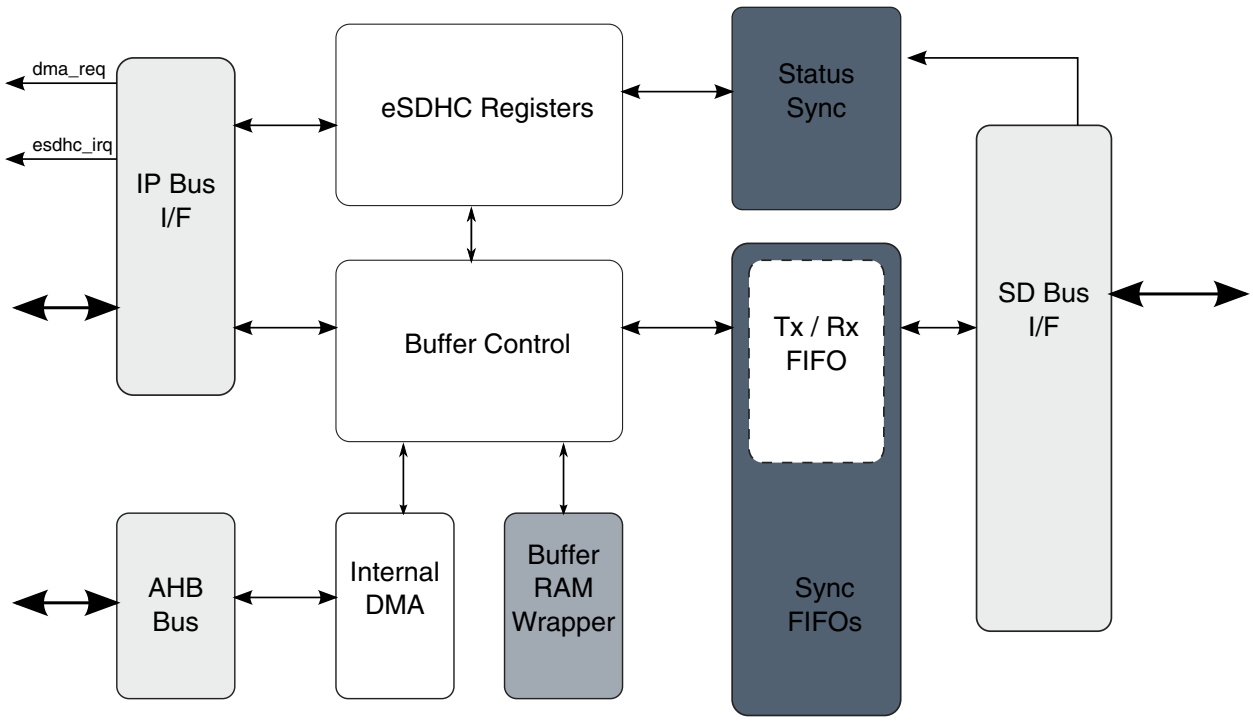


Figure 29-3. ESDHC Buffer Scheme

There are 3 transfer modes to access the data buffer:

- ARM platform polling mode:
  - For a host read operation, when the number of words received in the buffer meets or exceeds the RD_WML watermark value, then by polling the BRR bit the Host Driver can read the Buffer Data Port register to fetch the amount of words set in the RD_WML register from the buffer. The write operation is similar.
- External DMA mode:
  - For a read operation, when there are more words received in the buffer than the amount set in the RD_WML register, a DMA request is sent out to inform the external DMA to fetch the data. The request will be immediately de-asserted when there is an access on the Buffer Data Port register. If the number of words in the buffer after the current burst meets or exceeds RD_WML value, then the DMA request is asserted again. For instance, if there are twice as many words in the buffer than the RD_WML value, there are two successive DMA requests with only one cycle of de-assertion between. The write operation is similar.

NOTE

Data buffer accesses by the ARM platform polling mode and external DMA mode both use the IP Bus. In these mode, if the external DMA is enabled, an external DMA request is sent out every time the buffer is ready.

- Internal DMA mode (includes simple and advanced DMA accesses):
  - The internal DMA access, either by simple or advanced DMA, is over the AHB bus. For internal DMA access mode, the external DMA request will never be sent out.

    For a read operation, when there are more words in the buffer than the amount set in the RD_WML register, the internal DMA starts fetching data over the AHB bus. Except INCR4 and INCR8, the burst type is always INCR mode and the burst length depends on the shortest of following factors:

1. Burst length configured in the burst length field of the Watermark Level register
2. Watermark Level boundary
3. Block size boundary
4. Data boundary configured in the current descriptor (if the ADMA is active)
5. 1 Kbyte address boundary defined in the AHB protocol

Write operation is similar.

Sequential and contiguous access is necessary to ensure the pointer address value is correct. Random or skipped access is not possible. The byte order, by reset, is little endian mode. The actually byte order is swapped inside the buffer, according to the endian mode configured by software. See Figure 29-4 and Figure 29-5. For a host write operation, byte order is swapped after data is fetched from the buffer and ready to send to the SD Bus. For a host read operation, byte order is swapped before the data is stored into the buffer.

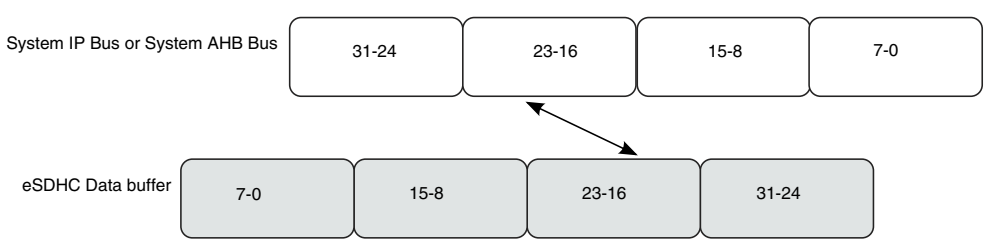


Figure 29-4. Data Swap between System Bus and ESDHC Data Buffer in Byte Little Endian Mode

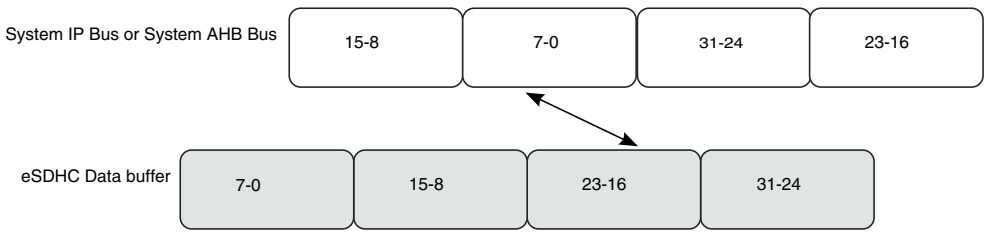


Figure 29-5. Data Swap between System Bus and ESDHC Data Buffer in Half Word Big Endian Mode

29.3.1.1 Write Operation Sequence

There are three ways to write data into the buffer when the user transfers data to the card:

1. By using external DMA through the ESDHC DMA request signal.
2. By processor core polling through the BWR bit in Interrupt Status register (interrupt or polling).
3. By using the internal DMA.

When the internal DMA is not used, (i.e. the DMAEN bit in the Transfer Type register is not set when the command is sent), the ESDHC asserts a DMA request when the amount of buffer space exceeds the value set in the WR_WML register, and is ready for receiving new data. At the same time, the ESDHC would set the BWR bit. The buffer write ready interrupt will be generated if it is enabled by software.

When internal DMA is used, the ESDHC will not inform the system before all the required number of bytes are transferred (if no error was encountered). When an error occurs during the data transfer, the ESDHC will abort the data transfer and abandon the current block. The Host Driver should read the contents of the DMA System Address register to get the starting address of the abandoned data block. If the current data transfer is in multi block mode, the ESDHC will not automatically send CMD12, even though the AC12EN bit in the Transfer Type register is set. The Host Driver shall send CMD12 in this scenario and re-start the write operation from that address. It is recommended that a Software Reset for Data be applied before the transfer is re-started after error recovery.

The ESDHC will not start data transmission until the number of words set in the WR_WML register can be held in the buffer. If the buffer is empty and the Host System does not write data in time, the ESDHC will stop the SD_CLK to avoid the data buffer under-run situation.

29.3.1.2 Read Operation Sequence

There are three ways to read data from the buffer when the user transfers data from the card:

1. By using the external DMA through the ESDHC DMA request signal
2. By processor core polling through the BRR bit in Interrupt Status register (interrupt or polling)
3. By using the internal DMA

When internal DMA is not used (i.e. DMAEN bit in Transfer Type register is not set when the command is sent), the ESDHC asserts a DMA request when the amount of data exceeds the value set in the RD_WML register (that is, available and ready for system fetching data). At the same time, the ESDHC would set the BRR bit. The buffer read ready interrupt will be generated if it is enabled by software.

When internal DMA is used, the ESDHC will not inform the system before all the required number of bytes are transferred (if no error was encountered). When an error occurs during the data transfer, the ESDHC will abort the data transfer and abandon the current block. The Host Driver should read the content of the DMA System Address register to get the starting address of the abandoned data block. If the current data transfer is in multi block mode, the ESDHC will not automatically send CMD12, even though the AC12EN bit in the Transfer Type register is set. The Host Driver shall send CMD12 in this scenario and re-start the read operation from that address. It is recommended that a Software Reset for Data be applied before the transfer is re-started after error recovery.

For any read transfer mode, the ESDHC will not start data transmission until the number of words set in the RD_WML register are in the buffer. If the buffer is full and the Host System does not read data in time, the ESDHC will stop the SD_CLK to avoid the data buffer over-run situation.

29.3.1.3 Data Buffer and Block Size

The user needs to know the buffer size, for the buffer operation during a data transfer, to utilize it in the most optimized way. In the ESDHC, the only data buffer can hold up to 128 words (32-bit), and the watermark levels for write and read can be configured respectively. For both read and write, the watermark level can be from 1 word to the maximum of 128 words. For both read and write, the burst length, can be from 1 word to the maximum of 31 words. The Host Driver may configure the value according to the system situation and requirement.

During a multi-block data transfer, the block length may be set to any value between 1 and 4096 bytes inclusive which satisfies the requirements of the external card. The only restriction is from the external card. It might not support that large of a block or it does not support a partial block access (which is not the integer times of 512 bytes).

For block size not times of 4, that is, not word aligned, ESDHC requires stuff bytes at the end of each block, because ESDHC treats each block individually. For example, if the block size is 7 bytes and there are 12 blocks to write, the system side must write two times for each block. For each block, the ending byte will be abandoned by ESDHC because it only sends 7 bytes to the card and picks data from the following system write, making a total of 24 beats of write access.

29.3.1.4 Dividing Large Data Transfer

This SDIO command CMD53 definition, limits the maximum data size of data transfers according to the following formula:

Max data size = Block size x Block count

The length of a multiple block transfer needs to be in block size units. If the total data length can't be divided evenly into a multiple of the block size, then there are two ways to transfer the data which depend on the function and the card design. Option 1 is for the Host Driver to split the transaction. The remainder of the block size data is then transferred by using a single block command at the end. Option 2 is to add dummy data in the last block to fill the block size. For option 2, the card must manage the removal of the dummy data.

See figure below for an example showing the dividing of large data transfers. Assuming a kind of WLAN SDIO card only supports block size up to 64 bytes. Although the ESDHC supports a block size of up to 4096 bytes, the SDIO can only accept a block size less than64 bytes, so the data must be divided (see example below).

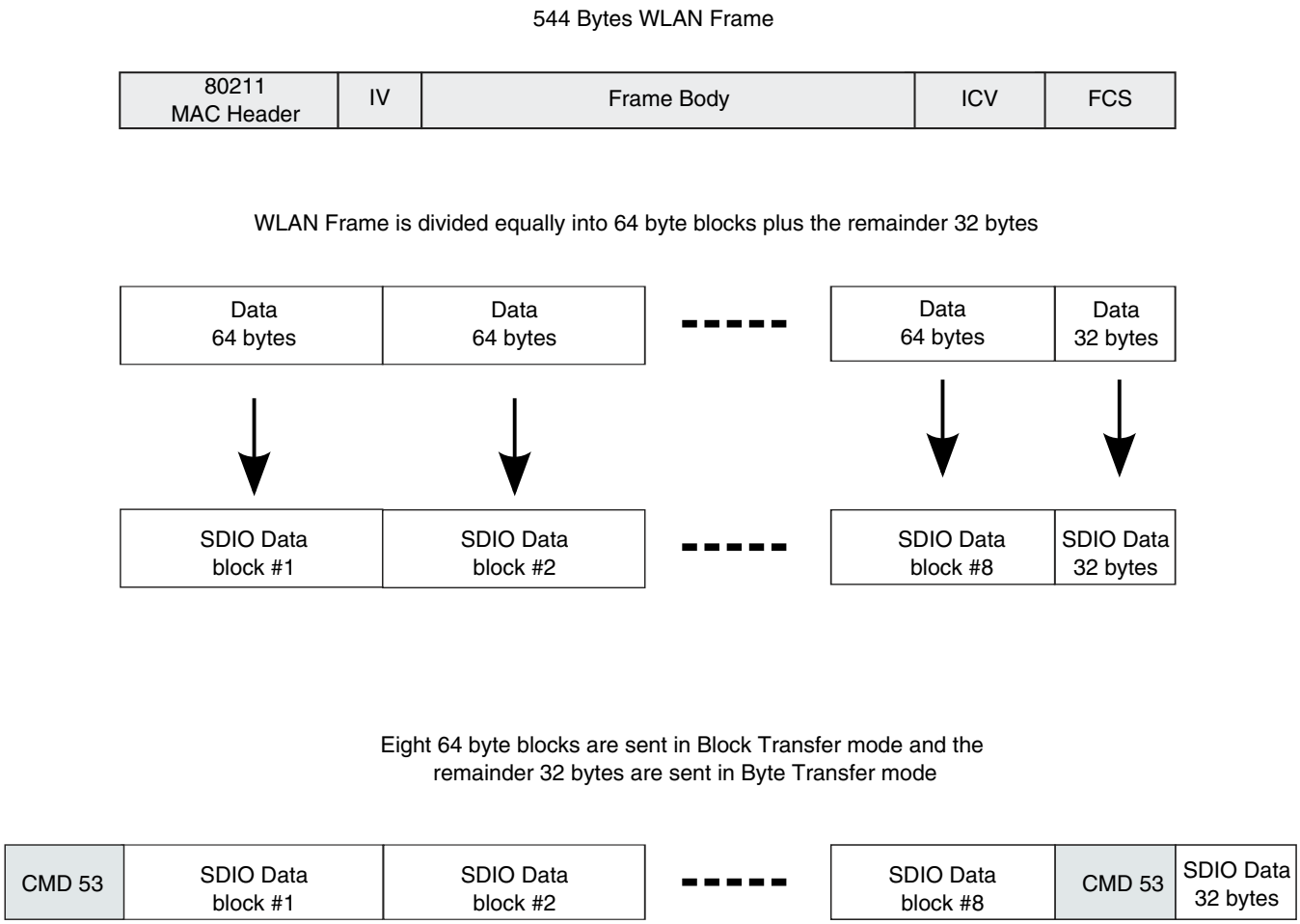


Figure 29-6. Example for Dividing Large Data Transfers

29.3.1.5 External DMA Request

When the internal DMA is not in use, and external DMA request is enabled, the Data Buffer will generate a DMA request to the system. During a write operation, when the number of WR_WML words can be held in the buffer free space, the signal esdhc_dreq_b is asserted to 0, informing the Host System of a DMA write. The BWR bit in the Interrupt Status register is also set, as long as the BWRSEN bit in the Interrupt Status Enable register is set. The DMA request is immediately de-asserted when an access to the Data Port register is made. If the buffer's free space still meets the watermark condition, the DMA request is asserted again after a cycle.

On read operation, when the number of RD_WML words are already in the buffer, the signal esdhc_dreq_b is asserted to 0, informing the Host System for a DMA read. The BRR bit in the Interrupt Status register is also set, as long as the BRRSEN bit in the

Interrupt Status Enable register is set. The DMA request is immediately de-asserted when an access to the Data Port register is made. If the buffer's data still meets the watermark condition, the DMA request is asserted again after a cycle.

Because the DMA burst length can not change during a data transfer for an external DMA transfer, the watermark level (read or write) must be a divisor of the block size. If it is not, transferring of the block may cause buffer under-run (read operation) or over-run (write operation). For example, if the block size is 512 bytes, the watermark level of read (or write) must be a power of two between 1 and 128. For processor core polling access, as the last access in the block transfer can be controlled by software, there is no such issue. The watermark level can be any value, even larger than the block size (but no greater than 128 words). This is because the actual number of bytes transferred by the software can be controlled and does not exceed the block size in each transfer.

The ESDHC also supports non-word aligned block size, as long as the card supports that block size. In this case, the watermark level should be set as the number of words. For example, if the block size is 31 bytes, the watermark level can be set to any number of word. For this case, the BLKSIZE bits of the Block Attribute register shall be set as 1fh. For the ARM platform polling access, the burst length can be 1 to 128 words, without restriction. This is because the software will transfer 8 words, and the ESDHC will also set the BWR or BRR bits when the remaining data does not violate data buffer. Refer to DMA Burst Length for more details about the dynamic watermark level of the data buffer. For the above example, even though 8 words are transferred through the Data Port register, the ESDHC will transfer only 31 bytes over the SD Bus, as required by the BLKSIZE bits. In this data transfer, with non-word aligned block size, the endian mode should be set cautiously, or invalid data will be transferred to/from the card.

29.3.2 DMA AHB Interface

The internal DMA implements a DMA engine and the AHB master. When the internal DMA is enabled, the esdhc_dreq_b will not be asserted during the transfer, but the BWR and BRR bits will be set if the BWRSEN and BRRSEN bits have been set in the Interrupt Status Enable register. See the figure below for an illustration of the DMA AHB interface block.

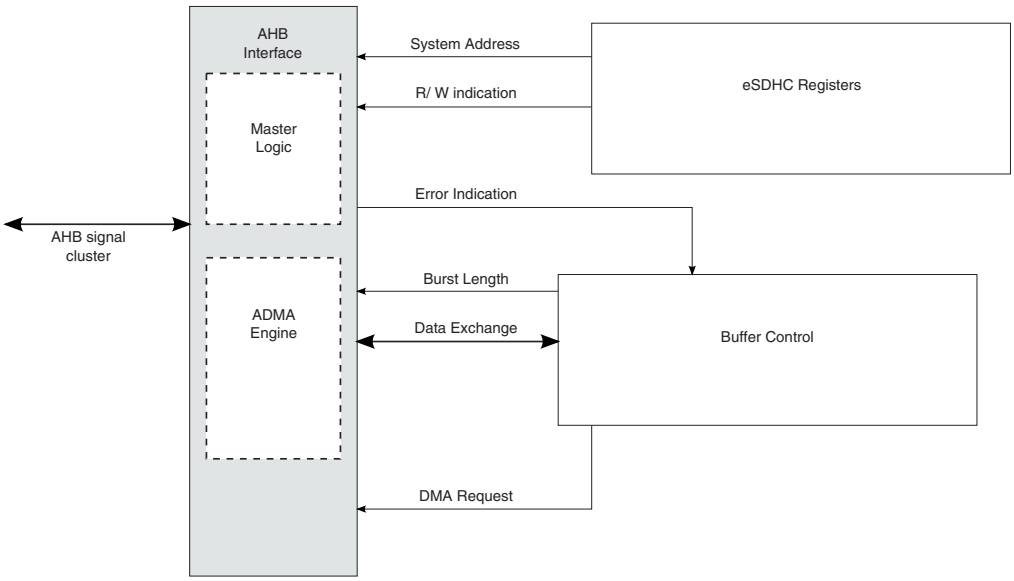


Figure 29-7. DMA AHB Interface Block

29.3.2.1 Internal DMA Request

If the watermark level requirement is met in data transfer, and the Internal DMA is enabled, the Data Buffer block will send a DMA request to AHB interface. Meanwhile, the external DMA request signal (esdhc_dreq_b) is disabled. The delay in response from the internal DMA engine depends on the system AHB bus loading and the priority assigned to the ESDHC. The DMA engine does not respond to the request during its burst transfer, but is ready to serve as soon as the burst is over. The Data Buffer de-asserts the request once an access to the buffer is made. Upon access to the buffer by internal DMA, the Data Buffer updates its internal buffer pointer, and when the watermark level is satisfied, another DMA request is sent.

The data transfer is in the block unit, and the subsequent watermark level is always set as the remaining number of words. For instance, for a multi block data read with each block size of 31 bytes, and the burst length set to 6 words. After the first burst transfer, if there are more than 2 words in the buffer (which might contain some data of the next block), another DMA request read is sent. This is because the remaining number of words to send for the current block is (31 - 6 * 4) / 4 = 2. The ESDHC will read 2 words out of the buffer, with 7 valid bytes and 1 stuff byte.

29.3.2.2 DMA Burst Length

Just like a ARM platform polling access, the DMA burst length for the internal DMA engine can be from 1 to 16 words. The actual burst length for the DMA depends on the lesser of the configured burst length or the remaining words of the current block. Take the example in Internal DMA Request again. The following burst length after 6 words are read will be 2 words, and the next burst length will be 6 words again. This is because the next block starts, which is 31 bytes, more than 6 words. The Host Driver writer may take this variable burst length into account. It is also acceptable to configure the burst length as the divisor of the block size, so that each time the burst length will be the same.

29.3.2.3 AHB Master Interface

It is possible that the internal AHB DMA engine could fail during the data transfer. When this error occurs, the DMA engine stops the transfer and goes to the idle state as well as the internal data buffer stops accepting incoming data. The DMAE bit in the Interrupt Status register is set to inform the driver.

Once the DMAE interrupt is received, the software shall send a CMD12 to abort the current transfer and read the DS_ADDR bits of the DMA System Address register to get the starting address of the corrupted block. After the DMA error is fixed, the software should apply a data reset and re-start the transfer from this address to recover the corrupted block.

29.3.2.4 ADMA Engine

In the SD Host Controller Standard, the new DMA transfer algorithm called the ADMA (Advanced DMA) is defined. For Simple DMA, once the page boundary is reached, a DMA interrupt will be generated and the new system address shall be programmed by the Host Driver. The ADMA defines the programmable descriptor table in the system memory. The Host Driver can calculate the system address at the page boundary and program the descriptor table before executing ADMA. It reduces the frequency of interrupts to the host system. Therefore, higher speed DMA transfers could be realized since the Host MCU intervention would not be needed during long DMA based data transfers.

There are two types of ADMA: ADMA1 and ADMA2 in Host Controller. ADMA1 can support data transfer of 4KB aligned data in system memory. ADMA2 removes the restriction so that data of any location and any size can be transferred in system memory. Their formats of Descriptor Table are different.

ADMA engine can recognize all kinds of descriptors define in SD Host Controller Standard, and if 'End' flag is detected in the descriptor, ADMA engine will stop after this descriptor is processed.

29.3.2.4.1 ADMA Concept and Descriptor Format

For ADMA1, including the following descriptors:

- Valid/Invalid descriptor.
- Nop descriptor.
- Set data length descriptor.
- Set data address descriptor.
- Link descriptor.
- Interrupt flag and End flag in descriptor.

For ADMA2, including the following descriptors:

- Valid/Invalid descriptor.
- Nop descriptor.
- Rsv descriptor.
- Set data length & address descriptor.
- Link descriptor.
- Interrupt flag and End flag in descriptor.

ADMA2 deals with the lower 32-bit first, and then the higher 32-bit. If the 'Valid' flag of descriptor is 0, it will ignore the high 32-bit. Address field shall be set on word aligned (lower 2-bit is always set to 0). Data length is in byte unit.

ADMA will start read/write operation after it reaches the Tran state, using the data length and data address analyzed from most recent descriptor(s).

For ADMA1, the valid data length descriptor is the last Set type descriptor before Tran type descriptor. Every Tran type will trigger a transfer, and the transfer data length is extracted from the most recent Set type descriptor. If there is no Set type descriptor after the previous Trans descriptor, the data length will be the value for previous transfer, or 0 if no Set descriptor is ever met.

For ADMA2, Tran type descriptor contains both data length and transfer data address, so only a Tran type descriptor can start a data transfer.

Address/ Page Field		Address/ Page Field		Attribute Field					
31	12	11	6	5	4	3	2	1	0
Address or Data Length		000000		Act 2	Act 1	0	Int	End	Valid

Act 2	Act1	Symbol	Comment	31- 28	27- 12
0	0	Nop	No Operation	Don't Care	
0	1	Set	Set Data Length	0000	Data Length
1	0	Tran	Transfer Data	Data Address	
1	1	Link	Link Descriptor	Descriptor Address	

Valid	Valid = 1 indicates this line of descriptor is effective. If Valid = 0 generate ADMA Error Interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA Interrupt when this descriptor is processed.

Figure 29-8. Format of the ADMA1 Descriptor Table

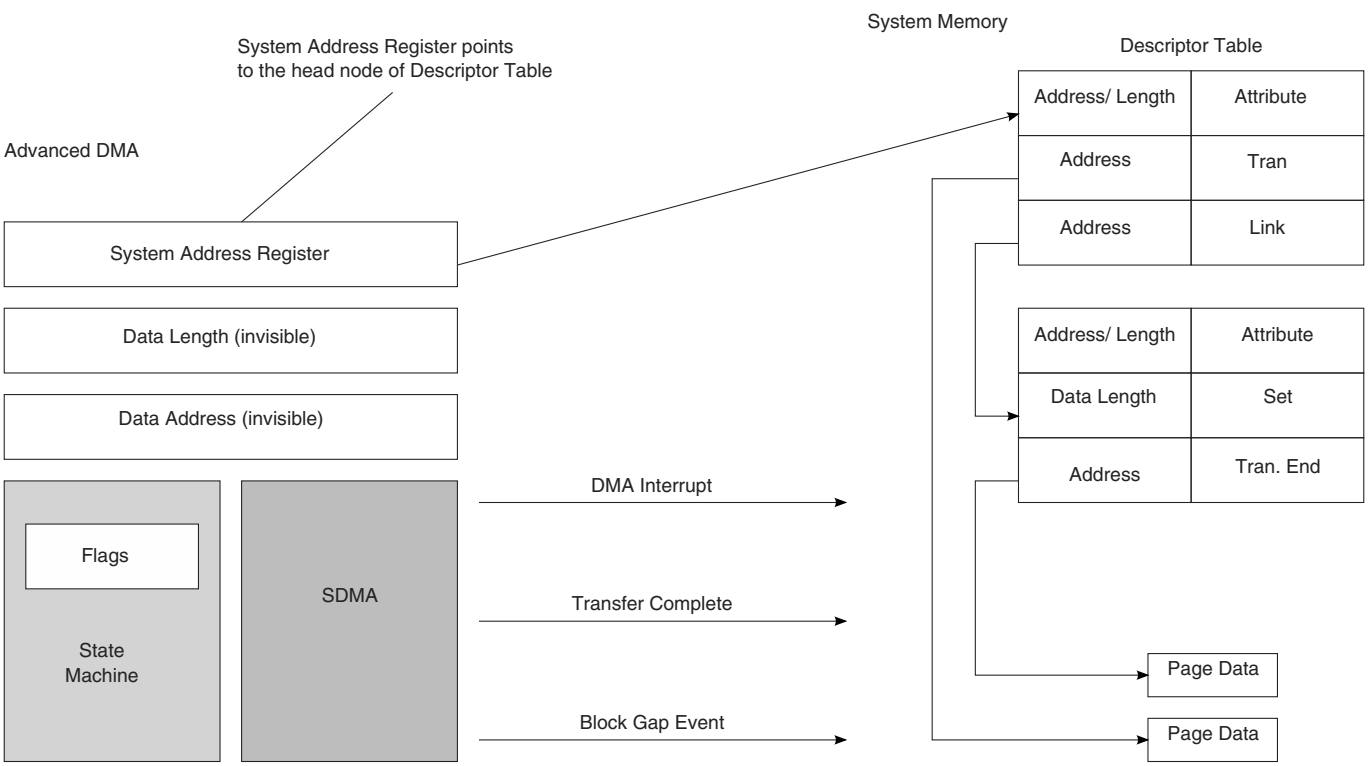


Figure 29-9. Concept and Access Method of ADMA1 Descriptor Table

Address/ Page Field		Address/ Page Field		Address/ Page Field		Attribute Field					
63	32	31	16	15	06	05	04	03	02	01	00
32-bit Address		16-bit Length		0000000000		Act 2	Act 1	0	Int	End	Valid

Act 2	Act1	Symbol	Comment	Operation
0	0	Nop	No Operation	Don't Care
0	1	Rsv	Reserved	Same as Nop. Read this line and go to next one
1	0	Tran	Transfer Data	Transfer data with address and length set in this descriptor line
1	1	Link	Link Descriptor	Link to another descriptor

Valid	Valid = 1indicates this line of descriptor is effective. If Valid = 0 generate ADMA Error Interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA Interrupt when this descriptor is done.

Figure 29-10. Format of the ADMA2 Descriptor Table

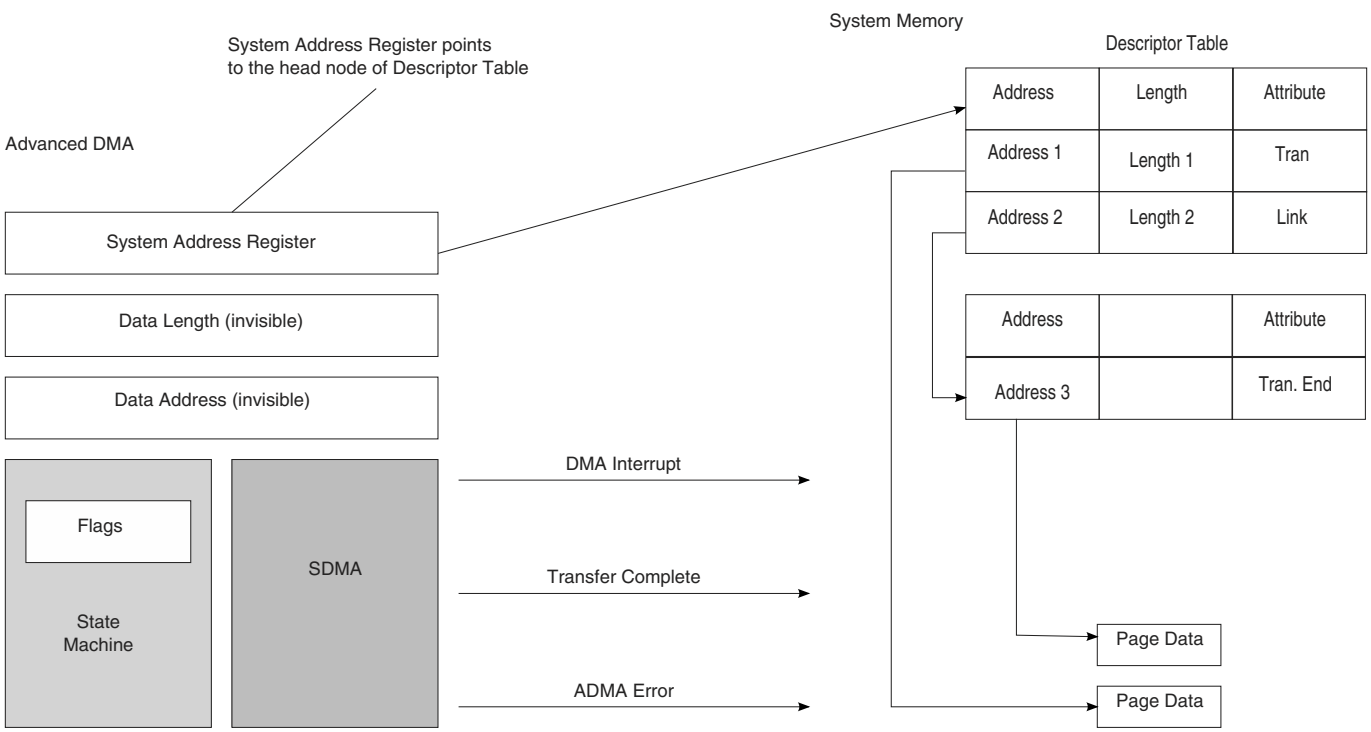


Figure 29-11. Concept and Access Method of ADMA2 Descriptor Table

29.3.2.4.2 ADMA Interrupt

If the 'Interrupt' flag of descriptor is set, ADMA will generate an interrupt according to different type descriptor:

For ADMA1:

- Set type descriptor: interrupt is generated when data length is set.
- Tran type descriptor: interrupt is generated when this transfer is complete.
- Link type descriptor: interrupt is generated when new descriptor address is set.
- Nop type descriptor: interrupt is generated just after this descriptor is fetched.

For ADMA2:

- Tran type descriptor: interrupt is generated when this transfer is complete.
- Link type descriptor: interrupt is generated when new descriptor address is set.
- Nop/Rsv type descriptor: interrupt is generated just after fetch this descriptor.

29.3.2.4.3 ADMA Error-DMA

The ADMA will stop whenever any error is encountered. These errors include:

- Fetching descriptor error
- AHB response error
- Data length mismatch error

ADMA descriptor error will be generated when it fails to detect 'Valid' flag in the descriptor. If adma descriptor error occurs, the interrupt is not generated even if the 'Interrupt' flag of this descriptor is set.

When BLKCNTEN bit is set, data length set in Block Attributes register must equal to the whole data length set in descriptor nodes, otherwise data length mismatch error will be generated.

If BLKCNTEN bit is not set, the whole data length set in descriptor should be times of block length, otherwise, when all data set in the descriptor nodes are done not at block boundary, the data mismatch error will occur.

29.3.3 Register Bank with IP Bus Interface

Register accesses via the IP Bus interface are actually on the Register Bank. See the figure below for the block diagram.

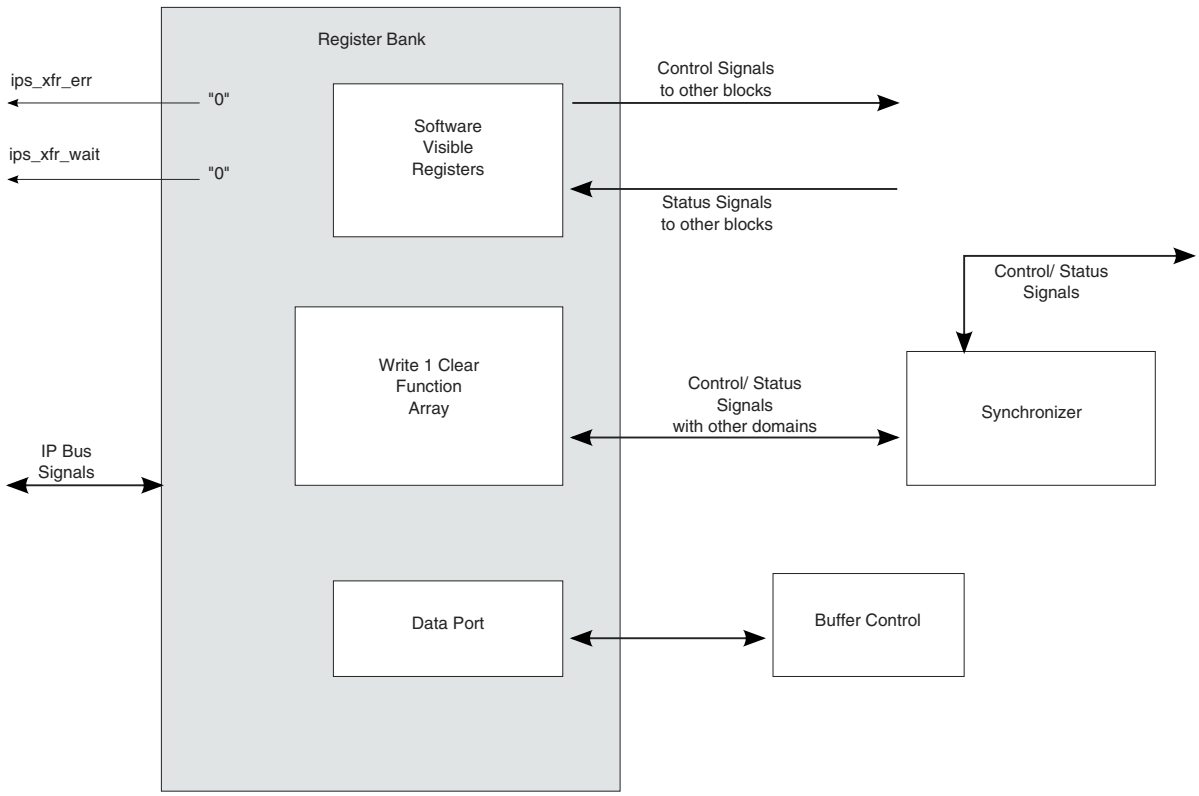


Figure 29-12. Register Bank Diagram

Only 32-bit access is allowed, and no partial read / write is supported, thus all accesses are word aligned.

29.3.4 SD Protocol Unit

The SD protocol unit deals with all SD protocol affairs.

The SD Protocol Unit performs the following functions:

- Acts as the bridge between the internal buffer and the SD bus
- Sends the command data as well as its argument serially
- Stores the serial response bit stream into corresponding registers
- Detects the bus state on the DAT[0] line
- Monitors the interrupt from the SDIO card
- Asserts the read wait signal
- Gates off the SD clock when buffer is announcing danger status
- Detects the write protect state

The SD Protocol Unit consists of four sub-blocks:

1. SD transceiver.

2. SD clock and monitor.
3. Command agent.
4. Data agent.

29.3.4.1 SD Transceiver

In the SD protocol unit, the transceiver is the main control sub-block. It consists of an FSM and control sub-block, from which the control signals for all other three sub-blocks are generated.

29.3.4.2 SD Clock and Monitor

This sub-block monitors the signal level on all 8 data lines, the command lines, and directly routes the level values into the Register Bank. The driver can use this for debug purposes.

The sub-block also detects the Card Detection (CD) line as well as the DAT[3] line. The transceiver reports the card insertion state according to the CD state, or the signal level on the DAT[3] line, when the D3CD bit in the Protocol Control register is set.

The sub-block detects the Write Protect (WP) line. With the information of the WP state, the Register Bank will ignore the command, accompanied by a write operation, when the WP switch is on.

If the internal data buffer is in danger, and the SD clock must be gated off to avoid buffer over/under-run, this sub-block will assert the gate of the output SD clock to shut the clock off. After the buffer danger has recovered, and when the system access of the buffer catches up, the clock gate of this sub-block will open and the SD clock will be active again.

This sub-block also drive SD_LCTL output signal when the LCTL bit is set by the driver.

29.3.4.3 Command Agent

The Command Agent deals with the transactions on the CMD line. See figure below for an illustration of the structure for the Command CRC Shift Register.

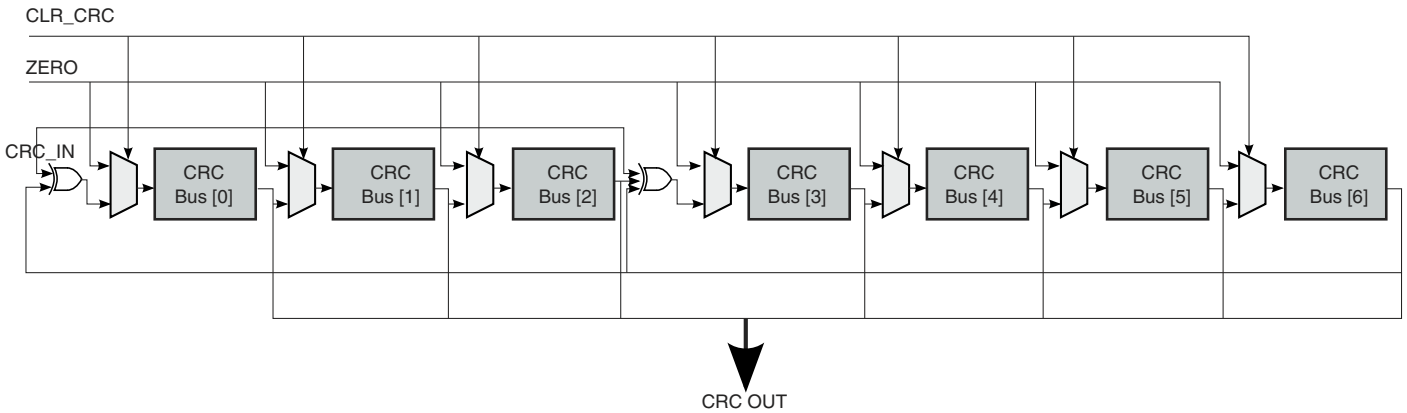


Figure 29-13. Command CRC Shift Register

The CRC polynomials for the CMD are as follows:

```
Generator polynomial: G(x) = x^7 + x^3 + 1
M(x) = (first bit) * x^n + (second bit) * x^n-1 +...+ (last bit) * x^0
CRC[6:0] = Remainder [(M(x) * x^7) / G(x)]
```

29.3.4.4 Data Agent

The Data Agent deals with the transactions on the eight data lines. Moreover, this sub-block also detects the busy state on the DAT[0] line, and generate the Read Wait state by the request from the Transceiver. The CRC polynomials for the DAT are as follows:

```
Generator polynomial: G(x) = x^16 + x^12 + x^5 +1
M(x) = (first bit) * x^n + (second bit) * x^n-1 +...+ (last bit) * x^0
CRC[15:0] = Remainder [(M(x) * x^16) / G(x)]
```

29.3.5 Clock and Reset Manager

This sub-block controls all the reset signals within the ESDHC.

There are four kinds of reset signals within ESDHC:

1. Hardware reset.
2. Software reset for all.
3. Software reset for the data part.
4. Software reset for the command part.

All these signals are fed into this sub-block and stable signals are generated inside the module to reset all other modules. The sub-block also gates off all the inside signals.

There are three clocks inside the ESDHC:

1. ipg_clk.

2. ipg_perclk.
3. hclk.

The sub-block monitors the activities of all other sub-blocks, supplies the clocks for them, and when enabled, automatically gates off the corresponding clocks.

29.3.6 Clock Generator

The Clock Generator generates the SD_CLK by peripheral source clock in two stages. Refer to Figure 29-14 for the structure of the divider. The term "Base" represents the frequency of peripheral source clock.

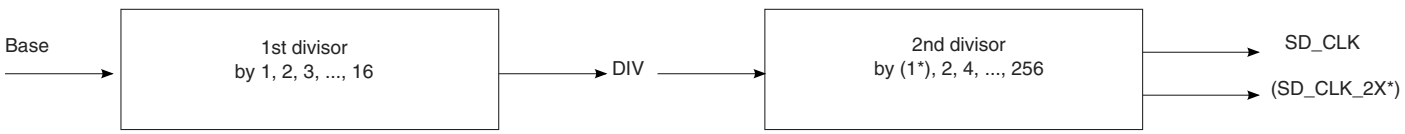


Figure 29-14. Two Stages of the Clock Divider

The first stage outputs an intermediate clock (DIV), which can be Base, Base/2, Base/3, ..., or Base/16.

The second stage is a prescaler, and outputs the actual clock (SD_CLK).This clock is the driving clock for all sub-blocks of the SD Protocol Unit, and the sync FIFOs (see Figure 29-3) to synchronize with the data rate from the internal data buffer. The frequency of the clock output from this stage, can beDIV, DIV/2, DIV/4,..., or DIV/256. Thus the highest frequency of the SD_CLK is Base, and the next highest is Base/2, while the lowest frequency is Base/4096. If the Base clock is of equal duty ratio (usually true), the duty cycle of SD_CLK is also 50%, even when the compound divisor is an odd value.

29.3.7 SDIO Card Interrupt

29.3.7.1 Interrupts in 1-bit Mode

In this case the DAT[1] pin is dedicated to providing the interrupt function. An interrupt is asserted by pulling the DAT[1] low from the SDIO card, until the interrupt service is finished to clear the interrupt.

29.3.7.2 Interrupt in 4-bit Mode

Since the interrupt and data line 1 share Pin 8 in 4-bit mode, an interrupt will only be sent by the card and recognized by the host during a specific time. This is known as the Interrupt Period. The ESDHC will only sample the level on Pin 8 during the Interrupt Period. At all other times, the host will ignore the level on Pin 8, and treat it as the data signal. The definition of the Interrupt Period is different for operations with single block and multiple block data transfers.

In the case of normal single data block transmissions, the Interrupt Period becomes active two clock cycles after the completion of a data packet. This Interrupt Period lasts until after the card receives the end bit of the next command that has a data block transfer associated with it.

For multiple block data transfers in 4-bit mode, there is only a limited period of time that the Interrupt Period can be active due to the limited period of data line availability between the multiple blocks of data. This requires a more strict definition of the Interrupt Period. For this case, the Interrupt Period is limited to two clock cycles. This begins two clocks after the end bit of the previous data block. During this 2-clock cycle interrupt period, if an interrupt is pending, the DAT[1] line will be held low for one clock cycle with the last clock cycle pulling DAT[1] high. On completion of the Interrupt Period, the card releases the DAT[1] line into the high Z state. The ESDHC samples the DAT[1] during the Interrupt Period when the IABG bit in the Protocol Control register is set.

Refer to SDIO Card Specification v1.10f for further information about the SDIO card interrupt.

29.3.7.3 Card Interrupt Handling

When the CINTIEN bit in the Interrupt Signal Enable Register is set to 0, the ESDHC clears the interrupt request to the Host System. The Host Driver should clear this bit before servicing the SDIO Interrupt and should set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts.

The SDIO Status bit is cleared by resetting the SDIO interrupt. Writing to this bit would have no effects. In 1-bit mode, the ESDHC will detect the SDIO Interrupt with or without the SD clock (to support wakeup). In 4-bit mode, the interrupt signal is sampled during the Interrupt Period, so there are some sample delays between the interrupt signal from the SDIO card and the interrupt to the Host System Interrupt Controller. When the SDIO status has been set, and the Host Driver needs to service this interrupt, so the SDIO bit in the Interrupt Control Register of SDIO card will be cleared. This is required to clear the SDIO interrupt status latched in the ESDHC and to stop driving the interrupt signal to the

System Interrupt Controller. The Host Driver must issue a CMD52 to clear the card interrupt. After completion of the card interrupt service, the SDIO Interrupt Enable bit is set to 1, and the ESDHC starts sampling the interrupt signal again.

See figure below for an illustration of the SDIO card interrupt scheme and for the sequences of software and hardware events that take place during a card interrupt handling procedure.

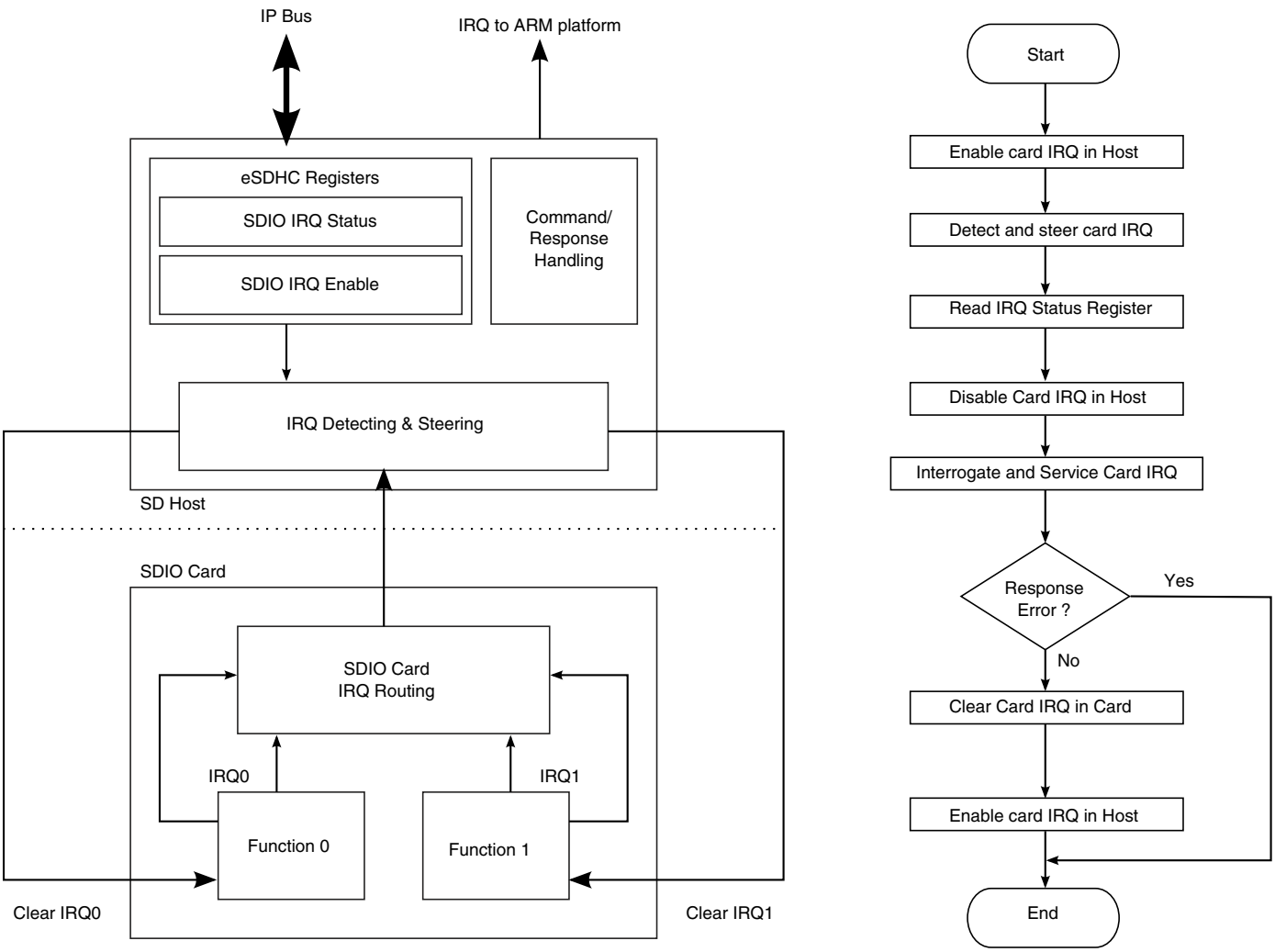


Figure 29-15. Card Interrupt Scheme and Card Interrupt Detection and Handling Procedure

29.3.8 Card Insertion and Removal Detection

The ESDHC uses either the DAT[3] pin or the CD pin to detect card insertion or removal. When there is no card on the MMC/SD bus, the DAT[3] will be pulled to a low voltage level by default. When any card is inserted to or removed from the socket, the ESDHC detects the logic value changes on the DAT[3] pin and generates an interrupt. When the DAT[3] pin is not used for card detection (for example, it is implemented in GPIO), the CD pin must be connected for card detection. Whether DAT[3] is configured for card detection or not, the CD pin is always a reference for card detection. Whether the DAT[3] pin or the CD pin is used to detect card insertion, the ESDHC will send an interrupt (if enabled) to inform the Host system that a card is inserted.

29.3.9 Power Management and Wake Up Events

When there is no operation between the ESDHC and the card through the SD bus, the user can completely disable the ipg_clk and ipg_perclk in the chip level clock control module to save power. When the user needs to use the ESDHC to communicate with the card, it can enable the clock and start the operation.

In some circumstances, when the clocks to the ESDHC are disabled, for instance, when the system is in low power mode, there are some events for which the user needs to enable the clock and handle the event. These events are called wakeup interrupts. The ESDHC can generate these interrupt even when there are no clocks enabled. The three interrupts which can be used as wake up events are:

1. Card Removal Interrupt
2. Card Insertion Interrupt
3. Interrupt from SDIO card

The ESDHC offers a power management feature. By clearing the clock enabled bits in the System Control Register (ESDHC_SYSCTL), the clocks are gated in the low position to the ESDHC. For maximum power saving, the user can disable all the clocks to the ESDHC when there is no operation in progress.

These three wake up events (or wakeup interrupts) can also be used to wake up the system from low-power modes.

NOTE

To make the interrupt a wakeup event, when all the clocks to the ESDHC are disabled or when the whole system is in low power mode, the corresponding wakeup enabled bit needs to be set. Refer to Protocol Control (ESDHCV2_PROCTL) for more information on the ESDHC Protocol Control register.

29.3.9.1 Setting Wake Up Events

For the ESDHC to respond to a wakeup event, the software must set the respective wakeup enable bit before the ARM platform enters sleep mode. Before the software disables the host clock, it should ensure that all of the following conditions have been met:

- No Read or Write Transfer is active
- Data and Command lines are not active
- No interrupts are pending
- Internal data buffer is empty

29.3.10 MMC Fast Boot

The Embedded Multimedia Card (eMMC 4.3) has a fast boot feature. In boot operation mode, the master (MultiMediaCard host) can read boot data from the slave (MMC device) by keeping CMD line low after power-on, or sending CMD0 with argument + 0xFFFFFFFA (optional for slave), before issuing CMD1.

There are two types of fast boot mode in the eMMC4.3 spec: 'Boot operation' and 'Alternative boot operation.' Each type also has with acknowledge and without acknowledge modes. eSDHCv2 supports both fast boot modes.

NOTE

For the eMMC4.3 card setting, please refer to the eMMC4.3 spec.

29.3.10.1 Boot Operation

NOTE

In this block guide, this fast boot is called normal fast boot mode.

If the CMD line is held LOW for 74 clock cycles and more after power-up before the first command is issued, the slave recognizes that boot mode is being initiated and starts preparing boot data internally.

Within 1 second after the CMD line goes LOW, the slave starts to send the first boot data to the master on the DAT line(s). The master must keep the CMD line LOW to read all of the boot data.

If boot acknowledge is enabled, the slave has to send acknowledge pattern '010' to the master within 50ms after the CMD line goes LOW. If boot acknowledge is disabled, the slave will not send out acknowledge pattern '010'.

The master can terminate boot mode with the CMD line HIGH.

Boot operation will be terminated when all contents of the enabled boot data are sent to the master. After boot operation is executed, the slave shall be ready for CMD1 operation and the master needs to start a normal MMC initialization sequence by sending CMD1.

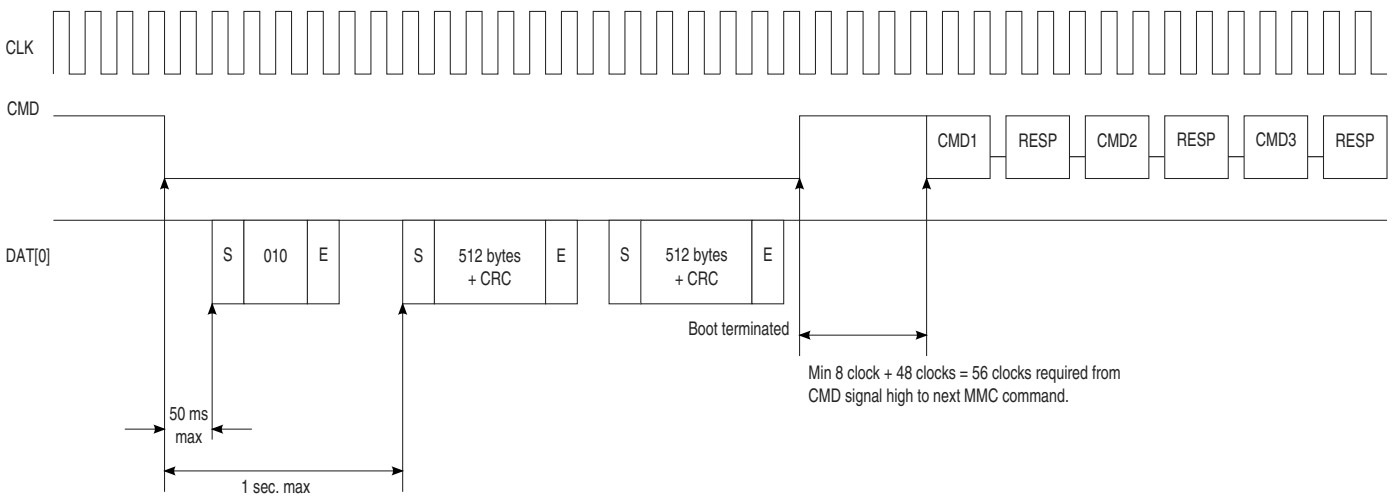


Figure 29-16. MultiMediaCard state diagram (normal boot mode)

29.3.10.2 Alternative Boot Operation

If bit 0 in the extended CSD byte[228] is set to '1', the device supports the alternative boot operation.

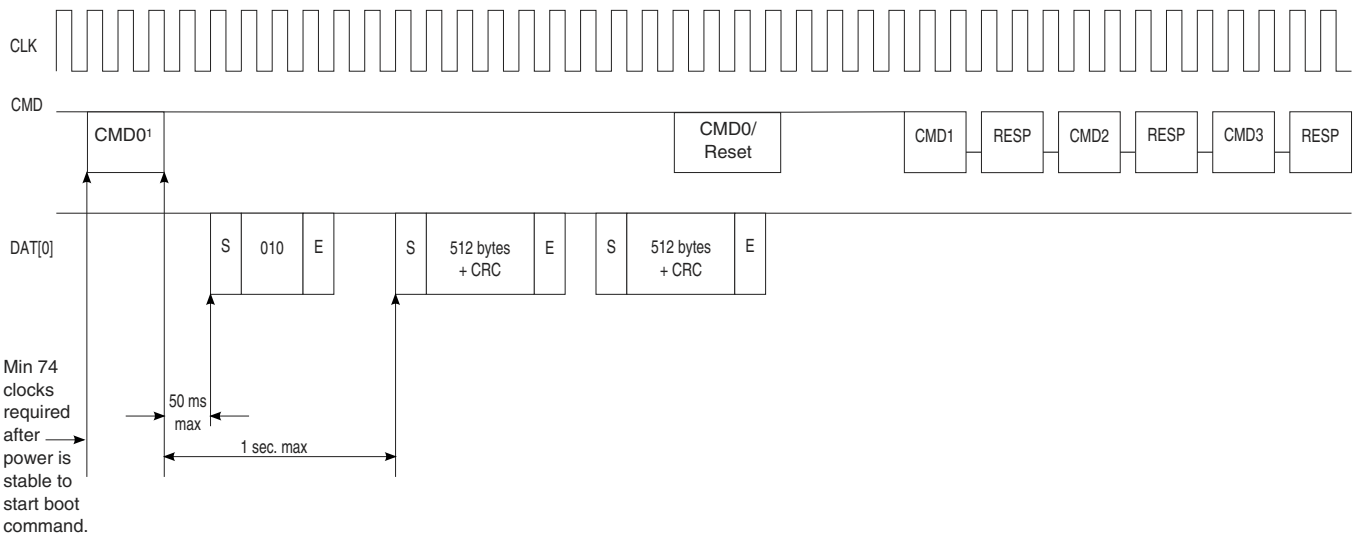
After power-up, if the host issues CMD0 with the argument of 0xFFFFFFFA after 74 clock cycles, before CMD1is issued or the CMD line goes low, the slave recognizes that boot mode is being initiated and starts preparing boot data internally.

Within 1 second after CMD0 with the argument of 0xFFFFFFFA is issued, the slave starts to send the first boot data to the master on the DAT line(s).

If boot acknowledge is enabled, the slave has to send the acknowledge pattern '010' to the master within 50ms after the CMD0 with the argument of 0xFFFFFFFA is received. If boot acknowledge is disabled, the slave will not send out acknowledge pattern '010'.

The master can terminate boot mode by issuing CMD0 (Reset).

Boot operation will be terminated when all contents of the enabled boot data are sent to the master. After boot operation is executed, the slave shall be ready for CMD1 operation and the master needs to start a normal MMC initialization sequence by sending CMD1.



NOTE1. CMD0 with argument 0xFFFFFFFA

Figure 29-17. MultiMediaCard state diagram (alternative boot mode)

29.4 Initialization/Application of ESDHC

All communication between system and cards are controlled by the host. The host sends commands of two types: broadcast and addressed (point-to-point).

Broadcast commands are intended for all cards, such as "GO_IDLE_STATE", "SEND_OP_COND", "ALL_SEND_CID" and etc. In Broadcast mode, all cards are in the open-drain mode to avoid bus contention. Refer to Commands for MMC/SD/SDIO for the commands of bc and bcr categories.

After the Broadcast command CMD3 is issued, the cards enter standby mode. Addressed type commands are used from this point. In this mode, the CMD/DAT I/O pads will turn to push-pull mode, to have the driving capability for maximum frequency operation. Refer to Commands for MMC/SD/SDIO for the commands of ac and adtc categories.

29.4.1 Command Send and Response Receive Basic Operation

Assuming the data type WORD is an unsigned 32-bit integer, the below flow is a guideline for sending a command to the card(s):

```
send_command(cmd_index, cmd_arg, other requirements)
{
WORD wCmd; // 32-bit integer to make up the data to write into Transfer Type register, it is
recommended to implement in a bit-field manner
wCmd = (<cmd_index> & 0x3f) >> 24; // set the first 8 bits as '00'+<cmd_index>
set CMDTYP, DPSEL, CICEN, CCCEN, RSTTYP, DTDSEL accorind to the command index;
if (internal DMA is used) wCmd |= 0x1;
if (multi-block transfer) {
          set MSBSEL bit;
          if (finite block number) {
                    set BCEN bit;
                    if (auto12 command is to use) set AC12EN bit;
          }
}
write_reg(CMDARG, <cmd_arg>); // configure the command argument
write_reg(XFERTYP, wCmd); // set Transfer Type register as wCmd value to issue the command
}
wait_for_response(cmd_index)
{
while (CC bit in IRQ Status register is not set); // wait until Command Complete bit is set
read IRQ Status register and check if any error bits about Command are set
if (any error bits are set) report error;
write 1 to clear CC bit and all Command Error bits;
}
```

For the sake of simplicity, the function wait_for_response is implemented here by means of polling. For an effective and formal way, the response is usually checked after the Command Complete Interrupt is received. By doing this, make sure the corresponding interrupt status bits are enabled.

For some scenarios, the response time-out is expected. For instance, after all cards respond to CMD3 and go to the Standby State, no response to the Host when CMD2 is sent. The Host Driver shall deal with 'fake' errors like this with caution.

29.4.2 Card Identification Mode

When a card is inserted to the socket or the card was reset by the host, the host needs to validate the operation voltage range, identify the cards, request the cards to publish the Relative Card Address (RCA) or to set the RCA for the MMC cards.

29.4.2.1 Card Detect

See figure below for a flow diagram showing the detection of MMC, SD and SDIO cards using the ESDHC.

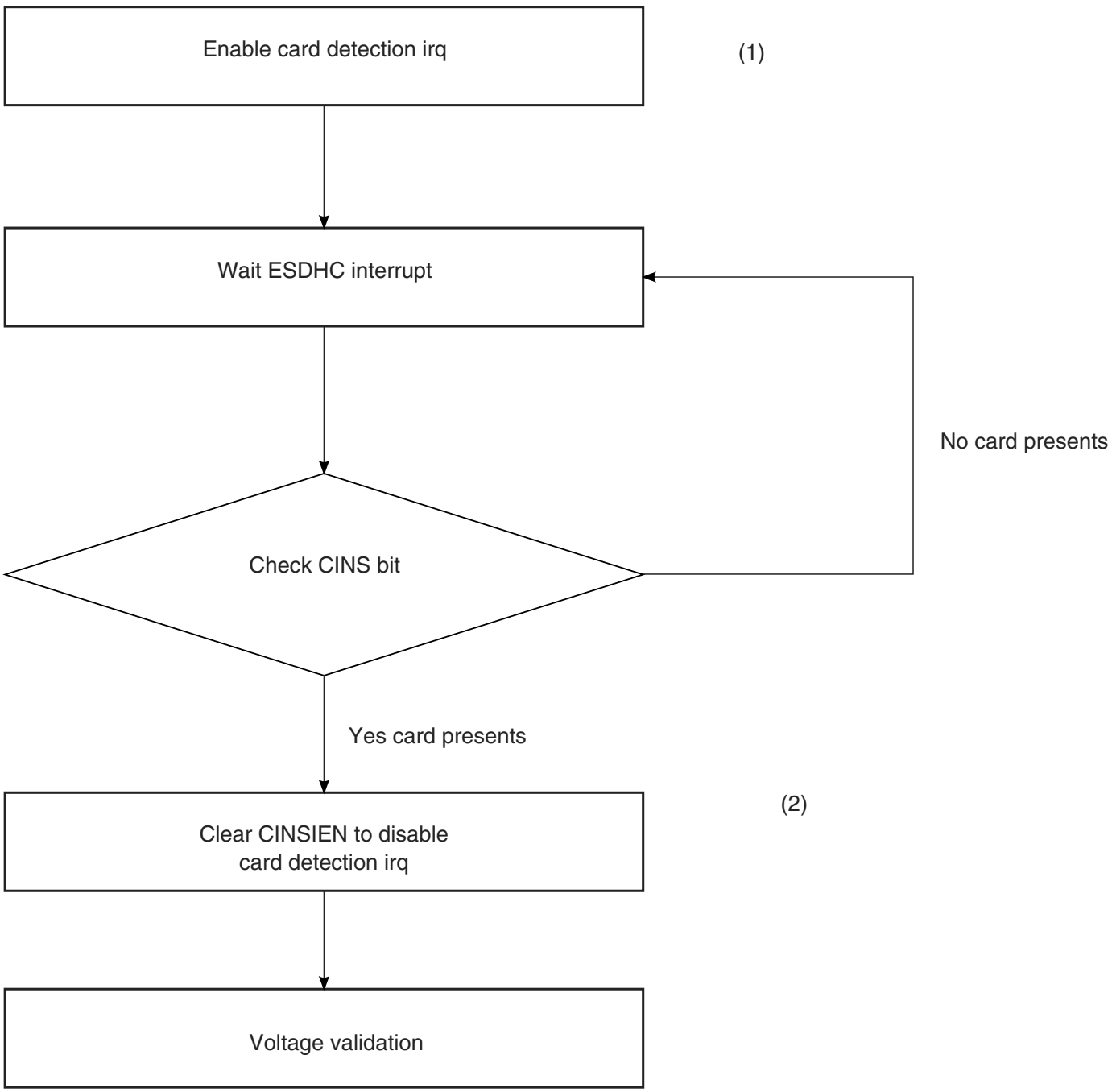


Figure 29-18. Flow Diagram for Card Detection

Here is the card detect sequence:

- Set the CINSIEN bit to enable card detection interrupt
- When an interrupt from the ESDHC is received, check the CINS bit in the Interrupt Status register to see if it was caused by card insertion
- Clear the CINSIEN bit to disable the card detection interrupt and ignore all card insertion interrupts afterwards

29.4.2.2 Reset

The host consists of three types of resets:

- Hardware reset (Card and Host) which is driven by POR (Power On Reset)
- Software reset (Host Only) is proceed by the write operation on the RSTD, RSTC, or RSTA bits of the System Control register to reset the data part, command part, or all parts of the Host Controller, respectively
- Card reset (Card Only). The command, "Go_Idle_State" (CMD0), is the software reset command for all types of MMC cards, SD Memory cards. This command sets each card into the Idle State regardless of the current card state. For an SD I/O Card, CMD52 is used to write an I/O reset in the CCCR. The cards are initialized with a default relative card address (RCA=0x0000) and with a default driver stage register setting (lowest speed, highest driving current capability).

After the card is reset, the host needs to validate the voltage range of the card. See figure below for the software flow to reset both the ESDHC and the card.

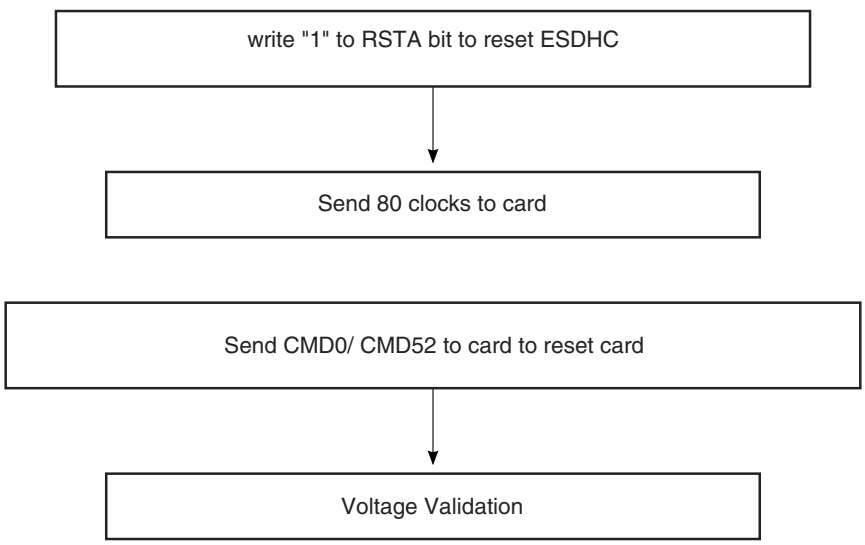


Figure 29-19. Flow Chart for Reset of the ESDHC and SD I/O Card

```
software_reset()
{
set_bit(SYSCTRL, RSTA); // software reset the Host
set DVS and SDCLKFS bit fields to get the SD_CLK of frequency around 400kHz
configure IO pad to set the power voltage of external card to around 3.0V
poll bits CIHB and CDIHB bits of PRSSTAT to wait both bits are cleared
set_bit(SYSCTRL, INTIA); // send 80 clock ticks for card to power up
send_command(CMD_GO_IDLE_STATE, <other parameters>); // reset the card with CMD0
or send_command(CMD_IO_RW_DIRECT, <other parameters>);
}
```

29.4.2.3 Voltage Validation

All cards should be able to establish communication with the host using any operation voltage in the maximum allowed voltage range specified in the card specification. However, the supported minimum and maximum values for Vdd are defined in the Operation Conditions Register (OCR) and may not cover the whole range. Cards that store the CID and CSD data in the preload memory are only able to communicate this information under data transfer Vdd conditions. This means if the host and card have non-common Vdd ranges, the card will not be able to complete the identification cycle, nor will it be able to send CSD data.

Therefore, a special command Send_Op_Cont (CMD1 for MMC), SD_Send_Op_Cont (ACMD41 for SD Memory) and IO_Send_Op_Cont (CMD5 for SD I/O) is used. The voltage validation procedure is designed to provide a mechanism to identify and reject cards which do not match the Vdd range(s) desired by the host. This is accomplished by the host sending the desired Vdd voltage window as the operand of this command. Cards that can't perform the data transfer in the specified range must discard themselves from further bus operations and go into the Inactive State. By omitting the voltage range in the command, the host can query each card and determine the common voltage range before sending out-of-range cards into the Inactive State. This query should be used if the host is able to select a common voltage range or if a notification shall be sent to the system when a non-usable card in the stack is detected.

The following steps show how to perform voltage validation when a card is inserted:

```
voltage_validation(voltage_range_arguement)
{
label the card as UNKNOWN;
send_command(IO_SEND_OP_COND, 0x0, <other parameters are omitted>); // CMD5, check SDIO
operation voltage, command argument is zero
if (RESP_TIMEOUT != wait_for_response(IO_SEND_OP_COND)) { // SDIO command is accepted
          if (0 < number of IO functions) {
                    label the card as SDIO;
                    IORDY = 0;
                    while (!(IORDY in IO OCR response)) { // set voltage range for each IO
function
                              send_command(IO_SEND_OP_COND, <voltage range>, <other
parameter>);
                              wait_for_response(IO_SEND_OP_COND);
                    } // end of while ...
          } // end of if (0 < ...
          if (memory part is present inside SDIO card) Label the card as SDCombo; // this is
an
SD-Combo card
} // end of if (RESP_TIMEOUT ...
if (the card is labelled as SDIO card) return; // card type is identified and voltage range
is
set, so exit the function;
send_command(APP_CMD, 0x0, <other parameters are omitted>); // CMD55, Application specific
CMD
prefix
if (no error calling wait_for_response(APP_CMD, <...>) { // CMD55 is accepted
          send_command(SD_APP_OP_COND, <voltage range>, <...>); // ACMD41
```

```
range
for memory part or SD card
          wait_for_response(SD_APP_OP_COND); // voltage range is set
          if (card type is UNKNOWN) label the card as SD;
          return; //
} // end of if (no error ...
else if (errors other than time-out occur) { // command/response pair is corrupted
          deal with it by program specific manner;
} // of else if (response time-out
else { // CMD55 is refuse, it must be MMC card           if (card is already labelled as
SDCombo) { // change label
                    re-label the card as SDIO;
                    ignore the error or report it;
                    return; // card is identified as SDIO card
          } // of if (card is ...
          send_command(SEND_OP_COND, <voltage range>, <...>);
          if (RESP_TIMEOUT == wait_for_response(SEND_OP_COND)) { // CMD1 is not accepted,
either
                    label the card as UNKNOWN;
                    return;
          } // of if (RESP_TIMEOUT ...
          if (check for CE-ATA signature succeeded) { // the card is CE-ATA
                    store CE-ATA specific info from the signature;
                    label the card as CE-ATA;
          } // of if (check for CE-ATA ...
          else label the card as MMC;
} // of else
}
```

29.4.2.4 Card Registry

Card registry for the MMC and SD/SDIO/SD Combo cards are different.

For the SD Card, the Identification process starts at a clock rate lower than 400 kHz and the power voltage higher than 2.7 V (as defined by the Card spec). At this time, the CMD line output drives are push-pull drivers instead of open-drain. After the bus is activated, the host will request the card to send their valid operation conditions. The response to ACMD41 is the operation condition register of the card. The same command shall be send to all of the new cards in the system. Incompatible cards are put into the Inactive State. The host then issues the command, All_Send_CID (CMD2), to each card to get its unique card identification (CID) number. Cards that are currently unidentified (in the Ready State), send their CID number as the response. After the CID is sent by the card, the card goes into the Identification State.

The host then issues Send_Relative_Addr (CMD3), requesting the card to publish a new relative card address (RCA) that is shorter than the CID. This RCA will be used to address the card for future data transfer operations. Once the RCA is received, the card changes its state to the Standby State. At this point, if the host wants the card to have an alternative RCA number, it may ask the card to publish a new number by sending another Send_Relative_Addr command to the card. The last published RCA is the actual RCA of the card.

The host repeats the identification process with CMD2 and CMD3 for each card in the system until the last CMD2 gets no response from any of the cards in system.

For MMC operation, the host starts the card identification process in open-drain mode with the identification clock rate lower than 400 kHz and the power voltage higher than 2.7 V. The open drain driver stages on the CMD line allow parallel card operation during card identification. After the bus is activated the host will request the cards to send their valid operation conditions (CMD1). The response to CMD1 is the "wired OR" operation on the condition restrictions of all cards in the system. Incompatible cards are sent into the Inactive State. The host then issues the broadcast command All_Send_CID (CMD2), asking all cards for their unique card identification (CID) number. All unidentified cards (the cards in Ready State) simultaneously start sending their CID numbers serially, while bit-wise monitoring their outgoing bit stream. Those cards, whose outgoing CID bits do not match the corresponding bits on the command line in any one of the bit periods, stop sending their CID immediately and must wait for the next identification cycle. Since the CID is unique for each card, only one card can be successfully send its full CID to the host. This card then goes into the Identification State. Thereafter, the host issues Set_Relative_Addr (CMD3) to assign to the card a relative card address (RCA). Once the RCA is received the card state changes to the Stand-by State, and the card does not react in further identification cycles, and its output driver switches from open-drain to push-pull. The host repeats the process, mainly CMD2 and CMD3, until the host receives a time-out condition to recognize the completion of the identification process.

For operation as MMC cards:

```
card_registry()
{
do { // decide RCA for each card until response time-out
          if(card is labelled as SDCombo or SDIO) { // for SDIO card like device
                    send_command(SET_RELATIVE_ADDR, 0x00, <...>); // ask SDIO card to
publish its
RCA
                    retrieve RCA from response;
          } // end if (card is labelled as SDCombo ...
          else if (card is labelled as SD) { // for SD card
                    send_command(ALL_SEND_CID, <...>);
                    if (RESP_TIMEOUT == wait_for_response(ALL_SEND_CID)) break;
                    send_command(SET_RELATIVE_ADDR, <...>);
                    retrieve RCA from response;
          } // else if (card is labelled as SD ...
          else if (card is labelled as MMC or CE-ATA) { // treat CE-ATA as MMC
                    send_command(ALL_SEND_CID, <...>);
                    rca = 0x1; // arbitrarily set RCA, 1 here for example, this RCA is also
the
relative address to access the CE-ATA card
                    send_command(SET_RELATIVE_ADDR, 0x1 << 16, <...>); // send RCA at upper
16
bits
          } // end of else if (card is labelled as MMC ...
} while (response is not time-out);
}
```

29.4.3 Card Access

29.4.3.1 Block Write

29.4.3.1.1 Normal Write

During a block write (CMD24 - 27, CMD60, CMD61), one or more blocks of data are transferred from the host to the card with a CRC appended to the end of each block by the host. If the CRC fails, the card shall indicate the failure on the DAT line. The transferred data will be discarded and not written, and all further transmitted blocks (in multiple block write mode) will be ignored.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed (CSD parameter WRITE_BLK_MISALIGN is not set), the card detects the block misalignment error and aborts the programming before the beginning of the first misaligned block. The card sets the ADDRESS_ERROR error bit in the status register, and while ignoring all further data transfer, waits in the Receive-data-State for a stop command. The write operation is also aborted if the host tries to write over a write protected area.

For MMC and SD cards, programming of the CID and CSD registers does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If this match fails, then the card will report an error and not change any register contents.

For all types of cards, some may require long and unpredictable periods of time to write a block of data. After receiving a block of data and completing the CRC check, the card will begin writing and hold the DAT line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command (CMD13) or other means for SDIO cards at any time, and the card will respond with its status. The responded status indicates whether the card can accept new data or whether the write process is still in progress. The host may deselect the card by issuing a CMD7 (to select a different card) to place the card into the Standby State and release the DAT line without interrupting the write operation. When re-selecting the card, it will reactivate the busy indication by pulling DAT to low if the programming is still in progress and the write buffer is unavailable.

The software flow to write to a card incorporates the internal DMA and the write operation is a multi-block write with the Auto CMD12 enabled. For the other two methods (by means of external DMA or ARM platform polling status) with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to write to a card is described below:

1. Check the card status, wait until the card is ready for data.
2. Set the card block length/size:
   a. For SD/MMC cards, use SET_BLOCKLEN (CMD16)
   b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT (CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7)
3. Set the ESDHC block length register to be the same as the block length set for the card in Step 2.
4. Set the ESDHC block attribute register, block number is 5 (for instance).
5. Disable the buffer write ready interrupt, configure the DMA settings and enable the ESDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a write CRC error occurred, or some another error, that occurred during the auto12 command sending and response receiving.

29.4.3.1.2 Write with Pause

The write operation can be paused during the transfer. Instead of stopping the SD_CLK at any time to pause all the operations, which is also inaccessible to the Host Driver, the Driver can set the SABGREQ bit in the Protocol Control register to pause the transfer between the data blocks. As there is no time-out condition in a write operation during the data blocks, a write to all types of cards can be paused in this way, and if the DAT0 line is not required to de-assert to release the busy state, no suspend command is needed.

Like in the flow described in Normal Write, the write with pause is shown with the same kind of write operation:

1. Check the card status, wait until card is ready for data.
2. Set the card block length/size:
   a. For SD/MMC, use SET_BLOCKLEN (CMD16)
   b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7)
3. Set the ESDHC block length register to be the same as the block length set for the card in Step 2.
4. Set the ESDHC number block register (NOB), nob is 5 (for instance).
5. Disable the buffer write ready interrupt, configure the DMA settings and enable the ESDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Set the SABGREQ bit.

7. Wait for the Transfer Complete interrupt.
8. Clear the SABGREQ bit.
9. Check the status bit to see if a write CRC error occurred.
10. Set the CREQ bit to continue the write operation.
11. Wait for the Transfer Complete interrupt.
12. Check the status bit to see if a write CRC error occurred, or some another error, that occurred during the auto12 command sending and response receiving.

The number of blocks left during the data transfer is accessible by reading the contents of the BLKCNT field in the Block Attribute register. As the data transfer and the setting of the SABGREQ bit are concurrent, and the delay of register read and the register setting, the actual number of blocks left may not be exactly the value read earlier. The Driver shall read the value of BLKCNT after the transfer is paused and the Transfer Complete interrupt is received.

It is also possible the last block has begun when the Stop At Block Gap Request is sent to the buffer. In this case, the next block gap is actually the end of the transfer. These types of requests are ignored and the Driver should treat this as a non-pause transfer and deal with it as a common write operation.

When the write operation is paused, the data transfer inside the Host System is not stopped, and the transfer is active until the data buffer is full. Because of this (if not needed), it is recommended to avoid using the Suspend Command for the SDIO card; when such a command is sent, the ESDHC thinks the System will switch to another function on the SDIO card, and flush the data buffer. The ESDHC takes the Resume Command as a normal command with data transfer, and it is left for the Driver to set all the relevant registers before the transfer is resumed. If there is only one block to send when the transfer is resumed, the MSBSEL and BCEN bits of the Transfer Type register are set as well as the AC12EN bit. However, the ESDHC will automatically send a CMD12 to mark the end of the multi-block transfer.

29.4.3.2 Block Read

29.4.3.2.1 Normal Read

For block reads, the basic unit of data transfer is a block whose maximum size is stored in areas defined by the corresponding card specification. A CRC is appended to the end of each block, ensuring data transfer integrity. The CMD17, CMD18, CMD53, CMD60, CMD61, and so on, can initiate a block read. After completing the transfer, the card returns to the Transfer State. For multi blocks read, data blocks will be continuously transferred until a stop command is issued.

The software flow to read from a card incorporates the internal DMA and the read operation is a multi-block read with the Auto CMD12 enabled. For the other two methods (by means of external DMA or ARM platform polling status) with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to read from a card is described below:

1. Check the card status, wait until card is ready for data.
2. Set the card block length/size:
   a. For SD/MMC, use SET_BLOCKLEN (CMD16)
   b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7)
3. Set the ESDHC block length register to be the same as the block length set for the card in Step 2.
4. Set the ESDHC number block register (NOB), nob is 5 (for instance).
5. Disable the buffer read ready interrupt, configure the DMA settings and enable the ESDHC DMA when sending the command with data transfer. The AC12EN bit should also be set:
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a read CRC error occurred, or some another error, occurred during the auto12 command sending and response receiving.

29.4.3.2.2 Read with Pause

The read operation is not generally able to pause. Only the SDIO card (and SDCombo card working under I/O mode) supporting the Read Wait feature can pause during the read operation. If the SDIO card supports Read Wait (SRW bit in CCCR register is 1), the Driver can set the SABGREQ bit in the Protocol Control register to pause the transfer between the data blocks. Before setting the SABGREQ bit, make sure the RWCTL bit in the Protocol Control register is set, otherwise the ESDHC will not assert the Read Wait signal during the block gap and data corruption occurs. It is recommended that the RWCTL bit be set once the Read Wait capability of the SDIO card is recognized.

Like in the flow described in Normal Read, the read with pause is shown with the same kind of read operation:

1. Check the SRW bit in the CCR register on the SDIO card to confirm the card supports Read Wait.
2. Set the RWCTL bit.
3. Check the card status and wait until the card is ready for data.
4. Set the card block length/size:

a. For SD/MMC, use SET_BLOCKLEN (CMD16)
b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7)

5. Set the ESDHC block length register to be the same as the block length set for the card in Step 2.
6. Set the ESDHC number block register (NOB), nob is 5 (for instance).
7. Disable the buffer read ready interrupt, configure the DMA setting and enable the ESDHC DMA when sending the command with data transfer. The AC12EN bit should also be set
8. Set the SABGREQ bit.
9. Wait for the Transfer Complete interrupt.
10. Clear the SABGREQ bit.
11. Check the status bit to see if read CRC error occurred.
12. Set the CREQ bit to continue the read operation.
13. Wait for the Transfer Complete interrupt.
14. Check the status bit to see if a read CRC error occurred, or some another error, occurred during the auto12 command sending and response receiving.

Like the write operation, it is possible to meet the ending block of the transfer when paused. In this case, the ESDHC will ignore the Stop At Block Gap Request and treat it as a command read operation.

Unlike the write operation, there is no remaining data inside the buffer when the transfer is paused. All data received before the pause will be transferred to the Host System. No matter if the Suspend Command is sent or not, the internal data buffer is not flushed.

If the Suspend Command is sent and the transfer is later resumed by means of a Resume Command, the ESDHC takes the command as a normal one accompanied with data transfer. It is left for the Driver to set all the relevant registers before the transfer is resumed. If there is only one block to send when the transfer is resumed, the MSBSEL and BCEN bits of the Transfer Type register are set, as well as the AC12EN bit. However, the ESDHC will automatically send the CMD12 to mark the end of multi-block transfer.

29.4.3.3 Suspend Resume

The ESDHC supports the Suspend Resume operations of SDIO cards, although slightly different than the suggested implementation of Suspend in the SDIO card specification.

Suspend

After setting the SABGREQ bit, the Host Driver may send a Suspend command to switch to another function of the SDIO card. The ESDHC does not monitor the content of the response, so it doe not know whether or not the Suspend command has succeeded. Accordingly, it does not de-assert Read Wait for read pause. To solve this problem, the Driver does not set the ESDHC_XFERTYP[CMDTYP] register to 01, that is, Suspend option. Instead, the Driver sends this command as if it were a normal command (that is, sets CMDTYP to b00). Only when the command succeeds, and the BS bit is set in the response, does the Driver send another command marked as "Suspend" to inform the ESDHC that the current transfer is suspended. This is shown in the following sequence for Suspend operation:

1. Set the SABREQ bit to pause the current data transfer at block gap.
2. After the BGE bit is set, send the Suspend command to suspend the active function. The CMDTYP bit field must be 2'b00.
3. Check the BS bit of the CCCR in the response. If it is 1, repeat this step until the BS bit is cleared or abandon the suspend operation according to the Driver strategy.
4. Send another normal I/O command to the suspended function. The CMDTYP of this command must be 2'b01, so the ESDHC can detect this special setting and be informed that the paused operation has successfully suspended. If the paused transfer is a read operation, the ESDHC stops driving DAT2 and goes to the idle state.
5. Save the context registers in the system memory for later use, including the DMA System Address Register (for internal DMA operation), and the Block Attribute Register.
6. Begin operation for another function on the SDIO card.

29.4.3.3.1 Resume

To resume the data transfer, a Resume command shall be issued:

1. To resume the suspended function, restore the context register with the saved value in step #5 of the Suspend operation above.
2. Send the Resume command. In the Transfer Type register, all bit fields are set to the value as if this were another ordinary data transfer, instead of a transfer resume (except the CMDTYP is set to 2'b10).
3. If the Resume command has responded, the data transfer will be resumed.

29.4.3.4 ADMA1 Usage

To use the ADMA1 in a data transfer, the Host Driver must prepare the correct descriptor chain prior to sending the read/write command. The steps to accomplish this are:

1. Create a descriptor to set the data length that the current descriptor group is about to transfer. The data length should be even numbers of the block size.
2. Create another descriptor to transfer the data from the address setting in this descriptor. The data address must be at a page boundary (4kB address aligned).
3. If necessary, create a Link descriptor containing the address of the next descriptor. The descriptor group is created in steps 1 ~ 3.
4. Repeat steps 1 ~ 3 until all descriptors are created.
5. In the last descriptor, set the End flag to 1 and make sure the total length of all descriptors match the product of the block size and block number configured in the Block Attribute Register.
6. Set the ADMA System Address Register to the address of the first descriptor and set the DMAS field in the Protocol Control Register to 01 to select the ADMA.
7. Issue a write or read command with the DMAEN bit set to 1 in the Transfer Type Register.

Steps 1 ~ 5 are independent of step 6, so step 6 can finish before steps 1 ~ 5. Regarding the descriptor configuration, it is recommended not to use the Link descriptor as it requires extra system memory access.

29.4.3.5 Transfer Error

29.4.3.5.1 CRC Error

It is possible at the end of a block transfer, that a write CRC status error or read CRC error occurs. For this type of error the latest block received shall be discarded. This is because the integrity of the data block is not guaranteed. It is recommended to discard the following data blocks and re-transfer the block from the corrupted one. For a multi-block transfer, the Host Driver shall issue a CMD12 to abort the current process and start the transfer by a new data command. In this scenario, even when the AC12EN and BCEND bits are set, the ESDHC does not automatically send a CMD12 because the last block is not transferred. On the other hand, if it is within the last block that the CRC error occurs, an Auto CMD12 will be sent by the ESDHC. In this case, the Driver shall re-send or re-obtain the last block with a single block transfer.

29.4.3.5.2 Internal DMA Error

During the data transfer with internal Simple DMA, if the DMA engine encounters some error on the AHB bus, the DMA operation is aborted and DMA Error interrupt is sent to the Host System. When acknowledged by such an interrupt, the Driver shall calculate the start address of data block in which the error occurs. The start address can be calculated by either:

1. Read the DMA System Address register. The error occurs during the previous burst. Taking the block size, the previous burst length and the start address of the next burst transfer into account, it is straight forward to obtain the start address of the corrupted block.
2. Read the BLKCNT field of the Block Attribute register. By the number of blocks left, the total number to transfer, the start address of transfer, and the size of each block, the start address of corrupted block can be determined. When the BCEN bit is not set, the contents of the Block Attribute register does not change, so this method does not work.

When a DMA error occurs, it is recommended to abort the current transfer by means of a CMD12 (for multi block transfer), apply a reset for data, and re-start the transfer from the corrupted block to recover from the error.

29.4.3.5.3 ADMA Error-Card Access

There are three kinds of possible ADMA errors. The AHB transfer, invalid descriptor, and data-length mismatch errors. When these errors occur, the DMA transfer stops and the corresponding error status bit is set. For acknowledging the status, the Host Driver should recover the error as shown below and re-transfer from the place of interruption.

1. AHB transfer error: Such errors may occur during data transfer or descriptor fetch. For either scenario, it is recommended to retrieve the transfer context, reset for the data part and re-transfer the block that was corrupted, or the next block if no block is corrupted.
2. Invalid descriptor error: For such errors, it is recommended to retrieve the transfer context, reset for the data part and re-create the descriptor chain from the invalid descriptor and issue a new transfer. As the data to transfer now may be less than the previous setting, the data length configured in the new descriptor chain should match the new value.
3. Data-length mismatch error: It is similar to recover from this error. The Host Driver polls relating registers to retrieve the transfer context, apply a reset for the data part, configure a new descriptor chain, and make another transfer if there is data left. Like the previous scenario of the invalid descriptor error, the data length must match the new transfer.

29.4.3.5.4 Auto CMD12 Error

After the last block of the multi block transfer is sent or received, and the AC12EN bit is set when the data transfer is initiated by the data command, the ESDHC automatically sends a CMD12 to the card to stop the transfer. When errors with this command occur, it is recommended to the Driver to deal with the situations in the following manner:

1. Auto CMD12 response time-out. It is not certain whether the command is accepted by the card or not. The Driver should clear the Auto CMD12 error status bits and re-send the CMD12 until it is accepted by the card.
2. Auto CMD12 response CRC error. Since card responds to the CMD12, the card will abort the transfer. The Driver may ignore the error and clear the error status bit.
3. Auto CMD12 conflict error or not sent. The command is not sent, so the Driver shall send a CMD12 manually.

29.4.3.6 Card Interrupt

The external cards can inform the Host Controller by means of some special signals. For the SDIO card, it can be the low level on the DAT[1] line during some special period. The ESDHC only monitors the DAT[1] line and supports the SDIO interrupt.

When the SDIO interrupt is captured by the ESDHC, and the Host System is informed by the ESDHC asserting the ESDHC interrupt line, the interrupt service from the Host Driver is called.

As the interrupt factor is controlled by the external card, the interrupt from the SDIO card must be served before the CINT bit is cleared by written 1. Refer to Card Interrupt Handling for the card interrupt handling flow.

29.4.4 Switch Function

MMC cards transferring data at bus widths other than 1-bit is a feature in MMC spec. The high speed timing mode for all card devices, was also defined in various card specifications. To enable these features, a "switch" command shall be issued by the Host Driver.

For SDIO cards, the high speed mode is enabled by writing the EHS bit in the CCCR register after the SHS bit is confirmed. For SD cards, the high speed mode is queried and enabled by a CMD6 (with the mnemonic symbol as SWITCH_FUNC). For MMC cards, the high speed mode is queried by a CMD8 and enabled by a CMD6 (with the mnemonic symbol as SWITCH).

The 4-bit and 8-bit bus width of the MMC is also enabled by the SWITCH command, but with a different argument.

DDR mode is also selected by setting ddr mode bus width via SWITCH.

These new functions can also be disabled by a software reset. For SDIO cards it can be done by setting the RES bit in the CCCR register. For other cards, it can be accomplished by issuing a CMD0. This method of restoring to the normal mode is not recommended because a complete identification process is needed before the card is ready for data transfer.

For the sake of simplicity, the following flowcharts do not show current capability check, which is recommended in the function switch process.

29.4.4.1 Query, Enable and Disable SDIO High Speed Mode

```
enable_sdio_high_speed_mode(void)
{
send CMD52 to query bit SHS at address 0x13;
if (SHS bit is '0') report the SDIO card does not support high speed mode and return;
send CMD52 to set bit EHS at address 0x13 and read after write to confirm EHS bit is set;
change clock divisor value or configure the system clock feeding into ESDHC to generate the
card_clk of around 50MHz;
(data transactions like normal peers)
}
disable_sdio_high_speed_mode(void)
{
send CMD52 to clear bit EHS at address 0x13 and read after write to confirm EHS bit is
cleared;
change clock divisor value or configure the system clock feeding into ESDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

29.4.4.2 Query, Enable and Disable SD High Speed Mode

```
enable_sd_high_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0xFFFFF1 and read 64 bytes of data accompanying the R1 response;
wait data transfer done bit is set;
check if the bit 401 of received 512 bit is set;
if (bit 401 is '0') report the SD card does not support high speed mode and return;
send CMD6, with argument 0x80FFFFF1 and read 64 bytes of data accompanying the R1 response;
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into ESDHC to generate the
card_clk of around 50MHz;
(data transactions like normal peers)
}
disable_sd_high_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0x80FFFFF0 and read 64 bytes of data accompanying the R1 response;
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into ESDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

29.4.4.3 Query, Enable and Disable MMC High Speed Mode

```
enable_mmc_high_speed_mode(void)
{
send CMD9 to get CSD value of MMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the MMC does not support high speed mode and
return;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of MMC;
extract the value of CARD_TYPE field to check the 'high speed mode' in this MMC is 26MHz or
52MHz;
send CMD6 with argument 0x1B90100;
send CMD13 to wait card ready (busy line released);
send CMD8 to get EXT_CSD value of MMC;
check if HS_TIMING byte (byte number 185) is 1;
if (HS_TIMING is not 1) report MMC switching to high speed mode failed and return;
change clock divisor value or configure the system clock feeding into ESDHC to generate the
card_clk of around 26MHz or 52MHz according to the CARD_TYPE;
(data transactions like normal peers)
}
disable_mmc_high_speed_mode(void)
{
send CMD6 with argument 0x2B90100;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of MMC;
check if HS_TIMING byte (byte number 185) is 0;
if (HS_TIMING is not 0) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into ESDHC to generate the
card_clk of the desired value below 20MHz;
(data transactions like normal peers)
}
```

29.4.4.4 Set MMC Bus Width

```
change_mmc_bus_width(void)
{
send CMD9 to get CSD value of MMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the MMC does not support multiple bit width and
return;
send CMD6 with argument 0x3B70x00; (8-bit, x=2; 4-bit, x=1; 1-bit, x=0)
send CMD13 to wait card ready (busy line released);
(data transactions like normal peers)
}
```

29.4.5 ADMA Operation

29.4.5.1 ADMA1 Operation

```
Set_adma1_descriptor
{
if (to start data transfer) {
// Make sure the address is 4KB align.
Set 'Set' type descriptor;
{
```

```
Set Act bits to 01;
Set [31:12] bits data length (byte unit);
}
Set 'Tran' type descriptor;
{
Set Act bits to 10;
Set [31:12] bits address (4KB align);
}
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to 11;
Set [31:12] bits the next descriptor address (4KB align);
}
else { // other types of descriptor
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set End bit to 1;
}
if (to generate interrupt for this descriptor) {
Set Int bit to 1;
}
Set Valid bit to 1;
}
```

29.4.5.2 ADMA2 Operation

```
Set_adma2_descriptor
{
if (to start data transfer) {
// Make sure the address is 32-bit boundary (lower 2-bit are always '00').
Set higher 32-bit of descriptor for this data transfer initial address;
Set [31:16] bits data length (byte unit);
Set Act bits to '10';
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to '11';
// Make sure the address is 32-bit boundary (lower 2-bit are always set to '00').
Set higher 32-bit of descriptor for the next descriptor address;
}
else { // other types of descriptor
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set 'End' bit '1';
}
if (to generate interrupt for this descriptor) {
Set 'Int' bit '1';
}
Set the 'Valid' bit to '1';
}
```

29.4.6 Fast Boot Operation

29.4.6.1 Normal fast boot flow

1. Software must configure INITA bit ESDHC_SYSCTL[27] to make sure 74 card clocks are finished.

2. Software must configure MMC Boot Register (ESDHC_MMCBOOT) (offset 0xc4) bit 6 to 1 (enable boot), and bit 5 to 0 (normal fast boot), and bit 4 to select the ack mode or not. If it is necessary to send through DMA mode, bit 7 must be configured to enable automatic stop at block gap feature. And need to configure bit 3-bit0 must be configured to select the ack timeout value according to the sd clk frequence.
3. Software must then configure Block Attributes Register to set block size/no.
4. Software must configure Protocol control register to set DTW (data transfer width).
5. Software must configure Command Argument Register to set argument if needed (no need in normal fast boot).
6. Software must configure Transfer Type Register to start the boot process. In normal boot mode, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN and DMAEN are kept default value. DPSEL bit is set to 1, DTDSEL is set to 1, MSBSEL is set to 1.

NOTE

DMAEN should be configured as 0 in polling mode. If BCEN is configured as 1, it is best to configure blk no in Bock Attributes Register to the max value.

7. When step 6 is configured, the boot process will begin. Software must poll the data buffer ready status to read the data from buffer in time. If boot time-out happens (ack time out or the first data read time out), interrupt will be triggered, and software must configure MMC Boot Register, bit 6 to 0 to disable boot. This will make CMD high, and after at least 56 clocks, it is ready to begin normal initialization process.
8. If no time out occurs, the software must decide if the data read is finished and then configure MMC Boot Register bit 6 to 0 to disable boot. This will make CMD line high and command completed is asserted. After at least 56 clock cycles, it is ready to begin normal initialization process.
9. Reset the host and then can begin the normal process.

29.4.6.2 Alternative fast boot flow

1. Software must configure init_active bit (system control register bit 27) to make sure 74 card clocks are finished.
2. Software must configure MMC Boot Register (ESDHC_MMCBOOT) (offset 0xc4) bit 6 to 1 (enable boot), and bit 5 to 1 (alternative boot), and bit 4 to select the ack mode or not. If it is necessary to send through DMA mode, bit 7 must be configured to enable automatic stop at block gap feature. Bit 3 through bit0 must be configured to select the ack timeout value according to the sd clk frequence.
3. Software must then configure Block Attributes Register to set block size/no.
4. Software mustconfigure Protocol control register to set DTW (data transfer width).

5. Software must configure Command Argument Register to set argument to 0xFFFFFFFA.
6. Software must configure Transfer Type Register to start the boot process by CMD0 with 0xFFFFFFFA argument . In alternative boot, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN and DMAEN are kept default value. DPSEL bit is set to 1, DTDSEL is set to 1, MSBSEL is set to 1.

NOTE

DMAEN should be configured as 0 in polling mode. And if BCEN is configured as 1 in polling mode, it is best to configure blk no in Bock Attributes Register to the max value.

7. When the step 6 is configured, boot process begins. Software must poll the data buffer ready status to read the data from buffer in time. If boot time out (acknowledge data time out in 50ms or data time out in 1s), host will send out the interrupt and software need to send CMD0 with reset and then configure boot enable bit to 0 to stop this process. After command completed, configure MMC Boot Register bit 6 to 0 to disable boot. After at least 8 clocks from command completed, card is ready for identification step.
8. If no time out occurs, software must decide when to stop the boot process, and send out the CMD0 with reset. After this command is completed, configure MMC Boot Register bit 6 to stop the process. After 8 clocks from command are completed, the slave (card) is ready for the identification step.
9. Reset the host and then can begin the normal process.

29.4.6.3 Fast boot application case (in DMA mode)

In the boot application case, because the image destination and the image size are contained in the

beginning of the image, DMA parameters must be switched on the fly during MMC fast boot.

In fast boot, host can use ADMA2 (Advanced DMA2) with two destinations.

The detail flow:

1. Software must configure init_active bit (system control register bit 27) to make sure 74 card clocks are finished.
2. Software must configure MMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot); and bit 5 to 0 (normal fast boot), to 1 (alternative boot); and bit 4 to select the acknowledge mode or not. In DMA mode, configure bit 7 to 1 for enable

automatically stop at block gap feature. Also configure bit31-bit16 to set the VALUE1 (value of block count that need to transfer first time), that host will stop at block gap when card block counter is equal to this value. Bit 3 through bit0 must be configured to select the acknowledge timeout value according to the SD clock frequency.

3. Software must then configure Block Attributes Register to set block size/no. In DMA mode, it is better to set block number to the max value (16'hFFFF).
4. Software must configure Protocol control register to set DTW (data transfer width).
5. Software enbables ADMA2 by configuring protocol control register bit9-bit8.
6. Software must set at least three pairs ADMA2 descriptor in boot memory (i.e. in IRAM, at least 6 words). The first pair descriptor define the start address (i.e. IRAM) and data length (that is, 512 bytes*VALUE1) of first part boot code. Software also must set the second pair descriptor, the second start address (any value that is writeable), data length is suggest to set 1~2word (record as VAULE2).

NOTE

The second couple descriptor also transfers useful data. (At lease 1 word because the ADMA2 can't support a 0 data_length data transfer descriptor.)

7. Software must configure Command Argument Register to set argument to 0xFFFFFFFA in alternative fast boot, and do not need set in normal fast boot.
8. Software must configure Transfer Type Register to start the boot process. CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN and DMAEN are kept default value. DPSEL bit is set to 1, DTDSEL is set to 1, and MSBSEL is set to 1. DMAEN is configured as 1 in DMA mode. And if BCEN is configured as 1, it is best to configure block no in Bock Attributes Register to the max value.
9. When the step 8 is configured, boot process will begin. The first VAULE1 block number data has transfer. Software must poll TC bit (bit1 in Interrupt Status Register) to determine if the first transfer has ended. Software must poll BGE bit (bit2 in Interrupt Status Register) to determine if first transfer has stopped at block gap.
10. When TC, BGE bit is 1, SW can analyze the first code of VAULE1 block, initialize the new memory device, if required, and set the third pair of descriptors to define the start address and length of the remaining part of boot code (VAULE3 the remain boot code block). Remember to set the last descriptor with END.
11. Software must configure MMC Boot Register (ESDHC_MMCBOOT) (offset 0xc4) again. Set bit 6 to 1 (enable boot); and bit 5 to 0 (normal fast boot), to 1 (alternative boot); and bit 4 to select the acknowledge mode or not. In DMA mode, configure bit 7 to 1 to enable automatic stop at block gap feature. Bit31-bit16 must be configured to set the (VAULE1+1+VAULE3) so that host will stop at block gap when card

block counter is equal to this value. And must configure bit 3-bit0 to select the acknowledge timeout value according to the SD clock frequency.

12. Software must clear TC and BGE bit. It must also clear SABGREQ (bit 16 in Protocol control register), and set CREQ (bit17 Protocol control register) to 1 to resume the data transfer. Host will transfer the VALUE2 and VAULE3 data to the destination that is set by descriptor.
13. Software must poll BGE bit to determine if the fast boot is over.

NOTE

When ADMA boot flow is started, for ESDHC, it is like a normal ADMA read operation. In order to keep descriptor, must have a memory length of at least a few words.For the 1~2 word data in second descriptor setting, it is the useful data, so software need to deal the data due to the application case.

29.5 Commands for MMC/SD/SDIO

See table below for the list of commands for the MMC/SD/SDIO cards.

Refer to the corresponding specifications for more details about the command information.

There are four kinds of commands defined to control the MultiMediaCard:

1. broadcast commands (bc), no response.
2. broadcast commands with response (bcr), response from all cards simultaneously.
3. addressed (point-to-point) commands (ac), no data transfer on the DAT.
4. addressed (point-to-point) data transfer commands (adtc).

The Access Bits for the EXT_CSD Access Modes are shown in table below.

Table 29-2. Commands for MMC/SD/SDIO Cards

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD0	bc	[31:0] stuff bits	-	GO_IDLE_STATE	Resets all MMC and SD memory cards to idle state.
CMD1	bcr	[31:0] OCR without busy	R3	SEND_OP_COND	Asks all MMC and SD Memory cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks all cards to send their CID numbers on the CMD line.

Table continues on the next page...

Table 29-2. Commands for MMC/SD/SDIO Cards (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD3[1]	ac	[31:6] RCA [15:0] stuff bits	R1 R6 (SDIO)	SET/ SEND_RELATIVE_ADDR	Assigns relative address to the card.
CMD4	bc	[31:0] DSR [15:0] stuff bits	-	SET_DSR	Programs the DSR of all cards.
CMD5	bc	[31:0] OCR without busy	R4	IO_SEND_OP_COND	Asks all SDIO cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD6[2]	adtc	[31] Mode 0: Check function 1: Switch function [30:8] Reserved for function groups 6 ~ 3 (All 0 or 0xFFFF) [7:4] Function group1 for command system [3:0] Function group2 for access mode	R1	SWITCH_FUNC	Checks switch ability (mode 0) and switch card function (mode 1). Refer to "SD Physical Specification V1.1" for more details.
CMD6[3]	ac	[31:26] Set to 0 [25:24] Access [23:16] Index [15:8] Value [7:3] Set to 0 [2:0] Cmd Set	R1b	SWITCH	Switches the mode of operation of the selected card or modifies the EXT_CSD registers. Refer to "The MultiMediaCard System Specification Version 4.0 Final draft 2" for more details.
CMD7	ac	[31:6] RCA [15:0] stuff bits	R1b	SELECT/ DESELECT_CARD	Toggles a card between the stand-by and transfer states or between the programming and disconnect states. In both cases, the card is selected by its own relative address and gets deselected by any other address. Address 0 deselects all.
CMD8	adtc	[31:0] stuff bits	R1	SEND_EXT_CSD	The card sends its EXT_CSD register as a block of data, with a block size of 512 bytes.
CMD9	ac	[31:6] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:6] RCA [15:0] stuff bits	R2	SEND_CID	Addressed card sends its card-identification (CID) on the CMD line.

Table continues on the next page...

Table 29-2. Commands for MMC/SD/SDIO Cards (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD11	adtc	[31:0] data address	R1	READ_DAT_UNTIL_STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD12	ac	[31:0] stuff bits	R1b	STOP_TRANSMISSION	Forces the card to stop transmission.
CMD13	ac	[31:6] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	Reserved				
CMD15	ac	[31:6] RCA [15:0] stuff bits	-	GO_INACTIVE_STATE	Sets the card to inactive state in order to protect the card stack against communication breakdowns.
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	Sets the block length (in bytes) for all following block commands (read and write). Default block length is specified in the CSD.
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a stop command.
CMD19	Reserved				
CMD20	adtc	[31:0] data address	R1	WRITE_DAT_UNTIL_STOP	Writes data stream from the host, starting at the given address, until a STOP_TRANSMISION follows.
CMD21-23	Reserved				
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows.
CMD26	adtc	[31:0] stuff bits	R1	PROGRAM_CID	Programming of the card identification register. This command shall be issued only once per card. The card contains hardware to prevent this operation after the first programming. Normally this command is reserved for the manufacturer.
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.

Table continues on the next page...

Table 29-2. Commands for MMC/SD/SDIO Cards (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD28	ac	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29	ac	[31:0] data address	R1b	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group.
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card provides write protection features, this command asks the card to send the status of the write protection bits.
CMD31	Reserved				
CMD32	ac	[31:0] data address	R1	TAG_SECTOR_START	Sets the address of the first sector of the erase group.
CMD33	ac	[31:0] data address	R1	TAG_SECTOR_END	Sets the address of the last sector in a continuous range within the selection of a single sector to be selected for erase.
CMD34	ac	[31:0] data address	R1	UNTAG_SECTOR	Removes one previously selected sector from the erase selection.
CMD35	ac	[31:0] data address	R1	TAG_ERASE_GROUP_START	Sets the address of the first erase group within a range to be selected for erase.
CMD36	ac	[31:0] data address	R1	TAG_ERASE_GROUP_END	Sets the address of the last erase group within a continuous range to be selected for erase.
CMD37	ac	[31:0] data address	R1	UNTAG_ERASE_GROUP	Removes one previously selected erase group from the erase selection.
CMD38	ac	[31:0] stuff bits	R1b	ERASE	Erase all previously selected sectors.
CMD39	ac	[31:0] RCA [15] register write flag [14:8] register address [7:0] register data	R4	FAST_IO	Used to write and read 8-bit (register) data fields. The command addresses a card, and a register, and provides the data for writing if the write flag is set. The R4 response contains data read from the address register. This command accesses application dependent registers which are not defined in the MMC standard.
CMD40	bcr	[31:0] stuff bits	R5	GO_IRQ_STATE	Sets the system into interrupt mode.

Table continues on the next page...

Table 29-2. Commands for MMC/SD/SDIO Cards (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD41	Reserved				
CDM42	adtc	[31:0] stuff bits	R1b	LOCK_UNLOCK	Used to set/reset the password or lock/unlock the card. The size of the data block is set by the SET_BLOCK_LEN command.
CMD43~51	Reserved				
CMD52	ac	[31:0] stuff bits	R5	IO_RW_DIRECT	Access a single register within the total 128k of register space in any I/O function.
CMD53	ac	[31:0] stuff bits	R5	IO_RW_EXTENDED	Accesses a multiple I/O register with a single command. Allows the reading or writing of a large number of I/O registers.
CMD54	Reserved				
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather that a standard command.
CMD56	adtc	[31:1] stuff bits [0]: RD/WR	R1b	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose / application specific commands. The size of the data block is set by the SET_BLOCK_LEN command.
CMD57~59	Reserved				
CMD60	adtc	[31] WR [30:24] stuff bits [23:16] address [15:8] stuff bits [7:0] byte count	R1b	RW_MULTIPLE_REGISTER	These registers are used to control the behavior of the device and to retrieve status information regarding the operation of the device. All Status and Control registers are WORD (32-bit) in size and are WORD aligned. CMD60 shall be used to read and write these registers.
CMD61	adtc	[31] WR [30:16] stuff bits [15:0] data unit count	R1b	RW_MULTIPLE_BLOCK	The host issues a RW_MULTIPLE_BLOCK (CMD61) to begin the data transfer.
CMD62~63	Reserved				
ACMD6[4]	ac	[31:2] stuff bits [1:0] bus width	R1	SET_BUS_WIDTH	Defines the data bus width ('00'=1bit or '10'=4bit bus) to be used for data transfer. The allowed data bus widths are given in SCR register.
ACMD13[4]	adtc	[31:0] stuff bits	R1	SD_STATUS	Send the SD Memory Card status.

Table continues on the next page...

Table 29-2. Commands for MMC/SD/SDIO Cards (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
ACMD22[4]	adtc	[31:0] stuff bits	R1	SEND_NUM_WR_SECTORS	Send the number of the written sectors (without errors). Responds with 32-bit plus the CRC data block.
ACMD23[4]	ac		R1	SET_WR_BLK_ERASE_COUNT	-
ACMD41[4]	bcr	[31:0] OCR	R3	SD_APP_OP_COND	Asks the accessed card to send its operating condition register (OCR) contents in the response on the CMD line.
ACMD42[4]	ac		R1	SET_CLR_CARD_DETECT	-
ACMD51[4]	adtc	[31:0] stuff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

1. CMD3 differs for MMC and SD cards. For MMC cards, it is referred to as SET_RELATIVE_ADDR, with a response type of R1. For SD cards, it is referred to as SEND_RELATIVE_ADDR, with a response type of R6 (with RCA inside).
2. CMD6 differs completely between high speed MMC cards and high speed SD cards. Command SWITCH_FUNC is for high speed SD cards.
3. Command SWITCH is for high speed MMC cards . The Index field can contain any value from 0-255, but only values 0-191 are valid. If the Index value is in the 192-255 range the card does not perform any modification and the SWITCH_ERROR status bit in the EXT_CSD register is set. The Access Bits are shown in Table 29-3.
4. ACMDs shall be preceded with the APP_CMD command. (Commands listed are used for SD only, other SD commands not listed are not supported on this block).

Table 29-3. EXT_CSD Access Modes

Bits	Access Name	Operation
00	Command Set	The command set is changed according to the Cmd Set field of the argument
01	Set Bits	The bits in the pointed byte are set, according to the 1 bits in the Value field.
10	Clear Bits	The bits in the pointed byte are cleared, according to the 1 bits in the Value field.
11	Write Byte	The Value field is written into the pointed byte.

29.6 Software Restrictions

29.6.1 Initialization Active

The driver cannot set INITA bit in System Control register when any of the command line or data lines is active, so the driver must ensure both CDIHB and CIHB bits are cleared. In order to auto clear the INITA bit, the SDCLKEN bit must be '1', otherwise no clocks can go out to the card and INITA will never clear.

29.6.2 Softeware Polling Procedure

For polling read or write, once the software begins a buffer read or write, it must access exactly the number of times as the values set in the Watermark Level Register; moreover, if the block size is not the times of the value in Watermark Level Register (read and write respectively), the software must access exactly the remaining number of words at the end of each block. For example, for read operation, if the RD_WML is 4, indicating the watermark level is 16 bytes, block size is 40 bytes, and the block number is 2, then the access times for the burst sequence in the whole transfer process must be 4, 4, 2, 4, 4, 2.

29.6.3 Suspend Operation

In order to suspend the data transfer, the software must inform ESDHC that the suspend command is successfully accepted. To achieve this, after the Suspend command is accepted by the SDIO card, software must send another normal command marked as suspend command (CMDTYP bits set as '01') to inform ESDHC that the transfer is suspended.

If software needs to resume the suspended trasnfer, it should read the value in BLKCNT register to save the remaining number of blocks before sending the normal command marked as suspend, otherwise on sending a 'suspend' command, ESDHC will regard the current transfer as aborted and change BLKCNT register to its original value, rather than keeping the remaining number of blocks.

29.6.4 Data Length Setting

For either ADMA (ADMA1 or ADMA2) transfer, the data in the data buffer must be word aligned, so the data length set in the descriptor must be a multiple of 4.

29.6.5 (A)DMA Address Setting

To configure ADMA1/ADMA2/DMA address register, when TC bit is set, the register will always update itself with the internal address value to support dynamic address synchronization, so software must make sure TC bit is cleared prior to configuring ADMA1/ADMA2/DMA address register.

29.6.6 Data Port Access

Data Port does not support parallel access. For example, during an external DMA access, it is not allowed to write any data to the Data Port by ARM platform; or during a ARM read operation, it is also prohibited to write any data to the Data Port, by either ARM or external DMA. Otherwise the data would be corrupted inside the ESDHC buffer.

29.6.7 Change Clock Frequency

ESDHC does not automatically gates off the card clock when the Host Driver changes the clock frequency. To remove possible glitch on the card clock, clear SDCLKEN bit when changing clock divisor value and set SDCLKEN bit to '1' after SDSTB bit is '1' again.

29.6.8 Multi-block Read

For pre-defined multi-block read operation, that is, the number of blocks to read has been defined by previous CMD23 for MMC, or pre-defined number of blocks in CMD53 for SDIO/SDCombo, or whatever multi-block read without abort command at card side, an abort command, either automatic or manual CMD12/CMD52, is still required by ESDHC after the pre-defined number of blocks are done, to drive the internal state machine to idle mode. In this case, the card may not respond to this extra abort command and ESDHC will get Response Timeout. It is recommended to manually send an abort command with RSPTYP[1:0] both bits cleared.

29.7 Programmable Registers

This section includes the block memory map and detailed descriptions of all registers. Each of these registers support only 32-bit accesses.

NOTE

Addresses greater than 0x44, except 0x50, 0x54, 0x58, 0x60, 0x64, 0xC0, 0xC4 and 0xFC, are reserved and read as all 0s. Write to these registers is ignored.

ESDHCV2 memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5000_4000	DMA System Address (ESDHCV2-1_DSADDR)	32	R/W	0000_0000h	29.7.1/ 1378
5000_4004	Block Attributes (ESDHCV2-1_BLKATTR)	32	R/W	0000_0000h	29.7.2/ 1379
5000_4008	Command Argument (ESDHCV2-1_CMDARG)	32	R/W	0000_0000h	29.7.3/ 1380
5000_400C	Command Transfer Type (ESDHCV2-1_XFERTYP)	32	R/W	0000_0000h	29.7.4/ 1381
5000_4010	Command Response 0 (ESDHCV2-1_CMDRSP0)	32	R	0000_0000h	29.7.5/ 1385
5000_4014	Command Response 1 (ESDHCV2-1_CMDRSP1)	32	R	0000_0000h	29.7.6/ 1386
5000_4018	Command Response 2 (ESDHCV2-1_CMDRSP2)	32	R	0000_0000h	29.7.7/ 1386
5000_401C	Command Response 3 (ESDHCV2-1_CMDRSP3)	32	R	0000_0000h	29.7.8/ 1386
5000_4020	Data Buffer Access Port (ESDHCV2-1_DATPORT)	32	R/W	0000_0000h	29.7.9/ 1388
5000_4024	Present State (ESDHCV2-1_PRSSTAT)	32	R	0000_0000h	29.7.10/ 1389
5000_4028	Protocol Control (ESDHCV2-1_PROCTL)	32	R/W	0000_0000h	29.7.11/ 1394
5000_402C	System Control (ESDHCV2-1_SYSCTL)	32	R/W	0000_8008h	29.7.12/ 1398
5000_4030	Interrupt Status (ESDHCV2-1_IRQSTAT)	32	w1c	0000_0000h	29.7.13/ 1401
5000_4034	Interrupt Status Enable (ESDHCV2-1_IRQSTATEN)	32	R/W	117F_013Fh	29.7.14/ 1407
5000_4038	Interrupt Signal Enable (ESDHCV2-1_IRQSIGEN)	32	R/W	0000_0000h	29.7.15/ 1410
5000_403C	Auto CMD12 Status (ESDHCV2-1_AUTOC12ERR)	32	R	0000_0000h	29.7.16/ 1412
5000_4040	Host Controller Capabilities (ESDHCV2-1_HOSTCAPBLT)	32	R	07F3_0000h	29.7.17/ 1415
5000_4044	Watermark Level (ESDHCV2-1_WML)	32	R/W	0810_0810h	29.7.18/ 1417
5000_4050	Force Event (ESDHCV2-1_FEVT)	32	W (always reads zero)	0000_0000h	29.7.19/ 1418
5000_4054	ADMA Error Status Register (ESDHCV2-1_ADMAES)	32	R	0000_0000h	29.7.20/ 1420

Table continues on the next page...

ESDHCV2 memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5000_4058	ADMA System Address (ESDHCV2-1_ADSADDR)	32	R/W	0000_0000h	29.7.21/ 1422
5000_40C0	Vendor Specific Register (ESDHCV2-1_VENDOR)	32	R/W	0000_0001h	29.7.22/ 1423
5000_40C4	MMC Boot Register (ESDHCV2-1_MMCBOOT)	32	R/W	0000_0000h	29.7.23/ 1424
5000_40FC	Host Controller Version (ESDHCV2-1_HOSTVER)	32	R	0000_1201h	29.7.24/ 1425
5000_8000	DMA System Address (ESDHCV2-2_DSADDR)	32	R/W	0000_0000h	29.7.1/ 1378
5000_8004	Block Attributes (ESDHCV2-2_BLKATTR)	32	R/W	0000_0000h	29.7.2/ 1379
5000_8008	Command Argument (ESDHCV2-2_CMDARG)	32	R/W	0000_0000h	29.7.3/ 1380
5000_800C	Command Transfer Type (ESDHCV2-2_XFERTYP)	32	R/W	0000_0000h	29.7.4/ 1381
5000_8010	Command Response 0 (ESDHCV2-2_CMDRSP0)	32	R	0000_0000h	29.7.5/ 1385
5000_8014	Command Response 1 (ESDHCV2-2_CMDRSP1)	32	R	0000_0000h	29.7.6/ 1386
5000_8018	Command Response 2 (ESDHCV2-2_CMDRSP2)	32	R	0000_0000h	29.7.7/ 1386
5000_801C	Command Response 3 (ESDHCV2-2_CMDRSP3)	32	R	0000_0000h	29.7.8/ 1386
5000_8020	Data Buffer Access Port (ESDHCV2-2_DATPORT)	32	R/W	0000_0000h	29.7.9/ 1388
5000_8024	Present State (ESDHCV2-2_PRSSTAT)	32	R	0000_0000h	29.7.10/ 1389
5000_8028	Protocol Control (ESDHCV2-2_PROCTL)	32	R/W	0000_0000h	29.7.11/ 1394
5000_802C	System Control (ESDHCV2-2_SYSCTL)	32	R/W	0000_8008h	29.7.12/ 1398
5000_8030	Interrupt Status (ESDHCV2-2_IRQSTAT)	32	w1c	0000_0000h	29.7.13/ 1401
5000_8034	Interrupt Status Enable (ESDHCV2-2_IRQSTATEN)	32	R/W	117F_013Fh	29.7.14/ 1407
5000_8038	Interrupt Signal Enable (ESDHCV2-2_IRQSIGEN)	32	R/W	0000_0000h	29.7.15/ 1410
5000_803C	Auto CMD12 Status (ESDHCV2-2_AUTOC12ERR)	32	R	0000_0000h	29.7.16/ 1412

Table continues on the next page...

ESDHCV2 memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5000_8040	Host Controller Capabilities (ESDHCV2-2_HOSTCAPBLT)	32	R	07F3_0000h	29.7.17/ 1415
5000_8044	Watermark Level (ESDHCV2-2_WML)	32	R/W	0810_0810h	29.7.18/ 1417
5000_8050	Force Event (ESDHCV2-2_FEVT)	32	W (always reads zero)	0000_0000h	29.7.19/ 1418
5000_8054	ADMA Error Status Register (ESDHCV2-2_ADMAES)	32	R	0000_0000h	29.7.20/ 1420
5000_8058	ADMA System Address (ESDHCV2-2_ADSADDR)	32	R/W	0000_0000h	29.7.21/ 1422
5000_80C0	Vendor Specific Register (ESDHCV2-2_VENDOR)	32	R/W	0000_0001h	29.7.22/ 1423
5000_80C4	MMC Boot Register (ESDHCV2-2_MMCBOOT)	32	R/W	0000_0000h	29.7.23/ 1424
5000_80FC	Host Controller Version (ESDHCV2-2_HOSTVER)	32	R	0000_1201h	29.7.24/ 1425
5002_4000	DMA System Address (ESDHCV2-4_DSADDR)	32	R/W	0000_0000h	29.7.1/ 1378
5002_4004	Block Attributes (ESDHCV2-4_BLKATTR)	32	R/W	0000_0000h	29.7.2/ 1379
5002_4008	Command Argument (ESDHCV2-4_CMDARG)	32	R/W	0000_0000h	29.7.3/ 1380
5002_400C	Command Transfer Type (ESDHCV2-4_XFERTYP)	32	R/W	0000_0000h	29.7.4/ 1381
5002_4010	Command Response 0 (ESDHCV2-4_CMDRSP0)	32	R	0000_0000h	29.7.5/ 1385
5002_4014	Command Response 1 (ESDHCV2-4_CMDRSP1)	32	R	0000_0000h	29.7.6/ 1386
5002_4018	Command Response 2 (ESDHCV2-4_CMDRSP2)	32	R	0000_0000h	29.7.7/ 1386
5002_401C	Command Response 3 (ESDHCV2-4_CMDRSP3)	32	R	0000_0000h	29.7.8/ 1386
5002_4020	Data Buffer Access Port (ESDHCV2-4_DATPORT)	32	R/W	0000_0000h	29.7.9/ 1388
5002_4024	Present State (ESDHCV2-4_PRSSTAT)	32	R	0000_0000h	29.7.10/ 1389
5002_4028	Protocol Control (ESDHCV2-4_PROCTL)	32	R/W	0000_0000h	29.7.11/ 1394
5002_402C	System Control (ESDHCV2-4_SYSCTL)	32	R/W	0000_8008h	29.7.12/ 1398

Table continues on the next page...

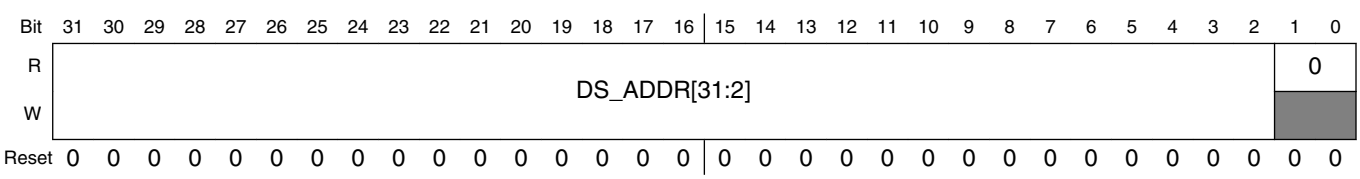
ESDHCV2 memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5002_4030	Interrupt Status (ESDHCV2-4_IRQSTAT)	32	w1c	0000_0000h	29.7.13/ 1401
5002_4034	Interrupt Status Enable (ESDHCV2-4_IRQSTATEN)	32	R/W	117F_013Fh	29.7.14/ 1407
5002_4038	Interrupt Signal Enable (ESDHCV2-4_IRQSIGEN)	32	R/W	0000_0000h	29.7.15/ 1410
5002_403C	Auto CMD12 Status (ESDHCV2-4_AUTOC12ERR)	32	R	0000_0000h	29.7.16/ 1412
5002_4040	Host Controller Capabilities (ESDHCV2-4_HOSTCAPBLT)	32	R	07F3_0000h	29.7.17/ 1415
5002_4044	Watermark Level (ESDHCV2-4_WML)	32	R/W	0810_0810h	29.7.18/ 1417
5002_4050	Force Event (ESDHCV2-4_FEVT)	32	W (always reads zero)	0000_0000h	29.7.19/ 1418
5002_4054	ADMA Error Status Register (ESDHCV2-4_ADMAES)	32	R	0000_0000h	29.7.20/ 1420
5002_4058	ADMA System Address (ESDHCV2-4_ADSADDR)	32	R/W	0000_0000h	29.7.21/ 1422
5002_40C0	Vendor Specific Register (ESDHCV2-4_VENDOR)	32	R/W	0000_0001h	29.7.22/ 1423
5002_40C4	MMC Boot Register (ESDHCV2-4_MMCBOOT)	32	R/W	0000_0000h	29.7.23/ 1424
5002_40FC	Host Controller Version (ESDHCV2-4_HOSTVER)	32	R	0000_1201h	29.7.24/ 1425

29.7.1 DMA System Address (ESDHCV2x_DSADDR)

This register contains the physical system memory address used for DMA transfers.

Addresses: ESDHCV2-1_DSADDR is 5000_4000h base + 0h offset = 5000_4000h
ESDHCV2-2_DSADDR is 5000_8000h base + 0h offset = 5000_8000h
ESDHCV2-4_DSADDR is 5002_4000h base + 0h offset = 5002_4000h



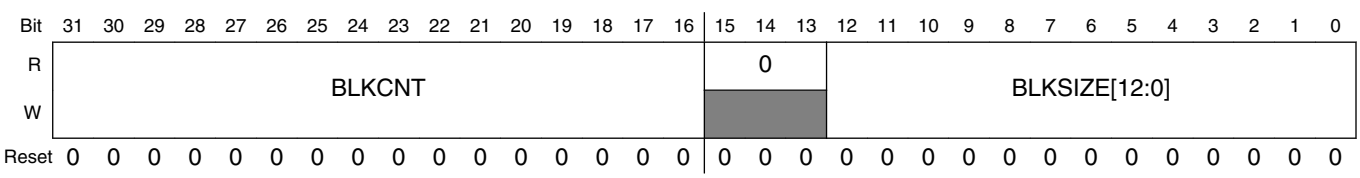
ESDHCV2x_DSADDR field descriptions

Field	Description
31–2 DS_ADDR[31:2]	DMA System Address: This register contains the 32-bit system memory address for a DMA transfer. As the address must be word (4 bytes) align, the least 2 bits are reserved, always 0. When the ESDHC stops a DMA transfer, this register points to the system address of the next contiguous data position. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). Read operation during transfers may return an invalid value. The Host Driver shall initialize this register before starting a DMA transaction. After DMA has stopped, the system address of the next contiguous data position can be read from this register. This register is protected during a data transfer. When data lines are active, write to this register is ignored. The Host driver shall wait, until ESDHCV2_PRSSTAT[DLA] bit is cleared, before writing to this register. The ESDHC internal DMA does not support a virtual memory system. It only supports continuous physical memory access. And due to AHB burst limitations, if the burst must cross the 1 KB boundary, ESDHC will automatically change SEQ burst type to NSEQ. As this register supports dynamic address reflecting, when TC bit is set, it automatically alters the value of internal address counter, so SW cannot change this register when TC bit is set. Such restriction is also listed in Software Restrictions.
1–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

29.7.2 Block Attributes (ESDHCV2x_BLKATTR)

This register is used to configure the number of data blocks and the number of bytes in each block.

Addresses: ESDHCV2-1_BLKATTR is 5000_4000h base + 4h offset = 5000_4004h
ESDHCV2-2_BLKATTR is 5000_8000h base + 4h offset = 5000_8004h
ESDHCV2-4_BLKATTR is 5002_4000h base + 4h offset = 5002_4004h



ESDHCV2x_BLKATTR field descriptions

Field	Description
31–16 BLKCNT	Blocks Count For Current Transfer: This register is enabled when ESDHCV2_XFERTYP[BCEN] bit is set to 1 and is valid only for multiple block transfers. For single block transfer, this register will always read as 1. The Host Driver shall set this register to a value between 1 and the maximum block count. The ESDHC decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to 0 results in no data blocks being transferred. This register should be accessed only when no transaction is executing (that is, after transactions have stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored.

Table continues on the next page...

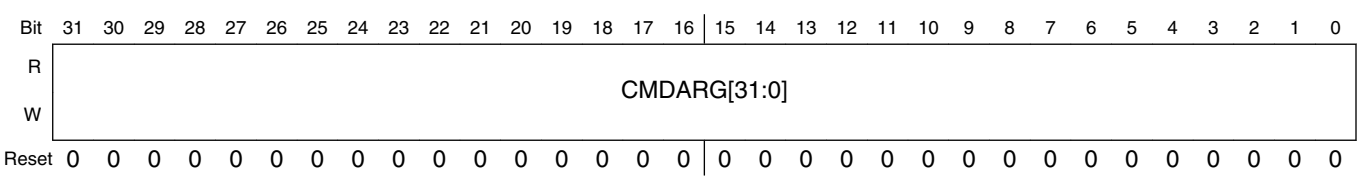
ESDHCV2x_BLKATTR field descriptions (continued)

Field	Description
	When saving transfer content as a result of a Suspend command, the number of blocks yet to be transferred can be determined by reading this register. The reading of this register should be applied after transfer is paused by stop at block gap operation and before sending the command marked as suspend. This is because when Suspend command is sent out, ESDHC will regard the current transfer is aborted and change BLKCNT register back to its original value instead of keeping the dynamical indicator of remained block count. When restoring transfer content prior to issuing a Resume command, the Host Driver shall restore the previously saved block count. 0xFFFF 65535 blocks 0x0002 2 blocks 0x0001 1 block 0x0000 Stop Count
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12–0 BLKSIZE[12:0]	Transfer Block Size: This register specifies the block size for block data transfers. Values ranging from 1 byte up to the maximum buffer size can be set. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations will be ignored. 0x1000 4096 Bytes 0x800 2048 Bytes 0x200 512 Bytes 0x1FF 511 Bytes 0x004 4 Bytes 0x003 3 Bytes 0x002 2 Bytes 0x001 1 Byte 0x000 No data transfer

29.7.3 Command Argument (ESDHCV2x_CMDARG)

This register contains the SD/MMC Command Argument.

Addresses: ESDHCV2-1_CMDARG is 5000_4000h base + 8h offset = 5000_4008h
ESDHCV2-2_CMDARG is 5000_8000h base + 8h offset = 5000_8008h
ESDHCV2-4_CMDARG is 5002_4000h base + 8h offset = 5002_4008h



ESDHCV2x_CMDARG field descriptions

Field	Description
31–0 CMDARG[31:0]	Command Argument: The SD/MMC Command Argument is specified as bits 39-8 of the Command Format in the SD or MMC Specification.This register is write protected when the Command Inhibit (CMD) bit in the Present State register is set.

29.7.4 Command Transfer Type (ESDHCV2x_XFERTYP)

This register is used to control the operation of data transfers. The Host Driver sets this register before issuing a command followed by a data transfer, or before issuing a Resume command. To prevent data loss, the ESDHC prevents writing to the bits, that are involved in the data transfer of this register, when data transfer is active. These bits are DPSEL, MBSEL, DTDSEL, AC12EN, BCEN and DMAEN.

The Host Driver shall check the Command Inhibit DAT bit (CDIHB) and the Command Inhibit CMD bit (CIHB) in the Present State register before writing to this register. When the CDIHB bit in the Present State register is set, any attempt to send a command with data by writing to this register is ignored; when the CIHB bit is set, any write to this register is ignored.

On sending commands with data transfer involved, it is mandatory that the block size is non-zero. Besides, block count must also be non-zero, or indicated as single block transfer (bit 5 of this register is '0' when written), or block count is disabled (bit 1 of this register is '0' when written), otherwise ESDHC will ignore the sending of this command and do nothing. For write command, with all above restrictions, it is also mandatory that the write protect switch is not active (WPSPL bit of Present State Register is '1), otherwise ESDHC will also ignore the command.

If the commands with data transfer does not receive the response in 64 clock cycles, that is, response time-out, ESDHC will regard the external device does not accept the command and abort the data transfer. In this scenario, the driver should issue the command again to re-try the transfer. It is also possible that for some reason the card responds the command but ESDHC does not receive the response, and if it is internal DMA (either simple DMA or ADMA) read operation, the external system memory is over-written by the internal DMA with data sent back from the card.

The CMDTYP field of this register is used to initiate three special commands these are:

- Suspend: ESDHC does not monitor the content of the Suspend command response and therefore assumes that the commnand succeeds when issued. It then operates assuming the card bus has been released and that it is permissible to issue the next command using the DAT line. It is the responsibility of S/W to ch[illegible]

the Suspend command and send another command marked as Suspend to inform the ESDHC that a Suspend command was successfully issued. Refer to Suspend Resume for more details. After the end bit of command is sent, ESDHC de-asserts Read Wait for read transactions and stops checking busy for write transactions. In 4-bit mode, the interrupt cycle starts. If the Suspend command fails, the ESDHC will maintain its current state and the Host Driver shall restart the transfer by setting the Continue Request bit in the Protocol Control register.

- Resume: S/W re-starts the data transfer by restoring the registers saved before sending the Suspend Command and then sends the Resume Command. The ESDHC will check for a pending busy state before starting write transfers.
- Abort: If this command is set when executing a read transfer, ESDHC will stop reads to the buffer. If this command is set when executing a write transfer, ESDHC will stop driving the DAT line. After issuing the Abort command, the Host Driver should issue a software reset (Abort Transaction).

Command Transfer Type (ESDHCV2_XFERTYP) shows the summary of how register settings determine the type of data transfer.

Table 29-42. Transfer Type Register Setting for Various Transfer Types

Multi/Single Block Select	Block Count Enable	Block Count	Function
0	Don't Care	Don't Care	Single Transfer
1	0	Don't Care	Infinite Transfer
1	1	Positive Number	Multiple Transfer
1	1	Zero	No Data Transfer

Table below shows the relationship between the Command Index Check Enable and the Command CRC Check Enable, in regards to the Response Type bits and the name of the response type.

Table 29-43. Relationship Between Parameters and the Name of the Response Type

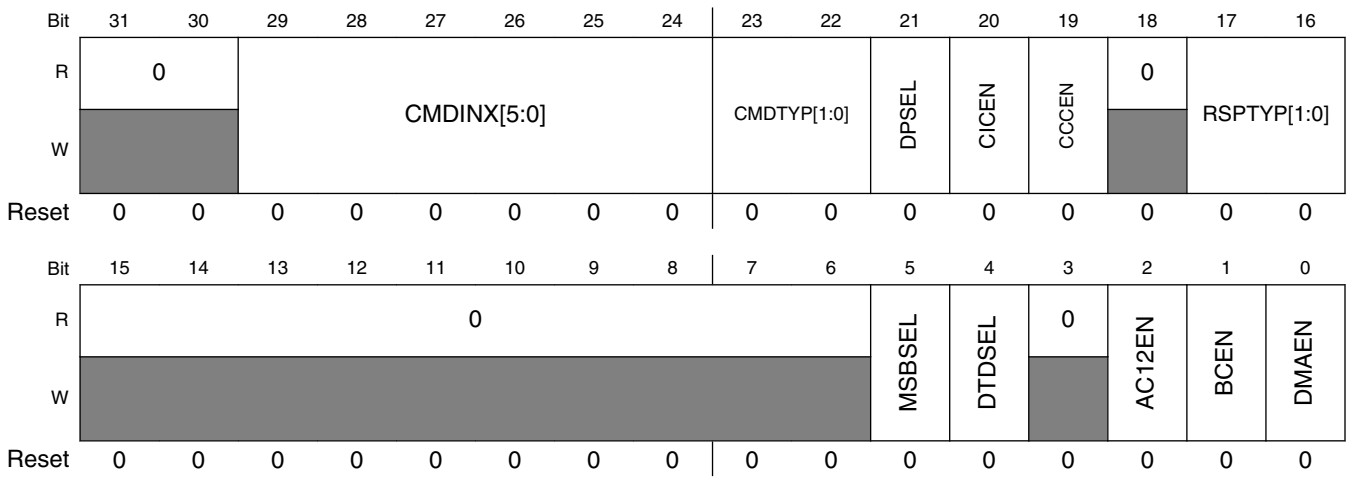
Response Type	Index Check Enable	CRC Check Enable	Name of Response Type
00	0	0	No Response
01	0	1	R2
10	0	0	R3,R4
10	1	1	R1,R5,R6
11	1	1	R1b,R5b

- In the SDIO specification, response type notation for R5b is not defined. R5 includes R5b in the SDIO specification. But R5b is defined in this specification to indicate

that ESDHC will check the busy status after receiving a response. For example, usually CMD52 is used with R5, but the I/O abort command will be used with R5b.

- The CRC field for R3 and R4 is expected to be all 1 bits. The CRC check shall be disabled for these response types.

Addresses: ESDHCV2-1_XFERTYP is 5000_4000h base + Ch offset = 5000_400Ch
ESDHCV2-2_XFERTYP is 5000_8000h base + Ch offset = 5000_800Ch
ESDHCV2-4_XFERTYP is 5002_4000h base + Ch offset = 5002_400Ch



ESDHCV2x_XFERTYP field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved
29–24 CMDINX[5:0]	Command Index: These bits shall be set to the command number that is specified in bits 45-40 of the Command-Format in the SD Memory Card Physical Layer Specification and SDIO Card Specification.
23–22 CMDTYP[1:0]	Command Type: 11 Abort CMD12, CMD52 for writing I/O Abort in CCCR 10 Resume CMD52 for writing Function Select in CCCR 01 Suspend CMD52 for writing Bus Suspend in CCCR 00 Normal mode. Used for all other commands
21 DPSEL	Data Present Select: This bit is set to 1 to indicate that data is present and shall be transferred using the DAT line. It is set to 0 for the following: • Commands using only the CMD line (for example, CMD52). • Commands with no data transfer, but using the busy signal on DAT[0] line (R1b or R5b such as, CMD38) **NOTE:** In resume command, this bit shall be set, and other bits in this register shall be set the same as when the transfer was initially launched. When the Write Protect switch is on, (that is, the WPSPL bit is active as '0'), any command with a write operation will be ignored. That is to say, when this bit is set, while the DTDSEL bit is 0, writes to the register Transfer Type are ignored.

Table continues on the next page...

ESDHCV2x_XFERTYP field descriptions (continued)

Field	Description
	1 Data Present 0 No Data Present
20 CICEN	Command Index Check Enable: If this bit is set to 1, the ESDHC will check the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this bit is set to 0, the Index field is not checked. 1 Enable 0 Disable
19 CCCEN	Command CRC Check Enable: If this bit is set to 1, the ESDHC shall check the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this bit is set to 0, the CRC field is not checked. The number of bits checked by the CRC field value changes according to the length of the response. (Refer to RSPTYP[1:0] and Command Transfer Type (ESDHCV2_XFERTYP).) 1 Enable 0 Disable
18 Reserved	This read-only field is reserved and always has the value zero. Reserved
17–16 RSPTYP[1:0]	Response Type Select: 00 No Response 01 Response Length 136 10 Response Length 48 11 Response Length 48, check Busy after response
15–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5 MSBSEL	Multi / Single Block Select: This bit enables multiple block DAT line data transfers. For any other commands, this bit shall be set to 0. If this bit is 0, it is not necessary to set the Block Count register. (See Command Transfer Type (ESDHCV2_XFERTYP).) 1 Multiple Blocks 0 Single Block
4 DTDSEL	Data Transfer Direction Select: This bit defines the direction of DAT line data transfers. The bit is set to 1 by the Host Driver to transfer data from the SD card to the ESDHC and is set to 0 for all other commands. 1 Read (Card to Host) 0 Write (Host to Card)
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2 AC12EN	Auto CMD12 Enable:

Table continues on the next page...

ESDHCV2x_XFERTYP field descriptions (continued)

Field	Description
	Multiple block transfers for memory require a CMD12 to stop the transaction. When this bit is set to 1, the ESDHC will issue a CMD12 automatically when the last block transfer has completed. The Host Driver shall not set this bit to issue commands that do not require CMD12 to stop a multiple block data transfer. In particular, secure commands defined in File Security Specification (see reference list) do not require CMD12. In single block transfer, the ESDHC will ignore this bit no matter if it is set or not. 1 Enable 0 Disable
1 BCEN	Block Count Enable: This bit is used to enable the Block Count register, which is only relevant for multiple block transfers. When this bit is 0, the internal counter for block is disabled, which is useful in executing an infinite transfer. 1 Enable 0 Disable
0 DMAEN	DMA Enable: This bit enables DMA functionality. If this bit is set to 1, a DMA operation shall begin when the Host Driver sets the DPSEL bit of this register. Whether the Simple DMA, or the Advanced DMA, is active depends on the DMA Select field of the Protocol Control register. 1 Enable 0 Disable

29.7.5 Command Response 0 (ESDHCV2x_CMDRSP0)

This register is used to store part 0 of the response bits from the card.

Addresses: ESDHCV2-1_CMDRSP0 is 5000_4000h base + 10h offset = 5000_4010h
ESDHCV2-2_CMDRSP0 is 5000_8000h base + 10h offset = 5000_8010h
ESDHCV2-4_CMDRSP0 is 5002_4000h base + 10h offset = 5002_4010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP0[31:0]																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

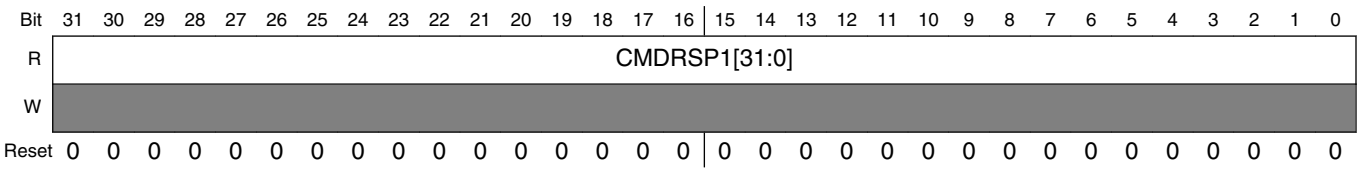
ESDHCV2x_CMDRSP0 field descriptions

Field	Description
31–0 CMDRSP0[31:0]	Command Response 0: Refer to Command Response 3 (ESDHCV2_CMDRSP3) for the mapping of command responses from the SD Bus to this register for each response type.

29.7.6 Command Response 1 (ESDHCV2x_CMDRSP1)

This register is used to store part 1 of the response bits from the card.

Addresses: ESDHCV2-1_CMDRSP1 is 5000_4000h base + 14h offset = 5000_4014h
ESDHCV2-2_CMDRSP1 is 5000_8000h base + 14h offset = 5000_8014h
ESDHCV2-4_CMDRSP1 is 5002_4000h base + 14h offset = 5002_4014h



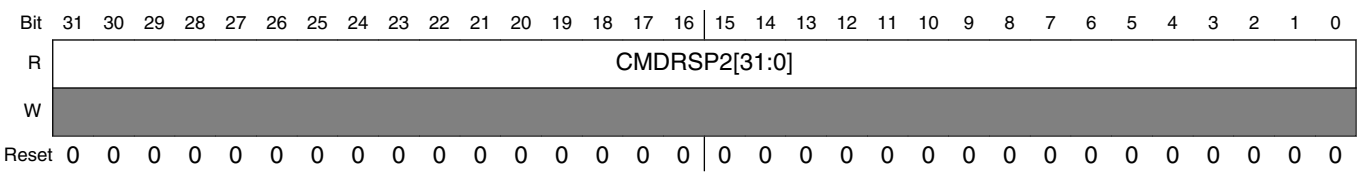
ESDHCV2x_CMDRSP1 field descriptions

Field	Description
31–0 CMDRSP1[31:0]	Command Response 1: Refer to Command Response 3 (ESDHCV2_CMDRSP3) for the mapping of command responses from the SD Bus to this register for each response type.

29.7.7 Command Response 2 (ESDHCV2x_CMDRSP2)

This register is used to store part 2 of the response bits from the card.

Addresses: ESDHCV2-1_CMDRSP2 is 5000_4000h base + 18h offset = 5000_4018h
ESDHCV2-2_CMDRSP2 is 5000_8000h base + 18h offset = 5000_8018h
ESDHCV2-4_CMDRSP2 is 5002_4000h base + 18h offset = 5002_4018h



ESDHCV2x_CMDRSP2 field descriptions

Field	Description
31–0 CMDRSP2[31:0]	Command Response 2: Refer to Command Response 3 (ESDHCV2_CMDRSP3) for the mapping of command responses from the SD Bus to this register for each response type.

29.7.8 Command Response 3 (ESDHCV2x_CMDRSP3)

This register is used to store part 3 of the response bits from the card.

Command Response 3 (ESDHCV2_CMDRSP3) describes the mapping of command responses from the SD Bus to Command Response registers for each response type. In the table, R[] refers to a bit range within the response data as transmitted on the SD Bus.

Table 29-48. Response Bit Definition for Each Response Type

Response Type	Meaning of Response	Response Field	Response Register
R1,R1b (normal response)	Card Status	R[39:8]	ESDHCV2_CMDRSP0
R1b (Auto CMD12 response)	Card Status for Auto CMD12	R[39:8]	ESDHCV2_CMDRSP3
R2 (CID, CSD register)	CID/CSD register [127:8]	R[127:8]	{ESDHCV2_CMDRSP3[23:0], ESDHCV2_CMDRSP2, ESDHCV2_CMDRSP1, ESDHCV2_CMDRSP0}
R3 (OCR register)	OCR register for memory	R[39:8]	ESDHCV2_CMDRSP0
R4 (OCR register)	OCR register for I/O etc.	R[39:8]	ESDHCV2_CMDRSP0
R5, R5b	SDIO response	R[39:8]	ESDHCV2_CMDRSP0
R6 (Publish RCA)	New Published RCA[31:16] and card status[15:0]	R[39:9]	ESDHCV2_CMDRSP0

This table shows that most responses with a length of 48 (R[47:0]) have 32-bits of the response data (R[39:8]) stored in the ESDHCV2_CMDRSP0 register. Responses of type R1b (Auto CMD12 responses) have response data bits (R[39:8]) stored in the ESDHCV2_CMDRSP3 register. Responses with length 136 (R[135:0]) have 120-bits of the response data (R[127:8]) stored in the ESDHCV2_CMDRSP0, 1, 2, and 3 registers.

To be able to read the response status efficiently, the ESDHC only stores part of the response data in the Command Response registers. This enables the Host Driver to efficiently read 32-bits of response data in one read cycle on a 32-bit bus system. Parts of the response, the Index field and the CRC, are checked by the ESDHC (as specified by the Command Index Check Enable and the Command CRC Check Enable bits in the Transfer Type register) and generate an error interrupt if any error is detected. The bit range for the CRC check depends on the response length. If the response length is 48, the ESDHC will check R[47:1], and if the response length is 136 the ESDHC will check R[119:1].

As ESDHC may have a multiple block data transfer executing concurrently with a CMD_wo_DAT command, the ESDHC stores the Auto CMD12 response in the ESDHCV2_CMDRSP3 register. The CMD_wo_DAT response is stored in ESDHCV2_CMDRSP0. This allows the ESDHC to avoid overwriting the Auto CMD12 response with the CMD_wo_DAT and vice versa. When the ESDHC modifies part of the Command Response registers, as shown in the table above, it preserves the unmodified bits.

Addresses: ESDHCV2-1_CMDRSP3 is 5000_4000h base + 1Ch offset = 5000_401Ch
ESDHCV2-2_CMDRSP3 is 5000_8000h base + 1Ch offset = 5000_801Ch
ESDHCV2-4_CMDRSP3 is 5002_4000h base + 1Ch offset = 5002_401Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP3[31:0]																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDHCV2x_CMDRSP3 field descriptions

Field	Description
31–0 CMDRSP3[31:0]	Command Response 3: Refer to Command Response 3 (ESDHCV2_CMDRSP3) for the mapping of command responses from the SD Bus to this register for each response type.

29.7.9 Data Buffer Access Port (ESDHCV2x_DATPORT)

This is a 32-bit data port register used to access the internal buffer.

Addresses: ESDHCV2-1_DATPORT is 5000_4000h base + 20h offset = 5000_4020h
ESDHCV2-2_DATPORT is 5000_8000h base + 20h offset = 5000_8020h
ESDHCV2-4_DATPORT is 5002_4000h base + 20h offset = 5002_4020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATCONT[31:0]																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDHCV2x_DATPORT field descriptions

Field	Description
31–0 DATCONT[31:0]	Data Content: The Buffer Data Port register is for 32-bit data access by the ARM platform or the external DMA. When the internal DMA is enabled, any write to this register is ignored, and any read from this register will always yield 0s.

29.7.10 Present State (ESDHCV2x_PRSSTAT)

The Host Driver can get status of the ESDHC from this 32-bit read only register. It issues CMD0, CMD12, CMD13 (for memory) and CMD52 (for SDIO) when the DAT lines are busy during a data transfer. These commands can be issued when Command Inhibit (CMD) is set to zero. Other commands will be issued when Command Inhibit (DAT) is set to zero. Possible changes to the SD Physical Specification may add other commands to this list in the future.

NOTE

The reset value of Present State Register depend on testbench connectivity.

Addresses: ESDHCV2-1_PRSSTAT is 5000_4000h base + 24h offset = 5000_4024h
ESDHCV2-2_PRSSTAT is 5000_8000h base + 24h offset = 5000_8024h
ESDHCV2-4_PRSSTAT is 5002_4000h base + 24h offset = 5002_4024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DLSL[7:0]								CLSL	0			WPSPL	CDPL	0	CINS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				BREN	BWEN	RTA	WTA	SDOFF	PEROFF	HCKOFF	IPGOFF	SDSTB	DLA	CDIHB	CIHB
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDHCV2x_PRSSTAT field descriptions

Field	Description
31–24 DLSL[7:0]	DAT[7:0] Line Signal Level: This status is used to check the DAT line level to recover from errors, and for debugging.This is especially useful in detecting the busy signal level from DAT[0]. The reset value is effected by the external pull-up/ pull-down resistors. By default, the read value of this bit field after reset is 8'b11110111, when DAT[3] is pulled down and the other lines are pulled up. DAT[7]: Data 7 line signal level DAT[6]: Data 6 line signal level

Table continues on the next page...

ESDHCV2x_PRSSTAT field descriptions (continued)

Field	Description
	DAT[5]: Data 5 line signal level DAT[4]: Data 4 line signal level DAT[3]: Data 3 line signal level DAT[2]: Data 2 line signal level DAT[1]: Data 1 line signal level DAT[0]: Data 0 line signal level
23 CLSL	CMD Line Signal Level: This status is used to check the CMD line level to recover from errors, and for debugging. The reset value is effected by the external pull-up/pull-down resistor, by default, the read value of this bit after reset is 1'b1, when the command line is pulled up.
22–20 Reserved	This read-only field is reserved and always has the value zero. Reserved
19 WPSPL	Write Protect Switch Pin Level: The Write Protect Switch is supported for memory and combo cards.This bit reflects the inverted value of the SD_WP pin of the card socket. A software reset does not affect this bit. The reset value is effected by the external write protect switch. If the SD_WP pin is not used, it should be tied low, so that the reset value of this bit is high and write is enabled. 1 Write enabled (SD_WP=0) 0 Write protected (SD_WP=1)
18 CDPL	Card Detect Pin Level: This bit reflects the inverse value of the SD_CD# pin for the card socket. Debouncing is not performed on this bit. This bit may be valid, but is not guaranteed, because of propagation delay. Use of this bit is limited to testing As it must be debounced by software. A software reset does not effect this bit. A write to the Force Event Register does not effect this bit. The reset value is effected by the external card detection pin. This bit shows the value on the SD_CD# pin (that is, when a card is inserted in the socket, it is 0 on the SD_CD# input, and consequently the CDPL reads 1.) 1 Card present (SD_CD#=0) 0 No card present (SD_CD#=1)
17 Reserved	This read-only field is reserved and always has the value zero. Reserved
16 CINS	Card Inserted: This bit indicates whether a card has been inserted. The ESDHC debounces this signal so that the Host Driver will not need to wait for it to stabilize. Changing from a 0 to 1 generates a Card Insertion interrupt in the Interrupt Status register. Changing from a 1 to 0 generates a Card Removal interrupt in the Interrupt Status register. A write to the Force Event Register does not effect this bit. The Software Reset For All in the System Control register does not effect this bit.A software reset does not effect this bit. 1 Card Inserted 0 Power on Reset or No Card
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

ESDHCV2x_PRSSTAT field descriptions (continued)

Field	Description
11 BREN	Buffer Read Enable: This status bit is used for non-DMA read transfers. The ESDHC may implement multiple buffers to transfer data efficiently. This read only flag indicates that valid data exists in the host side buffer. If this bit is high, valid data greater than the watermark level exist in the buffer. A change of this bit from 1 to 0 occurs when any read from the buffer is made. A change of this bit from 0 to1 occurs when there is enough valid data ready in the buffer and the Buffer Read Ready interrupt has been generated and enabled. 1 Read enable 0 Read disable
10 BWEN	Buffer Write Enable: This status bit is used for non-DMA write transfers. The ESDHC can implement multiple buffers to transfer data efficiently. This read only flag indicates if space is available for write data. If this bit is 1, valid data greater than the watermark level can be written to the buffer. A change of this bit from 1 to 0 occurs when any write to the buffer is made. A change of this bit from 0 to 1 occurs when the buffer can hold valid data greater than the write watermark level and the Buffer Write Ready interrupt is generated and enabled. 1 Write enable 0 Write disable
9 RTA	Read Transfer Active: This status bit is used for detecting completion of a read transfer. This bit is set for either of the following conditions: • After the end bit of the read command. • When writing a 1 to the Continue Request bit in the Protocol Control register to restart a read transfer. A Transfer Complete interrupt is generated when this bit changes to 0. This bit is cleared for either of the following conditions: • When the last data block as specified by block length is transferred to the System, that is, all data are read away from ESDHC internal buffer. • When all valid data blocks have been transferred from ESDHC internal buffer to the System and no current block transfers are being sent as a result of the Stop At Block Gap Request being set to 1. 1 Transferring data 0 No valid data
8 WTA	Write Transfer Active: This status bit indicates a write transfer is active. If this bit is 0, it means no valid write data exists in the ESDHC. This bit is set in either of the following cases: • After the end bit of the write command. • When writing 1 to the Continue Request bit in the Protocol Control register to restart a write transfer. This bit is cleared in either of the following cases: • After getting the CRC status of the last data block as specified by the transfer count (Single and Multiple). • After getting the CRC status of any block where data transmission is about to be stopped by a Stop At Block Gap Request.

Table continues on the next page...

ESDHCV2x_PRSSTAT field descriptions (continued)

Field	Description
	During a write transaction, a Block Gap Event interrupt is generated when this bit is changed to 0, as result of the Stop At Block Gap Request being set. This status is useful for the Host Driver in determining when to issue commands during Write Busy state. 1 Transferring data 0 No valid data
7 SDOFF	SD Clock Gated Off Internally: This status bit indicates that the SD Clock is internally gated off, because of buffer over/under-run or read pause without read wait assertion, or the driver has cleared SDCLKEN bit to stop the SD clock. This bit is for the Host Driver to debug data transaction on the SD bus. 1 SD Clock is gated off 0 SD Clock is active
6 PEROFF	ipg_perclk Gated Off Internally: This status bit indicates that the ipg_perclk is internally gated off. This bit is for the Host Driver to debug transaction on the SD bus. When INITA bit is set, ESDHC sending 80 clock cycles to the card, the SDCLKEN bit must be '1' to enable the output card clock, otherwise the ipg_perclk will never be gate off, so ipg_perclk and ipg_clk will be always active. 1 ipg_perclk is gated off 0 ipg_perclk is active
5 HCKOFF	hclk Gated Off Internally: This status bit indicates that the hclk is internally gated off. This bit is for the Host Driver to debug during a data transfer. 1 hclk is gated off 0 hclk is active
4 IPGOFF	ipg_clk Gated Off Internally: This status bit indicates that the ipg_clk is internally gated off. This bit is for the Host Driver to debug. 1 ipg_clk is gated off 0 ipg_clk is active
3 SDSTB	SD Clock Stable This status bit indicates that the internal card clock is stable. This bit is for the Host Driver to poll clock status when changing the clock frequency. It is recommended to clear SDCLKEN bit in System Control register to remove glitch on the card clock when the frequency is changing. 1 clock is stable 0 clock is changing frequency and not stable
2 DLA	Data Line Active This status bit indicates whether one of the DAT lines on the SD Bus is in use. In the case of read transactions: This status indicates if a read transfer is executing on the SD Bus. Changes in this value from 1 to 0, between data blocks, generates a Block Gap Event interrupt in the Interrupt Status register. This bit will be set in either of the following cases:

Table continues on the next page...

ESDHCV2x_PRSSTAT field descriptions (continued)

Field	Description
	• After the end bit of the read command. • When writing a 1 to the Continue Request bit in the Protocol Control register to restart a read transfer. This bit will be cleared in either of the following cases: (1) When the end bit of the last data block is sent from the SD Bus to the ESDHC. (2) When the Read Wait state is stopped by a Suspend command and the DAT2 line is released. The ESDHC will wait at the next block gap by driving Read Wait at the start of the interrupt cycle. If the Read Wait signal is already driven (data buffer cannot receive data), the ESDHC can wait for a current block gap by continuing to drive the Read Wait signal. It is necessary to support Read Wait in order to use the suspend / resume function. This bit will remain 1 during Read Wait. In the case of write transactions: This status indicates that a write transfer is executing on the SD Bus. Changes in this value from 1 to 0 generate a Transfer Complete interrupt in the Interrupt Status register. This bit will be set in either of the following cases: • After the end bit of the write command. • When writing to 1 to the Continue Request bit in the Protocol Control register to continue a write transfer. This bit will be cleared in either of the following cases: • When the SD card releases Write Busy of the last data block, the ESDHC will also detect if the output is not busy. If the SD card does not drive the busy signal after the CRC status is received, the ESDHC will assume the card drive "Not Busy". • When the SD card releases write busy, prior to waiting for write transfer, and as a result of a Stop At Block Gap Request. 1 DAT Line Active 0 DAT Line Inactive
1 CDIHB	Command Inhibit (DAT): This status bit is generated if either the DAT Line Active or the Read Transfer Active is set to 1. If this bit is 0, it indicates that the ESDHC can issue the next SD/MMC Command. Commands with a busy signal belong to Command Inhibit (DAT) (for example, R1b, R5b type). Except in the case when the command busy is finished, changing from 1 to 0 generates a Transfer Complete interrupt in the Interrupt Status register. **NOTE:** The SD Host Driver can save registers for a suspend transaction after this bit has changed from 1 to 0. 1 Cannot issue command which uses the DAT line 0 Can issue command which uses the DAT line
0 CIHB	Command Inhibit (CMD): If this status bit is 0, it indicates that the CMD line is not in use and the ESDHC can issue a SD/MMC Command using the CMD line. This bit is set also immediately after the Transfer Type register is written. This bit is cleared when the command response is received. Even if the Command Inhibit (DAT) is set to 1, Commands using only the CMD line can be issued if this bit is 0. Changing from 1 to 0 generates a Command Complete interrupt in the Interrupt Status register. If the ESDHC cannot issue the command because of a command conflict

Table continues on the next page...

ESDHCV2x_PRSSTAT field descriptions (continued)

Field	Description
	error (Refer to Command CRC Error) or because of a Command Not Issued By Auto CMD12 Error, this bit will remain 1 and the Command Complete is not set. The Status of issuing an Auto CMD12 does not show on this bit. 1 Cannot issue command 0 Can issue command using only CMD line

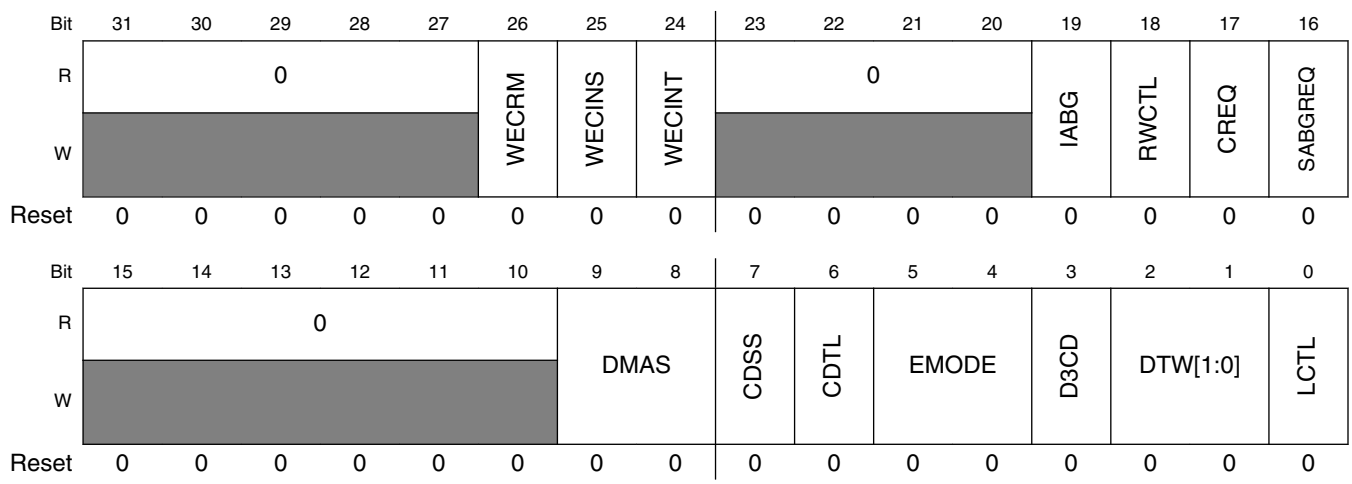
29.7.11 Protocol Control (ESDHCV2x_PROCTL)

There are three cases to restart the transfer after stop at the block gap. The appropriate case depends on whether ESDHC issues a Suspend command or the SD card accepts the Suspend command.

1. If the Host Driver does not issue a Suspend command, the Continue Request will be used to restart the transfer.
2. If the Host Driver issues a Suspend command and the SD card accepts it, a Resume command will be used to restart the transfer.
3. If the Host Driver issues a Suspend command and the SD card does not accept it, the Continue Request will be used to restart the transfer.

Any time Stop At Block Gap Request stops the data transfer, the Host Driver will wait for a Transfer Complete (in the Interrupt Status register), before attempting to restart the transfer. When restarting the data transfer by Continue Request, the Host Driver will clear the Stop At Block Gap Request before or simultaneously.

Addresses: ESDHCV2-1_PROCTL is 5000_4000h base + 28h offset = 5000_4028h
ESDHCV2-2_PROCTL is 5000_8000h base + 28h offset = 5000_8028h
ESDHCV2-4_PROCTL is 5002_4000h base + 28h offset = 5002_4028h



ESDHCV2x_PROCTL field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26 WECRM	Wakeup Event Enable On SD Card Removal: This bit enables a wakeup event, through a Card Removal, in the Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not effect this bit. When this bit is set, the Card Removal Status and the ESDHC interrupt can be asserted without SD_CLK toggling. When the wakeup feature is not enabled, the SD_CLK must be active in order to assert the Card Removal Status and the ESDHC interrupt. 1 Enable 0 Disable
25 WECINS	Wakeup Event Enable On SD Card Insertion: This bit enables a wakeup event, through a Card Insertion, in the Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not effect this bit. When this bit is set, the Card Insertion Status and the ESDHC interrupt can be asserted without SD_CLK toggling. When the wakeup feature is not enabled, the SD_CLK must be active in order to assert the Card Insertion Status and the ESDHC interrupt. 1 Enable 0 Disable
24 WECINT	Wakeup Event Enable On Card Interrupt: This bit enables a wakeup event, through a Card Interrupt, in the Interrupt Status register. This bit can be set to 1 if FN_WUS (Wake Up Support) in CIS is set to 1. When this bit is set, the Card Interrupt Status and the ESDHC interrupt can be asserted without SD_CLK toggling. When the wakeup feature is not enabled, the SD_CLK must be active in order to assert the Card Interrupt Status and the ESDHC interrupt. 1 Enable 0 Disable
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved
19 IABG	Interrupt At Block Gap: This bit is valid only in 4-bit mode, of the SDIO card, and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If the SDIO card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0 to avoid an inadvertent interrupt. When the Host Driver detects an SDIO card insertion, it sets this bit according to the CCCR of the card. 1 Enabled 0 Disabled
18 RWCTL	Read Wait Control: The read wait function is optional for SDIO cards. If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using the DAT[2] line. Otherwise the ESDHC has to stop the SD Clock to hold read data, which restricts commands generation. When the Host Driver detects an SDIO card insertion, it sets this bit according to the CCCR of the card. If the card does not support read wait, this bit will never be set to 1, otherwise DAT line conflicts may occur. If this bit is set to 0, stop at block gap during read operation is also supported, but the ESDHC will stop the SD Clock to pause reading operation.

Table continues on the next page...

ESDHCV2x_PROCTL field descriptions (continued)

Field	Description
	1 Enable Read Wait Control, and assert Read Wait without stopping SD Clock at block gap when SABGREQ bit is set 0 Disable Read Wait Control, and stop SD Clock at block gap when SABGREQ bit is set
17 CREQ	Continue Request: This bit is used to restart a transaction which was stopped using the Stop At Block Gap Request. When a Suspend operation is not accepted by the card, it is also by setting this bit to restart the paused transfer. To cancel stop at the block gap, set Stop At Block Gap Request to 0 and set this bit to 1 to restart the transfer. The ESDHC automatically clears this bit, therefore it is not necessary for the Host Driver to set this bit to 0. If both Stop At Block Gap Request and this bit are 1, the continue request is ignored. 1 Restart 0 No effect
16 SABGREQ	Stop At Block Gap Request: This bit is used to stop executing a transaction at the next block gap for both DMA and non-DMA transfers. Until the Transfer Complete is set to 1, indicating a transfer completion, the Host Driver will leave this bit set to 1. Clearing both the Stop At Block Gap Request and Continue Request does not cause the transaction to restart. Read Wait is used to stop the read transaction at the block gap. The ESDHC will honor the Stop At Block Gap Request for write transfers, but for read transfers it requires that the SDIO card support Read Wait. Therefore, the Host Driver will not set this bit during read transfers unless the SDIO card supports Read Wait and has set the Read Wait Control to 1, otherwise the ESDHC will stop the SD bus clock to pause the read operation during block gap. In the case of write transfers in which the Host Driver writes data to the Data Port register, the Host Driver will set this bit after all block data is written. If this bit is set to 1, the Host Driver will not write data to the Data Port register after a block is sent. Once this bit is set, the Host Driver will not clear this bit before the Transfer Complete bit in Interrupt Status Register is set, otherwise the ESDHCs behavior is undefined. This bit effects Read Transfer Active, Write Transfer Active, DAT Line Active and Command Inhibit (DAT) in the Present State register. 1 Stop 0 Transfer
15–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–8 DMAS	DMA Select: This field is valid while DMA (SDMA or ADMA) is enabled and selects the DMA operation. 00 No DMA or Simple DMA is selected 01 ADMA1 is selected 10 ADMA2 is selected 11 reserved
7 CDSS	Card Detect Signal Selection: This bit selects the source for the card detection. 1 Card Detection Test Level is selected (for test purpose) 0 Card Detection Level is selected (for normal purpose)
6 CDTL	Card Detect Test Level: This is bit is enabled while the Card Detection Signal Selection is set to 1 and it indicates card insertion.

Table continues on the next page...

ESDHCV2x_PROCTL field descriptions (continued)

Field	Description
	1 Card Detect Test Level is 1, card inserted 0 Card Detect Test Level is 0, no card inserted
5–4 EMODE	Endian Mode: The ESDHC supports all four endian modes in data transfer. Refer to Data Buffer for more details. 00 Big Endian Mode 01 Half Word Big Endian Mode 10 Little Endian Mode 11 Reserved
3 D3CD	DAT3 as Card Detection Pin: If this bit is set, DAT3 should be pulled down to act as a card detection pin. Be cautious when using this feature, because DAT3 is also a chip-select for the SPI mode. A pull-down on this pin and CMD0 may set the card into the SPI mode, which the ESDHC does not support. In case of SDIO application, if SD_CD signal is not multiplexed to ESDHC block, S/W should set D3CD bit. 1 DAT3 as Card Detection Pin 0 DAT3 does not monitor Card Insertion
2–1 DTW[1:0]	Data Transfer Width: This bit selects the data width of the SD bus for a data transfer. The Host Driver sets it to match the data width of the card. Possible Data transfer Width is 1-bit, 4-bits or 8-bits. 10 8-bit mode 01 4-bit mode 00 1-bit mode 11 Reserved
0 LCTL	LED Control: This bit, fully controlled by the Host Driver, is used to caution the user not to remove the card while the card is being accessed. If the software is going to issue multiple SD commands, this bit can be set during all these transactions. It is not necessary to change for each transaction. When the software issues multiple SD commands, setting the bit once before the first command is sufficient; it is not necessary to reset the bit between commands. 1 LED on 0 LED off

29.7.12 System Control (ESDHCV2x_SYSCTL)

Addresses: ESDHCV2-1_SYSCTL is 5000_4000h base + 2Ch offset = 5000_402Ch
ESDHCV2-2_SYSCTL is 5000_8000h base + 2Ch offset = 5000_802Ch
ESDHCV2-4_SYSCTL is 5002_4000h base + 2Ch offset = 5002_402Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				INITA	0	0	0	0				DTOCV			
W						RSTD	RSTC	RSTA								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SDCLKFS								DVS[3:0]				SDCLKEN	PEREN	HCKEN	IPGEN
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

ESDHCV2x_SYSCTL field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved
27 INITA	Initialization Active: When this bit is set, 80 SD-Clocks are sent to the card. After the 80 clocks are sent, this bit is self cleared. This bit is very useful during the card power-up period when 74 SD-Clocks are needed and the clock auto gating feature is enabled. Writing 1 to this bit when this bit is already 1 has no effect. Writing 0 to this bit at any time has no effect. When either of the CIHB and CDIHB bits in the Present State Register are set, writing 1 to this bit is ignored (that is, when command line or data lines are active, write to this bit is not allowed). On the other hand, when this bit is set, that is, during initialization active period, it is allowed to issue commands, and the command bit stream will appear on the CMD pad after all 80 clock cycles are completer. When this command ends, the driver can make sure that the 80 clock cycles are sent out, which is very useful when the driver needs to send 80 cycles to the card and does not want to wait until this bit is self cleared.
26 RSTD	Software Reset For DAT Line: Only part of the data circuit is reset. DMA circuit is also reset. The following registers and bits are cleared by this bit: • Data Port register • Buffer is cleared and initialized.Present State register • Buffer Read Enable • Buffer Write Enable • Read Transfer Active • Write Transfer Active • DAT Line Active • Command Inhibit (DAT) Protocol Control register • Continue Request

Table continues on the next page...

ESDHCV2x_SYSCTL field descriptions (continued)

Field	Description
	• Stop At Block Gap Request Interrupt Status register • Buffer Read Ready • Buffer Write Ready • DMA Interrupt • Block Gap Event • Transfer Complete 1 Reset 0 No Reset
25 RSTC	Software Reset For CMD Line: Only part of the command circuit is reset. The following registers and bits are cleared by this bit: • Present State register Command Inhibit (CMD) • Interrupt Status register Command Complete 1 Reset 0 No Reset
24 RSTA	Software Reset For ALL: This reset effects the entire Host Controller except for the card detection circuit. Register bits of type ROC, RW, RW1C, RWAC are cleared. During its initialization, the Host Driver will set this bit to 1 to reset the ESDHC. The ESDHC will reset this bit to 0 when the capabilities registers are valid and the Host Driver can read them. Additional use of Software Reset For All does not affect the value of the Capabilities registers. After this bit is set, it is recommended that the Host Driver reset the external card and re-initialize it. b Reset 0 No Reset
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved
19–16 DTOCV	Data Timeout Counter Value: This value determines the interval by which DAT line timeouts are detected. Refer to the Data Timeout Error bit in the Interrupt Status register for information on factors that dictate time-out generation. Time-out clock frequency will be generated by dividing the base clock SDCLK value by this value. The Host Driver can clear the Data Timeout Error Status Enable (in the Interrupt Status Enable register) to prevent inadvertent time-out events. 1111 Reserved 1110 SDCLK x 2^{27} 0001 SDCLK x 2^{14} 0000 SDCLK x 2^{13}
15–8 SDCLKFS	SDCLK Frequency Select: This register is used to select the frequency of the SDCLK pin. The frequency is not programmed directly, rather this register holds the prescaler (this register) and divisor (next register) of the Base Clock Frequency register. Setting 00h bypasses the frequency prescaler of the SD Clock. Multiple bits must not be set, or the behavior of this prescaler is undefined. The two default divider values can be calculated by the frequency of ipg_perclk and the following Divisor bits.

Table continues on the next page...

ESDHCV2x_SYSCTL field descriptions (continued)

Field	Description
	The frequency of SDCLK is set by the following formula: Clock Frequency = (Base Clock) / (prescaler x divisor) For example, if the Base Clock Frequency is 96 MHz, and the target frequency is 25 MHz, then choosing the prescaler value of 01h and divisor value of 1h will yield 24 MHz, which is the nearest frequency less than or equal to the target. Similarly, to approach a clock value of 400 kHz, the prescaler value of 08h and divisor value of eh yields the exact clock value of 400 kHz. The reset value of this bit field is 80h, so if the input Base Clock (ipg_perclk) is about 96 MHz, the default SD Clock after reset is 375 kHz. According to the SD Physical Specification Version 1.1 and the SDIO Card Specification Version 1.2, the maximum SD Clock frequency is 50 MHz and will never exceed this limit. Only the following settings are allowed: 0x80 Base clock divided by 256 0x40 Base clock divided by 128 0x20 Base clock divided by 64 0x10 Base clock divided by 32 0x08 Base clock divided by 16 0x04 Base clock divided by 8 0x02 Base clock divided by 4 0x01 Base clock divided by 2 0x00 Base clock (10 MHz-63 MHz)
7–4 DVS[3:0]	Divisor: This register is used to provide a more exact divisor to generate the desired SD clock frequency. **NOTE:** The divider can even support odd divisor without deterioration of duty cycle. The setting are as following: 0x0 Divisor by 1 0x1 Divisor by 2 0xe Divisor by 15 0xf Divisor by 16
3 SDCLKEN	SD Clock Enable The Host Controller will stop SDCLK when writing this bit to 0. SDCLK Frequency can be changed when this bit is 0. Then, the Host Controller will maintain the same clock frequency until SDCLK is stopped (Stop at SDCLK=0). If the Card Inserted in the Present State register is cleared, this bit should be cleared by the Host Driver to save power.
2 PEREN	Peripheral Clock Enable: If this bit is set, ipg_perclk will always be active and no automatic gating is applied. Thus the SDCLK is active except when auto gating-off during buffer danger (buffer about to over-run or under-run). When this bit is cleared, the ipg_perclk will be automatically off when there is no transaction on the SD bus. As this bit is only a feature enabling bit, clearing this bit does not stop SDCLK immediately. The ipg_perclk will be internally gated off, if none of the following factors are met: • The cmd part is reset, or • Data part is reset, or

Table continues on the next page...

ESDHCV2x_SYSCTL field descriptions (continued)

Field	Description
	• A soft reset, or • The cmd is about to send, or • Clock divisor is just updated, or • Continue request is just set, or • This bit is set, or • Card insertion is detected, or • Card removal is detected, or • Card external interrupt is detected, or • 80 clocks for initialization phase is ongoing 1 ipg_perclk will not be automatically gated off 0 ipg_perclk will be internally gated off
1 HCKEN	HCLK Enable: If this bit is set, hclk will always be active and no automatic gating is applied. When this bit is cleared, hclk will be automatically off when no data transfer is on the SD bus. 1 hclk will not be automatically gated off 0 hclk will be internally gated off
0 IPGEN	IPG Clock Enable: If this bit is set, ipg_clk will always be active and no automatic gating is applied. The ipg_clk will be internally gated off, if none of the following factors are met: • The cmd part is reset, or • Data part is reset, or • Soft reset, or • The cmd is about to send, or • Clock divisor is just updated, or • Continue request is just set, or • This bit is set, or • Card insertion is detected, or • Card removal is detected, or • Card external interrupt is detected, or • The ipg_perclk is not gated off **NOTE:** The ipg_clk will not be auto gated off if the ipg_perclk is not gated off. Clearing only this bit has no effect unless the PEREN bit is also cleared. 1 ipg_clk will not be automatically gated off 0 ipg_clk will be internally gated off

29.7.13 Interrupt Status (ESDHCV2x_IRQSTAT)

An interrupt is generated when the Normal Interrupt Signal Enable is enabled and at least one of the status bits is set to 1. For all bits, writing 1 to a bit clears it; writing to 0 keeps the bit unchanged. More than one status can be cleared with a single register write. Before clearing the Card Interrupt, make sure that the card stops asserting the interrupt, that is, the Card Driver stops servicing the interrupt condition. Otherwise, the CINT bit will be asserted again.

Interrupt Status (ESDHCV2_IRQSTAT) below shows the relationship between the Command Timeout Error and the Command Complete.

Table 29-54. ESDHC Status for Command Timeout Error/Command Complete Bit Combinations

Command Complete	Command Timeout Error	Meaning of the Status
0	0	X
X	1	Response not received within 64 SDCLK cycles
1	0	Response received

Table below shows the relationship between the Transfer Complete and the Data Timeout Error.

Table 29-55. ESDHC Status for Data Timeout Error/Transfer Complete Bit Combinations

Transfer Complete	Data Timeout Error	Meaning of the Status
0	0	X
0	1	Timeout occurred during transfer
1	X	Data Transfer Complete

Table below shows the relationship between the Command CRC Error and Command Timeout Error.

Table 29-56. ESDHC Status for Command CRC Error/Command Timeout Error Bit Combinations

Command Complete	Command Timeout Error	Meaning of the Status
0	0	No error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	CMD line conflict

Addresses: ESDHCV2-1_IRQSTAT is 5000_4000h base + 30h offset = 5000_4030h
ESDHCV2-2_IRQSTAT is 5000_8000h base + 30h offset = 5000_8030h
ESDHCV2-4_IRQSTAT is 5002_4000h base + 30h offset = 5002_4030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			DMAE	0			AC12E	0	DEBE	DCE	DTOE	CIE	CEBE	CCE	CTOE
W				w1c				w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CINT	CRM	CINS	BRR	BWR	DINT	BGE	TC	CC
W								w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDHCV2x_IRQSTAT field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28 DMAE	DMA Error: Occurs when an Internal DMA transfer has failed. This bit is set to 1, when some error occurs in the data transfer. This error can be caused by either Simple DMA or ADMA, depending on which DMA is in use. The value in DMA System Address register is the next fetch address where the error occurs. As any error corrupts the whole data block, the Host Driver will re-start the transfer from the corrupted block boundary. The address of the block boundary can be calculated either from the current DS_ADDR value or from the remaining number of blocks and the block size. 1 Error 0 No Error
27–25 Reserved	This read-only field is reserved and always has the value zero. Reserved
24 AC12E	Auto CMD12 Error: Occurs when detecting that one of the bits in the Auto CMD12 Error Status register has changed from 0 to 1. This bit is set to 1, not only when the errors in Auto CMD12 occur, but also when the Auto CMD12 is not executed due to the previous command error. 1 Error 0 No Error
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22 DEBE	Data End Bit Error: Occurs either when detecting 0 at the end bit position of read data, which uses the DAT line, or at the end bit position of the CRC.

Table continues on the next page...

ESDHCV2x_IRQSTAT field descriptions (continued)

Field	Description
	1 Error 0 No Error
21 DCE	Data CRC Error: Occurs when detecting a CRC error when transferring read data, which uses the DAT line, or when detecting the Write CRC status having a value other than 010. 1 Error 0 No Error
20 DTOE	Data Timeout Error: Occurs when detecting one of following time-out conditions. • Busy time-out for R1b,R5b type • Busy time-out after Write CRC status • Read Data time-out. 1 Time out 0 No Error
19 CIE	Command Index Error: Occurs if a Command Index error occurs in the command response. 1 Error 0 No Error
18 CEBE	Command End Bit Error: Occurs when detecting that the end bit of a command response is 0. 1 End Bit Error Generated 0 No Error
17 CCE	Command CRC Error: Command CRC Error is generated in two cases. • If a response is returned and the Command Timeout Error is set to 0 (indicating no time-out), this bit is set when detecting a CRC error in the command response. • The ESDHC detects a CMD line conflict by monitoring the CMD line when a command is issued. If the ESDHC drives the CMD line to 1, but detects 0 on the CMD line at the next SDCLK edge, then the ESDHC will abort the command (Stop driving CMD line) and set this bit to 1. The Command Timeout Error will also be set to 1 to distinguish CMD line conflict. 1 CRC Error Generated. 0 No Error
16 CTOE	Command Timeout Error: Occurs only if no response is returned within 64 SDCLK cycles from the end bit of the command. If the ESDHC detects a CMD line conflict, in which case a Command CRC Error will also be set (as shown in Interrupt Status (ESDHCV2_IRQSTAT)), this bit will be set without waiting for 64 SDCLK cycles. Because the command will be aborted by the ESDHC. 1 Time out 0 No Error

Table continues on the next page...

ESDHCV2x_IRQSTAT field descriptions (continued)

Field	Description
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 CINT	Card Interrupt: This status bit is set when an interrupt signal is detected from the external card. In 1-bit mode, the ESDHC will detect the Card Interrupt without the SD Clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so the interrupt from card can only be sampled during interrupt cycle, introducing some delay between the interrupt signal from the SDIO card and the interrupt to the Host System. Writing 1 to this bit will clear it, but as the interrupt factor from the SDIO card is not clear, this bit is set again. In order to clear this bit, reset the interrupt factor from the external card and then clear this bit. When this status bit is set, and the Host Driver needs to service this interrupt, the Card Interrupt Signal Enable in the Interrupt Signal Enable register should be cleared to deactivate the interrupt signal to the Host System. After completion of the card interrupt service, (it should reset the interrupt factors in the SDIO card and the interrupt signal should not be asserted), write 1 to clear this bit, enable the Card Interrupt Signal Enable and start sampling the interrupt signal again. 1 Generate Card Interrupt 0 No Card Interrupt
7 CRM	Card Removal: This status bit is set if the Card Inserted bit (PRSSTNT[CINS]) changes from 1 to 0. When the Host Driver writes this bit to 1 to clear this status, the status of CINS should be confirmed, because the card state may possibly change when the Host Driver clears this bit and the interrupt event may not be generated. When this bit is cleared, it will be set again if no card is inserted. In order to leave it cleared, clear the Card Removal Status Enable bit IRQSTATEN[CRMSEN]in Interrupt Status Enable register. 1 Card removed 0 Card state unstable or inserted
6 CINS	Card Insertion: This status bit is set if the Card Inserted bit in the Present State register changes from 0 to 1. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card state may possibly be changed when the Host Driver clears this bit and the interrupt event may not be generated. When this bit is cleared, it will be set again if a card is inserted. In order to leave it cleared, clear the Card Inserted Status Enable bit in Interrupt Status Enable register. 1 Card inserted 0 Card state unstable or removed
5 BRR	Buffer Read Ready: This status bit is set if the Buffer Read Enable bit, in the Present State register, changes from 0 to 1. Refer to the Buffer Read Enable bit in the Present State register for additional information. 1 Ready to read buffer 0 Not ready to read buffer
4 BWR	Buffer Write Ready: This status bit is set if the Buffer Write Enable bit, in the Present State register, changes from 0 to 1. Refer to the Buffer Write Enable bit in the Present State register for additional information. 1 Ready to write buffer: 0 Not ready to write buffer

Table continues on the next page...

ESDHCV2x_IRQSTAT field descriptions (continued)

Field	Description
3 DINT	DMA Interrupt: Occurs only when the internal DMA finishes the data transfer successfully. When errors occur during data transfer, this bit will not be set. Instead, the DMAE bit will be set. Either Simple DMA or ADMA finishes data transferring, this bit will be set. 1 DMA Interrupt is generated 0 No DMA Interrupt
2 BGE	Block Gap Event: If the Stop At Block Gap Request bit in the Protocol Control register is set, this bit is set when a read or write transaction is stopped at a block gap. If Stop At Block Gap Request is not set to 1, this bit is not set to 1. In the case of a Read Transaction: This bit is set at the falling edge of the DAT Line Active Status (When the transaction is stopped at SD Bus timing). The Read Wait must be supported in order to use this function. In the case of Write Transaction: This bit is set at the falling edge of Write Transfer Active Status (After getting CRC status at SD Bus timing). 1 Transaction stopped at block gap 0 No block gap event
1 TC	Transfer Complete: This bit is set when a read or write transfer is completed. In the case of a Read Transaction: This bit is set at the falling edge of the Read Transfer Active Status. There are two cases in which this interrupt is generated. The first is when a data transfer is completed as specified by the data length (after the last data has been read to the Host System). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request bit in the Protocol Control register (after valid data has been read to the Host System). In the case of a Write Transaction: This bit is set at the falling edge of the DAT Line Active Status. There are two cases in which this interrupt is generated. The first is when the last data is written to the SD card as specified by the data length and the busy signal is released. The second is when data transfers are stopped at the block gap, by setting the Stop At Block Gap Request bit in the Protocol Control register, and the data transfers are completed. (After valid data is written to the SD card and the busy signal released.) 1 Transfer complete 0 Transfer not complete
0 CC	Command Complete: This bit is set when you receive the end bit of the command response (except Auto CMD12). Refer to the Command Inhibit (CMD) in the Present State register. 1 Command complete 0 Command not complete

29.7.14 Interrupt Status Enable (ESDHCV2x_IRQSTATEN)

Setting the bits in this register to 1 enables the corresponding Interrupt Status to be set by the specified event. If any bit is cleared, the corresponding Interrupt Status bit is also cleared (that is, when the bit in this register is cleared, the corresponding bit in Interrupt Status Register is always 0).

- Depending on IABG bit setting, ESDHC may be programmed to sample the card interrupt signal during the interrupt period and hold its value in the flip-flop. There will be some delays on the Card Interrupt, asserted from the card, to the time the Host System is informed.
- To detect a CMD line conflict, the Host Driver must set both Command Timeout Error Status Enable and Command CRC Error Status Enable to 1.

Addresses: ESDHCV2-1_IRQSTATEN is 5000_4000h base + 34h offset = 5000_4034h
ESDHCV2-2_IRQSTATEN is 5000_8000h base + 34h offset = 5000_8034h
ESDHCV2-4_IRQSTATEN is 5002_4000h base + 34h offset = 5002_4034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				0			AC12ESEN	0	DEBESEN	DCESEN	DTOESEN	CIESEN	CEBESEN	CCESEN	CTOESEN
W				-												
Reset	0	0	0	1	0	0	0	1	0	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CINTSEN	CRMSEN	CINSEN	BRRSEN	BWRSEN	DINTSEN	BGESEN	TCSEN	CCSEN
W																
Reset	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1

ESDHCV2x_IRQSTATEN field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28 -	DMA Error Status Enable: 1 Enabled 0 Masked
27–25 Reserved	This read-only field is reserved and always has the value zero. Reserved
24 AC12ESEN	Auto CMD12 Error Status Enable:

Table continues on the next page...

ESDHCV2x_IRQSTATEN field descriptions (continued)

Field	Description
	1 Enabled 0 Masked
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22 DEBESEN	Data End Bit Error Status Enable: 1 Enabled 0 Masked
21 DCESEN	Data CRC Error Status Enable: 1 Enabled 0 Masked
20 DTOESEN	Data Timeout Error Status Enable: 1 Enabled 0 Masked
19 CIESEN	Command Index Error Status Enable: 1 Enabled 0 Masked
18 CEBESEN	Command End Bit Error Status Enable: 1 Enabled 0 Masked
17 CCESEN	Command CRC Error Status Enable: 1 Enabled 0 Masked
16 CTOESEN	Command Timeout Error Status Enable: 1 Enabled 0 Masked
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 CINTSEN	Card Interrupt Status Enable: If this bit is set to 0, the ESDHC will clear the interrupt request to the System. The Card Interrupt detection is stopped when this bit is cleared and restarted when this bit is set to 1. The Host Driver should clear the Card Interrupt Status Enable before servicing the Card Interrupt and should set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts. 1 Enabled 0 Masked
7 CRMSEN	Card Removal Status Enable: 1 Enabled 0 Masked

Table continues on the next page...

ESDHCV2x_IRQSTATEN field descriptions (continued)

Field	Description
6 CINSEN	Card Insertion Status Enable: 1 Enabled 0 Masked
5 BRRSEN	Buffer Read Ready Status Enable: 1 Enabled 0 Masked
4 BWRSEN	Buffer Write Ready Status Enable: 1 Enabled 0 Masked
3 DINTSEN	DMA Interrupt Status Enable: 1 Enabled 0 Masked
2 BGESEN	Block Gap Event Status Enable: 1 Enabled 0 Masked
1 TCSEN	Transfer Complete Status Enable: 1 Enabled 0 Masked
0 CCSEN	Command Complete Status Enable: 1 Enabled 0 Masked

29.7.15 Interrupt Signal Enable (ESDHCV2x_IRQSIGEN)

This register is used to select which interrupt status is indicated to the Host System as the interrupt. These status bits all share the same interrupt line. Setting any of these bits to 1 enables interrupt generation. The corresponding Status register bit will generate an interrupt when the corresponding interrupt signal enable bit is set.

Addresses: ESDHCV2-1_IRQSIGEN is 5000_4000h base + 38h offset = 5000_4038h
ESDHCV2-2_IRQSIGEN is 5000_8000h base + 38h offset = 5000_8038h
ESDHCV2-4_IRQSIGEN is 5002_4000h base + 38h offset = 5002_4038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			DMAEIEN	0			AC12EIEN	0	DEBEIEN	DCEIEN	DTOEIEN	CIEIEN	CEBEIEN	CCEIEN	CTOEIEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CINTIEN	CRMIEN	CINIEN	BRRIEN	BWRIEN	DINTIEN	BGEIEN	TCIEN	CCIEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDHCV2x_IRQSIGEN field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28 DMAEIEN	DMA Error Interrupt Enable: 1 Enable 0 Masked
27–25 Reserved	This read-only field is reserved and always has the value zero. Reserved
24 AC12EIEN	Auto CMD12 Error Interrupt Enable: 1 Enabled 0 Masked
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22 DEBEIEN	Data End Bit Error Interrupt Enable: 1 Enabled 0 Masked

Table continues on the next page...

ESDHCV2x_IRQSIGEN field descriptions (continued)

Field	Description
21 DCEIEN	Data CRC Error Interrupt Enable: 1 Enabled 0 Masked
20 DTOEIEN	Data Timeout Error Interrupt Enable: 1 Enabled 0 Masked
19 CIEIEN	Command Index Error Interrupt Enable: 1 Enabled 0 Masked
18 CEBEIEN	Command End Bit Error Interrupt Enable: 1 Enabled 0 Masked
17 CCEIEN	Command CRC Error Interrupt Enable: 1 Enabled 0 Masked
16 CTOEIEN	Command Timeout Error Interrupt Enable 1 Enabled 0 Masked
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 CINTIEN	Card Interrupt Interrupt Enable: 1 Enabled 0 Masked
7 CRMIEN	Card Removal Interrupt Enable: 1 Enabled 0 Masked
6 CINIEN	Card Insertion Interrupt Enable: 1 Enabled 0 Masked
5 BRRIEN	Buffer Read Ready Interrupt Enable: 1 Enabled 0 Masked
4 BWRIEN	Buffer Write Ready Interrupt Enable: 1 Enabled 0 Masked

Table continues on the next page...

ESDHCV2x_IRQSIGEN field descriptions (continued)

Field	Description
3 DINTIEN	DMA Interrupt Enable: 1 Enabled 0 Masked
2 BGEIEN	Block Gap Event Interrupt Enable: 1 Enabled 0 Masked
1 TCIEN	Transfer Complete Interrupt Enable: 1 Enabled 0 Masked
0 CCIEN	Command Complete Interrupt Enable: 1 Enabled 0 Masked

29.7.16 Auto CMD12 Status (ESDHCV2x_AUTOC12ERR)

When the Auto CMD12 Error Status bit in the Status register is set, the Host Driver will check this register to identify what kind of error the Auto CMD12 indicated. This register is valid only when the Auto CMD12 Error status bit is set.

Table below shows the relationship between the Auto CMGD12 CRC Error and the Auto CMD12 Command Timeout Error.

Table 29-60. Relationship Between Command CRC Error and Command Timeout Error for Auto CMD12

Auto CMD12 CRC Error	Auto CMD12 Timeout Error	Type of Error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	CMD line conflict

Changes in Auto CMD12 Error Status register can be classified in three scenarios:

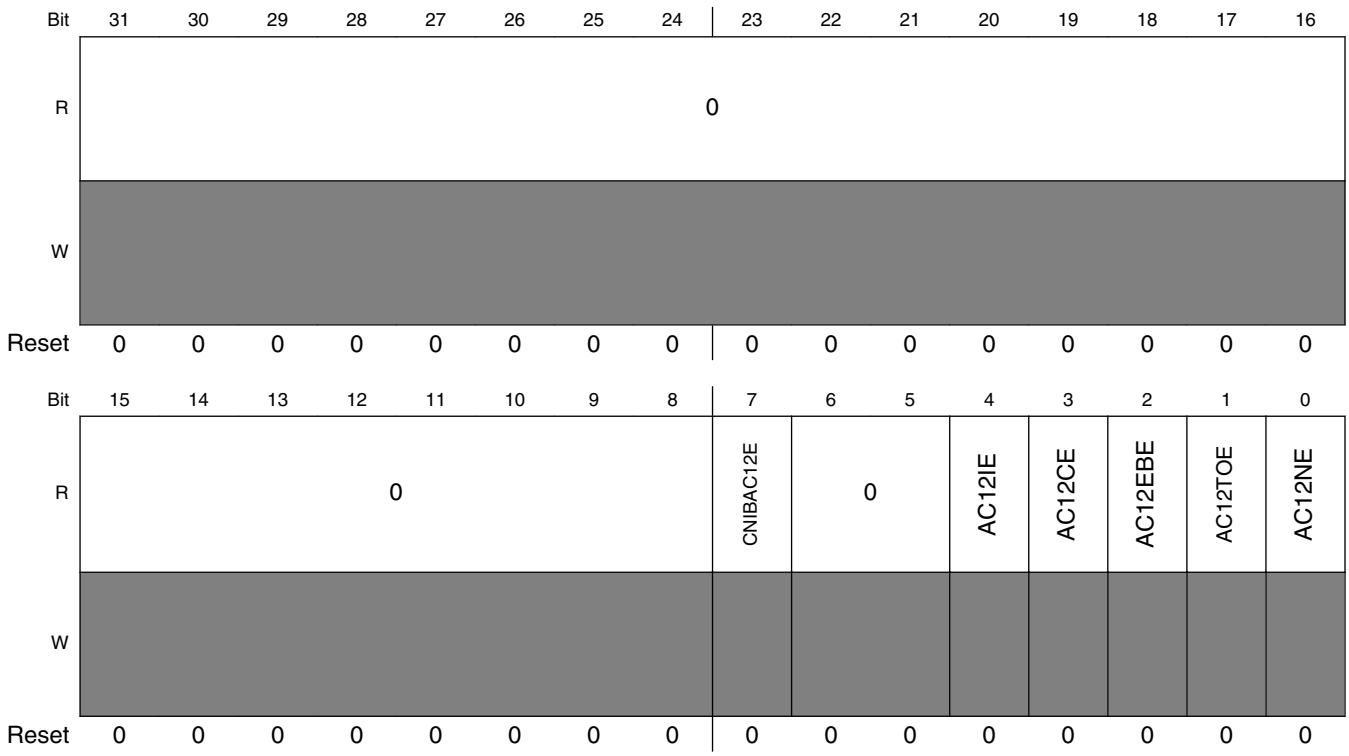
1. When the ESDHC is going to issue an Auto CMD12.
2. Set bit 0 to 1 if the Auto CMD12 cannot be issued due to an error in the previous command
3. Set bit 0 to 0 if the Auto CMD12 is issued
4. At the end bit of an Auto CMD12 response:
   - Check errors correspond to bits 1-4.

- Set bits 1-4 corresponding to detected errors.
- Clear bits 1-4 corresponding to detected errors

5. Before reading the Auto CMD12 Error Status bit 7:
   - Set bit 7 to 1 if there is a command that cannot be issued
   - Clear bit 7 if there is no command to issue

The timing for generating the Auto CMD12 Error and writing to the Command register are asynchronous. After that, bit 7 will be sampled when the driver is not writing to the Command register. It is suggested to read this register only when the AC12E bit in Interrupt Status register is set. An Auto CMD12 Error Interrupt is generated when one of the error bits (0-4) is set to 1. The Command Not Issued By Auto CMD12 Error does not generate an interrupt.

Addresses: ESDHCV2-1_AUTOC12ERR is 5000_4000h base + 3Ch offset = 5000_403Ch
ESDHCV2-2_AUTOC12ERR is 5000_8000h base + 3Ch offset = 5000_803Ch
ESDHCV2-4_AUTOC12ERR is 5002_4000h base + 3Ch offset = 5002_403Ch



ESDHCV2x_AUTOC12ERR field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7 CNIBAC12E	Command Not Issued By Auto CMD12 Error: Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 Error (D04-D01) in this register.

Table continues on the next page...

ESDHCV2x_AUTOC12ERR field descriptions (continued)

Field	Description
	1 Not Issued 0 No error
6–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 AC12IE	Auto CMD12 Index Error: Occurs if the Command Index error occurs in response to a command. 1 Error, the CMD index in response is not CMD12 0 No error
3 AC12CE	Auto CMD12 CRC Error: Occurs when detecting a CRC error in the command response. 1 CRC Error Met in Auto CMD12 Response 0 No CRC error
2 AC12EBE	Auto CMD12 End Bit Error: Occurs when detecting that the end bit of command response is 0 which should be 1. 1 End Bit Error Generated 0 No error
1 AC12TOE	Auto CMD12 Timeout Error: Occurs if no response is returned within 64 SDCLK cycles from the end bit of the command. If this bit is set to1, the other error status bits (2-4) have no meaning. 1 Time out 0 No error
0 AC12NE	Auto CMD12 Not Executed: If memory multiple block data transfer is not started, due to a command error, this bit is not set because it is not necessary to issue an Auto CMD12. Setting this bit to 1 means the ESDHC cannot issue the Auto CMD12 to stop a memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (1-4) have no meaning. 1 Not executed 0 Executed

29.7.17 Host Controller Capabilities (ESDHCV2x_HOSTCAPBLT)

This register provides the Host Driver with information specific to the ESDHC implementation. The value in this register is the power-on-reset value, and does not change with a software reset. Any write to this register is ignored.

Addresses: ESDHCV2-1_HOSTCAPBLT is 5000_4000h base + 40h offset = 5000_4040h
ESDHCV2-2_HOSTCAPBLT is 5000_8000h base + 40h offset = 5000_8040h
ESDHCV2-4_HOSTCAPBLT is 5002_4000h base + 40h offset = 5002_4040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					VS18	VS30	VS33	SRS	DMAS	HSS	ADMAS	0	MBL[2:0]		
W																
Reset	0	0	0	0	0	1	1	1	1	1	1	1	0	0	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDHCV2x_HOSTCAPBLT field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26 VS18	Voltage Support 1.8V: This bit depends on the Host System ability. 1 1.8V supported 0 1.8V not supported
25 VS30	Voltage Support 3.0V: This bit depends on the Host System ability. 1 3.0V supported 0 3.0V not supported
24 VS33	Voltage Support 3.3V: This bit depends on the Host System ability. 1 3.3V supported 0 3.3V not supported

Table continues on the next page...

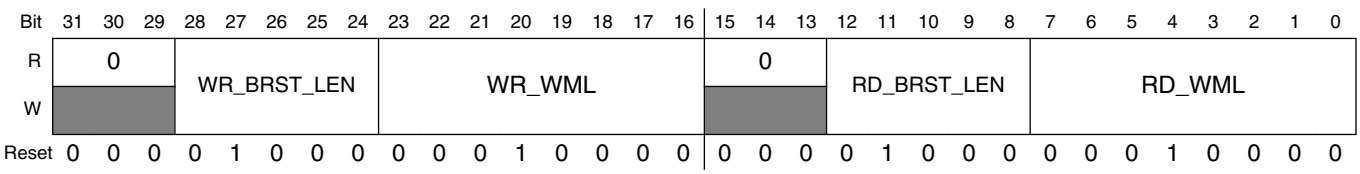
ESDHCV2x_HOSTCAPBLT field descriptions (continued)

Field	Description
23 SRS	Suspend / Resume Support: This bit indicates whether the ESDHC supports Suspend / Resume functionality. If this bit is 0, the Suspend and Resume mechanism, as well as the Read Wait, are not supported, and the Host Driver will not issue either Suspend or Resume commands. 1 Supported 0 Not supported
22 DMAS	DMA Support: This bit indicates whether the ESDHC is capable of using the internal DMA to transfer data between system memory and the data buffer directly. 1 DMA Supported 0 DMA not supported
21 HSS	High Speed Support: This bit indicates whether the ESDHC supports High Speed mode and the Host System can supply a SD Clock frequency from 25 MHz to 50 MHz. 1 High Speed Supported 0 High Speed Not Supported
20 ADMAS	ADMA Support: This bit indicates whether the ESDHC supports the ADMA feature. 1 Advanced DMA Supported 0 Advanced DMA Not supported
19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18–16 MBL[2:0]	Max Block Length: This value indicates the maximum block size that the Host Driver can read and write to the buffer in the ESDHC. The buffer will transfer block size without wait cycles. 000 512 bytes 001 1024 bytes 010 2048 bytes 011 4096 bytes
15–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

29.7.18 Watermark Level (ESDHCV2x_WML)

Both write and read watermark levels (FIFO threshold) are configurable. There value can range from 1 to 128 words. Both write and read burst lengths are also configurable. There value can range from 1 to 31 words.

Addresses: ESDHCV2-1_WML is 5000_4000h base + 44h offset = 5000_4044h
ESDHCV2-2_WML is 5000_8000h base + 44h offset = 5000_8044h
ESDHCV2-4_WML is 5002_4000h base + 44h offset = 5002_4044h



ESDHCV2x_WML field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28–24 WR_BRST_LEN	Write Burst Length:[1] The number of words the ESDHC writes in a single burst. The write burst length must be less than or equal to the write watermark level, and all bursts within a watermark level transfer will be in back-to-back mode. On reset, this field will be 8. Writing 0 to this field will result in '01000' (that is, it is not able to clear this field).
23–16 WR_WML	Write Watermark Level: The number of words used as the watermark level (FIFO threshold) in a DMA write operation. Also the number of words as a sequence of write bursts in back-to-back mode. The maximum legal value for the write watermark level is 128.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12–8 RD_BRST_LEN	Read Burst Length:[2] The number of words the ESDHC reads in a single burst. The read burst length must be less than or equal to the read watermark level, and all bursts within a watermark level transfer will be in back-to-back mode. On reset, this field will be 8. Writing 0 to this field will result in '01000' (that is, it is not able to clear this field.
7–0 RD_WML	Read Watermark Level: The number of words used as the watermark level (FIFO threshold) in a DMA read operation. Also the number of words as a sequence of read bursts in back-to-back mode. The maximum legal value for the read water mark level is 128.

1. Due to system restriction, the actual burst length may not exceed 16.
2. Due to system restriction, the actual burst length may not exceed 16.

29.7.19 Force Event (ESDHCV2x_FEVT)

The Force Event Register is not a physically implemented register. Rather, it is an address at which the Interrupt Status Register can be written if the corresponding bit of the Interrupt Status Enable Register is set. This register is a write only register and writing 0 to it has no effect. Writing 1 to this register sets the corresponding bit of Interrupt Status Register. A read from this register always results in 0's. In order to change the corresponding status bits in the Interrupt Status Register, set IPGEN bit in System Control Register so that ipg_clk is always active.

Forcing a card interrupt will generate a short pulse on the DAT[1] line, and the driver will treat this interrupt as a normal interrupt. The interrupt service routine will skip polling the card interrupt factor as the interrupt is self cleared.

Addresses: ESDHCV2-1_FEVT is 5000_4000h base + 50h offset = 5000_4050h
ESDHCV2-2_FEVT is 5000_8000h base + 50h offset = 5000_8050h
ESDHCV2-4_FEVT is 5002_4000h base + 50h offset = 5002_4050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0		0	0			0	0	0	0	0	0	0	0	0
W	FEVTCINT			FEVTDMAE				FEVTAC12E		FEVTDEBE	FEVTDCE	FEVTDTOE	FEVTCIE	FEVTCEBE	FEVTCCE	FEVTCCE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0	0		0	0	0	0	0
W									FEVTCNIBAC12E			FEVTAC12IE	FEVTAC12EBE	FEVTAC12CE	FEVTAC12TOE	FEVTAC12NE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDHCV2x_FEVT field descriptions

Field	Description
31 FEVTCINT	Force Event Card Interrupt: Writing 1 to this bit generates a short low-level pulse on the internal DAT[1] line, as if a self clearing interrupt was received from the external card. If enabled, the CINT bit will be set and the interrupt service routine will treat this interrupt as a normal interrupt from the external card.
30–29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28 FEVTDMAE	Force Event DMA Error: Forces the DMAE bit of Interrupt Status Register to be set
27–25 Reserved	This read-only field is reserved and always has the value zero. Reserved
24 FEVTAC12E	Force Event Auto Command 12 Error: Forces the AC12E bit of Interrupt Status Register to be set
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22 FEVTDEBE	Force Event Data End Bit Error: Forces the DEBE bit of Interrupt Status Register to be set
21 FEVTDCE	Force Event Data CRC Error: Forces the DCE bit of Interrupt Status Register to be set
20 FEVTDTOE	Force Event Data Time Out Error: Force the DTOE bit of Interrupt Status Register to be set
19 FEVTCIE	Force Event Command Index Error: Forces the CCE bit of Interrupt Status Register to be set
18 FEVTCEBE	Force Event Command End Bit Error: Forces the CEBE bit of Interrupt Status Register to be set
17 FEVTCCE	Force Event Command CRC Error: Forces the CCE bit of Interrupt Status Register to be set
16 FEVTCCE	Force Event Command Time Out Error: Forces the CTOE bit of Interrupt Status Register to be set
15–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7 FEVTCNIBAC12E	Force Event Command Not Executed By Auto Command 12 Error: Forces the CNIBAC12E bit in the Auto Command12 Error Status Register to be set
6–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 FEVTAC12IE	Force Event Auto Command 12 Index Error: Forces the AC12IE bit in the Auto Command12 Error Status Register to be set
3 FEVTAC12EBE	Force Event Auto Command 12 End Bit Error: Forces the AC12EBE bit in the Auto Command12 Error Status Register to be set

Table continues on the next page...

ESDHCV2x_FEVT field descriptions (continued)

Field	Description
2 FEVTAC12CE	Force Event Auto Command 12 CRC Error: Forces the AC12CE bit in the Auto Command12 Error Status Register to be set
1 FEVTAC12TOE	Force Event Auto Command 12 Time Out Error: Forces the AC12TOE bit in the Auto Command12 Error Status Register to be set
0 FEVTAC12NE	Force Event Auto Command 12 Not Executed: Forces the AC12NE bit in the Auto Command12 Error Status Register to be set

29.7.20 ADMA Error Status Register (ESDHCV2x_ADMAES)

When an ADMA Error Interrupt occurrs, the ADMA Error States field (ADMAES) in this register holds the ADMA state and the ADMA System Address register (ADSADDR) holds the address of the error descriptor.

To recover from this error, the Host Driver requires the ADMA state to identify the error descriptor address as follows:

- ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address
- ST_FDS: Current location set in the ADMA System Address register is the error descriptor address
- ST_CADR: This state is never set because it only increments the descriptor pointer and doesn't generate an ADMA error
- ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address

In case of a write operation, the Host Driver should use ACMD22 command (see field descriptions table) to get the number of the written block, because unwritten data may exist in the Host Controller.

The Host Controller generates the ADMA Error Interrupt when it detects invalid descriptor data (Valid=0) in the ST_FDS state. The Host Driver can distinguish this error by reading the Valid bit of the error descriptor.

Table 29-65. ADMA Error State Coding

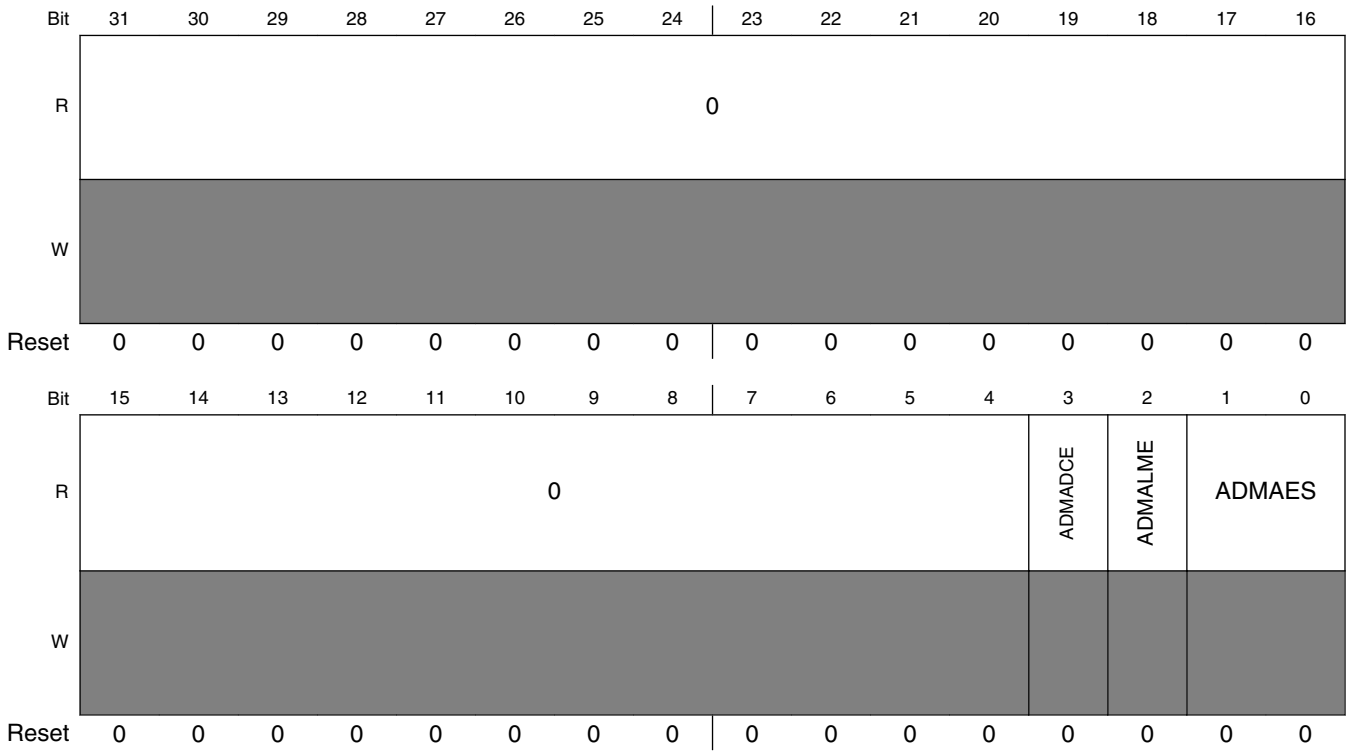
D01-D00	ADMA Error State (when error has occurred)	Contents of ADMA System Address Register
00	ST_STOP (Stop DMA)	Holds the address of the next executable Descriptor command
01	ST_FDS (Fetch Descriptor)	Holds the valid Descriptor address

Table continues on the next page...

Table 29-65. ADMA Error State Coding (continued)

D01-D00	ADMA Error State (when error has occurred)	Contents of ADMA System Address Register
10	ST_CADR (Change Address)	No ADMA Error is generated
11	ST_TFR (Transfer Data)	Holds the address of the next executable Descriptor command

Addresses: ESDHCV2-1_ADMAES is 5000_4000h base + 54h offset = 5000_4054h
ESDHCV2-2_ADMAES is 5000_8000h base + 54h offset = 5000_8054h
ESDHCV2-4_ADMAES is 5002_4000h base + 54h offset = 5002_4054h



ESDHCV2*x*_ADMAES field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value zero. Reserved
3 ADMADCE	ADMA Descritor Error: This error occurs when invalid descriptor fetched by ADMA: 1 Error 0 No Error
2 ADMALME	ADMA Length Mismatch Error: This error occurs in the following 2 cases:

Table continues on the next page...

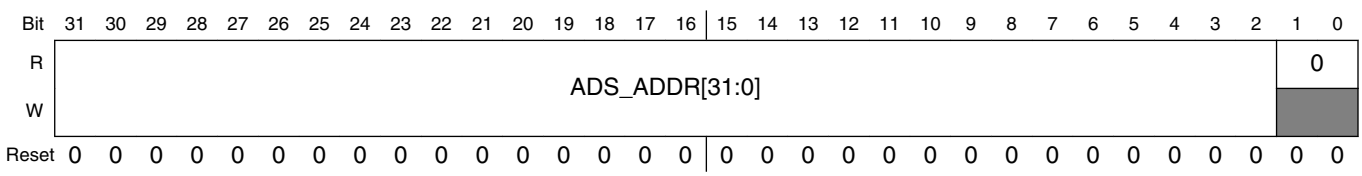
ESDHCV2x_ADMAES field descriptions (continued)

Field	Description
	• While the Block Count Enable is being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length • Total data length can not be divided by the block length 1 Error 0 No Error
1–0 ADMAES	ADMA Error State (when ADMA Error is occurred.): This field indicates the state of the ADMA when an error has occurred during an ADMA data transfer. Refer to ADMA Error Status Register (ESDHCV2_ADMAES) for more details.

29.7.21 ADMA System Address (ESDHCV2x_ADSADDR)

This register contains the physical system memory address used for ADMA transfers.

Addresses: ESDHCV2-1_ADSADDR is 5000_4000h base + 58h offset = 5000_4058h
ESDHCV2-2_ADSADDR is 5000_8000h base + 58h offset = 5000_8058h
ESDHCV2-4_ADSADDR is 5002_4000h base + 58h offset = 5002_4058h



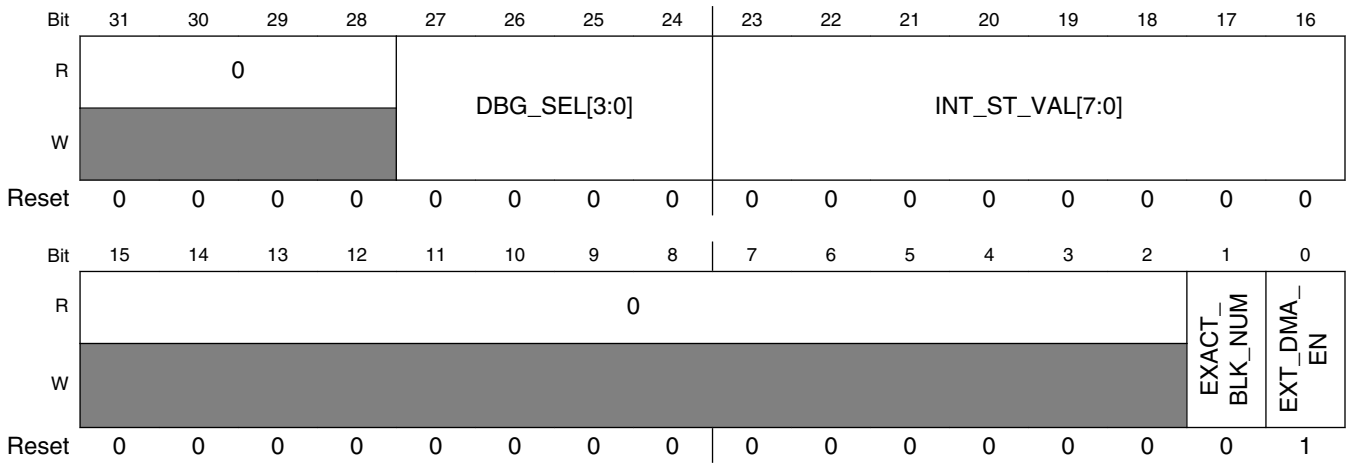
ESDHCV2x_ADSADDR field descriptions

Field	Description
31–2 ADS_ ADDR[31:0]	ADMA System Address: This register holds the word address of the executing command in the Descriptor table. At the start of ADMA, the Host Driver will set the start address of the Descriptor table. The ADMA engine increments this register address whenever fetching a Descriptor command. When the ADMA is stopped at the Block Gap, this register indicates the address of the next executable Descriptor command. When the ADMA Error Interrupt is generated, this register will hold the valid Descriptor address depending on the ADMA state. The lower 2 bits of this register is tied to '0' so the ADMA address is always word aligned. As this register supports dynamic address reflecting, when TC bit is set, it automatically alters the value of internal address counter, so this register value cannot be changed when the TC bit is set. Such restriction is also listed in Software Restrictions.
1–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

29.7.22 Vendor Specific Register (ESDHCV2x_VENDOR)

This register contains the vendor specific control/status register.

Addresses: ESDHCV2-1_VENDOR is 5000_4000h base + C0h offset = 5000_40C0h
ESDHCV2-2_VENDOR is 5000_8000h base + C0h offset = 5000_80C0h
ESDHCV2-4_VENDOR is 5002_4000h base + C0h offset = 5002_40C0h



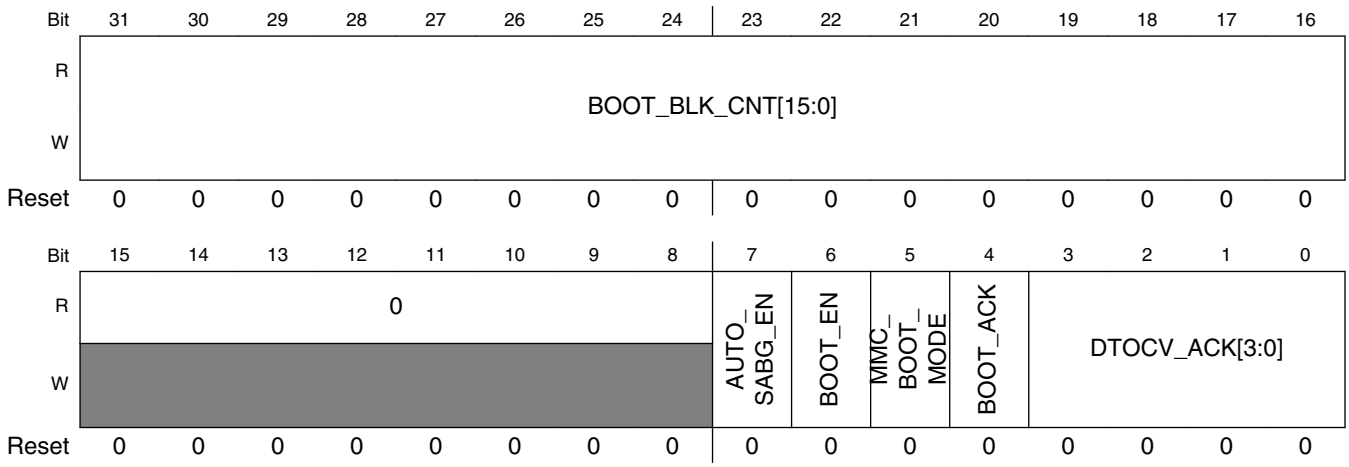
ESDHCV2x_VENDOR field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved
27–24 DBG_SEL[3:0]	Debug Select Select the internal sub-block to show its internal state value.
23–16 INT_ST_VAL[7:0]	Internal State Value Internal state value, reflecting the corresponding state value selected by Debug Select field. This field is read-only and write to this field does not have effect.
15–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1 EXACT_BLK_NUM	Exact block number block read enable for SDIO CMD53 This bit should be set before S/W issues CMD53 multi-block read with exact block number. This bit should not be set if the CMD53 multi-block read is not exact block number. 0 none exact block read 1 exact block read for SDIO CMD53
0 EXT_DMA_EN	External DMA Request Enable Enable the request to external DMA. When the internal DMA (either Simple DMA or Advanced DMA) is not in use, and this bit is set, ESDHC will send out DMA request when the internal buffer is ready. This bit is particularly useful when transferring data by ARM platform polling mode, and it is not allowed to send out the external DMA request. By default, this bit is set. 0 In any scenario, ESDHC does not send out external DMA request 1 When internal DMA is not active, the external DMA request will be sent out

29.7.23 MMC Boot Register (ESDHCV2x_MMCBOOT)

This register contains the MMC Fast Boot control register.

Addresses: ESDHCV2-1_MMCBOOT is 5000_4000h base + C4h offset = 5000_40C4h
ESDHCV2-2_MMCBOOT is 5000_8000h base + C4h offset = 5000_80C4h
ESDHCV2-4_MMCBOOT is 5002_4000h base + C4h offset = 5002_40C4h



ESDHCV2x_MMCBOOT field descriptions

Field	Description
31–16 BOOT_BLK_CNT[15:0]	The value defines the Stop At Block Gap value of automatic mode. When received card block cnt is equal to BOOT_BLK_CNT and AUTO_SABG_EN is 1, then Stop At Block Gap.
15–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7 AUTO_SABG_EN	When boot, enable auto stop at block gap function. This function will be triggered, and host will stop at block gap when received card block cnt is equal to BOOT_BLK_CNT.
6 BOOT_EN	Boot mode enable 0: fast boot disable 1: fast boot enable
5 MMC_BOOT_MODE	Boot mode select 0: normal boot 1: alternative boot
4 BOOT_ACK	Boot ack mode select 0: no ack 1: ack
3–0 DTOCV_ACK[3:0]	Boot ACK time out counter value. 0000 SDCLK x 2^8 0001 SDCLK x 2^9 0010 SDCLK x 2^10 0011 SDCLK x 2^11 0100 SDCLK x 2^12 0101 SDCLK x 2^13

Table continues on the next page...

ESDHCV2x_MMCBOOT field descriptions (continued)

Field	Description
	0110 SDCLK x 2^14 0111 SDCLK x 2^15 1110 SDCLK x 2^22 1111 Reserved

29.7.24 Host Controller Version (ESDHCV2x_HOSTVER)

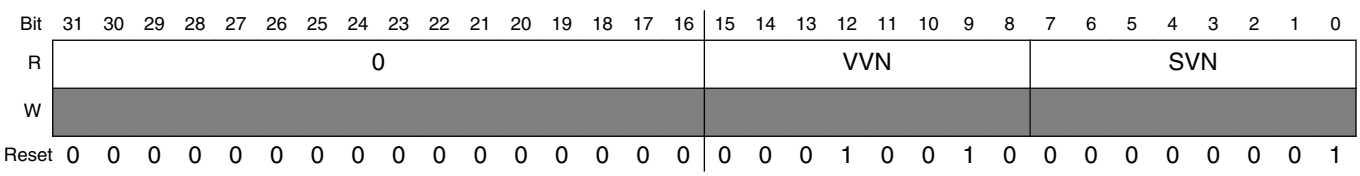
This register contains the vendor Host Controller version information. All bits are read only and when read return the reset value.

NOTE

The VVN field is 0x12 for both ESDHCV2 and ESDHCV3. Todetermine whether the controller is ESDHCV2 or ESDHCV3 do the following:

Set DLLCTRL [10]. This bit only exists in ESDHCV3. If it is set to 1 successfully, then the controller is ESDHCV3, otherwise it is ESDHCV2.

Addresses: ESDHCV2-1_HOSTVER is 5000_4000h base + FCh offset = 5000_40FCh
ESDHCV2-2_HOSTVER is 5000_8000h base + FCh offset = 5000_80FCh
ESDHCV2-4_HOSTVER is 5002_4000h base + FCh offset = 5002_40FCh



ESDHCV2x_HOSTVER field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–8 VVN	Vendor Version Number: These status bits are reserved for the vendor version number. The Host Driver will not use this status. Settings not shown are reserved. 00 Freescale ESDHC Version 1.0 10 Freescale ESDHC Version 2.0 11 Freescale ESDHC Version 2.1 12 Freescale ESDHC Version 2.2
7–0 SVN	Specification Version Number: These status bits indicate the Host Controller Specification Version.

Table continues on the next page...

ESDHCV2x_HOSTVER field descriptions (continued)

Field	Description
	Settings not shown are reserved. — 01 SD Host Specification Version 2.0, supports Test Event Register and ADMA.

Chapter 30
Enhanced Secured Digital Host Controller (eSDHCv3)

30.1 Overview

The Enhanced Secured Digital Host Controller Version 3 (ESDHCV3, referred to as ESDHC hereafter) provides the interface between the host system and the SD/SDIO/MMC cards, as depicted in Figure 30-1. The ESDHC acts as a bridge, passing host bus transactions to the SD/SDIO/MMC cards by sending commands and performing data accesses to/from the cards. It handles the SD/SDIO/MMC protocols at the transmission level.

Two versions of the ESDHC block are utilized in the i.MX53 SoC: Version 2 and Version 3 (ESDHCV2 and ESDHCV3). There are three instances of the block ESDHCV2, and one instance of the block ESDHCV3. The text will use the designation ESDHC when referring to features that are common to both.

The types of cards supported by the ESDHC are described briefly as follows:

The Multi Media Card (MMC) is a universal low cost data storage and communication media that is designed to cover a wide area of applications including mobile video and gaming. Old MMC cards are based on a 7-pin serial bus with a single data pin, while the new high speed MMC communication is based on an advanced 11-pin serial bus designed to operate in the low voltage range.

The Secure Digital Card (SD) is an evolution of the old MMC technology. It is specifically designed to meet the security, capacity, performance, and environment requirements inherent in newly emerging audio and video consumer electronic devices. The physical form factor, pin assignment and data transfer protocol are forward compatible with the old MMC (with some additions).

Under the SD protocol, it can be categorized into Memory card, I/O card and Combo card, which has both memory and I/O functions. The memory card invokes a copyright protection mechanism that complies with the security of the SDMI standard. The I/O

card, which is also known as SDIO card, provides high-speed data I/O with low power consumption for mobile electronic devices. For the sake of simplicity, Figure 30-1 does not show cards with reduced size or mini cards.

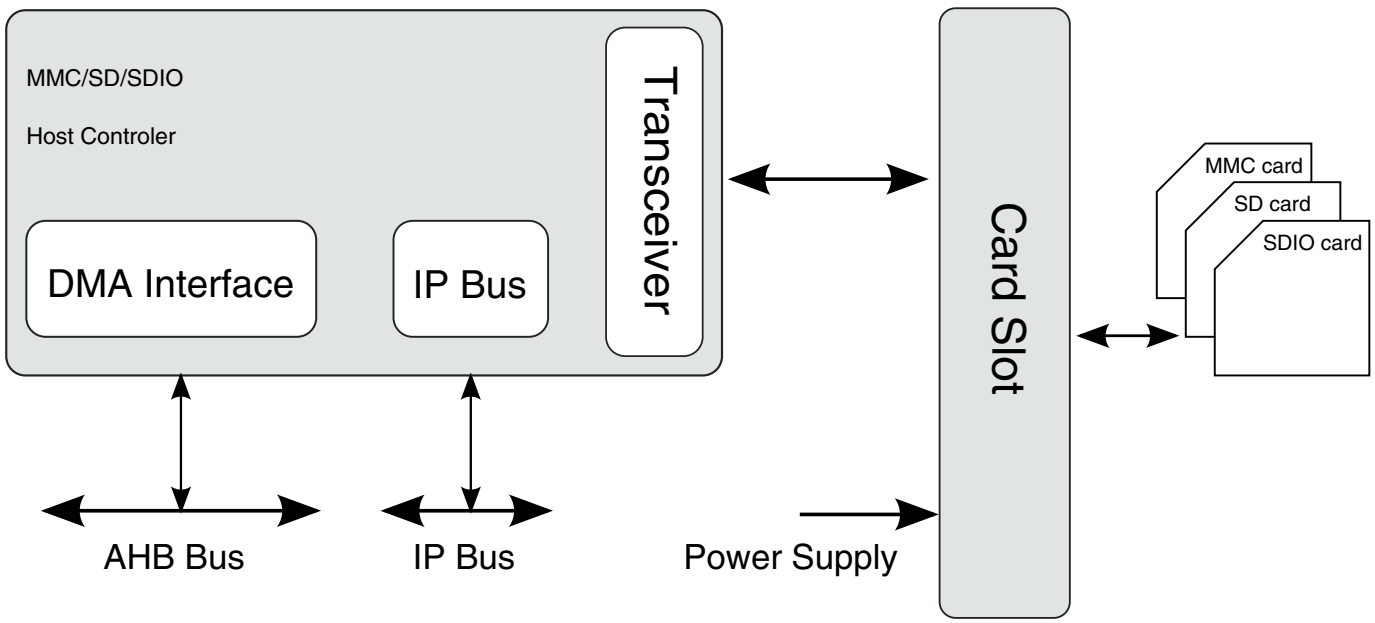


Figure 30-1. System Connection of the ESDHC

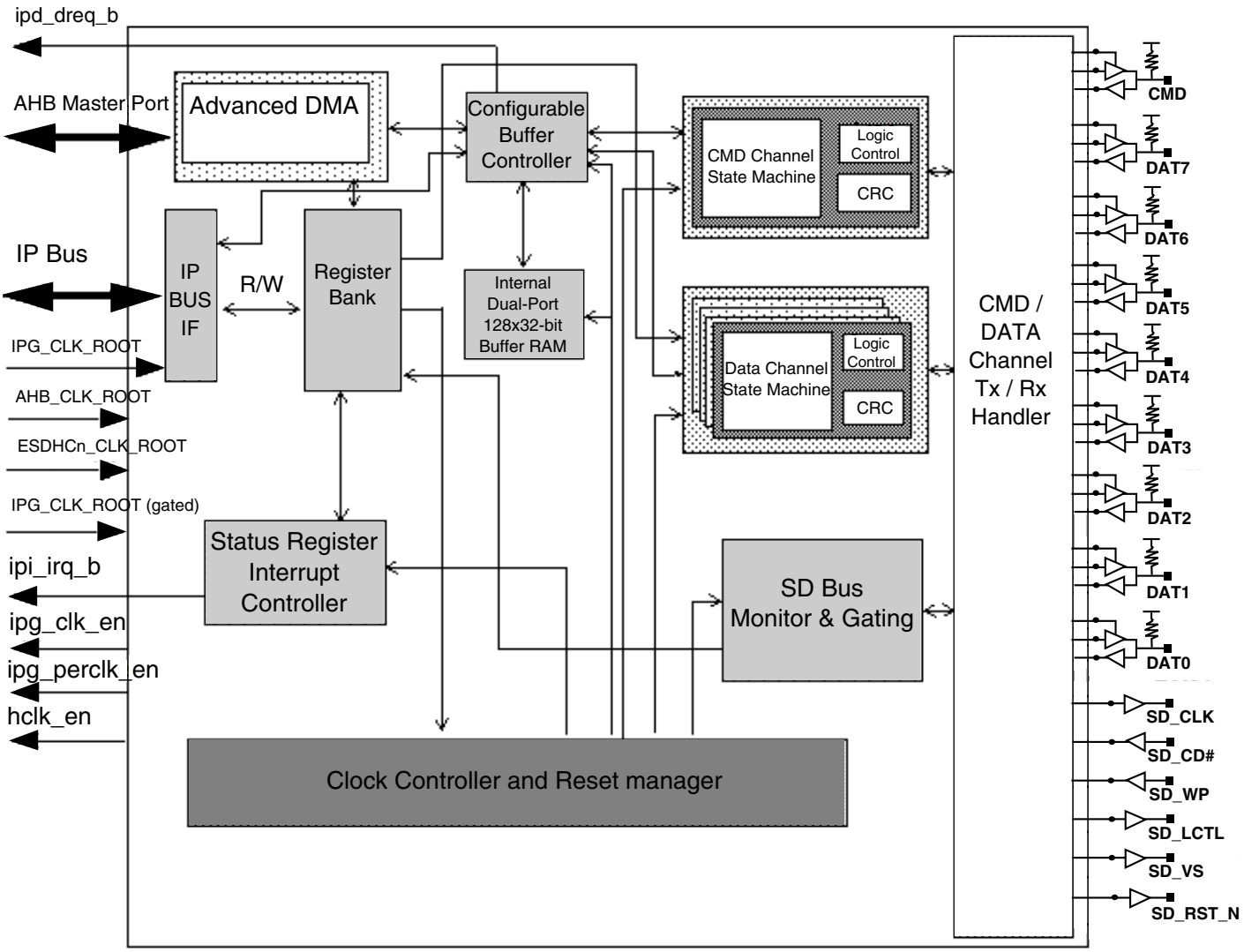


Figure 30-2. enhanced Secure Digital Host Controller Block Diagram

30.1.1 Features

The features of the ESDHC include the following:

- Conforms to the SD Host Controller Standard Specification version 2.0 including Test Event register support
- Compatible with the MMC System Specification version 4.2/4.3/4.4
- Compatible with the SD Memory Card Specification version 2.0 and supports the High Capacity SD Memory Card
- Compatible with the SDIO Card Specification version 2.0
- Designed to work with SD Memory, miniSD Memory, SDIO, miniSDIO, SD Combo, MMC, MMC plus, and MMC RS cards

- Card bus clock frequency up to 52 MHz
- Supports 1-bit / 4-bit SD and SDIO modes, 1-bit / 4-bit / 8-bit MMC modes devices
  - Up to 200 Mbps of data transfer for SD/SDIO cards using 4 parallel data lines
  - Up to 416 Mbps of data transfer for MMC cards using 8 parallel data lines in SDR (Single Data Rate) mode
  - Up to 832 Mbps of data transfer for MMC cards using 8 parallel data lines in DDR (Dual Data Rate) mode
- Supports Single Block, Multi Block read and write
- Supports block sizes of 1 ~ 4096 bytes
- Supports the write protection switch for write operations
- Supports both synchronous and asynchronous abort
- Supports pause during the data transfer at block gap
- Supports SDIO Read Wait and Suspend Resume operations
- Supports Auto CMD12 for multi-block transfer
- Host can initiate non-data transfer command while data transfer is in progress
- Allows cards to interrupt the host in 1-bit and 4-bit SDIO modes, also supports interrupt period
- Embodies a fully configurable 128x32-bit FIFO for read/write data
- Supports internal and external DMA capabilities
- Supports Advanced DMA to perform linked memory access

30.1.2 Modes and Operations

30.1.2.1 Data transfer Modes

The ESDHC can select the following modes for data transfer:

- SD 1-bit
- SD 4-bit
- MMC 1-bit
- MMC 4-bit
- MMC 8-bit
- Identification Mode (up to 400 kHz)
- MMC full speed mode (up to 20 MHz)
- MMC high speed mode (up to 52 MHz)
- MMC DDR mode (up to 52MHz, sampled on both clock edges)
- SD/SDIO full speed mode (up to 25 MHz)
- SD/SDIO high speed mode (up to 50 MHz)

30.2 External Signals

30.2.1 Signals Overview

The ESDHC has 15 associate I/O signals.

- The SD_CLK is an internally generated clock used to drive the MMC, SD, SDIO cards.
- The CMD I/O is used to send commands and receive responses to/from the card. Eight data lines (DAT7~DAT0) are used to perform data transfers between the ESDHC and the card.
- The SD_CD# and SD_WP are card detection and write protection signals directly routed from the socket. A low on SD_CD# means that a card is inserted and a high on SD_WP means that the write protect switch is active.
- SD_OD is an output signal generated in SoC level outside ESDHC and is used to select the external open drain resistor.
- SD_LCTL is an output signal used to drive an external LED to indicate that the SD interface is busy.
- SD_RST_N is an output signal used to reset MMC card. This should be supported by card.

SD_CD#, SD_WP, SD_OD, SD_LCTL and SD_RST_N are all optional for system implementation. If the ESDHC is desired to support a 4-bit data transfer, DAT7~DAT4 can also be optional and tied to high.

30.2.2 Ports Table

Table 30-1 shows the signal properties of the I/Os.

Table 30-1. Properties of I/O Signals

Name	Port	Function	Reset State	Pull up
SD_CLK	O	Clock for MMC/SD/SDIO card	0	N/A
SD_CMD	I/O	CMD line connect to card	1	Pull up
SD_DAT7	I/O	DAT7 line in 8-bit mode Not used in other modes	1	Pull up
SD_DAT6	I/O	DAT6 line in 8-bit mode Not used in other modes	1	Pull up

Table continues on the next page...

Table 30-1. Properties of I/O Signals (continued)

Name	Port	Function	Reset State	Pull up
SD_DAT5	I/O	DAT5 line in 8-bit mode Not used in other modes	1	Pull up
SD_DAT4	I/O	DAT4 line in 8-bit mode Not used in other modes	1	Pull up
SD_DAT3	I/O	DAT3 line in 4/8-bit mode or configured as card detection pin May be configured as card detection pin in 1-bit mode	0	Should be pull-up/pull-down configurable as this port may be used for card detection
SD_DAT2	I/O	DAT2 line or Read Wait in 4-bit mode Read Wait in 1-bit mode	1	Pull up
SD_DAT1	I/O	DAT1 line in 4/8-bit mode Also used to detect interrupt in 1/4-bit mode	1	Pull up
SD_DAT0	I/O	DAT0 line in all modes Also used to detect busy state	1	Pull up
SD_CD#	I	Card detection pin If not used tie high	N/A	N/A
SD_WP	I	Card write protect detect If not used tie low	N/A	N/A
SD_OD	O	Open drain select (not generated within the ESDHC). Optional output	N/A	N/A
SD_LCTL	O	LED control used to drive an external LED Active high Fully controlled by the driver Optional output	0	N/A
SD_RST_N	O	Card hardware reset signal, active LOW	1	N/A

30.3 Functional Description

The following sections provide a brief functional description of the major system blocks, including the Data Buffer, DMA AHB interface, register bank as well as IP Bus interface, dual-port memory wrapper, data/command controller, clock & reset manager and clock generator.

30.3.1 Data Buffer

The ESDHC uses one configurable data buffer, so that data can be transferred between the system bus (IP Bus or AHB Bus) and the SD card, with an optimized manner to maximize throughput between the two clock domains (that is, the IP peripheral clock, and the master clock). See the table below for illustration of the buffer scheme. The buffer is used as temporary storage for data being transferred between the host system and the card. The watermark levels for read and write are both configurable, and can be any number from 1 to 128 words. The burst lengths for read and write are also configurable, and can be any number from 1 to 31 words.

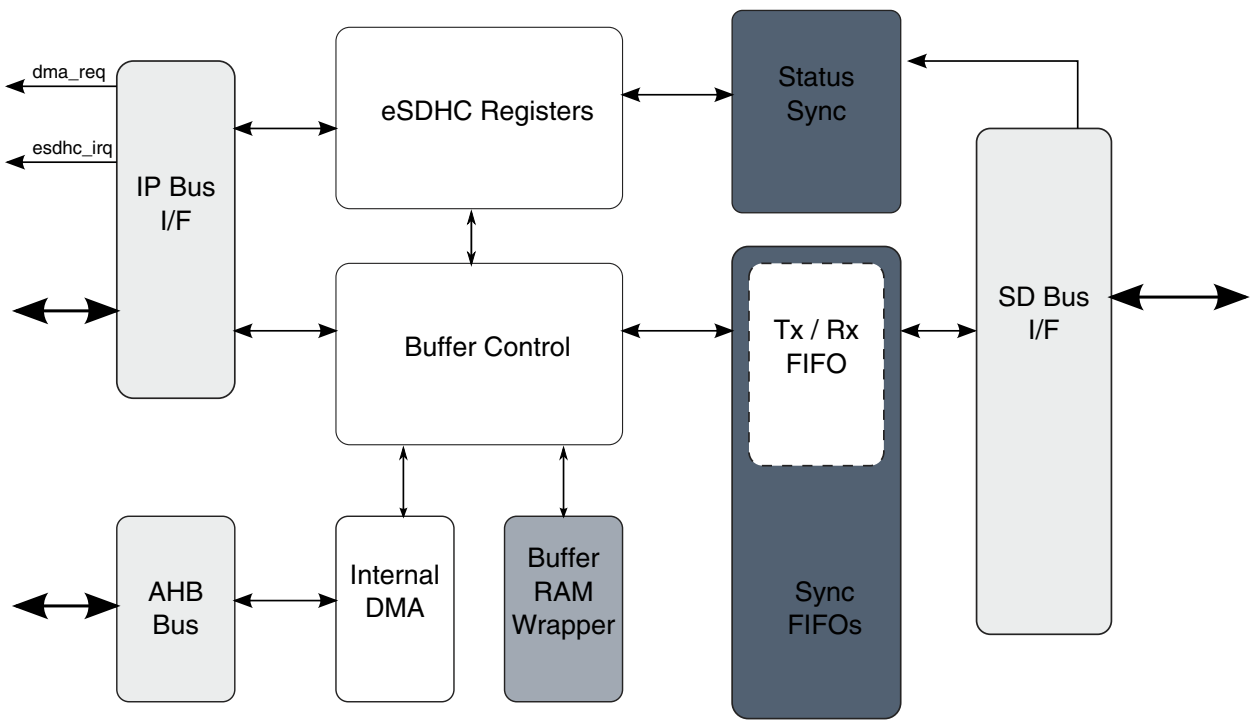


Figure 30-3. ESDHC Buffer Scheme

There are 3 transfer modes to access the data buffer:

- ARM platform polling mode:
  - For a host read operation, when the number of words received in the buffer meets or exceeds the RD_WML watermark value, then by polling the BRR bit the Host Driver can read the Buffer Data Port register to fetch the amount of words set in the RD_WML register from the buffer. The write operation is similar.
- External DMA mode:
  - For a read operation, when there are more words received in the buffer than the amount set in the RD_WML register, a DMA request is sent out to inform the external DMA to fetch the data. The request will be immediately de-asserted

when there is an access on the Buffer Data Port register. If the number of words in the buffer after the current burst meets or exceeds RD_WML value, then the DMA request is asserted again. For instance, if there are twice as many words in the buffer than the RD_WML value, there are two successive DMA requests with only one cycle of de-assertion between. The write operation is similar.

NOTE

Data buffer accesses by the ARM platform polling mode and external DMA mode both use the IP Bus. In these mode, if the external DMA is enabled, an external DMA request is sent out every time the buffer is ready.

- Internal DMA mode (includes simple and advanced DMA accesses):
  - The internal DMA access, either by simple or advanced DMA, is over the AHB bus. For internal DMA access mode, the external DMA request will never be sent out.

    For a read operation, when there are more words in the buffer than the amount set in the RD_WML register, the internal DMA starts fetching data over the AHB bus. Except INCR4 and INCR8, the burst type is always INCR mode and the burst length depends on the shortest of following factors:

1. Burst length configured in the burst length field of the Watermark Level register
2. Watermark Level boundary
3. Block size boundary
4. Data boundary configured in the current descriptor (if the ADMA is active)
5. 1 Kbyte address boundary defined in the AHB protocol

Write operation is similar.

Sequential and contiguous access is necessary to ensure the pointer address value is correct. Random or skipped access is not possible. The byte order, by reset, is little endian mode. The actually byte order is swapped inside the buffer, according to the endian mode configured by software. See Figure 29-4 and Figure 29-5. For a host write operation, byte order is swapped after data is fetched from the buffer and ready to send to the SD Bus. For a host read operation, byte order is swapped before the data is stored into the buffer.

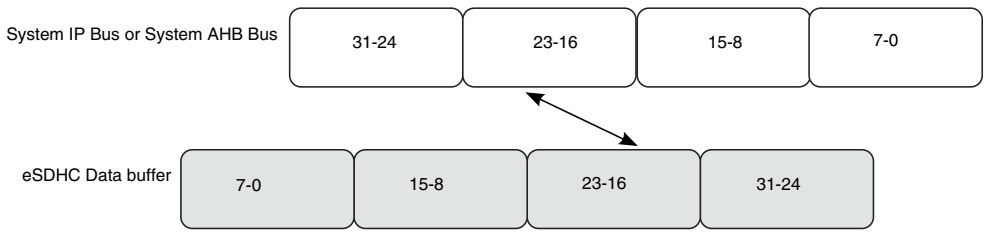


Figure 30-4. Data Swap between System Bus and ESDHC Data Buffer in Byte Little Endian Mode

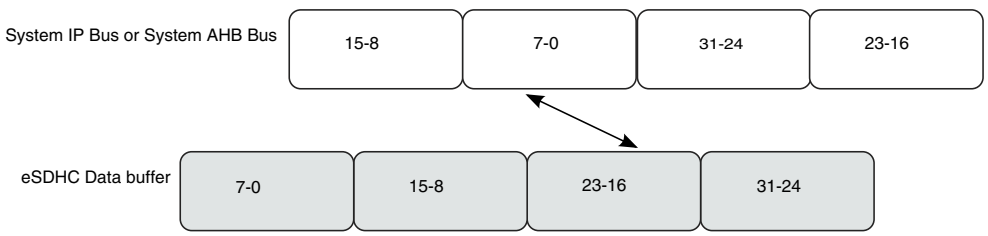


Figure 30-5. Data Swap between System Bus and ESDHC Data Buffer in Half Word Big Endian Mode

30.3.1.1 Write Operation Sequence

There are three ways to write data into the buffer when the user transfers data to the card:

1. By using external DMA through the ESDHC DMA request signal.
2. By processor core polling through the BWR bit in Interrupt Status register (interrupt or polling).
3. By using the internal DMA.

When the internal DMA is not used, (i.e. the DMAEN bit in the Transfer Type register is not set when the command is sent), the ESDHC asserts a DMA request when the amount of buffer space exceeds the value set in the WR_WML register, and is ready for receiving new data. At the same time, the ESDHC would set the BWR bit. The buffer write ready interrupt will be generated if it is enabled by software.

When internal DMA is used, the ESDHC will not inform the system before all the required number of bytes are transferred (if no error was encountered). When an error occurs during the data transfer, the ESDHC will abort the data transfer and abandon the current block. The Host Driver should read the contents of the DMA System Address register to get the starting address of the abandoned data block. If the current data transfer is in multi block mode, the ESDHC will not automatically send CMD12, even though the

AC12EN bit in the Transfer Type register is set. The Host Driver shall send CMD12 in this scenario and re-start the write operation from that address. It is recommended that a Software Reset for Data be applied before the transfer is re-started after error recovery.

The ESDHC will not start data transmission until the number of words set in the WR_WML register can be held in the buffer. If the buffer is empty and the Host System does not write data in time, the ESDHC will stop the SD_CLK to avoid the data buffer under-run situation.

30.3.1.2 Read Operation Sequence

There are three ways to read data from the buffer when the user transfers data from the card:

1. By using the external DMA through the ESDHC DMA request signal
2. By processor core polling through the BRR bit in Interrupt Status register (interrupt or polling)
3. By using the internal DMA

When internal DMA is not used (i.e. DMAEN bit in Transfer Type register is not set when the command is sent), the ESDHC asserts a DMA request when the amount of data exceeds the value set in the RD_WML register (that is, available and ready for system fetching data). At the same time, the ESDHC would set the BRR bit. The buffer read ready interrupt will be generated if it is enabled by software.

When internal DMA is used, the ESDHC will not inform the system before all the required number of bytes are transferred (if no error was encountered). When an error occurs during the data transfer, the ESDHC will abort the data transfer and abandon the current block. The Host Driver should read the content of the DMA System Address register to get the starting address of the abandoned data block. If the current data transfer is in multi block mode, the ESDHC will not automatically send CMD12, even though the AC12EN bit in the Transfer Type register is set. The Host Driver shall send CMD12 in this scenario and re-start the read operation from that address. It is recommended that a Software Reset for Data be applied before the transfer is re-started after error recovery.

For any read transfer mode, the ESDHC will not start data transmission until the number of words set in the RD_WML register are in the buffer. If the buffer is full and the Host System does not read data in time, the ESDHC will stop the SD_CLK to avoid the data buffer over-run situation.

30.3.1.3 Data Buffer and Block Size

The user needs to know the buffer size, for the buffer operation during a data transfer, to utilize it in the most optimized way. In the ESDHC, the only data buffer can hold up to 128 words (32-bit), and the watermark levels for write and read can be configured respectively. For both read and write, the watermark level can be from 1 word to the maximum of 128 words. For both read and write, the burst length, can be from 1 word to the maximum of 31 words. The Host Driver may configure the value according to the system situation and requirement.

During a multi-block data transfer, the block length may be set to any value between 1 and 4096 bytes inclusive which satisfies the requirements of the external card. The only restriction is from the external card. It might not support that large of a block or it does not support a partial block access (which is not the integer times of 512 bytes).

For block size not times of 4, that is, not word aligned, ESDHC requires stuff bytes at the end of each block, because ESDHC treats each block individually. For example, if the block size is 7 bytes and there are 12 blocks to write, the system side must write two times for each block. For each block, the ending byte will be abandoned by ESDHC because it only sends 7 bytes to the card and picks data from the following system write, making a total of 24 beats of write access.

30.3.1.4 Dividing Large Data Transfer

This SDIO command CMD53 definition, limits the maximum data size of data transfers according to the following formula:

Max data size = Block size x Block count

The length of a multiple block transfer needs to be in block size units. If the total data length can't be divided evenly into a multiple of the block size, then there are two ways to transfer the data which depend on the function and the card design. Option 1 is for the Host Driver to split the transaction. The remainder of the block size data is then transferred by using a single block command at the end. Option 2 is to add dummy data in the last block to fill the block size. For option 2, the card must manage the removal of the dummy data.

See figure below for an example showing the dividing of large data transfers. Assuming a kind of WLAN SDIO card only supports block size up to 64 bytes. Although the ESDHC supports a block size of up to 4096 bytes, the SDIO can only accept a block size less than64 bytes, so the data must be divided (see example below).

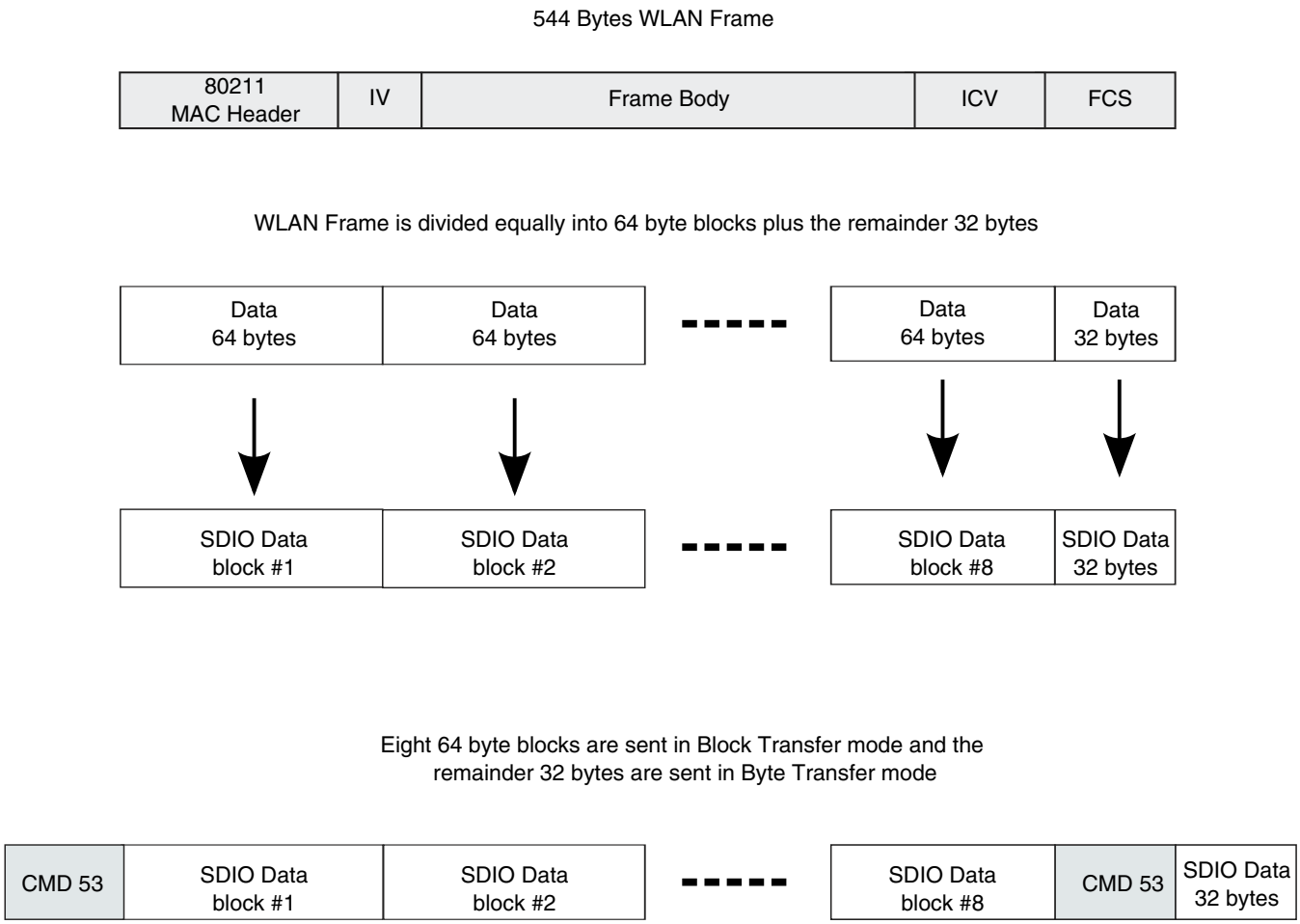


Figure 30-6. Example for Dividing Large Data Transfers

30.3.1.5 External DMA Request

When the internal DMA is not in use, and external DMA request is enabled, the Data Buffer will generate a DMA request to the system. During a write operation, when the number of WR_WML words can be held in the buffer free space, the signal esdhc_dreq_b is asserted to 0, informing the Host System of a DMA write. The BWR bit in the Interrupt Status register is also set, as long as the BWRSEN bit in the Interrupt Status Enable register is set. The DMA request is immediately de-asserted when an access to the Data Port register is made. If the buffer's free space still meets the watermark condition, the DMA request is asserted again after a cycle.

On read operation, when the number of RD_WML words are already in the buffer, the signal esdhc_dreq_b is asserted to 0, informing the Host System for a DMA read. The BRR bit in the Interrupt Status register is also set, as long as the BRRSEN bit in the

Interrupt Status Enable register is set. The DMA request is immediately de-asserted when an access to the Data Port register is made. If the buffer's data still meets the watermark condition, the DMA request is asserted again after a cycle.

Because the DMA burst length can not change during a data transfer for an external DMA transfer, the watermark level (read or write) must be a divisor of the block size. If it is not, transferring of the block may cause buffer under-run (read operation) or over-run (write operation). For example, if the block size is 512 bytes, the watermark level of read (or write) must be a power of two between 1 and 128. For processor core polling access, as the last access in the block transfer can be controlled by software, there is no such issue. The watermark level can be any value, even larger than the block size (but no greater than 128 words). This is because the actual number of bytes transferred by the software can be controlled and does not exceed the block size in each transfer.

The ESDHC also supports non-word aligned block size, as long as the card supports that block size. In this case, the watermark level should be set as the number of words. For example, if the block size is 31 bytes, the watermark level can be set to any number of word. For this case, the BLKSIZE bits of the Block Attribute register shall be set as 1fh. For the ARM platform polling access, the burst length can be 1 to 128 words, without restriction. This is because the software will transfer 8 words, and the ESDHC will also set the BWR or BRR bits when the remaining data does not violate data buffer. Refer to DMA Burst Length for more details about the dynamic watermark level of the data buffer. For the above example, even though 8 words are transferred through the Data Port register, the ESDHC will transfer only 31 bytes over the SD Bus, as required by the BLKSIZE bits. In this data transfer, with non-word aligned block size, the endian mode should be set cautiously, or invalid data will be transferred to/from the card.

30.3.2 DMA AHB Interface

The internal DMA implements a DMA engine and the AHB master. When the internal DMA is enabled, the esdhc_dreq_b will not be asserted during the transfer, but the BWR and BRR bits will be set if the BWRSEN and BRRSEN bits have been set in the Interrupt Status Enable register. See the figure below for an illustration of the DMA AHB interface block.

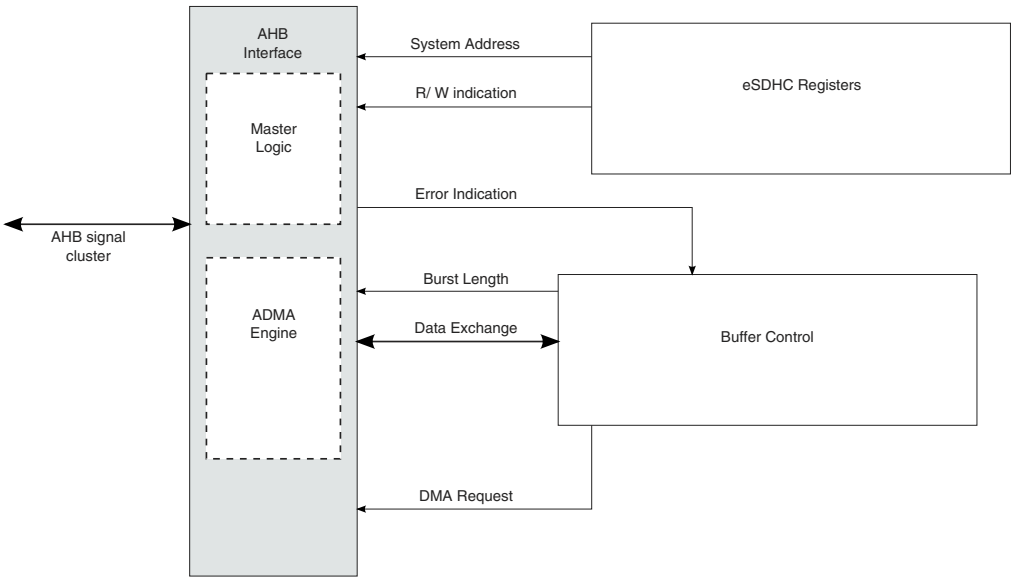


Figure 30-7. DMA AHB Interface Block

30.3.2.1 Internal DMA Request

If the watermark level requirement is met in data transfer, and the Internal DMA is enabled, the Data Buffer block will send a DMA request to AHB interface. Meanwhile, the external DMA request signal (esdhc_dreq_b) is disabled. The delay in response from the internal DMA engine depends on the system AHB bus loading and the priority assigned to the ESDHC. The DMA engine does not respond to the request during its burst transfer, but is ready to serve as soon as the burst is over. The Data Buffer de-asserts the request once an access to the buffer is made. Upon access to the buffer by internal DMA, the Data Buffer updates its internal buffer pointer, and when the watermark level is satisfied, another DMA request is sent.

The data transfer is in the block unit, and the subsequent watermark level is always set as the remaining number of words. For instance, for a multi block data read with each block size of 31 bytes, and the burst length set to 6 words. After the first burst transfer, if there are more than 2 words in the buffer (which might contain some data of the next block), another DMA request read is sent. This is because the remaining number of words to send for the current block is (31 - 6 * 4) / 4 = 2. The ESDHC will read 2 words out of the buffer, with 7 valid bytes and 1 stuff byte.

30.3.2.2 DMA Burst Length

Just like a ARM platform polling access, the DMA burst length for the internal DMA engine can be from 1 to 16 words. The actual burst length for the DMA depends on the lesser of the configured burst length or the remaining words of the current block. Take the example in Internal DMA Request again. The following burst length after 6 words are read will be 2 words, and the next burst length will be 6 words again. This is because the next block starts, which is 31 bytes, more than 6 words. The Host Driver writer may take this variable burst length into account. It is also acceptable to configure the burst length as the divisor of the block size, so that each time the burst length will be the same.

30.3.2.3 AHB Master Interface

It is possible that the internal AHB DMA engine could fail during the data transfer. When this error occurs, the DMA engine stops the transfer and goes to the idle state as well as the internal data buffer stops accepting incoming data. The DMAE bit in the Interrupt Status register is set to inform the driver.

Once the DMAE interrupt is received, the software shall send a CMD12 to abort the current transfer and read the DS_ADDR bits of the DMA System Address register to get the starting address of the corrupted block. After the DMA error is fixed, the software should apply a data reset and re-start the transfer from this address to recover the corrupted block.

30.3.2.4 ADMA Engine

In the SD Host Controller Standard, the new DMA transfer algorithm called the ADMA (Advanced DMA) is defined. For Simple DMA, once the page boundary is reached, a DMA interrupt will be generated and the new system address shall be programmed by the Host Driver. The ADMA defines the programmable descriptor table in the system memory. The Host Driver can calculate the system address at the page boundary and program the descriptor table before executing ADMA. It reduces the frequency of interrupts to the host system. Therefore, higher speed DMA transfers could be realized since the Host MCU intervention would not be needed during long DMA based data transfers.

There are two types of ADMA: ADMA1 and ADMA2 in Host Controller. ADMA1 can support data transfer of 4KB aligned data in system memory. ADMA2 removes the restriction so that data of any location and any size can be transferred in system memory. Their formats of Descriptor Table are different.

ADMA engine can recognize all kinds of descriptors define in SD Host Controller Standard, and if 'End' flag is detected in the descriptor, ADMA engine will stop after this descriptor is processed.

30.3.2.4.1 ADMA Concept and Descriptor Format

For ADMA1, including the following descriptors:

- Valid/Invalid descriptor.
- Nop descriptor.
- Set data length descriptor.
- Set data address descriptor.
- Link descriptor.
- Interrupt flag and End flag in descriptor.

For ADMA2, including the following descriptors:

- Valid/Invalid descriptor.
- Nop descriptor.
- Rsv descriptor.
- Set data length & address descriptor.
- Link descriptor.
- Interrupt flag and End flag in descriptor.

ADMA2 deals with the lower 32-bit first, and then the higher 32-bit. If the 'Valid' flag of descriptor is 0, it will ignore the high 32-bit. Address field shall be set on word aligned (lower 2-bit is always set to 0). Data length is in byte unit.

ADMA will start read/write operation after it reaches the Tran state, using the data length and data address analyzed from most recent descriptor(s).

For ADMA1, the valid data length descriptor is the last Set type descriptor before Tran type descriptor. Every Tran type will trigger a transfer, and the transfer data length is extracted from the most recent Set type descriptor. If there is no Set type descriptor after the previous Trans descriptor, the data length will be the value for previous transfer, or 0 if no Set descriptor is ever met.

For ADMA2, Tran type descriptor contains both data length and transfer data address, so only a Tran type descriptor can start a data transfer.

Address/ Page Field		Address/ Page Field		Attribute Field					
31	12	11	6	5	4	3	2	1	0
Address or Data Length		000000		Act 2	Act 1	0	Int	End	Valid

Act 2	Act1	Symbol	Comment	31- 28	27- 12
0	0	Nop	No Operation	Don't Care	
0	1	Set	Set Data Length	0000	Data Length
1	0	Tran	Transfer Data	Data Address	
1	1	Link	Link Descriptor	Descriptor Address	

Valid	Valid = 1 indicates this line of descriptor is effective. If Valid = 0 generate ADMA Error Interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA Interrupt when this descriptor is processed.

Figure 30-8. Format of the ADMA1 Descriptor Table

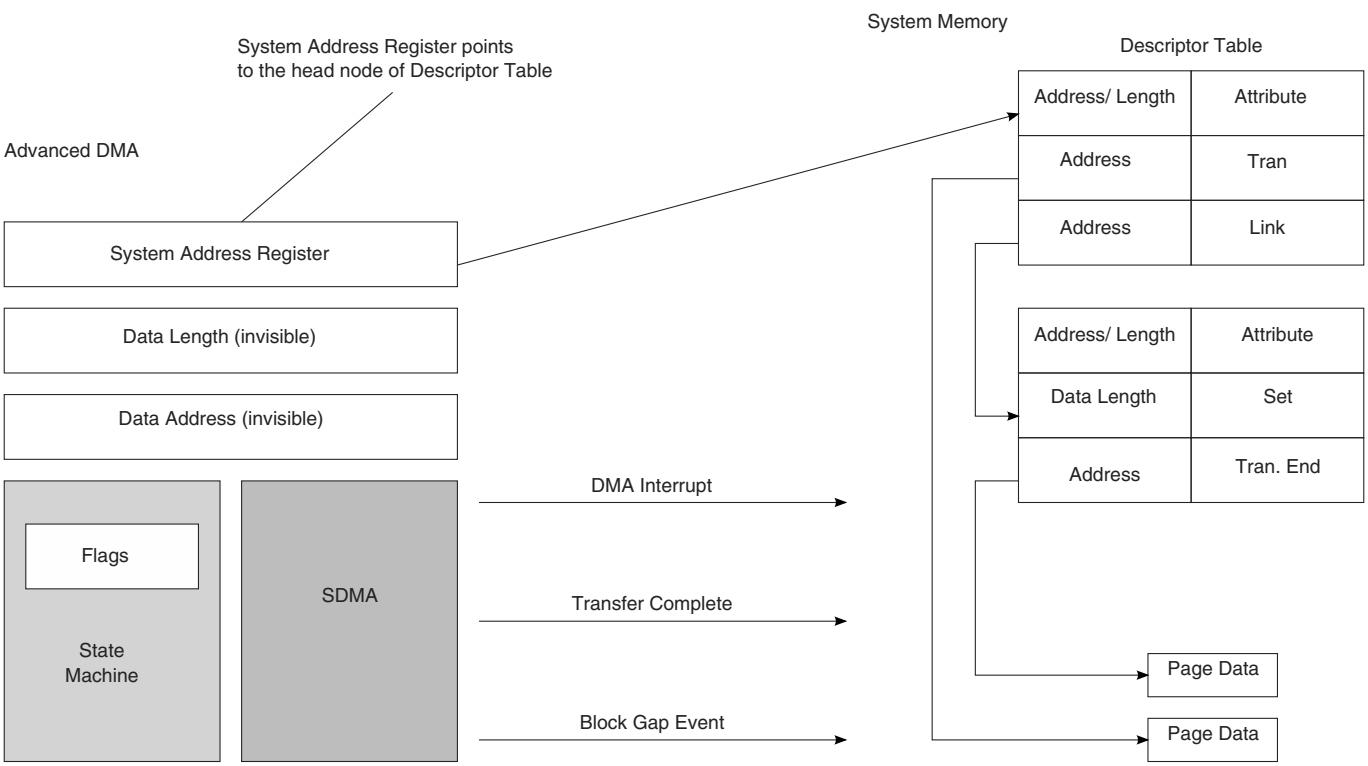


Figure 30-9. Concept and Access Method of ADMA1 Descriptor Table

Address/ Page Field		Address/ Page Field		Address/ Page Field		Attribute Field					
63	32	31	16	15	06	05	04	03	02	01	00
32-bit Address		16-bit Length		0000000000		Act 2	Act 1	0	Int	End	Valid

Act 2	Act1	Symbol	Comment	Operation
0	0	Nop	No Operation	Don't Care
0	1	Rsv	Reserved	Same as Nop. Read this line and go to next one
1	0	Tran	Transfer Data	Transfer data with address and length set in this descriptor line
1	1	Link	Link Descriptor	Link to another descriptor

Valid	Valid = 1indicates this line of descriptor is effective. If Valid = 0 generate ADMA Error Interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA Interrupt when this descriptor is done.

Figure 30-10. Format of the ADMA2 Descriptor Table

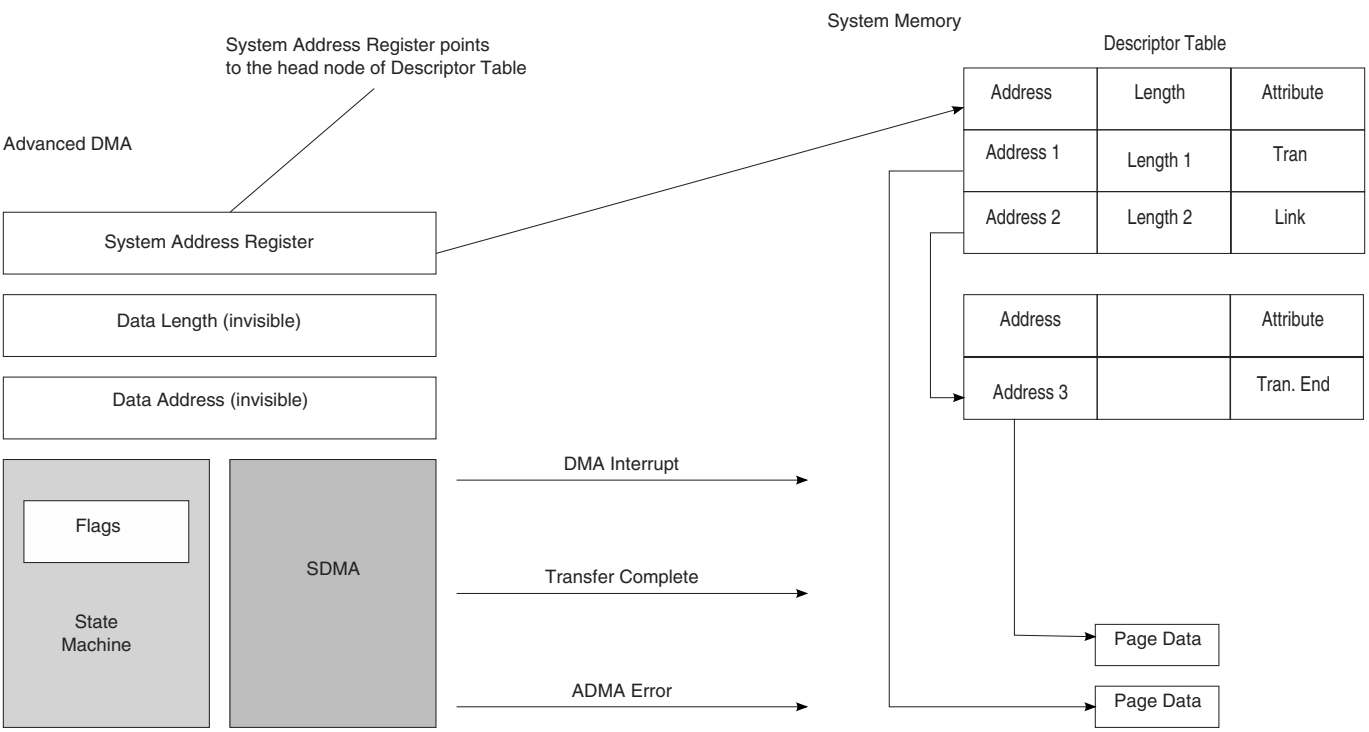


Figure 30-11. Concept and Access Method of ADMA2 Descriptor Table

30.3.2.4.2 ADMA Interrupt

If the 'Interrupt' flag of descriptor is set, ADMA will generate an interrupt according to different type descriptor:

For ADMA1:

- Set type descriptor: interrupt is generated when data length is set.
- Tran type descriptor: interrupt is generated when this transfer is complete.
- Link type descriptor: interrupt is generated when new descriptor address is set.
- Nop type descriptor: interrupt is generated just after this descriptor is fetched.

For ADMA2:

- Tran type descriptor: interrupt is generated when this transfer is complete.
- Link type descriptor: interrupt is generated when new descriptor address is set.
- Nop/Rsv type descriptor: interrupt is generated just after fetch this descriptor.

30.3.2.4.3 ADMA Error-DMA

The ADMA will stop whenever any error is encountered. These errors include:

- Fetching descriptor error
- AHB response error
- Data length mismatch error

ADMA descriptor error will be generated when it fails to detect 'Valid' flag in the descriptor. If adma descriptor error occurs, the interrupt is not generated even if the 'Interrupt' flag of this descriptor is set.

When BLKCNTEN bit is set, data length set in Block Attributes register must equal to the whole data length set in descriptor nodes, otherwise data length mismatch error will be generated.

If BLKCNTEN bit is not set, the whole data length set in descriptor should be times of block length, otherwise, when all data set in the descriptor nodes are done not at block boundary, the data mismatch error will occur.

30.3.3 Register Bank with IP Bus Interface

Register accesses via the IP Bus interface are actually on the Register Bank. See the figure below for the block diagram.

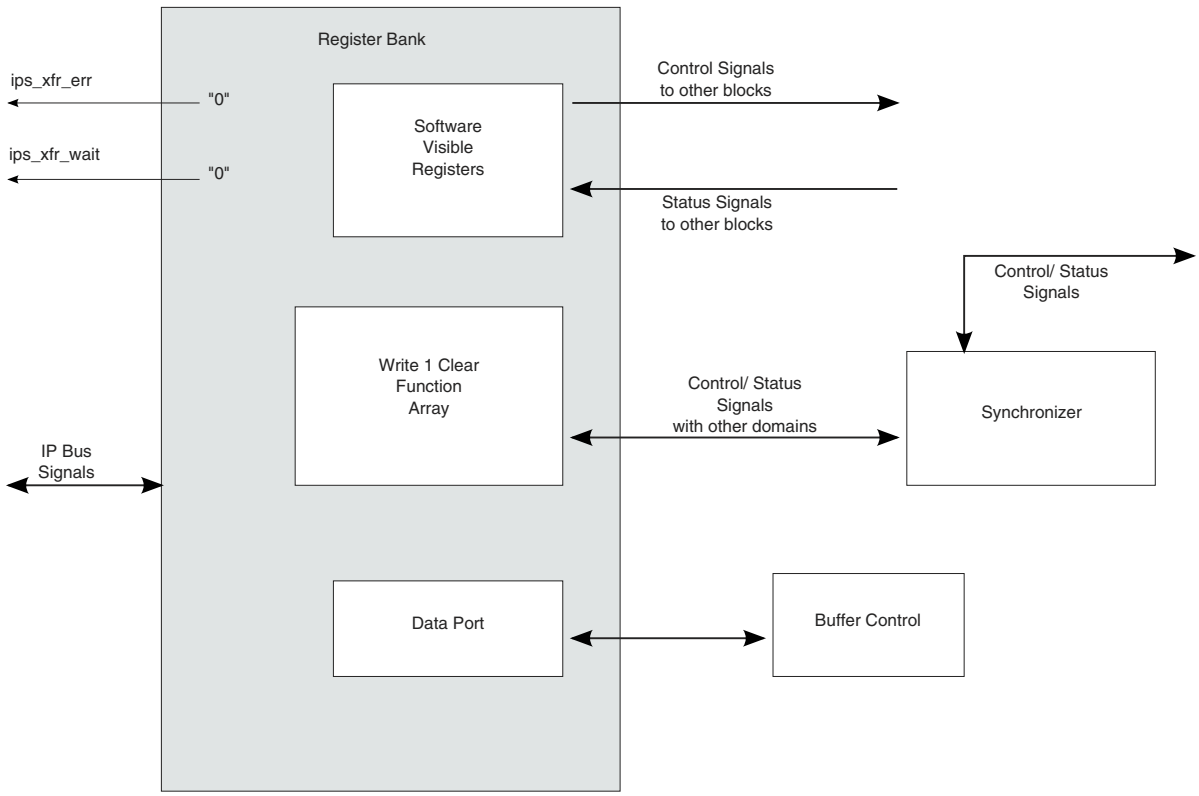


Figure 30-12. Register Bank Diagram

Only 32-bit access is allowed, and no partial read / write is supported, thus all accesses are word aligned.

30.3.4 SD Protocol Unit

The SD protocol unit deals with all SD protocol affairs.

The SD Protocol Unit performs the following functions:

- Acts as the bridge between the internal buffer and the SD bus
- Sends the command data as well as its argument serially
- Stores the serial response bit stream into corresponding registers
- Detects the bus state on the DAT[0] line
- Monitors the interrupt from the SDIO card
- Asserts the read wait signal
- Gates off the SD clock when buffer is announcing danger status
- Detects the write protect state

The SD Protocol Unit consists of four sub-blocks:

1. SD transceiver.

2. SD clock and monitor.
3. Command agent.
4. Data agent.

30.3.4.1 SD Transceiver

In the SD protocol unit, the transceiver is the main control sub-block. It consists of an FSM and control sub-block, from which the control signals for all other three sub-blocks are generated.

30.3.4.2 SD Clock and Monitor

This sub-block monitors the signal level on all 8 data lines, the command lines, and directly routes the level values into the Register Bank. The driver can use this for debug purposes.

The sub-block also detects the Card Detection (CD) line as well as the DAT[3] line. The transceiver reports the card insertion state according to the CD state, or the signal level on the DAT[3] line, when the D3CD bit in the Protocol Control register is set.

The sub-block detects the Write Protect (WP) line. With the information of the WP state, the Register Bank will ignore the command, accompanied by a write operation, when the WP switch is on.

If the internal data buffer is in danger, and the SD clock must be gated off to avoid buffer over/under-run, this sub-block will assert the gate of the output SD clock to shut the clock off. After the buffer danger has recovered, and when the system access of the buffer catches up, the clock gate of this sub-block will open and the SD clock will be active again.

This sub-block also drive SD_LCTL output signal when the LCTL bit is set by the driver.

30.3.4.3 Command Agent

The Command Agent deals with the transactions on the CMD line. See figure below for an illustration of the structure for the Command CRC Shift Register.

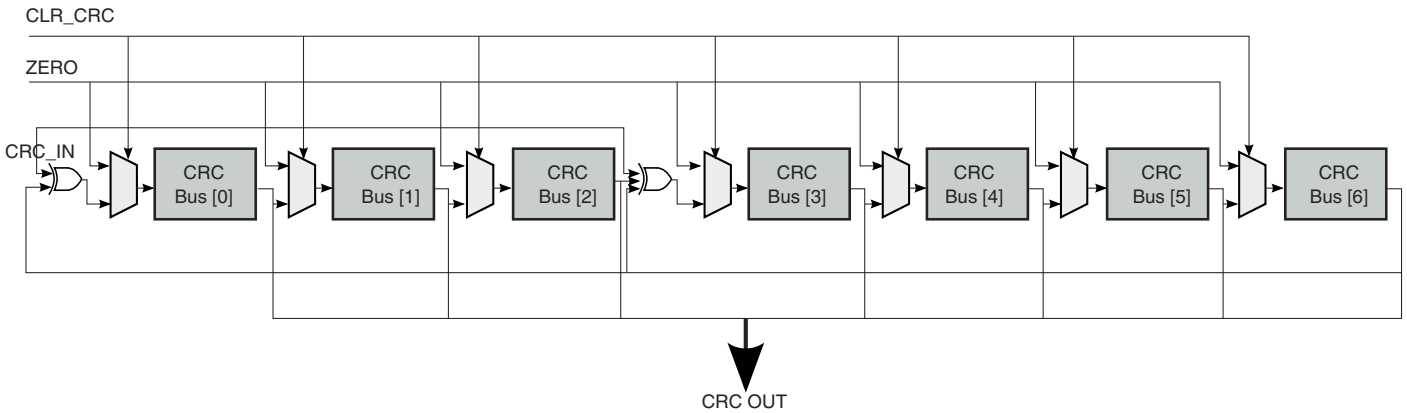


Figure 30-13. Command CRC Shift Register

The CRC polynomials for the CMD are as follows:

Generator polynomial: $G(x) = x^7 + x^3 + 1$
$M(x) = (\text{first bit}) * x^n + (\text{second bit}) * x^{n-1} + \ldots + (\text{last bit}) * x^0$
$CRC[6:0] = \text{Remainder } [(M(x) * x^7) / G(x)]$

30.3.4.4 Data Agent

The Data Agent deals with the transactions on the eight data lines. Moreover, this sub-block also detects the busy state on the DAT[0] line, and generate the Read Wait state by the request from the Transceiver. The CRC polynomials for the DAT are as follows:

Generator polynomial: $G(x) = x^{16} + x^{12} + x^5 + 1$
$M(x) = (\text{first bit}) * x^n + (\text{second bit}) * x^{n-1} + \ldots + (\text{last bit}) * x^0$
$CRC[15:0] = \text{Remainder } [(M(x) * x^{16}) / G(x)]$

30.3.5 Clock and Reset Manager

This sub-block controls all the reset signals within the ESDHC.

There are four kinds of reset signals within ESDHC:

1. Hardware reset.
2. Software reset for all.
3. Software reset for the data part.
4. Software reset for the command part.

All these signals are fed into this sub-block and stable signals are generated inside the module to reset all other modules. The sub-block also gates off all the inside signals.

There are three clocks inside the ESDHC:

1. ipg_clk.

2. ipg_perclk.
3. hclk.

The sub-block monitors the activities of all other sub-blocks, supplies the clocks for them, and when enabled, automatically gates off the corresponding clocks.

30.3.6 Clock Generator

The Clock Generator generates the SD_CLK by peripheral source clock in two stages. Refer to the figure below for the structure of the divider. The term "Base" represents the frequency of peripheral source clock.

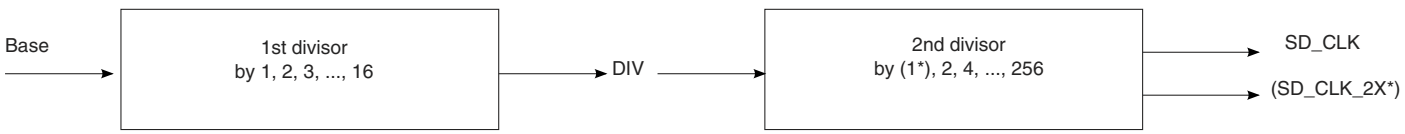


Figure 30-14. Two Stages of the Clock Divider

The first stage outputs an intermediate clock (DIV), which can be Base, Base/2, Base/3, ..., or Base/16.

The second stage is a prescaler, and outputs the actual clock (SD_CLK). Also it outputs DDR internal processing clock (SD_CLK_2X). These clocks are the driving clock for all sub-blocks of the SD Protocol Unit, and the sync FIFOs (see Figure 29-3) to synchronize with the data rate from the internal data buffer. The frequency of the clock output from this stage, can be DIV/2, DIV/4,..., or DIV/256. Thus the highest frequency of the SD_CLK is Base/2, while the lowest frequency is Base/4096. If the Base clock is of equal duty ratio (usually true), the duty cycle of SD_CLK/SD_CLK_2X is also 50%, even when the compound divisor is an odd value.

30.3.7 SDIO Card Interrupt

30.3.7.1 Interrupts in 1-bit Mode

In this case the DAT[1] pin is dedicated to providing the interrupt function. An interrupt is asserted by pulling the DAT[1] low from the SDIO card, until the interrupt service is finished to clear the interrupt.

30.3.7.2 Interrupt in 4-bit Mode

Since the interrupt and data line 1 share Pin 8 in 4-bit mode, an interrupt will only be sent by the card and recognized by the host during a specific time. This is known as the Interrupt Period. The ESDHC will only sample the level on Pin 8 during the Interrupt Period. At all other times, the host will ignore the level on Pin 8, and treat it as the data signal. The definition of the Interrupt Period is different for operations with single block and multiple block data transfers.

In the case of normal single data block transmissions, the Interrupt Period becomes active two clock cycles after the completion of a data packet. This Interrupt Period lasts until after the card receives the end bit of the next command that has a data block transfer associated with it.

For multiple block data transfers in 4-bit mode, there is only a limited period of time that the Interrupt Period can be active due to the limited period of data line availability between the multiple blocks of data. This requires a more strict definition of the Interrupt Period. For this case, the Interrupt Period is limited to two clock cycles. This begins two clocks after the end bit of the previous data block. During this 2-clock cycle interrupt period, if an interrupt is pending, the DAT[1] line will be held low for one clock cycle with the last clock cycle pulling DAT[1] high. On completion of the Interrupt Period, the card releases the DAT[1] line into the high Z state. The ESDHC samples the DAT[1] during the Interrupt Period when the IABG bit in the Protocol Control register is set.

Refer to SDIO Card Specification v1.10f for further information about the SDIO card interrupt.

30.3.7.3 Card Interrupt Handling

When the CINTIEN bit in the Interrupt Signal Enable Register is set to 0, the ESDHC clears the interrupt request to the Host System. The Host Driver should clear this bit before servicing the SDIO Interrupt and should set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts.

The SDIO Status bit is cleared by resetting the SDIO interrupt. Writing to this bit would have no effects. In 1-bit mode, the ESDHC will detect the SDIO Interrupt with or without the SD clock (to support wakeup). In 4-bit mode, the interrupt signal is sampled during the Interrupt Period, so there are some sample delays between the interrupt signal from the SDIO card and the interrupt to the Host System Interrupt Controller. When the SDIO status has been set, and the Host Driver needs to service this interrupt, so the SDIO bit in the Interrupt Control Register of SDIO card will be cleared. This is required to clear the SDIO interrupt status latched in the ESDHC and to stop driving the interrupt signal to the

System Interrupt Controller. The Host Driver must issue a CMD52 to clear the card interrupt. After completion of the card interrupt service, the SDIO Interrupt Enable bit is set to 1, and the ESDHC starts sampling the interrupt signal again.

See figure below for an illustration of the SDIO card interrupt scheme and for the sequences of software and hardware events that take place during a card interrupt handling procedure.

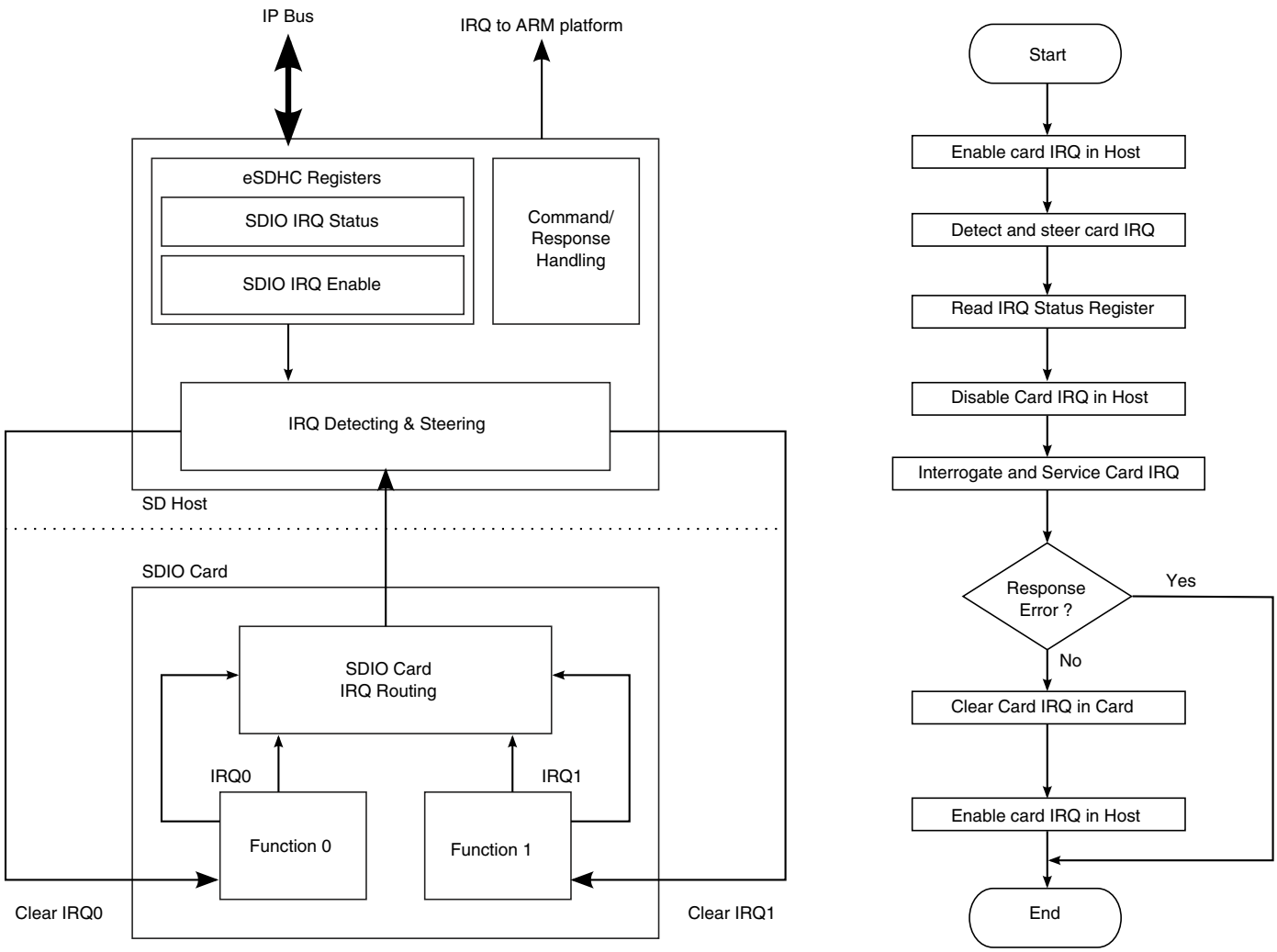


Figure 30-15. Card Interrupt Scheme and Card Interrupt Detection and Handling Procedure

30.3.8 Card Insertion and Removal Detection

The ESDHC uses either the DAT[3] pin or the CD pin to detect card insertion or removal. When there is no card on the MMC/SD bus, the DAT[3] will be pulled to a low voltage level by default. When any card is inserted to or removed from the socket, the ESDHC detects the logic value changes on the DAT[3] pin and generates an interrupt. When the DAT[3] pin is not used for card detection (for example, it is implemented in GPIO), the CD pin must be connected for card detection. Whether DAT[3] is configured for card detection or not, the CD pin is always a reference for card detection. Whether the DAT[3] pin or the CD pin is used to detect card insertion, the ESDHC will send an interrupt (if enabled) to inform the Host system that a card is inserted.

30.3.9 Power Management and Wake Up Events

When there is no operation between the ESDHC and the card through the SD bus, the user can completely disable the ipg_clk and ipg_perclk in the chip level clock control module to save power. When the user needs to use the ESDHC to communicate with the card, it can enable the clock and start the operation.

In some circumstances, when the clocks to the ESDHC are disabled, for instance, when the system is in low power mode, there are some events for which the user needs to enable the clock and handle the event. These events are called wakeup interrupts. The ESDHC can generate these interrupt even when there are no clocks enabled. The three interrupts which can be used as wake up events are:

1. Card Removal Interrupt
2. Card Insertion Interrupt
3. Interrupt from SDIO card

The ESDHC offers a power management feature. By clearing the clock enabled bits in the System Control Register (ESDHC_SYSCTL), the clocks are gated in the low position to the ESDHC. For maximum power saving, the user can disable all the clocks to the ESDHC when there is no operation in progress.

These three wake up events (or wakeup interrupts) can also be used to wake up the system from low-power modes.

NOTE

To make the interrupt a wakeup event, when all the clocks to the ESDHC are disabled or when the whole system is in low power mode, the corresponding wakeup enabled bit needs to be set. Refer to Protocol Control (ESDHCV2_PROCTL) for more information on the ESDHC Protocol Control register.

30.3.9.1 Setting Wake Up Events

For the ESDHC to respond to a wakeup event, the software must set the respective wakeup enable bit before the ARM platform enters sleep mode. Before the software disables the host clock, it should ensure that all of the following conditions have been met:

- No Read or Write Transfer is active
- Data and Command lines are not active
- No interrupts are pending
- Internal data buffer is empty

30.3.10 MMC Fast Boot

The Embedded Multimedia Card (eMMC 4.3) has a fast boot feature. In boot operation mode, the master (MultiMediaCard host) can read boot data from the slave (MMC device) by keeping CMD line low after power-on, or sending CMD0 with argument + 0xFFFFFFFA (optional for slave), before issuing CMD1.

There are two supported types of fast boot mode, boot operation and alternative boot operation. Each type has with-acknowledge and without-acknowledge modes.

NOTE

For the eMMC 4.3 card setting, please see the eMMC 4.3 spec.

30.3.10.1 Boot Operation

NOTE

In this block guide, this fast boot is called normal fast boot mode.

If the CMD line is held LOW for 74 clock cycles and more after power-up before the first command is issued, the slave recognizes that boot mode is being initiated and starts preparing boot data internally.

Within 1 second after the CMD line goes LOW, the slave starts to send the first boot data to the master on the DAT line(s). The master must keep the CMD line LOW to read all of the boot data.

If boot acknowledge is enabled, the slave has to send acknowledge pattern '010' to the master within 50ms after the CMD line goes LOW. If boot acknowledge is disabled, the slave will not send out acknowledge pattern '010'.

The master can terminate boot mode with the CMD line HIGH.

Boot operation will be terminated when all contents of the enabled boot data are sent to the master. After boot operation is executed, the slave shall be ready for CMD1 operation and the master needs to start a normal MMC initialization sequence by sending CMD1.

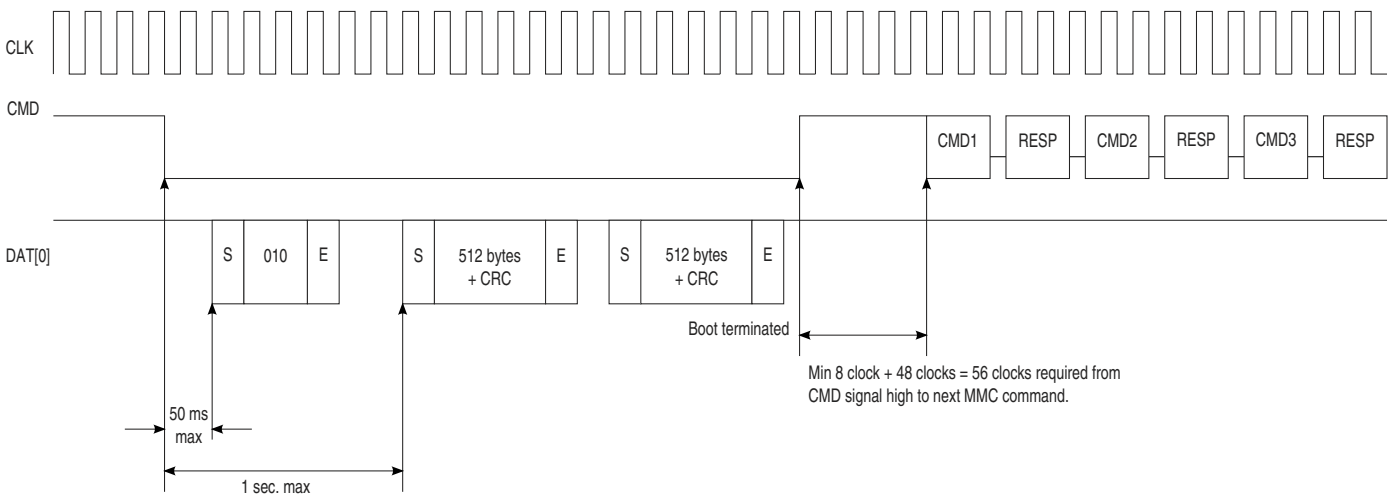


Figure 30-16. MultiMediaCard state diagram (normal boot mode)

30.3.10.2 Alternative Boot Operation

If bit 0 in the extended CSD byte[228] is set to '1', the device supports the alternative boot operation.

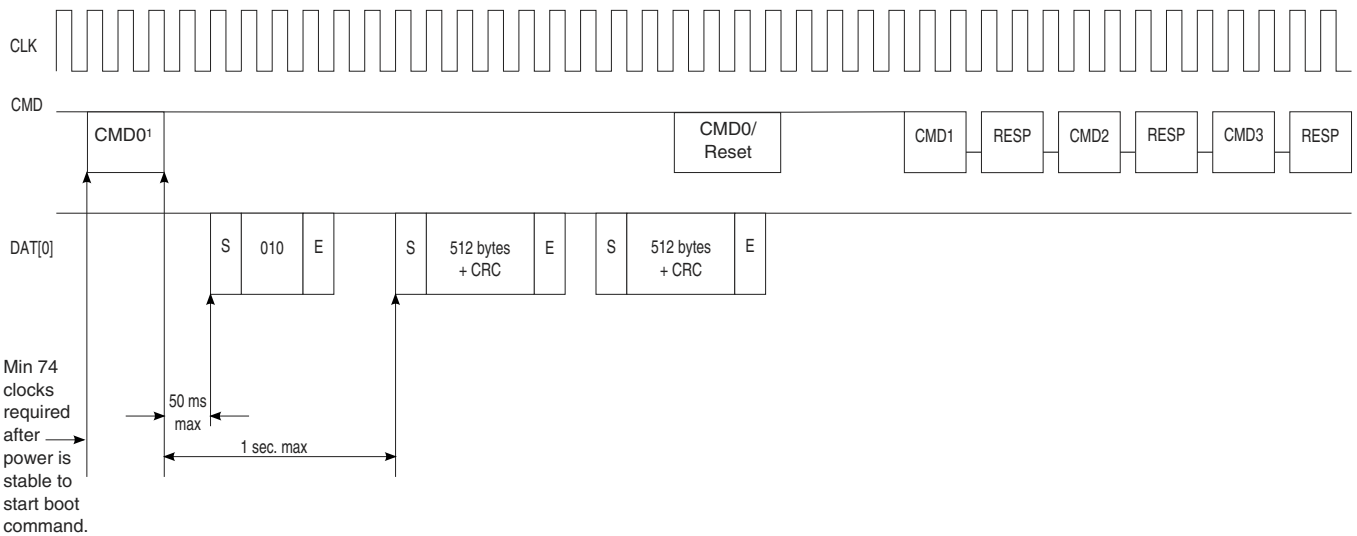
After power-up, if the host issues CMD0 with the argument of 0xFFFFFFFA after 74 clock cycles, before CMD1is issued or the CMD line goes low, the slave recognizes that boot mode is being initiated and starts preparing boot data internally.

Within 1 second after CMD0 with the argument of 0xFFFFFFFA is issued, the slave starts to send the first boot data to the master on the DAT line(s).

If boot acknowledge is enabled, the slave has to send the acknowledge pattern '010' to the master within 50ms after the CMD0 with the argument of 0xFFFFFFFA is received. If boot acknowledge is disabled, the slave will not send out acknowledge pattern '010'.

The master can terminate boot mode by issuing CMD0 (Reset).

Boot operation will be terminated when all contents of the enabled boot data are sent to the master. After boot operation is executed, the slave shall be ready for CMD1 operation and the master needs to start a normal MMC initialization sequence by sending CMD1.



NOTE1. CMD0 with argument 0xFFFFFFFA

Figure 30-17. MultiMediaCard state diagram (alternative boot mode)

30.4 Initialization/Application of ESDHC

All communication between system and cards are controlled by the host. The host sends commands of two types: broadcast and addressed (point-to-point).

Broadcast commands are intended for all cards, such as "GO_IDLE_STATE", "SEND_OP_COND", "ALL_SEND_CID" and etc. In Broadcast mode, all cards are in the open-drain mode to avoid bus contention. Refer to Commands for MMC/SD/SDIO for the commands of bc and bcr categories.

After the Broadcast command CMD3 is issued, the cards enter standby mode. Addressed type commands are used from this point. In this mode, the CMD/DAT I/O pads will turn to push-pull mode, to have the driving capability for maximum frequency operation. Refer to Commands for MMC/SD/SDIO for the commands of ac and adtc categories.

30.4.1 Command Send and Response Receive Basic Operation

Assuming the data type WORD is an unsigned 32-bit integer, the below flow is a guideline for sending a command to the card(s):

```
send_command(cmd_index, cmd_arg, other requirements)
{
WORD wCmd; // 32-bit integer to make up the data to write into Transfer Type register, it is
recommended to implement in a bit-field manner
wCmd = (<cmd_index> & 0x3f) >> 24; // set the first 8 bits as '00'+<cmd_index>
set CMDTYP, DPSEL, CICEN, CCCEN, RSTTYP, DTDSEL accorind to the command index;
if (internal DMA is used) wCmd |= 0x1;
if (multi-block transfer) {
          set MSBSEL bit;
          if (finite block number) {
                    set BCEN bit;
                    if (auto12 command is to use) set AC12EN bit;
          }
}
write_reg(CMDARG, <cmd_arg>); // configure the command argument
write_reg(XFERTYP, wCmd); // set Transfer Type register as wCmd value to issue the command
}
wait_for_response(cmd_index)
{
while (CC bit in IRQ Status register is not set); // wait until Command Complete bit is set
read IRQ Status register and check if any error bits about Command are set
if (any error bits are set) report error;
write 1 to clear CC bit and all Command Error bits;
}
```

For the sake of simplicity, the function wait_for_response is implemented here by means of polling. For an effective and formal way, the response is usually checked after the Command Complete Interrupt is received. By doing this, make sure the corresponding interrupt status bits are enabled.

For some scenarios, the response time-out is expected. For instance, after all cards respond to CMD3 and go to the Standby State, no response to the Host when CMD2 is sent. The Host Driver shall deal with 'fake' errors like this with caution.

30.4.2 Card Identification Mode

When a card is inserted to the socket or the card was reset by the host, the host needs to validate the operation voltage range, identify the cards, request the cards to publish the Relative Card Address (RCA) or to set the RCA for the MMC cards.

30.4.2.1 Card Detect

See figure below for a flow diagram showing the detection of MMC, SD and SDIO cards using the ESDHC.

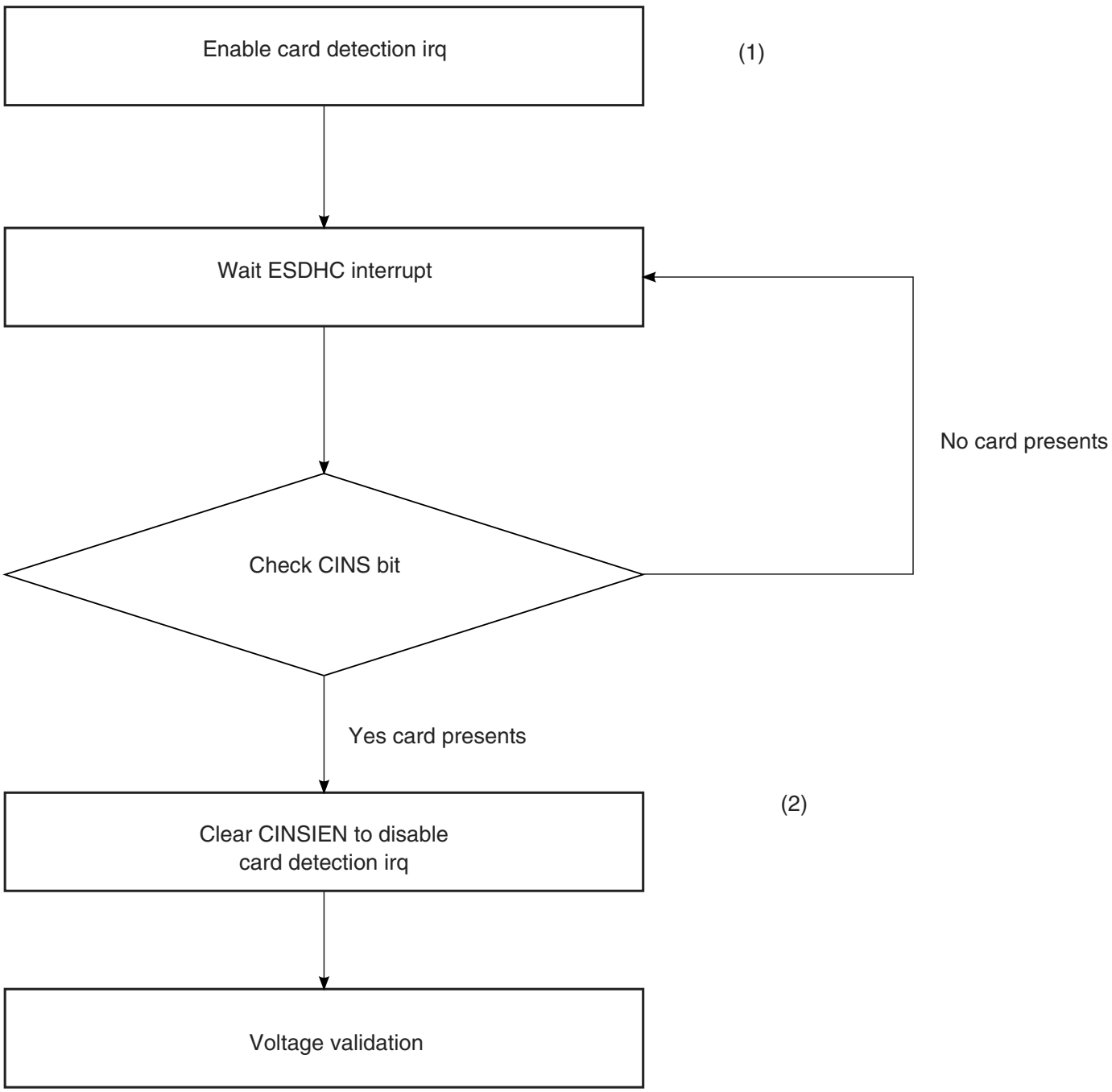


Figure 30-18. Flow Diagram for Card Detection

Here is the card detect sequence:

- Set the CINSIEN bit to enable card detection interrupt
- When an interrupt from the ESDHC is received, check the CINS bit in the Interrupt Status register to see if it was caused by card insertion
- Clear the CINSIEN bit to disable the card detection interrupt and ignore all card insertion interrupts afterwards

30.4.2.2 Reset

The host consists of three types of resets:

- Hardware reset (Card and Host) which is driven by POR (Power On Reset)
- Software reset (Host Only) is proceed by the write operation on the RSTD, RSTC, or RSTA bits of the System Control register to reset the data part, command part, or all parts of the Host Controller, respectively
- Card reset (Card Only). The command, "Go_Idle_State" (CMD0), is the software reset command for all types of MMC cards, SD Memory cards. This command sets each card into the Idle State regardless of the current card state. For an SD I/O Card, CMD52 is used to write an I/O reset in the CCCR. The cards are initialized with a default relative card address (RCA=0x0000) and with a default driver stage register setting (lowest speed, highest driving current capability).

After the card is reset, the host needs to validate the voltage range of the card. See figure below for the software flow to reset both the ESDHC and the card.

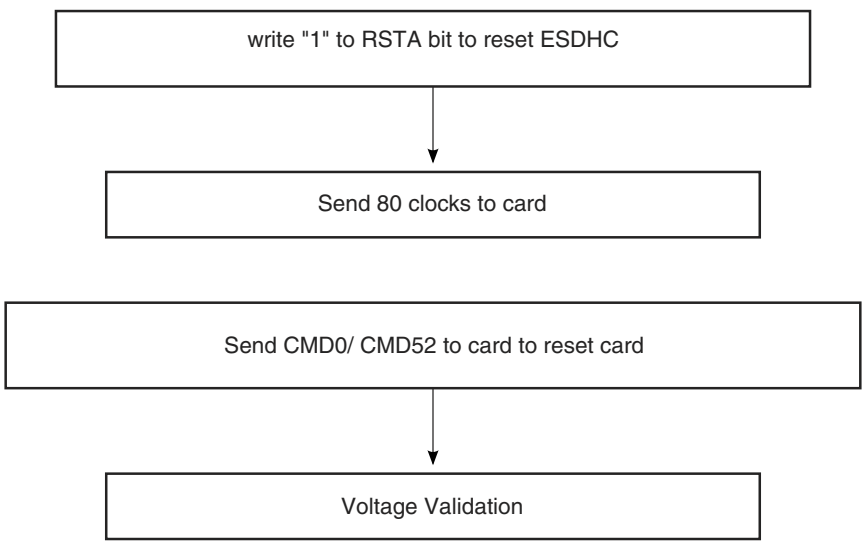


Figure 30-19. Flow Chart for Reset of the ESDHC and SD I/O Card

```
software_reset()
{
set_bit(SYSCTRL, RSTA); // software reset the Host
set DVS and SDCLKFS bit fields to get the SD_CLK of frequency around 400kHz
configure IO pad to set the power voltage of external card to around 3.0V
poll bits CIHB and CDIHB bits of PRSSTAT to wait both bits are cleared
set_bit(SYSCTRL, INTIA); // send 80 clock ticks for card to power up
send_command(CMD_GO_IDLE_STATE, <other parameters>); // reset the card with CMD0
or send_command(CMD_IO_RW_DIRECT, <other parameters>);
}
```

30.4.2.3 Voltage Validation

All cards should be able to establish communication with the host using any operation voltage in the maximum allowed voltage range specified in the card specification. However, the supported minimum and maximum values for Vdd are defined in the Operation Conditions Register (OCR) and may not cover the whole range. Cards that store the CID and CSD data in the preload memory are only able to communicate this information under data transfer Vdd conditions. This means if the host and card have non-common Vdd ranges, the card will not be able to complete the identification cycle, nor will it be able to send CSD data.

Therefore, a special command Send_Op_Cont (CMD1 for MMC), SD_Send_Op_Cont (ACMD41 for SD Memory) and IO_Send_Op_Cont (CMD5 for SD I/O) is used. The voltage validation procedure is designed to provide a mechanism to identify and reject cards which do not match the Vdd range(s) desired by the host. This is accomplished by the host sending the desired Vdd voltage window as the operand of this command. Cards that can't perform the data transfer in the specified range must discard themselves from further bus operations and go into the Inactive State. By omitting the voltage range in the command, the host can query each card and determine the common voltage range before sending out-of-range cards into the Inactive State. This query should be used if the host is able to select a common voltage range or if a notification shall be sent to the system when a non-usable card in the stack is detected.

The following steps show how to perform voltage validation when a card is inserted:

```
voltage_validation(voltage_range_arguement)
{
label the card as UNKNOWN;
send_command(IO_SEND_OP_COND, 0x0, <other parameters are omitted>); // CMD5, check SDIO
operation voltage, command argument is zero
if (RESP_TIMEOUT != wait_for_response(IO_SEND_OP_COND)) { // SDIO command is accepted
          if (0 < number of IO functions) {
                    label the card as SDIO;
                    IORDY = 0;
                    while (!(IORDY in IO OCR response)) { // set voltage range for each IO
function
                              send_command(IO_SEND_OP_COND, <voltage range>, <other
parameter>);
                              wait_for_response(IO_SEND_OP_COND);
                    } // end of while ...
          } // end of if (0 < ...
          if (memory part is present inside SDIO card) Label the card as SDCombo; // this is
an
SD-Combo card
} // end of if (RESP_TIMEOUT ...
if (the card is labelled as SDIO card) return; // card type is identified and voltage range
is
set, so exit the function;
send_command(APP_CMD, 0x0, <other parameters are omitted>); // CMD55, Application specific
CMD
prefix
if (no error calling wait_for_response(APP_CMD, <...>) { // CMD55 is accepted
          send_command(SD_APP_OP_COND, <voltage range>, <...>); // ACMD41
```

```
range
for memory part or SD card
          wait_for_response(SD_APP_OP_COND); // voltage range is set
          if (card type is UNKNOWN) label the card as SD;
          return; //
} // end of if (no error ...
else if (errors other than time-out occur) { // command/response pair is corrupted
          deal with it by program specific manner;
} // of else if (response time-out
else { // CMD55 is refuse, it must be MMC card           if (card is already labelled as
SDCombo) { // change label
                    re-label the card as SDIO;
                    ignore the error or report it;
                    return; // card is identified as SDIO card
          } // of if (card is ...
          send_command(SEND_OP_COND, <voltage range>, <...>);
          if (RESP_TIMEOUT == wait_for_response(SEND_OP_COND)) { // CMD1 is not accepted,
either
                    label the card as UNKNOWN;
                    return;
          } // of if (RESP_TIMEOUT ...
          if (check for CE-ATA signature succeeded) { // the card is CE-ATA
                    store CE-ATA specific info from the signature;
                    label the card as CE-ATA;
          } // of if (check for CE-ATA ...
          else label the card as MMC;
} // of else
}
```

30.4.2.4 Card Registry

Card registry for the MMC and SD/SDIO/SD Combo cards are different.

For the SD Card, the Identification process starts at a clock rate lower than 400 kHz and the power voltage higher than 2.7 V (as defined by the Card spec). At this time, the CMD line output drives are push-pull drivers instead of open-drain. After the bus is activated, the host will request the card to send their valid operation conditions. The response to ACMD41 is the operation condition register of the card. The same command shall be send to all of the new cards in the system. Incompatible cards are put into the Inactive State. The host then issues the command, All_Send_CID (CMD2), to each card to get its unique card identification (CID) number. Cards that are currently unidentified (in the Ready State), send their CID number as the response. After the CID is sent by the card, the card goes into the Identification State.

The host then issues Send_Relative_Addr (CMD3), requesting the card to publish a new relative card address (RCA) that is shorter than the CID. This RCA will be used to address the card for future data transfer operations. Once the RCA is received, the card changes its state to the Standby State. At this point, if the host wants the card to have an alternative RCA number, it may ask the card to publish a new number by sending another Send_Relative_Addr command to the card. The last published RCA is the actual RCA of the card.

The host repeats the identification process with CMD2 and CMD3 for each card in the system until the last CMD2 gets no response from any of the cards in system.

For MMC operation, the host starts the card identification process in open-drain mode with the identification clock rate lower than 400 kHz and the power voltage higher than 2.7 V. The open drain driver stages on the CMD line allow parallel card operation during card identification. After the bus is activated the host will request the cards to send their valid operation conditions (CMD1). The response to CMD1 is the "wired OR" operation on the condition restrictions of all cards in the system. Incompatible cards are sent into the Inactive State. The host then issues the broadcast command All_Send_CID (CMD2), asking all cards for their unique card identification (CID) number. All unidentified cards (the cards in Ready State) simultaneously start sending their CID numbers serially, while bit-wise monitoring their outgoing bit stream. Those cards, whose outgoing CID bits do not match the corresponding bits on the command line in any one of the bit periods, stop sending their CID immediately and must wait for the next identification cycle. Since the CID is unique for each card, only one card can be successfully send its full CID to the host. This card then goes into the Identification State. Thereafter, the host issues Set_Relative_Addr (CMD3) to assign to the card a relative card address (RCA). Once the RCA is received the card state changes to the Stand-by State, and the card does not react in further identification cycles, and its output driver switches from open-drain to push-pull. The host repeats the process, mainly CMD2 and CMD3, until the host receives a time-out condition to recognize the completion of the identification process.

For operation as MMC cards:

```
card_registry()
{
do { // decide RCA for each card until response time-out
          if(card is labelled as SDCombo or SDIO) { // for SDIO card like device
                    send_command(SET_RELATIVE_ADDR, 0x00, <...>); // ask SDIO card to
publish its
RCA
                    retrieve RCA from response;
          } // end if (card is labelled as SDCombo ...
          else if (card is labelled as SD) { // for SD card
                    send_command(ALL_SEND_CID, <...>);
                    if (RESP_TIMEOUT == wait_for_response(ALL_SEND_CID)) break;
                    send_command(SET_RELATIVE_ADDR, <...>);
                    retrieve RCA from response;
          } // else if (card is labelled as SD ...
          else if (card is labelled as MMC or CE-ATA) { // treat CE-ATA as MMC
                    send_command(ALL_SEND_CID, <...>);
                    rca = 0x1; // arbitrarily set RCA, 1 here for example, this RCA is also
the
relative address to access the CE-ATA card
                    send_command(SET_RELATIVE_ADDR, 0x1 << 16, <...>); // send RCA at upper
16
bits
          } // end of else if (card is labelled as MMC ...
} while (response is not time-out);
}
```

30.4.3 Card Access

30.4.3.1 Block Write

30.4.3.1.1 Normal Write

During a block write (CMD24 - 27, CMD60, CMD61), one or more blocks of data are transferred from the host to the card with a CRC appended to the end of each block by the host. If the CRC fails, the card shall indicate the failure on the DAT line. The transferred data will be discarded and not written, and all further transmitted blocks (in multiple block write mode) will be ignored.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed (CSD parameter WRITE_BLK_MISALIGN is not set), the card detects the block misalignment error and aborts the programming before the beginning of the first misaligned block. The card sets the ADDRESS_ERROR error bit in the status register, and while ignoring all further data transfer, waits in the Receive-data-State for a stop command. The write operation is also aborted if the host tries to write over a write protected area.

For MMC and SD cards, programming of the CID and CSD registers does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If this match fails, then the card will report an error and not change any register contents.

For all types of cards, some may require long and unpredictable periods of time to write a block of data. After receiving a block of data and completing the CRC check, the card will begin writing and hold the DAT line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command (CMD13) or other means for SDIO cards at any time, and the card will respond with its status. The responded status indicates whether the card can accept new data or whether the write process is still in progress. The host may deselect the card by issuing a CMD7 (to select a different card) to place the card into the Standby State and release the DAT line without interrupting the write operation. When re-selecting the card, it will reactivate the busy indication by pulling DAT to low if the programming is still in progress and the write buffer is unavailable.

The software flow to write to a card incorporates the internal DMA and the write operation is a multi-block write with the Auto CMD12 enabled. For the other two methods (by means of external DMA or ARM platform polling status) with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to write to a card is described below:

1. Check the card status, wait until the card is ready for data.
2. Set the card block length/size:
   a. For SD/MMC cards, use SET_BLOCKLEN (CMD16)
   b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT (CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7)
3. Set the ESDHC block length register to be the same as the block length set for the card in Step 2.
4. Set the ESDHC block attribute register, block number is 5 (for instance).
5. Disable the buffer write ready interrupt, configure the DMA settings and enable the ESDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a write CRC error occurred, or some another error, that occurred during the auto12 command sending and response receiving.

30.4.3.1.2 DDR Write

ESDHC supports dual data rate mode.

The software flow to write to a card in ddr mode is described as below:

1. Check the card status, wait until the card is ready for data.
2. For eMMC4.4 card, block length only can be set to 512byte.
3. Set the ESDHC number block register (NOB), nob is 5 (for instance).
4. Set eMMC4.4 card to high speed mode, use SWITCH(CMD6).
5. Set eMMC4.4 card bus with (4-bit /8-bit ddr mode), use SWITCH(CMD6).
6. Disable the buffer write ready interrupt, configure the DMA settings and enable the ESDHC DMA when sending the command with data transfer.The DDR_EN_IPG bit should be set. The AC12EN bit should also be set.
7. Wait for the Transfer Complete interrupt.
8. Check the status bit to see if a write CRC error occurred, or some another error, that occurred during the auto12 command sending and response receiving.

30.4.3.1.3 Write with Pause

The write operation can be paused during the transfer. Instead of stopping the SD_CLK at any time to pause all the operations, which is also inaccessible to the Host Driver, the Driver can set the SABGREQ bit in the Protocol Control register to pause the transfer

between the data blocks. As there is no time-out condition in a write operation during the data blocks, a write to all types of cards can be paused in this way, and if the DAT0 line is not required to de-assert to release the busy state, no suspend command is needed.

Like in the flow described in Normal Write, the write with pause is shown with the same kind of write operation:

1. Check the card status, wait until card is ready for data.
2. Set the card block length/size:
   a. For SD/MMC, use SET_BLOCKLEN (CMD16)
   b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7)
3. Set the ESDHC block length register to be the same as the block length set for the card in Step 2.
4. Set the ESDHC number block register (NOB), nob is 5 (for instance).
5. Disable the buffer write ready interrupt, configure the DMA settings and enable the ESDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Set the SABGREQ bit.
7. Wait for the Transfer Complete interrupt.
8. Clear the SABGREQ bit.
9. Check the status bit to see if a write CRC error occurred.
10. Set the CREQ bit to continue the write operation.
11. Wait for the Transfer Complete interrupt.
12. Check the status bit to see if a write CRC error occurred, or some another error, that occurred during the auto12 command sending and response receiving.

The number of blocks left during the data transfer is accessible by reading the contents of the BLKCNT field in the Block Attribute register. As the data transfer and the setting of the SABGREQ bit are concurrent, and the delay of register read and the register setting, the actual number of blocks left may not be exactly the value read earlier. The Driver shall read the value of BLKCNT after the transfer is paused and the Transfer Complete interrupt is received.

It is also possible the last block has begun when the Stop At Block Gap Request is sent to the buffer. In this case, the next block gap is actually the end of the transfer. These types of requests are ignored and the Driver should treat this as a non-pause transfer and deal with it as a common write operation.

When the write operation is paused, the data transfer inside the Host System is not stopped, and the transfer is active until the data buffer is full. Because of this (if not needed), it is recommended to avoid using the Suspend Command for the SDIO card; when such a command is sent, the ESDHC thinks the System will switch to another

function on the SDIO card, and flush the data buffer. The ESDHC takes the Resume Command as a normal command with data transfer, and it is left for the Driver to set all the relevant registers before the transfer is resumed. If there is only one block to send when the transfer is resumed, the MSBSEL and BCEN bits of the Transfer Type register are set as well as the AC12EN bit. However, the ESDHC will automatically send a CMD12 to mark the end of the multi-block transfer.

30.4.3.2 Block Read

30.4.3.2.1 Normal Read

For block reads, the basic unit of data transfer is a block whose maximum size is stored in areas defined by the corresponding card specification. A CRC is appended to the end of each block, ensuring data transfer integrity. The CMD17, CMD18, CMD53, CMD60, CMD61, and so on, can initiate a block read. After completing the transfer, the card returns to the Transfer State. For multi blocks read, data blocks will be continuously transferred until a stop command is issued.

The software flow to read from a card incorporates the internal DMA and the read operation is a multi-block read with the Auto CMD12 enabled. For the other two methods (by means of external DMA or ARM platform polling status) with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to read from a card is described below:

1. Check the card status, wait until card is ready for data.
2. Set the card block length/size:
   a. For SD/MMC, use SET_BLOCKLEN (CMD16)
   b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7)
3. Set the ESDHC block length register to be the same as the block length set for the card in Step 2.
4. Set the ESDHC number block register (NOB), nob is 5 (for instance).
5. Disable the buffer read ready interrupt, configure the DMA settings and enable the ESDHC DMA when sending the command with data transfer. The AC12EN bit should also be set:
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a read CRC error occurred, or some another error, occurred during the auto12 command sending and response receiving.

30.4.3.2.2 DDR Read

ESDHC supports dual data rate mode.

The software flow to write to a card in ddr mode is described below:

1. Check the card status, wait until the card is ready for data.
2. For eMMC4.4 card, block length only can be set to 512byte.
3. Set the ESDHC number block register (NOB), nob is 5 (for instance).
4. Set eMMC4.4 card to high speed mode, use SWITCH(CMD6).
5. Set eMMC4.4 card bus with (4-bit /8-bit ddr mode), use SWITCH(CMD6).
6. Disable the buffer write ready interrupt, configure the DMA settings and enable the ESDHC DMA when sending the command with data transfer.The DDR_EN_IPG bit should be set. The AC12EN bit should also be set.
7. Wait for the Transfer Complete interrupt.
8. Check the status bit to see if a write CRC error occurred, or some another error, that occurred during the auto12 command sending and response receiving.

30.4.3.2.3 Read with Pause

The read operation is not generally able to pause. Only the SDIO card (and SDCombo card working under I/O mode) supporting the Read Wait feature can pause during the read operation. If the SDIO card supports Read Wait (SRW bit in CCCR register is 1), the Driver can set the SABGREQ bit in the Protocol Control register to pause the transfer between the data blocks. Before setting the SABGREQ bit, make sure the RWCTL bit in the Protocol Control register is set, otherwise the ESDHC will not assert the Read Wait signal during the block gap and data corruption occurs. It is recommended that the RWCTL bit be set once the Read Wait capability of the SDIO card is recognized.

Like in the flow described in Normal Read, the read with pause is shown with the same kind of read operation:

1. Check the SRW bit in the CCR register on the SDIO card to confirm the card supports Read Wait.
2. Set the RWCTL bit.
3. Check the card status and wait until the card is ready for data.
4. Set the card block length/size:
   a. For SD/MMC, use SET_BLOCKLEN (CMD16)
   b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7)
5. Set the ESDHC block length register to be the same as the block length set for the card in Step 2.

6. Set the ESDHC number block register (NOB), nob is 5 (for instance).
7. Disable the buffer read ready interrupt, configure the DMA setting and enable the ESDHC DMA when sending the command with data transfer. The AC12EN bit should also be set
8. Set the SABGREQ bit.
9. Wait for the Transfer Complete interrupt.
10. Clear the SABGREQ bit.
11. Check the status bit to see if read CRC error occurred.
12. Set the CREQ bit to continue the read operation.
13. Wait for the Transfer Complete interrupt.
14. Check the status bit to see if a read CRC error occurred, or some another error, occurred during the auto12 command sending and response receiving.

Like the write operation, it is possible to meet the ending block of the transfer when paused. In this case, the ESDHC will ignore the Stop At Block Gap Request and treat it as a command read operation.

Unlike the write operation, there is no remaining data inside the buffer when the transfer is paused. All data received before the pause will be transferred to the Host System. No matter if the Suspend Command is sent or not, the internal data buffer is not flushed.

If the Suspend Command is sent and the transfer is later resumed by means of a Resume Command, the ESDHC takes the command as a normal one accompanied with data transfer. It is left for the Driver to set all the relevant registers before the transfer is resumed. If there is only one block to send when the transfer is resumed, the MSBSEL and BCEN bits of the Transfer Type register are set, as well as the AC12EN bit. However, the ESDHC will automatically send the CMD12 to mark the end of multi-block transfer.

30.4.3.2.4 DLL (Delay Line) in Read Path

The DLL (Delay Line) is added to assist in sampling read data. As in The DLL provides the ability to programmatically select a quantized delay (in fractions of the clock period) regardless of on-chip variations such as process, voltage and temperature (PVT). The reasons why the DLL is needed for ESDHC are 1.) the path of read data traveling from card to host varies. 2.) in MMC DDR mode the minimum input setup and hold time are both at 2.5 ns. The data sampling window is so small that the delay of loopback clock needs to be accurate and consistent regardless of PVT. The DLL takes the divided perclk as the reference clock and loopback clock as the input clock. It then generates a delayed version of the input clock according to the programmed target delay.

The DLL can be disabled or bypassed, and it can also be manually set for a fixed delay in override mode. The override value set is the number of delay cells. In override mode, the DLL_enable is no need to set. And another working mode of DLL is target value mode. In this mode, DLL will automatically adjust the number of delay cells according to the target value your set value and PVT changes. Be aware that target value is in the unit of 1/32 clock reference period. If the perclk is 100MHz, then the reference clock period is 10ns, setting target value of 16 means 5ns=(16/32)*10ns. Software can disable automatically update by setting dll_gate_update bit. Please refer to Figure 30-20.

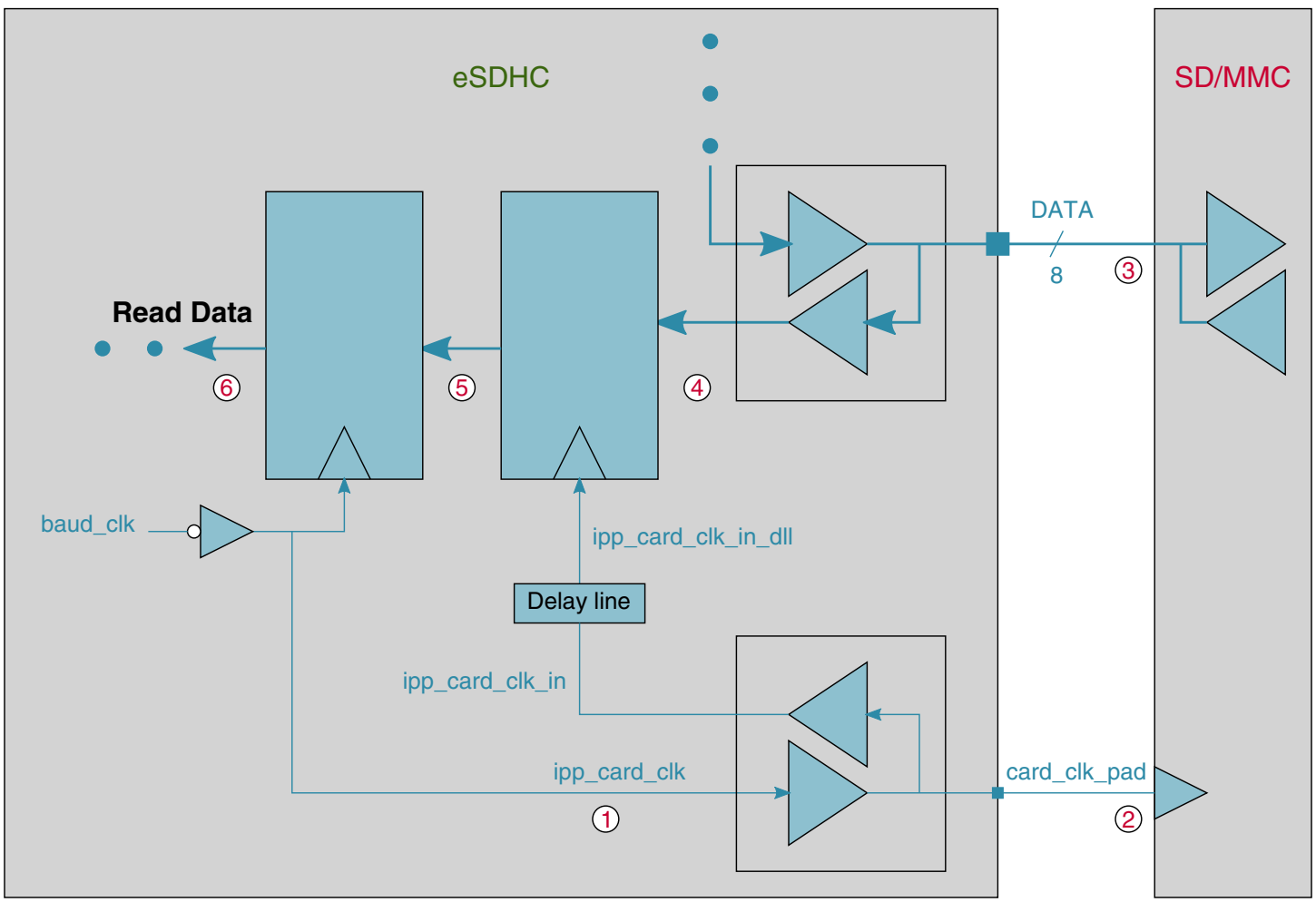


Figure 30-20. DLL (Delay Line) in Read Path

30.4.3.3 Suspend Resume

The ESDHC supports the Suspend Resume operations of SDIO cards, although slightly different than the suggested implementation of Suspend in the SDIO card specification.

Suspend

After setting the SABGREQ bit, the Host Driver may send a Suspend command to switch to another function of the SDIO card. The ESDHC does not monitor the content of the response, so it doe not know whether or not the Suspend command has succeeded. Accordingly, it does not de-assert Read Wait for read pause. To solve this problem, the Driver does not set the ESDHC_XFERTYP[CMDTYP] register to 01, that is, Suspend option. Instead, the Driver sends this command as if it were a normal command (that is, sets CMDTYP to b00). Only when the command succeeds, and the BS bit is set in the response, does the Driver send another command marked as "Suspend" to inform the ESDHC that the current transfer is suspended. This is shown in the following sequence for Suspend operation:

1. Set the SABREQ bit to pause the current data transfer at block gap.
2. After the BGE bit is set, send the Suspend command to suspend the active function. The CMDTYP bit field must be 2'b00.
3. Check the BS bit of the CCCR in the response. If it is 1, repeat this step until the BS bit is cleared or abandon the suspend operation according to the Driver strategy.
4. Send another normal I/O command to the suspended function. The CMDTYP of this command must be 2'b01, so the ESDHC can detect this special setting and be informed that the paused operation has successfully suspended. If the paused transfer is a read operation, the ESDHC stops driving DAT2 and goes to the idle state.
5. Save the context registers in the system memory for later use, including the DMA System Address Register (for internal DMA operation), and the Block Attribute Register.
6. Begin operation for another function on the SDIO card.

30.4.3.3.1 Resume

To resume the data transfer, a Resume command shall be issued:

1. To resume the suspended function, restore the context register with the saved value in step #5 of the Suspend operation above.
2. Send the Resume command. In the Transfer Type register, all bit fields are set to the value as if this were another ordinary data transfer, instead of a transfer resume (except the CMDTYP is set to 2'b10).
3. If the Resume command has responded, the data transfer will be resumed.

30.4.3.4 ADMA1 Usage

To use the ADMA1 in a data transfer, the Host Driver must prepare the correct descriptor chain prior to sending the read/write command. The steps to accomplish this are:

1. Create a descriptor to set the data length that the current descriptor group is about to transfer. The data length should be even numbers of the block size.
2. Create another descriptor to transfer the data from the address setting in this descriptor. The data address must be at a page boundary (4kB address aligned).
3. If necessary, create a Link descriptor containing the address of the next descriptor. The descriptor group is created in steps 1 ~ 3.
4. Repeat steps 1 ~ 3 until all descriptors are created.
5. In the last descriptor, set the End flag to 1 and make sure the total length of all descriptors match the product of the block size and block number configured in the Block Attribute Register.
6. Set the ADMA System Address Register to the address of the first descriptor and set the DMAS field in the Protocol Control Register to 01 to select the ADMA.
7. Issue a write or read command with the DMAEN bit set to 1 in the Transfer Type Register.

Steps 1 ~ 5 are independent of step 6, so step 6 can finish before steps 1 ~ 5. Regarding the descriptor configuration, it is recommended not to use the Link descriptor as it requires extra system memory access.

30.4.3.5 Transfer Error

30.4.3.5.1 CRC Error

It is possible at the end of a block transfer, that a write CRC status error or read CRC error occurs. For this type of error the latest block received shall be discarded. This is because the integrity of the data block is not guaranteed. It is recommended to discard the following data blocks and re-transfer the block from the corrupted one. For a multi-block transfer, the Host Driver shall issue a CMD12 to abort the current process and start the transfer by a new data command. In this scenario, even when the AC12EN and BCEND bits are set, the ESDHC does not automatically send a CMD12 because the last block is not transferred. On the other hand, if it is within the last block that the CRC error occurs, an Auto CMD12 will be sent by the ESDHC. In this case, the Driver shall re-send or re-obtain the last block with a single block transfer.

30.4.3.5.2 Internal DMA Error

During the data transfer with internal Simple DMA, if the DMA engine encounters some error on the AHB bus, the DMA operation is aborted and DMA Error interrupt is sent to the Host System. When acknowledged by such an interrupt, the Driver shall calculate the start address of data block in which the error occurs. The start address can be calculated by either:

1. Read the DMA System Address register. The error occurs during the previous burst. Taking the block size, the previous burst length and the start address of the next burst transfer into account, it is straight forward to obtain the start address of the corrupted block.
2. Read the BLKCNT field of the Block Attribute register. By the number of blocks left, the total number to transfer, the start address of transfer, and the size of each block, the start address of corrupted block can be determined. When the BCEN bit is not set, the contents of the Block Attribute register does not change, so this method does not work.

When a DMA error occurs, it is recommended to abort the current transfer by means of a CMD12 (for multi block transfer), apply a reset for data, and re-start the transfer from the corrupted block to recover from the error.

30.4.3.5.3 ADMA Error-Card Access

There are three kinds of possible ADMA errors. The AHB transfer, invalid descriptor, and data-length mismatch errors. When these errors occur, the DMA transfer stops and the corresponding error status bit is set. For acknowledging the status, the Host Driver should recover the error as shown below and re-transfer from the place of interruption.

1. AHB transfer error: Such errors may occur during data transfer or descriptor fetch. For either scenario, it is recommended to retrieve the transfer context, reset for the data part and re-transfer the block that was corrupted, or the next block if no block is corrupted.
2. Invalid descriptor error: For such errors, it is recommended to retrieve the transfer context, reset for the data part and re-create the descriptor chain from the invalid descriptor and issue a new transfer. As the data to transfer now may be less than the previous setting, the data length configured in the new descriptor chain should match the new value.
3. Data-length mismatch error: It is similar to recover from this error. The Host Driver polls relating registers to retrieve the transfer context, apply a reset for the data part, configure a new descriptor chain, and make another transfer if there is data left. Like the previous scenario of the invalid descriptor error, the data length must match the new transfer.

30.4.3.5.4 Auto CMD12 Error

After the last block of the multi block transfer is sent or received, and the AC12EN bit is set when the data transfer is initiated by the data command, the ESDHC automatically sends a CMD12 to the card to stop the transfer. When errors with this command occur, it is recommended to the Driver to deal with the situations in the following manner:

1. Auto CMD12 response time-out. It is not certain whether the command is accepted by the card or not. The Driver should clear the Auto CMD12 error status bits and re-send the CMD12 until it is accepted by the card.
2. Auto CMD12 response CRC error. Since card responds to the CMD12, the card will abort the transfer. The Driver may ignore the error and clear the error status bit.
3. Auto CMD12 conflict error or not sent. The command is not sent, so the Driver shall send a CMD12 manually.

30.4.3.6 Card Interrupt

The external cards can inform the Host Controller by means of some special signals. For the SDIO card, it can be the low level on the DAT[1] line during some special period. The ESDHC only monitors the DAT[1] line and supports the SDIO interrupt.

When the SDIO interrupt is captured by the ESDHC, and the Host System is informed by the ESDHC asserting the ESDHC interrupt line, the interrupt service from the Host Driver is called.

As the interrupt factor is controlled by the external card, the interrupt from the SDIO card must be served before the CINT bit is cleared by written 1. Refer to Card Interrupt Handling for the card interrupt handling flow.

30.4.4 Switch Function

MMC cards transferring data at bus widths other than 1-bit is a feature in MMC spec. The high speed timing mode for all card devices, was also defined in various card specifications. To enable these features, a "switch" command shall be issued by the Host Driver.

For SDIO cards, the high speed mode is enabled by writing the EHS bit in the CCCR register after the SHS bit is confirmed. For SD cards, the high speed mode is queried and enabled by a CMD6 (with the mnemonic symbol as SWITCH_FUNC). For MMC cards, the high speed mode is queried by a CMD8 and enabled by a CMD6 (with the mnemonic symbol as SWITCH).

The 4-bit and 8-bit bus width of the MMC is also enabled by the SWITCH command, but with a different argument.

DDR mode is also selected by setting ddr mode bus width via SWITCH.

These new functions can also be disabled by a software reset. For SDIO cards it can be done by setting the RES bit in the CCCR register. For other cards, it can be accomplished by issuing a CMD0. This method of restoring to the normal mode is not recommended because a complete identification process is needed before the card is ready for data transfer.

For the sake of simplicity, the following flowcharts do not show current capability check, which is recommended in the function switch process.

30.4.4.1 Query, Enable and Disable SDIO High Speed Mode

```
enable_sdio_high_speed_mode(void)
{
send CMD52 to query bit SHS at address 0x13;
if (SHS bit is '0') report the SDIO card does not support high speed mode and return;
send CMD52 to set bit EHS at address 0x13 and read after write to confirm EHS bit is set;
change clock divisor value or configure the system clock feeding into ESDHC to generate the
card_clk of around 50MHz;
(data transactions like normal peers)
}
disable_sdio_high_speed_mode(void)
{
send CMD52 to clear bit EHS at address 0x13 and read after write to confirm EHS bit is
cleared;
change clock divisor value or configure the system clock feeding into ESDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

30.4.4.2 Query, Enable and Disable SD High Speed Mode

```
enable_sd_high_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0xFFFFF1 and read 64 bytes of data accompanying the R1 response;
wait data transfer done bit is set;
check if the bit 401 of received 512 bit is set;
if (bit 401 is '0') report the SD card does not support high speed mode and return;
send CMD6, with argument 0x80FFFFF1 and read 64 bytes of data accompanying the R1 response;
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into ESDHC to generate the
card_clk of around 50MHz;
(data transactions like normal peers)
}
disable_sd_high_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0x80FFFFF0 and read 64 bytes of data accompanying the R1 response;
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into ESDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

30.4.4.3 Query, Enable and Disable MMC High Speed Mode

```
enable_mmc_high_speed_mode(void)
{
send CMD9 to get CSD value of MMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the MMC does not support high speed mode and
return;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of MMC;
extract the value of CARD_TYPE field to check the 'high speed mode' in this MMC is 26MHz or
52MHz;
send CMD6 with argument 0x1B90100;
send CMD13 to wait card ready (busy line released);
send CMD8 to get EXT_CSD value of MMC;
check if HS_TIMING byte (byte number 185) is 1;
if (HS_TIMING is not 1) report MMC switching to high speed mode failed and return;
change clock divisor value or configure the system clock feeding into ESDHC to generate the
card_clk of around 26MHz or 52MHz according to the CARD_TYPE;
(data transactions like normal peers)
}
disable_mmc_high_speed_mode(void)
{
send CMD6 with argument 0x2B90100;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of MMC;
check if HS_TIMING byte (byte number 185) is 0;
if (HS_TIMING is not 0) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into ESDHC to generate the
card_clk of the desired value below 20MHz;
(data transactions like normal peers)
}
```

30.4.4.4 Set MMC Bus Width

```
change_mmc_bus_width(void)
{
send CMD9 to get CSD value of MMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the MMC does not support multiple bit width and
return;
send CMD6 with argument 0x3B70x00; (8-bit, x=2; 4-bit, x=1; 1-bit, x=0)
send CMD13 to wait card ready (busy line released);
(data transactions like normal peers)
}
```

30.4.5 ADMA Operation

30.4.5.1 ADMA1 Operation

```
Set_adma1_descriptor
{
if (to start data transfer) {
// Make sure the address is 4KB align.
Set 'Set' type descriptor;
{
```

```
Set Act bits to 01;
Set [31:12] bits data length (byte unit);
}
Set 'Tran' type descriptor;
{
Set Act bits to 10;
Set [31:12] bits address (4KB align);
}
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to 11;
Set [31:12] bits the next descriptor address (4KB align);
}
else { // other types of descriptor
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set End bit to 1;
}
if (to generate interrupt for this descriptor) {
Set Int bit to 1;
}
Set Valid bit to 1;
}
```

30.4.5.2 ADMA2 Operation

```
Set_adma2_descriptor
{
if (to start data transfer) {
// Make sure the address is 32-bit boundary (lower 2-bit are always '00').
Set higher 32-bit of descriptor for this data transfer initial address;
Set [31:16] bits data length (byte unit);
Set Act bits to '10';
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to '11';
// Make sure the address is 32-bit boundary (lower 2-bit are always set to '00').
Set higher 32-bit of descriptor for the next descriptor address;
}
else { // other types of descriptor
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set 'End' bit '1';
}
if (to generate interrupt for this descriptor) {
Set 'Int' bit '1';
}
Set the 'Valid' bit to '1';
}
```

30.4.6 Fast Boot Operation

30.4.6.1 Normal fast boot flow

1. Software need to configure init_active bit (system control register bit 27) to make sure 74 card clocks are finished.

2. Software need to configure MMC Boot Register (offset 0xc4) bit 6 to 1(enable boot), and bit 5 to 0(normal fast boot), and bit 4 to select the ack mode or not. If need to send through DMA mode, need to configure bit 7 to enable automatically stop at block gap feature. And need to configure bit 3-bit0 to select the ack timeout value according to the sd clk frequence.
3. Software then need to configure Block Attributes Register to set block size/no. If in ddr fast boot mode, block size only can be configure to 512byte.
4. Software need to configure Protocol control register to set DTW (data transfer width). If in ddr fast boot mode, DTW only can be configure to 4-bit/8-bit dataline mode.
5. Software need to configure Command Argument Register to set argument if needed (no need in normal fast boot).
6. Software need to configure Transfer Type Register to start the boot process. In normal boot mode, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN and DMAEN are kept default value. DPSEL bit is set to 1, DTDSEL is set to 1, MSBSEL is set to 1.

NOTE

DMAEN should be configured as 0 in polling mode. And if BCEN is configured as 1, better to configure blk no in Bock Attributes Register to the max value. And if in ddr fast boot mode, DDR_EN need to be set to 1.

7. When the step 6 is configured, boot process will begin. Software need to poll the data buffer ready status to read the data from buffer in time. If boot time-out happened (ack time out or the first data read time out), Interrupt will be triggered, and software need to configure MMC Boot Register to bit 6 to 0 to disable boot. Thus will make CMD high, and then after at least 56 clocks, it is ready to begin normal initialization process.
8. If no time out, software need to decide the data read is finished and then configure MMC Boot Register bit 6 to 0 to disable boot. This will make CMD line high and command completed asserted. After at least 56 clocks, it is ready to begin normal initialization process.
9. Reset the host and then can begin the normal process.

30.4.6.2 Alternative fast boot flow

1. Software need to configure init_active bit (system control register bit 27) to make sure 74 card clocks are finished.
2. Software need to configure MMC Boot Register (offset 0xc4) bit 6 to 1(enable boot), and bit 5 to 1(alternative boot), and bit 4 to select the ack mode or not. If need to send through DMA mode, need to configure bit 7 to enable automatically stop at

block gap feature. And need to configure bit 3-bit0 to select the ack timeout value according to the sd clk frequence.

3. Software then need to configure Block Attributes Register to set block size/no. If in ddr fast boot mode, block size only can be configure to 512byte.
4. Software need to configure Protocol control register to set DTW(data transfer width). If in ddr fast boot mode, DTW only can be configure to 4-bit/8-bit dataline mode.
5. Software need to configure Command Argument Register to set argument to 0xFFFFFFFA.
6. Software need to configure Transfer Type Register to start the boot process by CMD0 with 0xFFFFFFFA argument . In alternative boot, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN and DMAEN are kept default value. DPSEL bit is set to 1, DTDSEL is set to 1, MSBSEL is set to 1.

NOTE

DMAEN should be configured as 0 in polling mode. And if BCEN is configured as 1 in polling mode, better to configure blk no in Bock Attributes Register to the max value. And if in ddr fast boot mode, DDR_EN need to be set to 1.

7. When the step 6 is configured, boot process will begin. Software need to poll the data buffer ready status to read the data from buffer in time. If boot time out(ack data time out in 50ms or data time out in 1s), host will send out the interrupt and software need to send CMD0 with reset and then configure boot enable bit to 0 to stop this process. After command completed, configure MMC Boot Register bit 6 to 0 to disable boot. After at least 8 clocks from command completed, card is ready for identification step.
8. If no time out, software need to decide when to stop the boot process, and send out the CMD0 with reset and then after command completed, configure MMC Boot Register bit 6 to stop the process. After 8 clocks from command completed, slave(card) is ready for identification step.
9. Reset the host and then can begin the normal process.

30.4.6.3 Fast boot application case (in DMA mode)

In the boot application case, because the image destination and the image size are contained in the

beginning of the image, need to switch DMA parameters on the fly during MMC fast boot.

In fast boot, host can use ADMA2 (Advanced DMA2) with two destinations.

The detail flow:

1. Software need to configure init_active bit (system control register bit 27) to make sure 74 card clocks are finished.
2. Software need to configure MMC Boot Register (offset 0xc4) bit 6 to 1(enable boot); and bit 5 to 0(normal fast boot), to 1(alternative boot); and bit 4 to select the acknowledge mode or not. In DMA mode, configure bit 7 to 1 for enable automatically stop at block gap feature. Also configure bit31-bit16 to set the VALUE1 (value of block count that need to transfer first time), that host will stop at block gap when card block counter is equal to this value. And need to configure bit 3-bit0 to select the acknowledge timeout value according to the SD clock frequency.
3. Software then need to configure Block Attributes Register to set block size/no. If in DDR fast boot mode, block size only can be configure to 512byte. In DMA mode, it is better to set block number to the max value (16'hFFFF).
4. Software need to configure Protocol control register to set DTW (data transfer width). If in DDR fast boot mode, DTW can be only configured to 4-bit/8-bit data line mode.
5. Software enbables ADMA2 by configuring protocol control register bit9-bit8.
6. Software need to set at least three pairs ADMA2 descriptor in boot memory (i.e. in IRAM, at least 6 words). The first pair descriptor define the start address (i.e. IRAM) and data length (i.e. 512 bytes*VALUE1) of first part boot code. Software also need to set the second pair descriptor, the second start address (any value that is writeable), data length is suggest to set 1~2word (record as VAULE2).

NOTE

The second couple descriptor also transfer useful data even at lease 1 word. Because our ADMA2 can't support 0 data_length data transfer descriptor.

7. Software need to configure Command Argument Register to set argument to 0xFFFFFFFA in alternative fast boot, and don't need set in normal fast boot.
8. Software need to configure Transfer Type Register to start the boot process. CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN and DMAEN are kept default value. DPSEL bit is set to 1, DTDSEL is set to 1, and MSBSEL is set to 1. DMAEN is configured as 1 in DMA mode. And if BCEN is configured as 1, better to configure block no in Bock Attributes Register to the max value. And if in DDR fast boot mode, DDR_EN need to be set to 1.
9. When the step 8 is configured, boot process will begin. The first VAULE1 block number data has transfer. Software need to polling TC bit (bit1 in Interrupt Status Register) to determine first transfer is end. Also software need to polling BGE bit (bit2 in Interrupt Status Register) to determine if first transfer stop at block gap.
10. When TC, BGE bit is 1, SW can analyzes the first code of VAULE1 block, initializes the new memory device, if required, and sets the third pair of descriptors to define

the start address and length of the remaining part of boot code(VAULE3 the remain boot code block). Remember set the last descriptor with END.

11. Software need to configure MMC Boot Register (offset 0xc4) again. Set bit 6 to 1(enable boot); and bit 5 to 0(normal fast boot), to 1(alternative boot); and bit 4 to select the acknowledge mode or not. In DMA mode, configure bit 7 to 1 for enable automatically stop at block gap feature. Also configure bit31-bit16 to set the (VAULE1+1+VAULE3), that host will stop at block gap when card block counter is equal to this value. And need to configure bit 3-bit0 to select the acknowledge timeout value according to the SD clock frequency.
12. Software need to clear TC and BGE bit. And software need to clear SABGREQ (bit 16 in Protocol control register), and set CREQ (bit17 Protocol control register) to 1 to resume the data transfer. Host will transfer the VALUE2 and VAULE3 data to the destination that is set by descriptor.
13. Software need to polling BGE bit to determine if the fast boot is over.

NOTE

- When ADMA boot flow is started, for ESDHC, it is like a normal ADMA read operation.
- Need a few words length memory to keep descriptor.
- For the 1~2 word data in second descriptor setting, it is the useful data, so software need to deal the data due to the application case.

30.5 Commands for MMC/SD/SDIO

See table below for the list of commands for the MMC/SD/SDIO cards.

Refer to the corresponding specifications for more details about the command information.

There are four kinds of commands defined to control the MultiMediaCard:

1. broadcast commands (bc), no response.
2. broadcast commands with response (bcr), response from all cards simultaneously.
3. addressed (point-to-point) commands (ac), no data transfer on the DAT.
4. addressed (point-to-point) data transfer commands (adtc).

The Access Bits for the EXT_CSD Access Modes are shown in table below.

Table 30-2. Commands for MMC/SD/SDIO Cards

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD0	bc	[31:0] stuff bits	-	GO_IDLE_STATE	Resets all MMC and SD memory cards to idle state.
CMD1	bcr	[31:0] OCR without busy	R3	SEND_OP_COND	Asks all MMC and SD Memory cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks all cards to send their CID numbers on the CMD line.
CMD3[1]	ac	[31:6] RCA [15:0] stuff bits	R1 R6 (SDIO)	SET/ SEND_RELATIVE_ADDR	Assigns relative address to the card.
CMD4	bc	[31:0] DSR [15:0] stuff bits	-	SET_DSR	Programs the DSR of all cards.
CMD5	bc	[31:0] OCR without busy	R4	IO_SEND_OP_COND	Asks all SDIO cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD6[2]	adtc	[31] Mode 0: Check function 1: Switch function [30:8] Reserved for function groups 6 ~ 3 (All 0 or 0xFFFF) [7:4] Function group1 for command system [3:0] Function group2 for access mode	R1	SWITCH_FUNC	Checks switch ability (mode 0) and switch card function (mode 1). Refer to "SD Physical Specification V1.1" for more details.
CMD6[3]	ac	[31:26] Set to 0 [25:24] Access [23:16] Index [15:8] Value [7:3] Set to 0 [2:0] Cmd Set	R1b	SWITCH	Switches the mode of operation of the selected card or modifies the EXT_CSD registers. Refer to "The MultiMediaCard System Specification Version 4.0 Final draft 2" for more details.
CMD7	ac	[31:6] RCA [15:0] stuff bits	R1b	SELECT/ DESELECT_CARD	Toggles a card between the stand-by and transfer states or between the programming and disconnect states. In both cases, the card is selected by its own relative address and gets deselected by any other address. Address 0 deselects all.

Table continues on the next page...

Table 30-2. Commands for MMC/SD/SDIO Cards (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD8	adtc	[31:0] stuff bits	R1	SEND_EXT_CSD	The card sends its EXT_CSD register as a block of data, with a block size of 512 bytes.
CMD9	ac	[31:6] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:6] RCA [15:0] stuff bits	R2	SEND_CID	Addressed card sends its card-identification (CID) on the CMD line.
CMD11	adtc	[31:0] data address	R1	READ_DAT_UNTIL_STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD12	ac	[31:0] stuff bits	R1b	STOP_TRANSMISSION	Forces the card to stop transmission.
CMD13	ac	[31:6] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	Reserved				
CMD15	ac	[31:6] RCA [15:0] stuff bits	-	GO_INACTIVE_STATE	Sets the card to inactive state in order to protect the card stack against communication breakdowns.
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	Sets the block length (in bytes) for all following block commands (read and write). Default block length is specified in the CSD.
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a stop command.
CMD19	Reserved				
CMD20	adtc	[31:0] data address	R1	WRITE_DAT_UNTIL_STOP	Writes data stream from the host, starting at the given address, until a STOP_TRANSMISION follows.
CMD21-23	Reserved				
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows.

Table continues on the next page...

Table 30-2. Commands for MMC/SD/SDIO Cards (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD26	adtc	[31:0] stuff bits	R1	PROGRAM_CID	Programming of the card identification register. This command shall be issued only once per card. The card contains hardware to prevent this operation after the first programming. Normally this command is reserved for the manufacturer.
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.
CMD28	ac	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29	ac	[31:0] data address	R1b	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group.
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card provides write protection features, this command asks the card to send the status of the write protection bits.
CMD31	Reserved				
CMD32	ac	[31:0] data address	R1	TAG_SECTOR_START	Sets the address of the first sector of the erase group.
CMD33	ac	[31:0] data address	R1	TAG_SECTOR_END	Sets the address of the last sector in a continuous range within the selection of a single sector to be selected for erase.
CMD34	ac	[31:0] data address	R1	UNTAG_SECTOR	Removes one previously selected sector from the erase selection.
CMD35	ac	[31:0] data address	R1	TAG_ERASE_GROUP_START	Sets the address of the first erase group within a range to be selected for erase.
CMD36	ac	[31:0] data address	R1	TAG_ERASE_GROUP_END	Sets the address of the last erase group within a continuous range to be selected for erase.
CMD37	ac	[31:0] data address	R1	UNTAG_ERASE_GROUP	Removes one previously selected erase group from the erase selection.
CMD38	ac	[31:0] stuff bits	R1b	ERASE	Erase all previously selected sectors.

Table continues on the next page...

Table 30-2. Commands for MMC/SD/SDIO Cards (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD39	ac	[31:0] RCA [15] register write flag [14:8] register address [7:0] register data	R4	FAST_IO	Used to write and read 8-bit (register) data fields. The command addresses a card, and a register, and provides the data for writing if the write flag is set. The R4 response contains data read from the address register. This command accesses application dependent registers which are not defined in the MMC standard.
CMD40	bcr	[31:0] stuff bits	R5	GO_IRQ_STATE	Sets the system into interrupt mode.
CMD41	Reserved				
CDM42	adtc	[31:0] stuff bits	R1b	LOCK_UNLOCK	Used to set/reset the password or lock/unlock the card. The size of the data block is set by the SET_BLOCK_LEN command.
CMD43~51	Reserved				
CMD52	ac	[31:0] stuff bits	R5	IO_RW_DIRECT	Access a single register within the total 128k of register space in any I/O function.
CMD53	ac	[31:0] stuff bits	R5	IO_RW_EXTENDED	Accesses a multiple I/O register with a single command. Allows the reading or writing of a large number of I/O registers.
CMD54	Reserved				
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather that a standard command.
CMD56	adtc	[31:1] stuff bits [0]: RD/WR	R1b	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose / application specific commands. The size of the data block is set by the SET_BLOCK_LEN command.
CMD57~59	Reserved				
CMD60	adtc	[31] WR [30:24] stuff bits [23:16] address [15:8] stuff bits [7:0] byte count	R1b	RW_MULTIPLE_REGISTER	These registers are used to control the behavior of the device and to retrieve status information regarding the operation of the device. All Status and Control registers are WORD (32-bit) in size and are WORD aligned. CMD60 shall be used to read and write these registers.

Table continues on the next page...

Table 30-2. Commands for MMC/SD/SDIO Cards (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD61	adtc	[31] WR [30:16] stuff bits [15:0] data unit count	R1b	RW_MULTIPLE_BLOCK	The host issues a RW_MULTIPLE_BLOCK (CMD61) to begin the data transfer.
CMD62~63	Reserved				
ACMD6[4]	ac	[31:2] stuff bits [1:0] bus width	R1	SET_BUS_WIDTH	Defines the data bus width ('00'=1bit or '10'=4bit bus) to be used for data transfer. The allowed data bus widths are given in SCR register.
ACMD13[4]	adtc	[31:0] stuff bits	R1	SD_STATUS	Send the SD Memory Card status.
ACMD22[4]	adtc	[31:0] stuff bits	R1	SEND_NUM_WR_SECTORS	Send the number of the written sectors (without errors). Responds with 32-bit plus the CRC data block.
ACMD23[4]	ac		R1	SET_WR_BLK_ERASE_COUNT	-
ACMD41[4]	bcr	[31:0] OCR	R3	SD_APP_OP_COND	Asks the accessed card to send its operating condition register (OCR) contents in the response on the CMD line.
ACMD42[4]	ac		R1	SET_CLR_CARD_DETECT	-
ACMD51[4]	adtc	[31:0] stuff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

1. CMD3 differs for MMC and SD cards. For MMC cards, it is referred to as SET_RELATIVE_ADDR, with a response type of R1. For SD cards, it is referred to as SEND_RELATIVE_ADDR, with a response type of R6 (with RCA inside).
2. CMD6 differs completely between high speed MMC cards and high speed SD cards. Command SWITCH_FUNC is for high speed SD cards.
3. Command SWITCH is for high speed MMC cards . The Index field can contain any value from 0-255, but only values 0-191 are valid. If the Index value is in the 192-255 range the card does not perform any modification and the SWITCH_ERROR status bit in the EXT_CSD register is set. The Access Bits are shown in Table 29-3.
4. ACMDs shall be preceded with the APP_CMD command. (Commands listed are used for SD only, other SD commands not listed are not supported on this block).

Table 30-3. EXT_CSD Access Modes

Bits	Access Name	Operation
00	Command Set	The command set is changed according to the Cmd Set field of the argument
01	Set Bits	The bits in the pointed byte are set, according to the 1 bits in the Value field.
10	Clear Bits	The bits in the pointed byte are cleared, according to the 1 bits in the Value field.
11	Write Byte	The Value field is written into the pointed byte.

30.6 Software Restrictions

30.6.1 Initialization Active

The driver cannot set INITA bit in System Control register when any of the command line or data lines is active, so the driver must ensure both CDIHB and CIHB bits are cleared. In order to auto clear the INITA bit, the SDCLKEN bit must be '1', otherwise no clocks can go out to the card and INITA will never clear.

30.6.2 Softeware Polling Procedure

For polling read or write, once the software begins a buffer read or write, it must access exactly the number of times as the values set in the Watermark Level Register; moreover, if the block size is not the times of the value in Watermark Level Register (read and write respectively), the software must access exactly the remaining number of words at the end of each block. For example, for read operation, if the RD_WML is 4, indicating the watermark level is 16 bytes, block size is 40 bytes, and the block number is 2, then the access times for the burst sequence in the whole transfer process must be 4, 4, 2, 4, 4, 2.

30.6.3 Suspend Operation

In order to suspend the data transfer, the software must inform ESDHC that the suspend command is successfully accepted. To achieve this, after the Suspend command is accepted by the SDIO card, software must send another normal command marked as suspend command (CMDTYP bits set as '01') to inform ESDHC that the transfer is suspended.

If software needs to resume the suspended trasnfer, it should read the value in BLKCNT register to save the remaining number of blocks before sending the normal command marked as suspend, otherwise on sending a 'suspend' command, ESDHC will regard the current transfer as aborted and change BLKCNT register to its original value, rather than keeping the remaining number of blocks.

30.6.4 Data Length Setting

For either ADMA (ADMA1 or ADMA2) transfer, the data in the data buffer must be word aligned, so the data length set in the descriptor must be a multiple of 4.

30.6.5 (A)DMA Address Setting

To configure ADMA1/ADMA2/DMA address register, when TC bit is set, the register will always update itself with the internal address value to support dynamic address synchronization, so software must make sure TC bit is cleared prior to configuring ADMA1/ADMA2/DMA address register.

30.6.6 Data Port Access

Data Port does not support parallel access. For example, during an external DMA access, it is not allowed to write any data to the Data Port by ARM platform; or during a ARM read operation, it is also prohibited to write any data to the Data Port, by either ARM or external DMA. Otherwise the data would be corrupted inside the ESDHC buffer.

30.6.7 Change Clock Frequency

ESDHC does not automatically gates off the card clock when the Host Driver changes the clock frequency. To remove possible glitch on the card clock, clear SDCLKEN bit when changing clock divisor value and set SDCLKEN bit to '1' after SDSTB bit is '1' again.

30.6.8 Multi-block Read

For pre-defined multi-block read operation, that is, the number of blocks to read has been defined by previous CMD23 for MMC, or pre-defined number of blocks in CMD53 for SDIO/SDCombo, or whatever multi-block read without abort command at card side, an abort command, either automatic or manual CMD12/CMD52, is still required by ESDHC after the pre-defined number of blocks are done, to drive the internal state machine to idle mode. In this case, the card may not respond to this extra abort command and ESDHC will get Response Timeout. It is recommended to manually send an abort command with RSPTYP[1:0] both bits cleared.

30.7 Programmable Registers

This section includes the block memory map and detailed descriptions of all registers. Each of these registers support only 32-bit accesses.

NOTE

Addresses greater than 0x44, except 0x50, 0x54, 0x58, 0x60, 0x64, 0xC0, 0xC4 and 0xFC, are reserved and read as all 0s. Write to these

ESDHCV3 memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5002_0000	DMA System Address (ESDHCV3-3_DSADDR)	32	R/W	0000_0000h	30.7.1/ 1488
5002_0004	Block Attributes (ESDHCV3-3_BLKATTR)	32	R/W	0000_0000h	30.7.2/ 1489
5002_0008	Command Argument (ESDHCV3-3_CMDARG)	32	R/W	0000_0000h	30.7.3/ 1490
5002_000C	Command Transfer Type (ESDHCV3-3_XFERTYP)	32	R/W	0000_0000h	30.7.4/ 1491
5002_0010	Command Response n (ESDHCV3-3_CMDRSP0)	32	R	0000_0000h	30.7.5/ 1495
5002_0014	Command Response n (ESDHCV3-3_CMDRSP1)	32	R	0000_0000h	30.7.5/ 1495
5002_0018	Command Response n (ESDHCV3-3_CMDRSP2)	32	R	0000_0000h	30.7.5/ 1495
5002_001C	Command Response n (ESDHCV3-3_CMDRSP3)	32	R	0000_0000h	30.7.5/ 1495
5002_0020	Data Buffer Access Port (ESDHCV3-3_DATPORT)	32	R/W	0000_0000h	30.7.6/ 1497
5002_0024	Present State (ESDHCV3-3_PRSSTAT)	32	R	0000_0000h	30.7.7/ 1497
5002_0028	Protocol Control (ESDHCV3-3_PROCTL)	32	R/W	0000_0000h	30.7.8/ 1502
5002_002C	System Control (ESDHCV3-3_SYSCTL)	32	R/W	0080_8008h	30.7.9/ 1506
5002_0030	Interrupt Status (ESDHCV3-3_IRQSTAT)	32	w1c	0000_0000h	30.7.10/ 1510

Table continues on the next page...

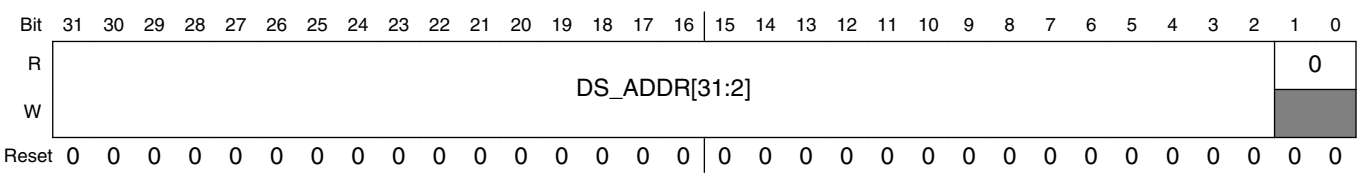
ESDHCV3 memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5002_0034	Interrupt Status Enable (ESDHCV3-3_IRQSTATEN)	32	R/W	117F_013Fh	30.7.11/ 1515
5002_0038	Interrupt Signal Enable (ESDHCV3-3_IRQSIGEN)	32	R/W	0000_0000h	30.7.12/ 1518
5002_003C	Auto CMD12 Status (ESDHCV3-3_AUTOC12ERR)	32	R	0000_0000h	30.7.13/ 1520
5002_0040	Host Controller Capabilities (ESDHCV3-3_HOSTCAPBLT)	32	R	07F3_0000h	30.7.14/ 1523
5002_0044	Watermark Level (ESDHCV3-3_WML)	32	R/W	0810_0810h	30.7.15/ 1525
5002_0050	Force Event (ESDHCV3-3_FEVT)	32	W (always reads zero)	0000_0000h	30.7.16/ 1526
5002_0054	ADMA Error Status Register (ESDHCV3-3_ADMAES)	32	R	0000_0000h	30.7.17/ 1528
5002_0058	ADMA System Address (ESDHCV3-3_ADSADDR)	32	R/W	0000_0000h	30.7.18/ 1530
5002_0060	DLL (Delay Line) Control (ESDHCV3-3_DLLCTRL)	32	R/W	0000_0000h	30.7.19/ 1531
5002_0064	DLL Status (ESDHCV3-3_DLLSTS)	32	R	0000_0000h	30.7.20/ 1532
5002_00C0	Vendor Specific Register (ESDHCV3-3_VENDOR)	32	R/W	0000_0001h	30.7.21/ 1533
5002_00C4	MMC Boot Register (ESDHCV3-3_MMCBOOT)	32	R/W	0000_0000h	30.7.22/ 1534
5002_00FC	Host Controller Version (ESDHCV3-3_HOSTVER)	32	R	0000_1201h	30.7.23/ 1535

30.7.1 DMA System Address (ESDHCV3x_DSADDR)

This register contains the physical system memory address used for DMA transfers.

Addresses: ESDHCV3-3_DSADDR is 5002_0000h base + 0h offset = 5002_0000h



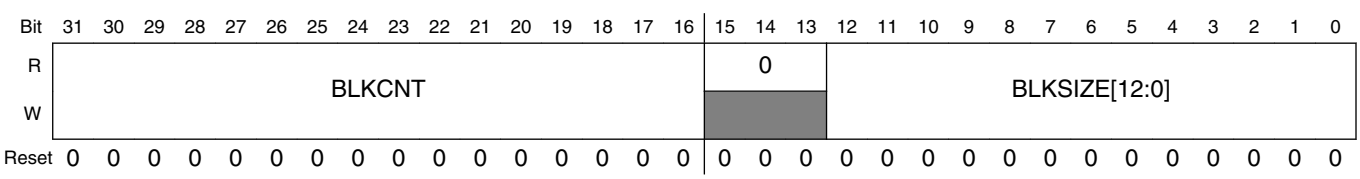
ESDHCV3x_DSADDR field descriptions

Field	Description
31–2 DS_ADDR[31:2]	DMA System Address: This register contains the 32-bit system memory address for a DMA transfer. As the address must be word (4 bytes) align, the least 2 bits are reserved, always 0. When the ESDHC stops a DMA transfer, this register points to the system address of the next contiguous data position. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). Read operation during transfers may return an invalid value. The Host Driver shall initialize this register before starting a DMA transaction. After DMA has stopped, the system address of the next contiguous data position can be read from this register. This register is protected during a data transfer. When data lines are active, write to this register is ignored. The Host driver shall wait, until ESDHCV3_PRSSTAT[DLA] bit is cleared, before writing to this register. The ESDHC internal DMA does not support a virtual memory system. It only supports continuous physical memory access. And due to AHB burst limitations, if the burst must cross the 1 KB boundary, ESDHC will automatically change SEQ burst type to NSEQ. As this register supports dynamic address reflecting, when TC bit is set, it automatically alters the value of internal address counter, so SW cannot change this register when TC bit is set. Such restriction is also listed in Software Restrictions.
1–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

30.7.2 Block Attributes (ESDHCV3x_BLKATTR)

This register is used to configure the number of data blocks and the number of bytes in each block.

Addresses: ESDHCV3-3_BLKATTR is 5002_0000h base + 4h offset = 5002_0004h



ESDHCV3x_BLKATTR field descriptions

Field	Description
31–16 BLKCNT	Blocks Count For Current Transfer: This register is enabled when ESDHCV3_XFERTYP[BCEN] bit is set to 1 and is valid only for multiple block transfers. For single block transfer, this register will always read as 1. The Host Driver shall set this register to a value between 1 and the maximum block count. The ESDHC decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to 0 results in no data blocks being transferred. This register should be accessed only when no transaction is executing (that is, after transactions have stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored. When saving transfer content as a result of a Suspend command, the number of blocks yet to be transferred can be determined by reading this register. The reading of this register should be applied after transfer is paused by stop at block gap operation and before sending the command marked as suspend.

Table continues on the next page...

ESDHCV3x_BLKATTR field descriptions (continued)

Field	Description
	This is because when Suspend command is sent out, ESDHC will regard the current transfer is aborted and change BLKCNT register back to its original value instead of keeping the dynamical indicator of remained block count. When restoring transfer content prior to issuing a Resume command, the Host Driver shall restore the previously saved block count. 0xFFFF 65535 blocks 0x0002 2 blocks 0x0001 1 blocks 0x0000 Stop Count
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12–0 BLKSIZE[12:0]	Transfer Block Size: This register specifies the block size for block data transfers. Values ranging from 1 byte up to the maximum buffer size can be set. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations will be ignored. 0x1000 4096 Bytes 0x800 2048 Bytes 0x200 512 Bytes 0x1FF 511 Bytes 0x004 4 Bytes 0x003 3 Bytes 0x002 2 Bytes 0x001 1 Bytes 0x000 No data transfer

30.7.3 Command Argument (ESDHCV3x_CMDARG)

This register contains the SD/MMC Command Argument.

Addresses: ESDHCV3-3_CMDARG is 5002_0000h base + 8h offset = 5002_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMDARG[31:0]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDHCV3x_CMDARG field descriptions

Field	Description
31–0 CMDARG[31:0]	Command Argument: The SD/MMC Command Argument is specified as bits 39-8 of the Command Format in the SD or MMC Specification.This register is write protected when the Command Inhibit (CMD) bit in the Present State register is set.

30.7.4 Command Transfer Type (ESDHCV3x_XFERTYP)

This register is used to control the operation of data transfers. The Host Driver sets this register before issuing a command followed by a data transfer, or before issuing a Resume command. To prevent data loss, the ESDHC prevents writing to the bits, that are involved in the data transfer of this register, when data transfer is active. These bits are DPSEL, MBSEL, DTDSEL, AC12EN, BCEN and DMAEN.

The Host Driver shall check the Command Inhibit DAT bit (CDIHB) and the Command Inhibit CMD bit (CIHB) in the Present State register before writing to this register. When the CDIHB bit in the Present State register is set, any attempt to send a command with data by writing to this register is ignored; when the CIHB bit is set, any write to this register is ignored.

On sending commands with data transfer involved, it is mandatory that the block size is non-zero. Besides, block count must also be non-zero, or indicated as single block transfer (bit 5 of this register is '0' when written), or block count is disabled (bit 1 of this register is '0' when written), otherwise ESDHC will ignore the sending of this command and do nothing. For write command, with all above restrictions, it is also mandatory that the write protect switch is not active (WPSPL bit of Present State Register is '1), otherwise ESDHC will also ignore the command.

If the commands with data transfer does not receive the response in 64 clock cycles, that is, response time-out, ESDHC will regard the external device does not accept the command and abort the data transfer. In this scenario, the driver should issue the command again to re-try the transfer. It is also possible that for some reason the card responds the command but ESDHC does not receive the response, and if it is internal DMA (either simple DMA or ADMA) read operation, the external system memory is over-written by the internal DMA with data sent back from the card.

The CMDTYP field of this register is used to initiate three special commands; these are:

- Suspend: ESDHC does not monitor the content of the Suspend command response and therefore assumes that the commnand succeeds when issued. It then operates assuming the card bus has been released and that it is permissible to issue the next command using the DAT line. It is the responsibility of S/W to check the status of the Suspend command and send another command marked as Suspend to inform the ESDHC that a Suspend command was successfully issued. Refer to Suspend Resume for more details. After the end bit of command is sent, ESDHC de-asserts Read Wait for read transactions and stops checking busy for write transactions. In 4-bit mode, the interrupt cycle starts. If the Suspend command fails, the ESDHC will maintain its

current state and the Host Driver shall restart the transfer by setting the Continue Request bit in the Protocol Control register.

- Resume: S/W re-starts the data transfer by restoring the registers saved before sending the Suspend Command and then sends the Resume Command. The ESDHC will check for a pending busy state before starting write transfers.
- Abort: If this command is set when executing a read transfer, ESDHC will stop reads to the buffer. If this command is set when executing a write transfer, ESDHC will stop driving the DAT line. After issuing the Abort command, the Host Driver should issue a software reset (Abort Transaction).

The following table shows the summary of how register settings determine the type of data transfer.

Table 30-49. Transfer Type Register Setting for Various Transfer Types

Multi/Single Block Select	Block Count Enable	Block Count	Function
0	Don't Care	Don't Care	Single Transfer
1	0	Don't Care	Infinite Transfer
1	1	Positive Number	Multiple Transfer
1	1	Zero	No Data Transfer

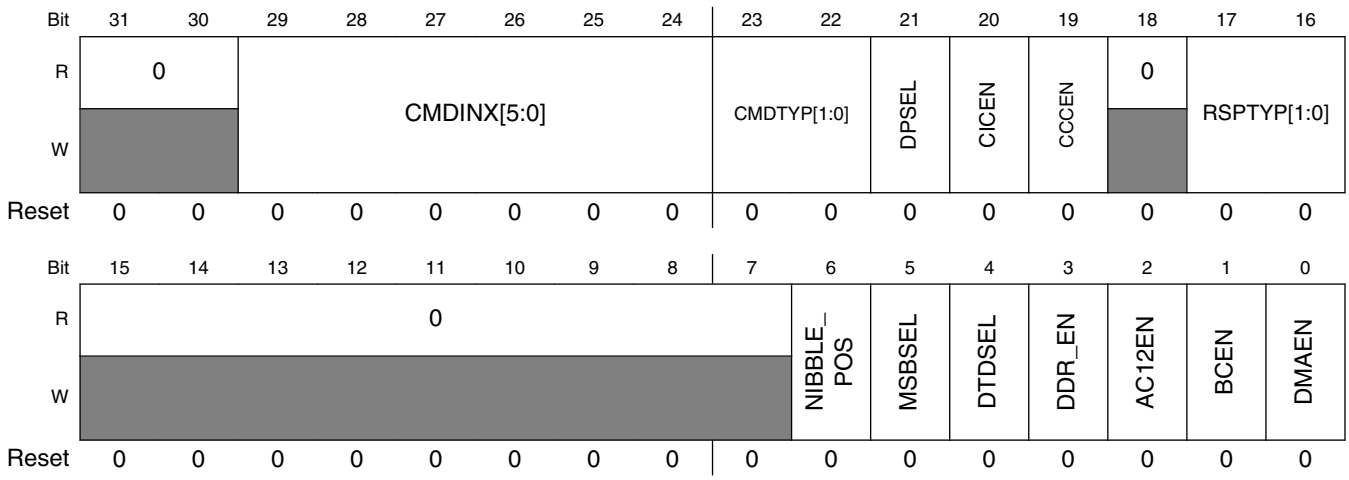
The following table shows the relationship between the Command Index Check Enable and the Command CRC Check Enable, in regards to the Response Type bits and the name of the response type.

Table 30-50. Relationship Between Parameters and the Name of the Response Type

Response Type	Index Check Enable	CRC Check Enable	Name of Response Type
00	0	0	No Response
01	0	1	R2
10	0	0	R3,R4
10	1	1	R1,R5,R6
11	1	1	R1b,R5b

- In the SDIO specification, response type notation for R5b is not defined. R5 includes R5b in the SDIO specification. But R5b is defined in this specification to indicate that ESDHC will check the busy status after receiving a response. For example, usually CMD52 is used with R5, but the I/O abort command will be used with R5b.
- The CRC field for R3 and R4 is expected to be all 1 bits. The CRC check shall be disabled for these response types.

Addresses: ESDHCV3-3_XFERTYP is 5002_0000h base + Ch offset = 5002_000Ch



ESDHCV3x_XFERTYP field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved
29–24 CMDINX[5:0]	Command Index: These bits shall be set to the command number that is specified in bits 45-40 of the Command-Format in the SD Memory Card Physical Layer Specification and SDIO Card Specification.
23–22 CMDTYP[1:0]	Command Type: 11 Abort CMD12, CMD52 for writing I/O Abort in CCCR 10 Resume CMD52 for writing Function Select in CCCR 01 Suspend CMD52 for writing Bus Suspend in CCCR 00 Normal mode. Used for all other commands
21 DPSEL	Data Present Select: This bit is set to 1 to indicate that data is present and shall be transferred using the DAT line. It is set to 0 for the following: • Commands using only the CMD line (for example, CMD52). • Commands with no data transfer, but using the busy signal on DAT[0] line (R1b or R5b such as, CMD38) **NOTE:** In resume command, this bit shall be set, and other bits in this register shall be set the same as when the transfer was initially launched. When the Write Protect switch is on, (that is, the WPSPL bit is active as '0'), any command with a write operation will be ignored. That is to say, when this bit is set, while the DTDSEL bit is 0, writes to the register Transfer Type are ignored. 1 Data Present 0 No Data Present
20 CICEN	Command Index Check Enable: If this bit is set to 1, the ESDHC will check the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this bit is set to 0, the Index field is not checked.

Table continues on the next page...

ESDHCV3x_XFERTYP field descriptions (continued)

Field	Description
	1 Enable 0 Disable
19 CCCEN	Command CRC Check Enable: If this bit is set to 1, the ESDHC shall check the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this bit is set to 0, the CRC field is not checked. The number of bits checked by the CRC field value changes according to the length of the response. (Refer to RSPTYP[1:0] and table "Relationship Between Parameters and the Name of the Response Type ".) 1 Enable 0 Disable
18 Reserved	This read-only field is reserved and always has the value zero. Reserved
17–16 RSPTYP[1:0]	Response Type Select: 00 No Response 01 Response Length 136 10 Response Length 48 11 Response Length 48, check Busy after response
15–7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6 NIBBLE_POS	In DDR 4 bit mode nibble position indication. 0- the sequence is 'odd high nibble -> even high nibble -> odd low nibble -> even low nibble'; 1- the sequence is 'odd high nibble -> odd low nibble -> even high nibble -> even low nibble'.
5 MSBSEL	Multi / Single Block Select: This bit enables multiple block DAT line data transfers. For any other commands, this bit shall be set to 0. If this bit is 0, it is not necessary to set the Block Count register. (See table above.) 1 Multiple Blocks 0 Single Block
4 DTDSEL	Data Transfer Direction Select: This bit defines the direction of DAT line data transfers. The bit is set to 1 by the Host Driver to transfer data from the SD card to the ESDHC and is set to 0 for all other commands. 1 Read (Card to Host) 0 Write (Host to Card)
3 DDR_EN	Dual Data Rate mode selection
2 AC12EN	Auto CMD12 Enable: Multiple block transfers for memory require a CMD12 to stop the transaction. When this bit is set to 1, the ESDHC will issue a CMD12 automatically when the last block transfer has completed. The Host Driver shall not set this bit to issue commands that do not require CMD12 to stop a multiple block data transfer. In particular, secure commands defined in File Security Specification (see reference list) do not require CMD12. In single block transfer, the ESDHC will ignore this bit no matter if it is set or not. 1 Enable 0 Disable

Table continues on the next page...

ESDHCV3x_XFERTYP field descriptions (continued)

Field	Description
1 BCEN	Block Count Enable: This bit is used to enable the Block Count register, which is only relevant for multiple block transfers. When this bit is 0, the internal counter for block is disabled, which is useful in executing an infinite transfer. 1 Enable 0 Disable
0 DMAEN	DMA Enable: This bit enables DMA functionality. If this bit is set to 1, a DMA operation shall begin when the Host Driver sets the DPSEL bit of this register. Whether the Simple DMA, or the Advanced DMA, is active depends on the DMA Select field of the Protocol Control register. 1 Enable 0 Disable

30.7.5 Command Response n (ESDHCV3x_CMDRSP*n*)

This register is used to store part n of the response bits from the card.

The table below describes the mapping of command responses from the SD Bus to Command Response registers for each response type. In the table, R[] refers to a bit range within the response data as transmitted on the SD Bus.

Table 30-52. Response Bit Definition for Each Response Type

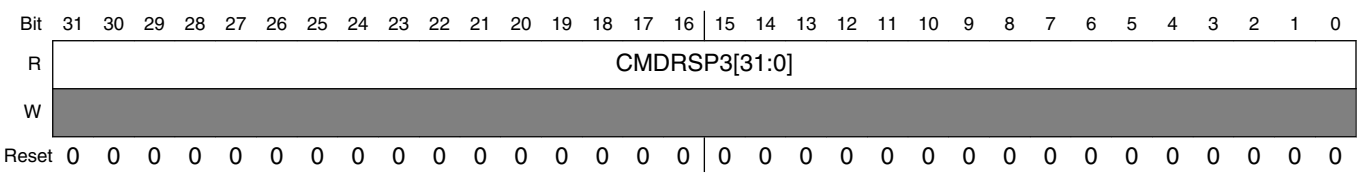
Response Type	Meaning of Response	Response Field	Response Register
R1,R1b (normal response)	Card Status	R[39:8]	ESDHCV3_CMDRSP0
R1b (Auto CMD12 response)	Card Status for Auto CMD12	R[39:8]	ESDHCV3_CMDRSP3
R2 (CID, CSD register)	CID/CSD register [127:8]	R[127:8]	{ESDHCV3_CMDRSP3[23:0] , ESDHCV3_CMDRSP2, ESDHCV3_CMDRSP1, ESDHCV3_CMDRSP0}
R3 (OCR register)	OCR register for memory	R[39:8]	ESDHCV3_CMDRSP0
R4 (OCR register)	OCR register for I/O etc.	R[39:8]	ESDHCV3_CMDRSP0
R5, R5b	SDIO response	R[39:8]	ESDHCV3_CMDRSP0
R6 (Publish RCA)	New Published RCA[31:16] and card status[15:0]	R[39:9]	ESDHCV3_CMDRSP0

This table shows that most responses with a length of 48 (R[47:0]) have 32-bits of the response data (R[39:8]) stored in the ESDHCV3_CMDRSP0 register. Responses of type R1b (Auto CMD12 responses) have response data bits (R[39:8]) stored in the ESDHCV3_CMDRSP3 register. Responses with length 136 (R[135:0]) have 120-bits of the response data (R[127:8]) stored in the ESDHCV3_CMDRSP0, 1, 2, and 3 registers.

To be able to read the response status efficiently, the ESDHC only stores part of the response data in the Command Response registers. This enables the Host Driver to efficiently read 32-bits of response data in one read cycle on a 32-bit bus system. Parts of the response, the Index field and the CRC, are checked by the ESDHC (as specified by the Command Index Check Enable and the Command CRC Check Enable bits in the Transfer Type register) and generate an error interrupt if any error is detected. The bit range for the CRC check depends on the response length. If the response length is 48, the ESDHC will check R[47:1], and if the response length is 136 the ESDHC will check R[119:1].

As ESDHC may have a multiple block data transfer executing concurrently with a CMD_wo_DAT command, the ESDHC stores the Auto CMD12 response in the ESDHCV3_CMDRSP3 register. The CMD_wo_DAT response is stored in ESDHCV3_CMDRSP0. This allows the ESDHC to avoid overwriting the Auto CMD12 response with the CMD_wo_DAT and vice versa. When the ESDHC modifies part of the Command Response registers, as shown in the table above, it preserves the unmodified bits.

Addresses: ESDHCV3-3_CMDRSP0 is 5002_0000h base + 10h offset = 5002_0010h
ESDHCV3-3_CMDRSP1 is 5002_0000h base + 14h offset = 5002_0014h
ESDHCV3-3_CMDRSP2 is 5002_0000h base + 18h offset = 5002_0018h
ESDHCV3-3_CMDRSP3 is 5002_0000h base + 1Ch offset = 5002_001Ch



ESDHCV3x_CMDRSP*n* field descriptions

Field	Description
31–0 CMDRSP3[31:0]	Command Response n: Refer to the table "Response Bit Definition for Each Response Type" above for the mapping of command responses from the SD Bus to this register for each response type.

30.7.6 Data Buffer Access Port (ESDHCV3x_DATPORT)

This is a 32-bit data port register used to access the internal buffer.

Addresses: ESDHCV3-3_DATPORT is 5002_0000h base + 20h offset = 5002_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATCONT[31:0]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDHCV3x_DATPORT field descriptions

Field	Description
31–0 DATCONT[31:0]	Data Content: The Buffer Data Port register is for 32-bit data access by the ARM platform or the external DMA. When the internal DMA is enabled, any write to this register is ignored, and any read from this register will always yield 0s.

30.7.7 Present State (ESDHCV3x_PRSTAT)

The Host Driver can get status of the ESDHC from this 32-bit read only register.

NOTE

The reset value of Present State Register depend on testbench connectivity.

- The Host Driver issues CMD0, CMD12, CMD13 (for memory) and CMD52 (for SDIO) when the DAT lines are busy during a data transfer. These commands can be issued when Command Inhibit (CMD) is set to zero. Other commands will be issued when Command Inhibit (DAT) is set to zero. Possible changes to the SD Physical Specification may add other commands to this list in the future.

Addresses: ESDHCV3-3_PRSSTAT is 5002_0000h base + 24h offset = 5002_0024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DLSL[7:0]								CLSL	0			WPSPL	CDPL	0	CINS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				BREN	BWEN	RTA	WTA	SDOFF	PEROFF	HCKOFF	IPGOFF	SDSTB	DLA	CDIHB	CIHB
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDHCV3x_PRSSTAT field descriptions

Field	Description
31–24 DLSL[7:0]	DAT[7:0] Line Signal Level: This status is used to check the DAT line level to recover from errors, and for debugging.This is especially useful in detecting the busy signal level from DAT[0]. The reset value is effected by the external pull-up/pull-down resistors. By default, the read value of this bit field after reset is 8'b11110111, when DAT[3] is pulled down and the other lines are pulled up. DAT[7]: Data 7 line signal level DAT[6]: Data 6 line signal level DAT[5]: Data 5 line signal level DAT[4]: Data 4 line signal level DAT[3]: Data 3 line signal level DAT[2]: Data 2 line signal level DAT[1]: Data 1 line signal level DAT[0]: Data 0 line signal level
23 CLSL	CMD Line Signal Level: This status is used to check the CMD line level to recover from errors, and for debugging. The reset value is effected by the external pull-up/pull-down resistor, by default, the read value of this bit after reset is 1'b1, when the command line is pulled up.
22–20 Reserved	This read-only field is reserved and always has the value zero. Reserved
19 WPSPL	Write Protect Switch Pin Level:

Table continues on the next page...

ESDHCV3x_PRSSTAT field descriptions (continued)

Field	Description
	The Write Protect Switch is supported for memory and combo cards.This bit reflects the inverted value of the SD_WP pin of the card socket. A software reset does not affect this bit. The reset value is effected by the external write protect switch. If the SD_WP pin is not used, it should be tied low, so that the reset value of this bit is high and write is enabled. 1 Write enabled (SD_WP=0) 0 Write protected (SD_WP=1)
18 CDPL	Card Detect Pin Level: This bit reflects the inverse value of the SD_CD# pin for the card socket. Debouncing is not performed on this bit. This bit may be valid, but is not guaranteed, because of propagation delay. Use of this bit is limited to testing As it must be debounced by software. A software reset does not effect this bit. A write to the Force Event Register does not effect this bit. The reset value is effected by the external card detection pin. This bit shows the value on the SD_CD# pin (that is, when a card is inserted in the socket, it is 0 on the SD_CD# input, and consequently the CDPL reads 1.) 1 Card present (SD_CD#=0) 0 No card present (SD_CD#=1)
17 Reserved	This read-only field is reserved and always has the value zero. Reserved
16 CINS	Card Inserted: This bit indicates whether a card has been inserted. The ESDHC debounces this signal so that the Host Driver will not need to wait for it to stabilize. Changing from a 0 to 1 generates a Card Insertion interrupt in the Interrupt Status register. Changing from a 1 to 0 generates a Card Removal interrupt in the Interrupt Status register. A write to the Force Event Register does not effect this bit. The Software Reset For All in the System Control register does not effect this bit.A software reset does not effect this bit. 1 Card Inserted 0 Power on Reset or No Card
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11 BREN	Buffer Read Enable: This status bit is used for non-DMA read transfers. The ESDHC may implement multiple buffers to transfer data efficiently. This read only flag indicates that valid data exists in the host side buffer. If this bit is high, valid data greater than the watermark level exist in the buffer. A change of this bit from 1 to 0 occurs when any read from the buffer is made. A change of this bit from 0 to1 occurs when there is enough valid data ready in the buffer and the Buffer Read Ready interrupt has been generated and enabled. 1 Read enable 0 Read disable
10 BWEN	Buffer Write Enable: This status bit is used for non-DMA write transfers. The ESDHC can implement multiple buffers to transfer data efficiently. This read only flag indicates if space is available for write data. If this bit is 1, valid data greater than the watermark level can be written to the buffer. A change of this bit from 1 to 0 occurs when any write to the buffer is made. A change of this bit from 0 to 1 occurs when the buffer can hold valid data greater than the write watermark level and the Buffer Write Ready interrupt is generated and enabled.

Table continues on the next page...

ESDHCV3x_PRSSTAT field descriptions (continued)

Field	Description
	1 Write enable 0 Write disable
9 RTA	Read Transfer Active: This status bit is used for detecting completion of a read transfer. This bit is set for either of the following conditions: • After the end bit of the read command. • When writing a 1 to the Continue Request bit in the Protocol Control register to restart a read transfer. A Transfer Complete interrupt is generated when this bit changes to 0. This bit is cleared for either of the following conditions: • When the last data block as specified by block length is transferred to the System, that is, all data are read away from ESDHC internal buffer. • When all valid data blocks have been transferred from ESDHC internal buffer to the System and no current block transfers are being sent as a result of the Stop At Block Gap Request being set to 1. 1 Transferring data 0 No valid data
8 WTA	Write Transfer Active: This status bit indicates a write transfer is active. If this bit is 0, it means no valid write data exists in the ESDHC. This bit is set in either of the following cases: • After the end bit of the write command. • When writing 1 to the Continue Request bit in the Protocol Control register to restart a write transfer. This bit is cleared in either of the following cases: • After getting the CRC status of the last data block as specified by the transfer count (Single and Multiple). • After getting the CRC status of any block where data transmission is about to be stopped by a Stop At Block Gap Request. During a write transaction, a Block Gap Event interrupt is generated when this bit is changed to 0, as result of the Stop At Block Gap Request being set. This status is useful for the Host Driver in determining when to issue commands during Write Busy state. 1 Transferring data 0 No valid data
7 SDOFF	SD Clock Gated Off Internally: This status bit indicates that the SD Clock is internally gated off, because of buffer over/under-run or read pause without read wait assertion, or the driver has cleared SDCLKEN bit to stop the SD clock. This bit is for the Host Driver to debug data transaction on the SD bus. 1 SD Clock is gated off 0 SD Clock is active
6 PEROFF	ipg_perclk Gated Off Internally:

Table continues on the next page...

ESDHCV3x_PRSSTAT field descriptions (continued)

Field	Description
	This status bit indicates that the ipg_perclk is internally gated off. This bit is for the Host Driver to debug transaction on the SD bus. When INITA bit is set, ESDHC sending 80 clock cycles to the card, the SDCLKEN bit must be '1' to enable the output card clock, otherwise the ipg_perclk will never be gate off, so ipg_perclk and ipg_clk will be always active. 1 ipg_perclk is gated off 0 ipg_perclk is active
5 HCKOFF	hclk Gated Off Internally: This status bit indicates that the hclk is internally gated off. This bit is for the Host Driver to debug during a data transfer. 1 hclk is gated off 0 hclk is active
4 IPGOFF	ipg_clk Gated Off Internally: This status bit indicates that the ipg_clk is internally gated off. This bit is for the Host Driver to debug. 1 ipg_clk is gated off 0 ipg_clk is active
3 SDSTB	SD Clock Stable This status bit indicates that the internal card clock is stable. This bit is for the Host Driver to poll clock status when changing the clock frequency. It is recommended to clear SDCLKEN bit in System Control register to remove glitch on the card clock when the frequency is changing. 1 clock is stable 0 clock is changing frequency and not stable
2 DLA	Data Line Active This status bit indicates whether one of the DAT lines on the SD Bus is in use. In the case of read transactions: This status indicates if a read transfer is executing on the SD Bus. Changes in this value from 1 to 0, between data blocks, generates a Block Gap Event interrupt in the Interrupt Status register. This bit will be set in either of the following cases: • After the end bit of the read command. • When writing a 1 to the Continue Request bit in the Protocol Control register to restart a read transfer. This bit will be cleared in either of the following cases: (1) When the end bit of the last data block is sent from the SD Bus to the ESDHC. (2) When the Read Wait state is stopped by a Suspend command and the DAT2 line is released. The ESDHC will wait at the next block gap by driving Read Wait at the start of the interrupt cycle. If the Read Wait signal is already driven (data buffer cannot receive data), the ESDHC can wait for a current block gap by continuing to drive the Read Wait signal. It is necessary to support Read Wait in order to use the suspend / resume function. This bit will remain 1 during Read Wait. In the case of write transactions: This status indicates that a write transfer is executing on the SD Bus. Changes in this value from 1 to 0 generate a Transfer Complete interrupt in the Interrupt Status register. This bit will be set in either of the following cases:

Table continues on the next page...

ESDHCV3x_PRSSTAT field descriptions (continued)

Field	Description
	• After the end bit of the write command. • When writing to 1 to the Continue Request bit in the Protocol Control register to continue a write transfer. This bit will be cleared in either of the following cases: • When the SD card releases Write Busy of the last data block, the ESDHC will also detect if the output is not busy. If the SD card does not drive the busy signal after the CRC status is received, the ESDHC will assume the card drive "Not Busy". • When the SD card releases write busy, prior to waiting for write transfer, and as a result of a Stop At Block Gap Request. In the case of command with busy pending: This status indicates that a busy state follows the command and the data line is in use. This bit will be cleared when the DAT0 line is released. 1 DAT Line Active 0 DAT Line Inactive
1 CDIHB	Command Inhibit (DAT): This status bit is generated if either the DAT Line Active or the Read Transfer Active is set to 1. If this bit is 0, it indicates that the ESDHC can issue the next SD/MMC Command. Commands with a busy signal belong to Command Inhibit (DAT) (for example, R1b, R5b type). Except in the case when the command busy is finished, changing from 1 to 0 generates a Transfer Complete interrupt in the Interrupt Status register. **NOTE:** The SD Host Driver can save registers for a suspend transaction after this bit has changed from 1 to 0. 1 Cannot issue command which uses the DAT line 0 Can issue command which uses the DAT line
0 CIHB	Command Inhibit (CMD): If this status bit is 0, it indicates that the CMD line is not in use and the ESDHC can issue a SD/MMC Command using the CMD line. This bit is set also immediately after the Transfer Type register is written. This bit is cleared when the command response is received. Even if the Command Inhibit (DAT) is set to 1, Commands using only the CMD line can be issued if this bit is 0. Changing from 1 to 0 generates a Command Complete interrupt in the Interrupt Status register. If the ESDHC cannot issue the command because of a command conflict error (Refer to Command CRC Error) or because of a Command Not Issued By Auto CMD12 Error, this bit will remain 1 and the Command Complete is not set. The Status of issuing an Auto CMD12 does not show on this bit. 1 Cannot issue command 0 Can issue command using only CMD line

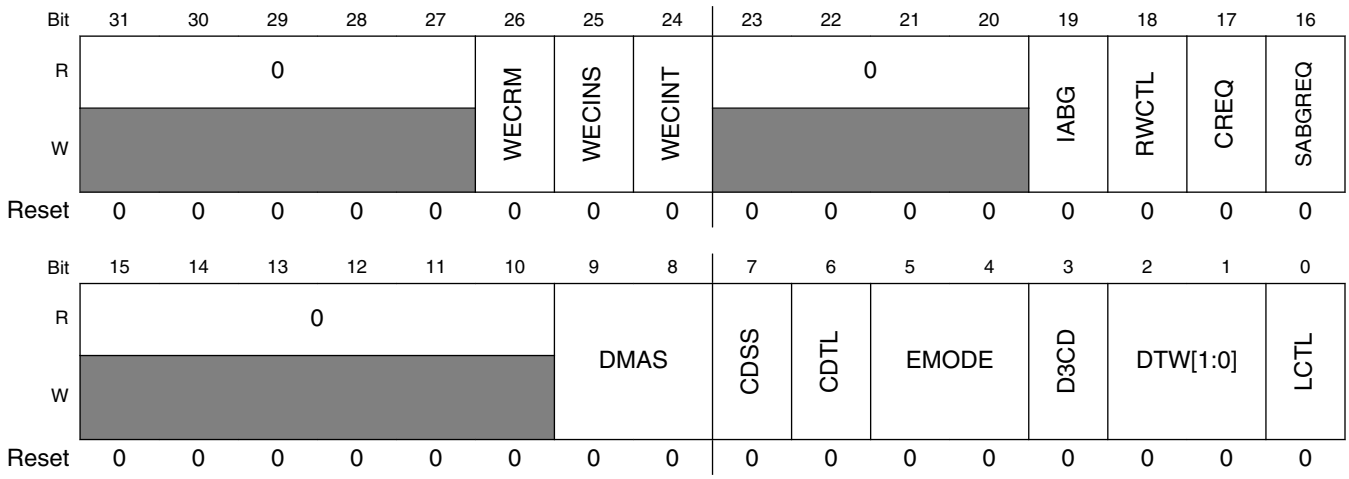
30.7.8 Protocol Control (ESDHCV3x_PROCTL)

There are three cases to restart the transfer after stop at the block gap. The appropriate case depends on whether ESDHC issues a Suspend command or the SD card accepts the Suspend command.

1. If the Host Driver does not issue a Suspend command, the Continue Request will be used to restart the transfer.
2. If the Host Driver issues a Suspend command and the SD card accepts it, a Resume command will be used to restart the transfer.
3. If the Host Driver issues a Suspend command and the SD card does not accept it, the Continue Request will be used to restart the transfer.

Any time Stop At Block Gap Request stops the data transfer, the Host Driver will wait for a Transfer Complete (in the Interrupt Status register), before attempting to restart the transfer. When restarting the data transfer by Continue Request, the Host Driver will clear the Stop At Block Gap Request before or simultaneously.

Addresses: ESDHCV3-3_PROCTL is 5002_0000h base + 28h offset = 5002_0028h



ESDHCV3x_PROCTL field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26 WECRM	Wakeup Event Enable On SD Card Removal: This bit enables a wakeup event, through a Card Removal, in the Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not effect this bit. When this bit is set, the Card Removal Status and the ESDHC interrupt can be asserted without SD_CLK toggling. When the wakeup feature is not enabled, the SD_CLK must be active in order to assert the Card Removal Status and the ESDHC interrupt. 1 Enable 0 Disable
25 WECINS	Wakeup Event Enable On SD Card Insertion: This bit enables a wakeup event, through a Card Insertion, in the Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not effect this bit. When this bit is set, the Card Insertion Status and the ESDHC interrupt can be asserted without SD_CLK toggling. When the wakeup feature is not enabled, the SD_CLK must be active in order to assert the Card Insertion Status and the ESDHC interrupt.

Table continues on the next page...

ESDHCV3x_PROCTL field descriptions (continued)

Field	Description
	1 Enable 0 Disable
24 WECINT	Wakeup Event Enable On Card Interrupt: This bit enables a wakeup event, through a Card Interrupt, in the Interrupt Status register. This bit can be set to 1 if FN_WUS (Wake Up Support) in CIS is set to 1. When this bit is set, the Card Interrupt Status and the ESDHC interrupt can be asserted without SD_CLK toggling. When the wakeup feature is not enabled, the SD_CLK must be active in order to assert the Card Interrupt Status and the ESDHC interrupt. 1 Enable 0 Disable
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved
19 IABG	Interrupt At Block Gap: This bit is valid only in 4-bit mode, of the SDIO card, and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If the SDIO card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0 to avoid an inadvertent interrupt. When the Host Driver detects an SDIO card insertion, it sets this bit according to the CCCR of the card. 1 Enabled 0 Disabled
18 RWCTL	Read Wait Control: The read wait function is optional for SDIO cards. If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using the DAT[2] line. Otherwise the ESDHC has to stop the SD Clock to hold read data, which restricts commands generation. When the Host Driver detects an SDIO card insertion, it sets this bit according to the CCCR of the card. If the card does not support read wait, this bit will never be set to 1, otherwise DAT line conflicts may occur. If this bit is set to 0, stop at block gap during read operation is also supported, but the ESDHC will stop the SD Clock to pause reading operation. 1 Enable Read Wait Control, and assert Read Wait without stopping SD Clock at block gap when SABGREQ bit is set 0 Disable Read Wait Control, and stop SD Clock at block gap when SABGREQ bit is set
17 CREQ	Continue Request: This bit is used to restart a transaction which was stopped using the Stop At Block Gap Request. When a Suspend operation is not accepted by the card, it is also by setting this bit to restart the paused transfer. To cancel stop at the block gap, set Stop At Block Gap Request to 0 and set this bit to 1 to restart the transfer. The ESDHC automatically clears this bit, therefore it is not necessary for the Host Driver to set this bit to 0. If both Stop At Block Gap Request and this bit are 1, the continue request is ignored. 1 Restart 0 No effect
16 SABGREQ	Stop At Block Gap Request: This bit is used to stop executing a transaction at the next block gap for both DMA and non-DMA transfers. Until the Transfer Complete is set to 1, indicating a transfer completion, the Host Driver will leave this bit set to 1. Clearing both the Stop At Block Gap Request and Continue Request does not

Table continues on the next page...

ESDHCV3x_PROCTL field descriptions (continued)

Field	Description
	cause the transaction to restart. Read Wait is used to stop the read transaction at the block gap. The ESDHC will honor the Stop At Block Gap Request for write transfers, but for read transfers it requires that the SDIO card support Read Wait. Therefore, the Host Driver will not set this bit during read transfers unless the SDIO card supports Read Wait and has set the Read Wait Control to 1, otherwise the ESDHC will stop the SD bus clock to pause the read operation during block gap. In the case of write transfers in which the Host Driver writes data to the Data Port register, the Host Driver will set this bit after all block data is written. If this bit is set to 1, the Host Driver will not write data to the Data Port register after a block is sent. Once this bit is set, the Host Driver will not clear this bit before the Transfer Complete bit in Interrupt Status Register is set, otherwise the ESDHCs behavior is undefined. This bit effects Read Transfer Active, Write Transfer Active, DAT Line Active and Command Inhibit (DAT) in the Present State register. 1 Stop 0 Transfer
15–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–8 DMAS	DMA Select: This field is valid while DMA (SDMA or ADMA) is enabled and selects the DMA operation. 00 No DMA or Simple DMA is selected 01 ADMA1 is selected 10 ADMA2 is selected 11 reserved
7 CDSS	Card Detect Signal Selection: This bit selects the source for the card detection. 1 Card Detection Test Level is selected (for test purpose) 0 Card Detection Level is selected (for normal purpose)
6 CDTL	Card Detect Test Level: This is bit is enabled while the Card Detection Signal Selection is set to 1 and it indicates card insertion. 1 Card Detect Test Level is 1, card inserted 0 Card Detect Test Level is 0, no card inserted
5–4 EMODE	Endian Mode: The ESDHC supports all four endian modes in data transfer. Refer to Data Buffer for more details. 00 Big Endian Mode 01 Half Word Big Endian Mode 10 Little Endian Mode 11 Reserved
3 D3CD	DAT3 as Card Detection Pin: If this bit is set, DAT3 should be pulled down to act as a card detection pin. Be cautious when using this feature, because DAT3 is also a chip-select for the SPI mode. A pull-down on this pin and CMD0 may set the card into the SPI mode, which the ESDHC does not support. In case of SDIO application, if SD_CD signal is not multiplexed to ESDHC block, S/W should set D3CD bit.

Table continues on the next page...

ESDHCV3x_PROCTL field descriptions (continued)

Field	Description
	1 DAT3 as Card Detection Pin 0 DAT3 does not monitor Card Insertion
2–1 DTW[1:0]	Data Transfer Width: This bit selects the data width of the SD bus for a data transfer. The Host Driver sets it to match the data width of the card. Possible Data transfer Width is 1-bit, 4-bits or 8-bits. 10 8-bit mode 01 4-bit mode 00 1-bit mode 11 Reserved
0 LCTL	LED Control: This bit, fully controlled by the Host Driver, is used to caution the user not to remove the card while the card is being accessed. If the software is going to issue multiple SD commands, this bit can be set during all these transactions. It is not necessary to change for each transaction. When the software issues multiple SD commands, setting the bit once before the first command is sufficient; it is not necessary to reset the bit between commands. 1 LED on 0 LED off

30.7.9 System Control (ESDHCV3x_SYSCTL)

Addresses: ESDHCV3-3_SYSCTL is 5002_0000h base + 2Ch offset = 5002_002Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				INITA	0	0	0	IPP_RST_N	0			DTOCV			
W						RSTD	RSTC	RSTA								
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SDCLKFS								DVS[3:0]				SDCLKEN	PEREN	HCKEN	IPGEN
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

ESDHCV3x_SYSCTL field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

ESDHCV3x_SYSCTL field descriptions (continued)

Field	Description
27 INITA	Initialization Active: When this bit is set, 80 SD-Clocks are sent to the card. After the 80 clocks are sent, this bit is self cleared. This bit is very useful during the card power-up period when 74 SD-Clocks are needed and the clock auto gating feature is enabled. Writing 1 to this bit when this bit is already 1 has no effect. Writing 0 to this bit at any time has no effect. When either of the CIHB and CDIHB bits in the Present State Register are set, writing 1 to this bit is ignored (that is, when command line or data lines are active, write to this bit is not allowed). On the other hand, when this bit is set, that is, during initialization active period, it is allowed to issue commands, and the command bit stream will appear on the CMD pad after all 80 clock cycles are completer. When this command ends, the driver can make sure that the 80 clock cycles are sent out, which is very useful when the driver needs to send 80 cycles to the card and does not want to wait until this bit is self cleared.
26 RSTD	Software Reset For DAT Line: Only part of the data circuit is reset. DMA circuit is also reset. The following registers and bits are cleared by this bit: • Data Port register • Buffer is cleared and initialized.Present State register • Buffer Read Enable • Buffer Write Enable • Read Transfer Active • Write Transfer Active • DAT Line Active • Command Inhibit (DAT) Protocol Control register • Continue Request • Stop At Block Gap Request Interrupt Status register • Buffer Read Ready • Buffer Write Ready • DMA Interrupt • Block Gap Event • Transfer Complete 1 Reset 0 No Reset
25 RSTC	Software Reset For CMD Line: Only part of the command circuit is reset. The following registers and bits are cleared by this bit: • Present State register Command Inhibit (CMD) • Interrupt Status register Command Complete 1 Reset 0 No Reset
24 RSTA	Software Reset For ALL: This reset effects the entire Host Controller except for the card detection circuit. Register bits of type ROC, RW, RW1C, RWAC are cleared. During its initialization, the Host Driver will set this bit to 1 to reset the ESDHC. The ESDHC will reset this bit to 0 when the capabilities registers are valid and the Host Driver can read them. Additional use of Software Reset For All does not affect the value of the Capabilities registers. After this bit is set, it is recommended that the Host Driver reset the external card and re-initialize it.

Table continues on the next page...

ESDHCV3x_SYSCTL field descriptions (continued)

Field	Description
	b Reset 0 No Reset
23 IPP_RST_N	If the card supports it, this Register value will be output directly to the CARD through the pad for hardware reset.
22–20 Reserved	This read-only field is reserved and always has the value zero. Reserved
19–16 DTOCV	Data Timeout Counter Value: This value determines the interval by which DAT line timeouts are detected. Refer to the Data Timeout Error bit in the Interrupt Status register for information on factors that dictate time-out generation. Time-out clock frequency will be generated by dividing the base clock SDCLK value by this value. The Host Driver can clear the Data Timeout Error Status Enable (in the Interrupt Status Enable register) to prevent inadvertent time-out events. 1111 Reserved 1110 SDCLK x 2^{27} 0001 SDCLK x 2^{14} 0000 SDCLK x 2^{13} 1111 Reserved (in DDR mode) 1110 SDCLK x 2^{26} (in DDR mode) 0001 SDCLK x 2^{13} (in DDR mode) 0000 SDCLK x 2^{12} (in DDR mode)
15–8 SDCLKFS	SDCLK Frequency Select: This register is used to select the frequency of the SDCLK pin. The frequency is not programmed directly, rather this register holds the prescaler (this register) and divisor (next register) of the Base Clock Frequency register. Setting 00h bypasses the frequency prescaler of the SD Clock. Multiple bits must not be set, or the behavior of this prescaler is undefined. The two default divider values can be calculated by the frequency of ipg_perclk and the following Divisor bits. **NOTE:** In ESDHC this register field can not be set to 00h(that is, bypass). The smallest value is 01h, that is divided by 2. The frequency of SDCLK is set by the following formula: Clock Frequency = (Base Clock) / (prescaler x divisor) For example, if the Base Clock Frequency is 96 MHz, and the target frequency is 25 MHz, then choosing the prescaler value of 01h and divisor value of 1h will yield 24 MHz, which is the nearest frequency less than or equal to the target. Similarly, to approach a clock value of 400 kHz, the prescaler value of 08h and divisor value of eh yields the exact clock value of 400 kHz. The reset value of this bit field is 80h, so if the input Base Clock (ipg_perclk) is about 96 MHz, the default SD Clock after reset is 375 kHz. According to the SD Physical Specification Version 1.1 and the SDIO Card Specification Version 1.2, the maximum SD Clock frequency is 50 MHz and will never exceed this limit. Only the following settings are allowed: 0x80 Base clock divided by 256 0x40 Base clock divided by 128 0x20 Base clock divided by 64

Table continues on the next page...

ESDHCV3x_SYSCTL field descriptions (continued)

Field	Description
	0x10 Base clock divided by 32 0x08 Base clock divided by 16 0x04 Base clock divided by 8 0x02 Base clock divided by 4 0x01 Base clock divided by 2
7–4 DVS[3:0]	Divisor: This register is used to provide a more exact divisor to generate the desired SD clock frequency. **NOTE:** The divider can even support odd divisor without deterioration of duty cycle. The setting are as following: 0x0 Divisor by 1 0x1 Divisor by 2 0xe Divisor by 15 0xf Divisor by 16
3 SDCLKEN	SD Clock Enable The Host Controller will stop SDCLK when writing this bit to 0. SDCLK Frequency can be changed when this bit is 0. Then, the Host Controller will maintain the same clock frequency until SDCLK is stopped (Stop at SDCLK=0). If the Card Inserted in the Present State register is cleared, this bit should be cleared by the Host Driver to save power.
2 PEREN	Peripheral Clock Enable: If this bit is set, ipg_perclk will always be active and no automatic gating is applied. Thus the SDCLK is active except when auto gating-off during buffer danger (buffer about to over-run or under-run). When this bit is cleared, the ipg_perclk will be automatically off when there is no transaction on the SD bus. As this bit is only a feature enabling bit, clearing this bit does not stop SDCLK immediately. The ipg_perclk will be internally gated off, if none of the following factors are met: • The cmd part is reset, or • Data part is reset, or • A soft reset, or • The cmd is about to send, or • Clock divisor is just updated, or • Continue request is just set, or • This bit is set, or • Card insertion is detected, or • Card removal is detected, or • Card external interrupt is detected, or • 80 clocks for initialization phase is ongoing 1 ipg_perclk will not be automatically gated off 0 ipg_perclk will be internally gated off
1 HCKEN	HCLK Enable: If this bit is set, hclk will always be active and no automatic gating is applied. When this bit is cleared, hclk will be automatically off when no data transfer is on the SD bus. 1 hclk will not be automatically gated off 0 hclk will be internally gated off

Table continues on the next page...

ESDHCV3x_SYSCTL field descriptions (continued)

Field	Description
0 IPGEN	IPG Clock Enable: If this bit is set, ipg_clk will always be active and no automatic gating is applied. The ipg_clk will be internally gated off, if none of the following factors are met: • The cmd part is reset, or • Data part is reset, or • Soft reset, or • The cmd is about to send, or • Clock divisor is just updated, or • Continue request is just set, or • This bit is set, or • Card insertion is detected, or • Card removal is detected, or • Card external interrupt is detected, or • The ipg_perclk is not gated off **NOTE:** The ipg_clk will not be auto gated off if the ipg_perclk is not gated off. Clearing only this bit has no effect unless the PEREN bit is also cleared. 1 ipg_clk will not be automatically gated off 0 ipg_clk will be internally gated off

30.7.10 Interrupt Status (ESDHCV3x_IRQSTAT)

An interrupt is generated when the Normal Interrupt Signal Enable is enabled and at least one of the status bits is set to 1. For all bits, writing 1 to a bit clears it; writing to 0 keeps the bit unchanged. More than one status can be cleared with a single register write. Before clearing the Card Interrupt, make sure that the card stops asserting the interrupt, that is, the Card Driver stops servicing the interrupt condition. Otherwise, the CINT bit will be asserted again.

The table below shows the relationship between the Command Timeout Error and the Command Complete.

Table 30-66. ESDHC Status for Command Timeout Error/Command Complete Bit Combinations

Command Complete	Command Timeout Error	Meaning of the Status
0	0	X
X	1	Response not received within 64 SDCLK cycles
1	0	Response received

The table below shows the relationship between the Transfer Complete and the Data Timeout Error.

Table 30-67. ESDHC Status for Data Timeout Error/Transfer Complete Bit Combinations

Transfer Complete	Data Timeout Error	Meaning of the Status
0	0	X
0	1	Timeout occurred during transfer
1	X	Data Transfer Complete

The table below shows the relationship between the Command CRC Error and Command Timeout Error.

Table 30-68. ESDHC Status for Command CRC Error/Command Timeout Error Bit Combinations

Command Complete	Command Timeout Error	Meaning of the Status
0	0	No error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	CMD line conflict

Addresses: ESDHCV3-3_IRQSTAT is 5002_0000h base + 30h offset = 5002_0030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			DMAE	0			AC12E	0	DEBE	DCE	DTOE	CIE	CEBE	CCE	CTOE
W				w1c				w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CINT	CRM	CINS	BRR	BWR	DINT	BGE	TC	CC
W								w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDHCV3x_IRQSTAT field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28 DMAE	DMA Error:

Table continues on the next page...

ESDHCV3x_IRQSTAT field descriptions (continued)

Field	Description
	Occurs when an Internal DMA transfer has failed. This bit is set to 1, when some error occurs in the data transfer. This error can be caused by either Simple DMA or ADMA, depending on which DMA is in use. The value in DMA System Address register is the next fetch address where the error occurs. As any error corrupts the whole data block, the Host Driver will re-start the transfer from the corrupted block boundary. The address of the block boundary can be calculated either from the current DS_ADDR value or from the remaining number of blocks and the block size. 1 Error 0 No Error
27–25 Reserved	This read-only field is reserved and always has the value zero. Reserved
24 AC12E	Auto CMD12 Error: Occurs when detecting that one of the bits in the Auto CMD12 Error Status register has changed from 0 to 1. This bit is set to 1, not only when the errors in Auto CMD12 occur, but also when the Auto CMD12 is not executed due to the previous command error. 1 Error 0 No Error
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22 DEBE	Data End Bit Error: Occurs either when detecting 0 at the end bit position of read data, which uses the DAT line, or at the end bit position of the CRC. 1 Error 0 No Error
21 DCE	Data CRC Error: Occurs when detecting a CRC error when transferring read data, which uses the DAT line, or when detecting the Write CRC status having a value other than 010. 1 Error 0 No Error
20 DTOE	Data Timeout Error: Occurs when detecting one of following time-out conditions. • Busy time-out for R1b,R5b type • Busy time-out after Write CRC status • Read Data time-out. 1 Time out 0 No Error
19 CIE	Command Index Error: Occurs if a Command Index error occurs in the command response. 1 Error 0 No Error
18 CEBE	Command End Bit Error: Occurs when detecting that the end bit of a command response is 0.

Table continues on the next page...

ESDHCV3x_IRQSTAT field descriptions (continued)

Field	Description
	1 End Bit Error Generated 0 No Error
17 CCE	Command CRC Error: Command CRC Error is generated in two cases. • If a response is returned and the Command Timeout Error is set to 0 (indicating no time-out), this bit is set when detecting a CRC error in the command response. • The ESDHC detects a CMD line conflict by monitoring the CMD line when a command is issued. If the ESDHC drives the CMD line to 1, but detects 0 on the CMD line at the next SDCLK edge, then the ESDHC will abort the command (Stop driving CMD line) and set this bit to 1. The Command Timeout Error will also be set to 1 to distinguish CMD line conflict. 1 CRC Error Generated. 0 No Error
16 CTOE	Command Timeout Error: Occurs only if no response is returned within 64 SDCLK cycles from the end bit of the command. If the ESDHC detects a CMD line conflict, in which case a Command CRC Error will also be set (as shown in the table above), this bit will be set without waiting for 64 SDCLK cycles. Because the command will be aborted by the ESDHC. 1 Time out 0 No Error
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 CINT	Card Interrupt: This status bit is set when an interrupt signal is detected from the external card. In 1-bit mode, the ESDHC will detect the Card Interrupt without the SD Clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so the interrupt from card can only be sampled during interrupt cycle, introducing some delay between the interrupt signal from the SDIO card and the interrupt to the Host System. Writing 1 to this bit will clear it, but as the interrupt factor from the SDIO card is not clear, this bit is set again. In order to clear this bit, reset the interrupt factor from the external card and then clear this bit. When this status bit is set, and the Host Driver needs to service this interrupt, the Card Interrupt Signal Enable in the Interrupt Signal Enable register should be cleared to deactivate the interrupt signal to the Host System. After completion of the card interrupt service, (it should reset the interrupt factors in the SDIO card and the interrupt signal should not be asserted), write 1 to clear this bit, enable the Card Interrupt Signal Enable and start sampling the interrupt signal again. 1 Generate Card Interrupt 0 No Card Interrupt
7 CRM	Card Removal: This status bit is set if the Card Inserted bit (PRSSTNT[CINS]) changes from 1 to 0. When the Host Driver writes this bit to 1 to clear this status, the status of CINS should be confirmed, because the card state may possibly change when the Host Driver clears this bit and the interrupt event may not be generated. When this bit is cleared, it will be set again if no card is inserted. In order to leave it cleared, clear the Card Removal Status Enable bit IRQSTATEN[CRMSEN]in Interrupt Status Enable register. 1 Card removed 0 Card state unstable or inserted

Table continues on the next page...

ESDHCV3x_IRQSTAT field descriptions (continued)

Field	Description
6 CINS	Card Insertion: This status bit is set if the Card Inserted bit in the Present State register changes from 0 to 1. When the Host Driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card state may possibly be changed when the Host Driver clears this bit and the interrupt event may not be generated. When this bit is cleared, it will be set again if a card is inserted. In order to leave it cleared, clear the Card Inserted Status Enable bit in Interrupt Status Enable register. 1 Card inserted 0 Card state unstable or removed
5 BRR	Buffer Read Ready: This status bit is set if the Buffer Read Enable bit, in the Present State register, changes from 0 to 1. Refer to the Buffer Read Enable bit in the Present State register for additional information. 1 Ready to read buffer 0 Not ready to read buffer
4 BWR	Buffer Write Ready: This status bit is set if the Buffer Write Enable bit, in the Present State register, changes from 0 to 1. Refer to the Buffer Write Enable bit in the Present State register for additional information. 1 Ready to write buffer: 0 Not ready to write buffer
3 DINT	DMA Interrupt: Occurs only when the internal DMA finishes the data transfer successfully. When errors occur during data transfer, this bit will not be set. Instead, the DMAE bit will be set. Either Simple DMA or ADMA finishes data transferring, this bit will be set. 1 DMA Interrupt is generated 0 No DMA Interrupt
2 BGE	Block Gap Event: If the Stop At Block Gap Request bit in the Protocol Control register is set, this bit is set when a read or write transaction is stopped at a block gap. If Stop At Block Gap Request is not set to 1, this bit is not set to 1. In the case of a Read Transaction: This bit is set at the falling edge of the DAT Line Active Status (When the transaction is stopped at SD Bus timing). The Read Wait must be supported in order to use this function. In the case of Write Transaction: This bit is set at the falling edge of Write Transfer Active Status (After getting CRC status at SD Bus timing). 1 Transaction stopped at block gap 0 No block gap event
1 TC	Transfer Complete: This bit is set when a read or write transfer is completed.

Table continues on the next page...

ESDHCV3x_IRQSTAT field descriptions (continued)

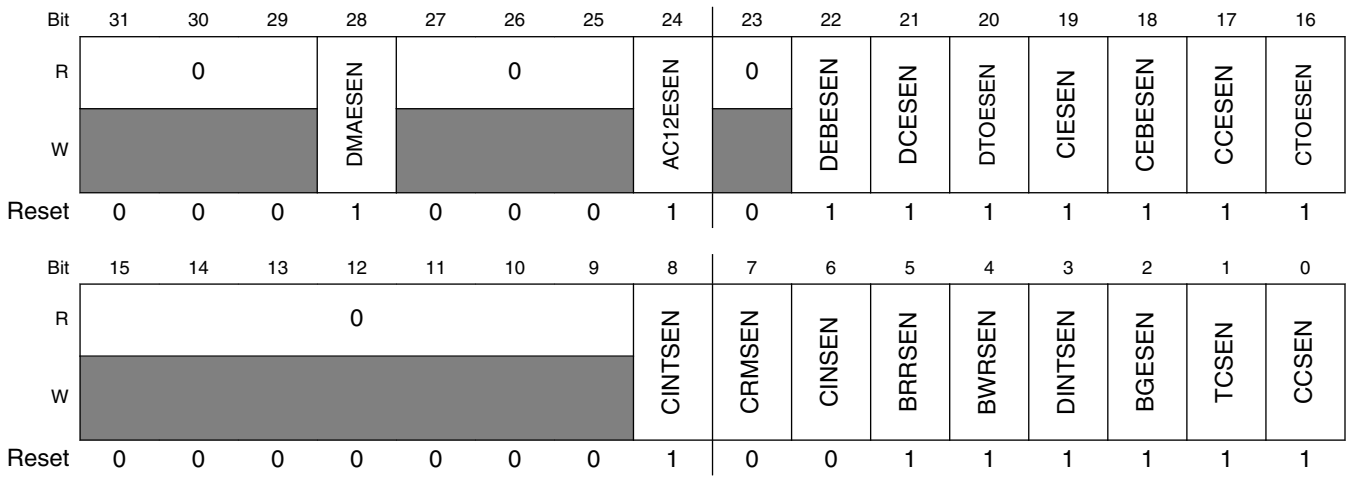
Field	Description
	In the case of a Read Transaction: This bit is set at the falling edge of the Read Transfer Active Status. There are two cases in which this interrupt is generated. The first is when a data transfer is completed as specified by the data length (after the last data has been read to the Host System). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request bit in the Protocol Control register (after valid data has been read to the Host System). In the case of a Write Transaction: This bit is set at the falling edge of the DAT Line Active Status. There are two cases in which this interrupt is generated. The first is when the last data is written to the SD card as specified by the data length and the busy signal is released. The second is when data transfers are stopped at the block gap, by setting the Stop At Block Gap Request bit in the Protocol Control register, and the data transfers are completed. (After valid data is written to the SD card and the busy signal released.) 1 Transfer complete 0 Transfer not complete
0 CC	Command Complete: This bit is set when you receive the end bit of the command response (except Auto CMD12). Refer to the Command Inhibit (CMD) in the Present State register. 1 Command complete 0 Command not complete

30.7.11 Interrupt Status Enable (ESDHCV3x_IRQSTATEN)

Setting the bits in this register to 1 enables the corresponding Interrupt Status to be set by the specified event. If any bit is cleared, the corresponding Interrupt Status bit is also cleared (that is, when the bit in this register is cleared, the corresponding bit in Interrupt Status Register is always 0).

- Depending on IABG bit setting, ESDHC may be programmed to sample the card interrupt signal during the interrupt period and hold its value in the flip-flop. There will be some delays on the Card Interrupt, asserted from the card, to the time the Host System is informed.
- To detect a CMD line conflict, the Host Driver must set both Command Timeout Error Status Enable and Command CRC Error Status Enable to 1.

Addresses: ESDHCV3-3_IRQSTATEN is 5002_0000h base + 34h offset = 5002_0034h



ESDHCV3x_IRQSTATEN field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28 DMAESEN	DMA Error Status Enable: 1 Enabled 0 Masked
27–25 Reserved	This read-only field is reserved and always has the value zero. Reserved
24 AC12ESEN	Auto CMD12 Error Status Enable: 1 Enabled 0 Masked
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22 DEBESEN	Data End Bit Error Status Enable: 1 Enabled 0 Masked
21 DCESEN	Data CRC Error Status Enable: 1 Enabled 0 Masked
20 DTOESEN	Data Timeout Error Status Enable: 1 Enabled 0 Masked
19 CIESEN	Command Index Error Status Enable: 1 Enabled 0 Masked

Table continues on the next page...

ESDHCV3x_IRQSTATEN field descriptions (continued)

Field	Description
18 CEBESEN	Command End Bit Error Status Enable: 1 Enabled 0 Masked
17 CCESEN	Command CRC Error Status Enable: 1 Enabled 0 Masked
16 CTOESEN	Command Timeout Error Status Enable: 1 Enabled 0 Masked
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 CINTSEN	Card Interrupt Status Enable: If this bit is set to 0, the ESDHC will clear the interrupt request to the System. The Card Interrupt detection is stopped when this bit is cleared and restarted when this bit is set to 1. The Host Driver should clear the Card Interrupt Status Enable before servicing the Card Interrupt and should set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts. 1 Enabled 0 Masked
7 CRMSEN	Card Removal Status Enable: 1 Enabled 0 Masked
6 CINSEN	Card Insertion Status Enable: 1 Enabled 0 Masked
5 BRRSEN	Buffer Read Ready Status Enable: 1 Enabled 0 Masked
4 BWRSEN	Buffer Write Ready Status Enable: 1 Enabled 0 Masked
3 DINTSEN	DMA Interrupt Status Enable: 1 Enabled 0 Masked
2 BGESEN	Block Gap Event Status Enable: 1 Enabled 0 Masked

Table continues on the next page...

ESDHCV3x_IRQSTATEN field descriptions (continued)

Field	Description
1 TCSEN	Transfer Complete Status Enable: 1 Enabled 0 Masked
0 CCSEN	Command Complete Status Enable: 1 Enabled 0 Masked

30.7.12 Interrupt Signal Enable (ESDHCV3x_IRQSIGEN)

This register is used to select which interrupt status is indicated to the Host System as the interrupt. These status bits all share the same interrupt line. Setting any of these bits to 1 enables interrupt generation. The corresponding Status register bit will generate an interrupt when the corresponding interrupt signal enable bit is set.

Addresses: ESDHCV3-3_IRQSIGEN is 5002_0000h base + 38h offset = 5002_0038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			DMAEIEN	0			AC12EIEN	0	DEBEIEN	DCEIEN	DTOEIEN	CIEIEN	CEBEIEN	CCEIEN	CTOEIEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CINTIEN	CRMIEN	CINIEN	BRRIEN	BWRIEN	DINTIEN	BGEIEN	TCIEN	CCIEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDHCV3x_IRQSIGEN field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28 DMAEIEN	DMA Error Interrupt Enable: 1 Enable 0 Masked
27–25 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

ESDHCV3x_IRQSIGEN field descriptions (continued)

Field	Description
24 AC12EIEN	Auto CMD12 Error Interrupt Enable: 1 Enabled 0 Masked
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22 DEBEIEN	Data End Bit Error Interrupt Enable: 1 Enabled 0 Masked
21 DCEIEN	Data CRC Error Interrupt Enable: 1 Enabled 0 Masked
20 DTOEIEN	Data Timeout Error Interrupt Enable: 1 Enabled 0 Masked
19 CIEIEN	Command Index Error Interrupt Enable: 1 Enabled 0 Masked
18 CEBEIEN	Command End Bit Error Interrupt Enable: 1 Enabled 0 Masked
17 CCEIEN	Command CRC Error Interrupt Enable: 1 Enabled 0 Masked
16 CTOEIEN	Command Timeout Error Interrupt Enable 1 Enabled 0 Masked
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 CINTIEN	Card Interrupt Interrupt Enable: 1 Enabled 0 Masked
7 CRMIEN	Card Removal Interrupt Enable: 1 Enabled 0 Masked
6 CINIEN	Card Insertion Interrupt Enable:

Table continues on the next page...

ESDHCV3x_IRQSIGEN field descriptions (continued)

Field	Description
	1 Enabled 0 Masked
5 BRRIEN	Buffer Read Ready Interrupt Enable: 1 Enabled 0 Masked
4 BWRIEN	Buffer Write Ready Interrupt Enable: 1 Enabled 0 Masked
3 DINTIEN	DMA Interrupt Enable: 1 Enabled 0 Masked
2 BGEIEN	Block Gap Event Interrupt Enable: 1 Enabled 0 Masked
1 TCIEN	Transfer Complete Interrupt Enable: 1 Enabled 0 Masked
0 CCIEN	Command Complete Interrupt Enable: 1 Enabled 0 Masked

30.7.13 Auto CMD12 Status (ESDHCV3x_AUTOC12ERR)

When the Auto CMD12 Error Status bit in the Status register is set, the Host Driver will check this register to identify what kind of error the Auto CMD12 indicated. This register is valid only when the Auto CMD12 Error status bit is set.

The table below shows the relationship between the Auto CMGD12 CRC Error and the Auto CMD12 Command Timeout Error.

Table 30-72. Relationship Between Command CRC Error and Command Timeout Error for Auto CMD12

Auto CMD12 CRC Error	Auto CMD12 Timeout Error	Type of Error
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	CMD line conflict

Changes in Auto CMD12 Error Status register can be classified in three scenarios:

1. When the ESDHC is going to issue an Auto CMD12.
   - Set bit 0 to 1 if the Auto CMD12 cannot be issued due to an error in the previous command
   - Set bit 0 to 0 if the Auto CMD12 is issued
2. At the end bit of an Auto CMD12 response.
   - Check errors correspond to bits 1-4.
   - Set bits 1-4 corresponding to detected errors.
   - Clear bits 1-4 corresponding to detected errors
3. Before reading the Auto CMD12 Error Status bit 7.
   - Set bit 7 to 1 if there is a command that cannot be issued
   - Clear bit 7 if there is no command to issue

The timing for generating the Auto CMD12 Error and writing to the Command register are asynchronous. After that, bit 7 will be sampled when the driver is not writing to the Command register. It is suggested to read this register only when the AC12E bit in Interrupt Status register is set. An Auto CMD12 Error Interrupt is generated when one of the error bits (0-4) is set to 1. The Command Not Issued By Auto CMD12 Error does not generate an interrupt.

Addresses: ESDHCV3-3_AUTOC12ERR is 5002_0000h base + 3Ch offset = 5002_003Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CNIBAC12E	0		AC12IE	AC12CE	AC12EBE	AC12TOE	AC12NE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDHCV3x_AUTOC12ERR field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7 CNIBAC12E	Command Not Issued By Auto CMD12 Error: Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 Error (D04-D01) in this register. 1 Not Issued 0 No error
6–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 AC12IE	Auto CMD12 Index Error: Occurs if the Command Index error occurs in response to a command. 1 Error, the CMD index in response is not CMD12 0 No error
3 AC12CE	Auto CMD12 CRC Error: Occurs when detecting a CRC error in the command response. 1 CRC Error Met in Auto CMD12 Response 0 No CRC error
2 AC12EBE	Auto CMD12 End Bit Error: Occurs when detecting that the end bit of command response is 0 which should be 1. 1 End Bit Error Generated 0 No error
1 AC12TOE	Auto CMD12 Timeout Error: Occurs if no response is returned within 64 SDCLK cycles from the end bit of the command. If this bit is set to1, the other error status bits (2-4) have no meaning. 1 Time out 0 No error
0 AC12NE	Auto CMD12 Not Executed: If memory multiple block data transfer is not started, due to a command error, this bit is not set because it is not necessary to issue an Auto CMD12. Setting this bit to 1 means the ESDHC cannot issue the Auto CMD12 to stop a memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (1-4) have no meaning. 1 Not executed 0 Executed

30.7.14 Host Controller Capabilities (ESDHCV3x_HOSTCAPBLT)

This register provides the Host Driver with information specific to the ESDHC implementation. The value in this register is the power-on-reset value, and does not change with a software reset. Any write to this register is ignored.

Addresses: ESDHCV3-3_HOSTCAPBLT is 5002_0000h base + 40h offset = 5002_0040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					VS18	VS30	VS33	SRS	DMAS	HSS	ADMAS	-	MBL[2:0]		
W																
Reset	0	0	0	0	0	1	1	1	1	1	1	1	0	0	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDHCV3x_HOSTCAPBLT field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26 VS18	Voltage Support 1.8V: This bit depends on the Host System ability. 1 1.8V supported 0 1.8V not supported
25 VS30	Voltage Support 3.0V: This bit depends on the Host System ability. 1 3.0V supported 0 3.0V not supported
24 VS33	Voltage Support 3.3V: This bit depends on the Host System ability. 1 3.3V supported 0 3.3V not supported
23 SRS	Suspend / Resume Support:

Table continues on the next page...

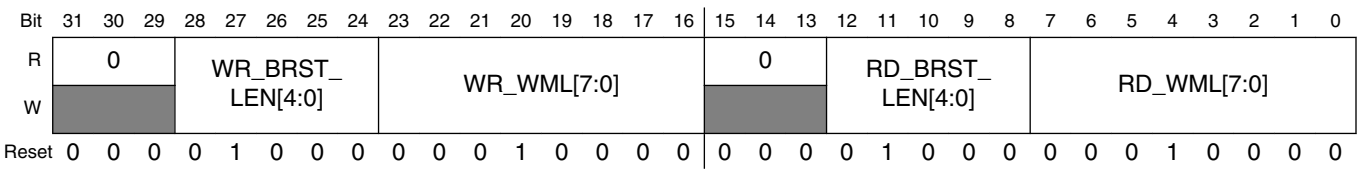
ESDHCV3x_HOSTCAPBLT field descriptions (continued)

Field	Description
	This bit indicates whether the ESDHC supports Suspend / Resume functionality. If this bit is 0, the Suspend and Resume mechanism, as well as the Read Wait, are not supported, and the Host Driver will not issue either Suspend or Resume commands. 1 Supported 0 Not supported
22 DMAS	DMA Support: This bit indicates whether the ESDHC is capable of using the internal DMA to transfer data between system memory and the data buffer directly. 1 DMA Supported 0 DMA not supported
21 HSS	High Speed Support: This bit indicates whether the ESDHC supports High Speed mode and the Host System can supply a SD Clock frequency from 25 MHz to 50 MHz. 1 High Speed Supported 0 High Speed Not Supported
20 ADMAS	ADMA Support: This bit indicates whether the ESDHC supports the ADMA feature. 1 Advanced DMA Supported 0 Advanced DMA Not supported
19 -	Reserved
18–16 MBL[2:0]	Max Block Length: This value indicates the maximum block size that the Host Driver can read and write to the buffer in the ESDHC. The buffer will transfer block size without wait cycles. 000 512 bytes 001 1024 bytes 010 2048 bytes 011 4096 bytes
15–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

30.7.15 Watermark Level (ESDHCV3x_WML)

Both write and read watermark levels (FIFO threshold) are configurable. There value can range from 1 to 128 words. Both write and read burst lengths are also configurable. There value can range from 1 to 31 words.

Addresses: ESDHCV3-3_WML is 5002_0000h base + 44h offset = 5002_0044h



ESDHCV3x_WML field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28–24 WR_BRST_ LEN[4:0]	Write Burst Length: The number of words the ESDHC writes in a single burst. The write burst length must be less than or equal to the write watermark level, and all bursts within a watermark level transfer will be in back-to-back mode. On reset, this field will be 8. Writing 0 to this field will result in '01000' (that is, it is not able to clear this field). **NOTE:** Due to a system restriction, the actual burst length may not exceed 16.
23–16 WR_WML[7:0]	Write Watermark Level: The number of words used as the watermark level (FIFO threshold) in a DMA write operation. Also the number of words as a sequence of write bursts in back-to-back mode. The maximum legal value for the write watermark level is 128.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12–8 RD_BRST_ LEN[4:0]	Read Burst Length: The number of words the ESDHC reads in a single burst. The read burst length must be less than or equal to the read watermark level, and all bursts within a watermark level transfer will be in back-to-back mode. On reset, this field will be 8. Writing 0 to this field will result in '01000' (that is, it is not able to clear this field. **NOTE:** Due to a system restriction, the actual burst length may not exceed 16.
7–0 RD_WML[7:0]	Read Watermark Level: The number of words used as the watermark level (FIFO threshold) in a DMA read operation. Also the number of words as a sequence of read bursts in back-to-back mode. The maximum legal value for the read water mark level is 128.

30.7.16 Force Event (ESDHCV3x_FEVT)

The Force Event Register is not a physically implemented register. Rather, it is an address at which the Interrupt Status Register can be written if the corresponding bit of the Interrupt Status Enable Register is set. This register is a write only register and writing 0 to it has no effect. Writing 1 to this register sets the corresponding bit of Interrupt Status Register. A read from this register always results in 0's. In order to change the corresponding status bits in the Interrupt Status Register, set IPGEN bit in System Control Register so that ipg_clk is always active.

Forcing a card interrupt will generate a short pulse on the DAT[1] line, and the driver will treat this interrupt as a normal interrupt. The interrupt service routine will skip polling the card interrupt factor as the interrupt is self cleared.

Addresses: ESDHCV3-3_FEVT is 5002_0000h base + 50h offset = 5002_0050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0		0	0			0	0	0	0	0	0	0	0	0
W	FEVTCINT			FEVTDMAE				FEVTAC12E		FEVTDEBE	FEVTDCE	FEVTDTOE	FEVTCIE	FEVTCEBE	FEVTCCE	FEVTCCE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0	0		0	0	0	0	0
W									FEVTCNIBAC12E			FEVTAC12IE	FEVTAC12EBE	FEVTAC12CE	FEVTAC12TOE	FEVTAC12NE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDHCV3x_FEVT field descriptions

Field	Description
31 FEVTCINT	Force Event Card Interrupt: Writing 1 to this bit generates a short low-level pulse on the internal DAT[1] line, as if a self clearing interrupt was received from the external card. If enabled, the CINT bit will be set and the interrupt service routine will treat this interrupt as a normal interrupt from the external card.
30–29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28 FEVTDMAE	Force Event DMA Error: Forces the DMAE bit of Interrupt Status Register to be set
27–25 Reserved	This read-only field is reserved and always has the value zero. Reserved
24 FEVTAC12E	Force Event Auto Command 12 Error: Forces the AC12E bit of Interrupt Status Register to be set
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22 FEVTDEBE	Force Event Data End Bit Error: Forces the DEBE bit of Interrupt Status Register to be set
21 FEVTDCE	Force Event Data CRC Error: Forces the DCE bit of Interrupt Status Register to be set
20 FEVTDTOE	Force Event Data Time Out Error: Force the DTOE bit of Interrupt Status Register to be set
19 FEVTCIE	Force Event Command Index Error: Forces the CCE bit of Interrupt Status Register to be set
18 FEVTCEBE	Force Event Command End Bit Error: Forces the CEBE bit of Interrupt Status Register to be set
17 FEVTCCE	Force Event Command CRC Error: Forces the CCE bit of Interrupt Status Register to be set
16 FEVTCCE	Force Event Command Time Out Error: Forces the CTOE bit of Interrupt Status Register to be set
15–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7 FEVTCNIBAC12E	Force Event Command Not Executed By Auto Command 12 Error: Forces the CNIBAC12E bit in the Auto Command12 Error Status Register to be set
6–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 FEVTAC12IE	Force Event Auto Command 12 Index Error: Forces the AC12IE bit in the Auto Command12 Error Status Register to be set
3 FEVTAC12EBE	Force Event Auto Command 12 End Bit Error: Forces the AC12EBE bit in the Auto Command12 Error Status Register to be set

Table continues on the next page...

ESDHCV3x_FEVT field descriptions (continued)

Field	Description
2 FEVTAC12CE	Force Event Auto Command 12 CRC Error: Forces the AC12CE bit in the Auto Command12 Error Status Register to be set
1 FEVTAC12TOE	Force Event Auto Command 12 Time Out Error: Forces the AC12TOE bit in the Auto Command12 Error Status Register to be set
0 FEVTAC12NE	Force Event Auto Command 12 Not Executed: Forces the AC12NE bit in the Auto Command12 Error Status Register to be set

30.7.17 ADMA Error Status Register (ESDHCV3x_ADMAES)

When an ADMA Error Interrupt occurrs, the ADMA Error States field (ADMAES) in this register holds the ADMA state and the ADMA System Address register (ADSADDR) holds the address of the error descriptor.

To recover from this error, the Host Driver requires the ADMA state to identify the error descriptor address as follows:

- ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address
- ST_FDS: Current location set in the ADMA System Address register is the error descriptor address
- ST_CADR: This state is never set because it only increments the descriptor pointer and doesn't generate an ADMA error
- ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address

In case of a write operation, the Host Driver should use ACMD22 command (see below table) to get the number of the written block, because unwritten data may exist in the Host Controller.

The Host Controller generates the ADMA Error Interrupt when it detects invalid descriptor data (Valid=0) in the ST_FDS state. The Host Driver can distinguish this error by reading the Valid bit of the error descriptor.

Table 30-77. ADMA Error State Coding

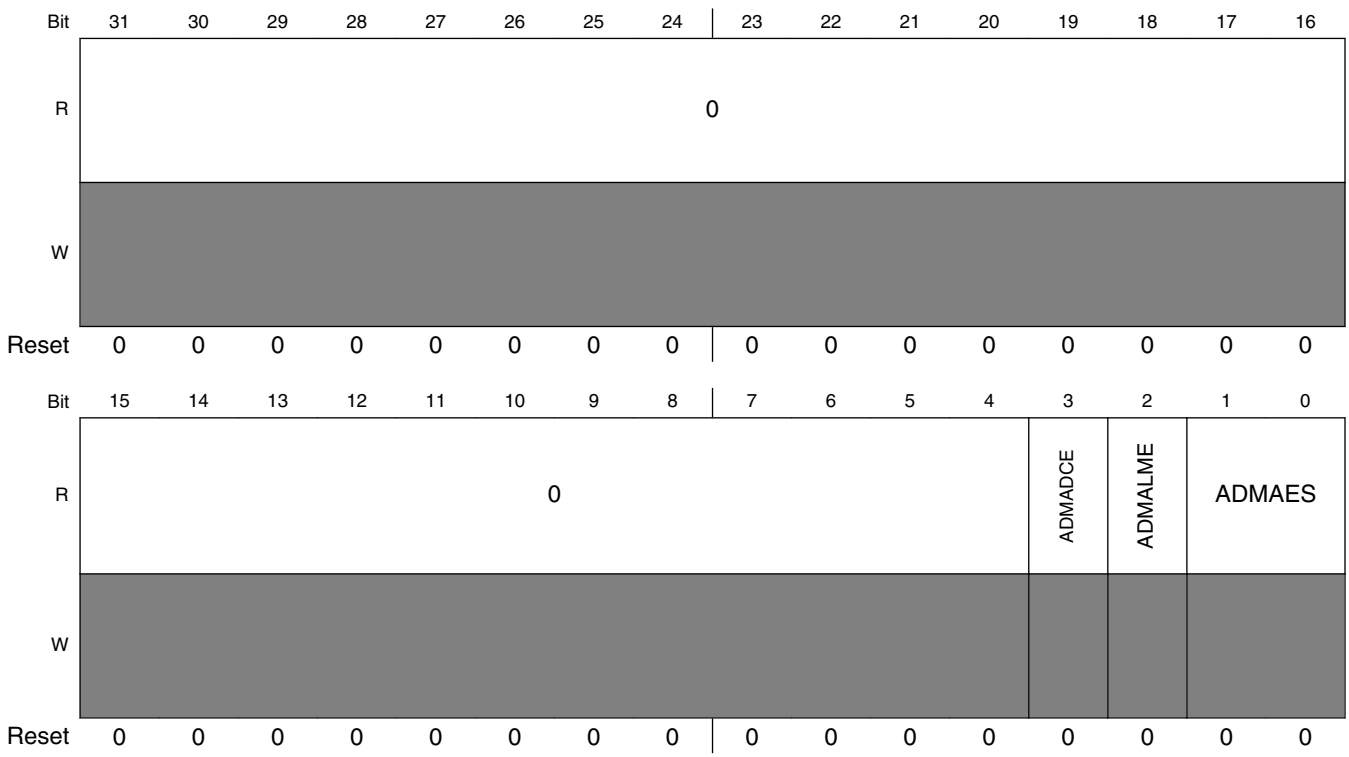
D01-D00	ADMA Error State (when error has occurred)	Contents of ADMA System Address Register
00	ST_STOP (Stop DMA)	Holds the address of the next executable Descriptor command
01	ST_FDS (Fetch Descriptor)	Holds the valid Descriptor address

Table continues on the next page...

Table 30-77. ADMA Error State Coding (continued)

D01-D00	ADMA Error State (when error has occurred)	Contents of ADMA System Address Register
10	ST_CADR (Change Address)	No ADMA Error is generated
11	ST_TFR (Transfer Data)	Holds the address of the next executable Descriptor command

Addresses: ESDHCV3-3_ADMAES is 5002_0000h base + 54h offset = 5002_0054h



ESDHCV3x_ADMAES field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value zero. Reserved
3 ADMADCE	ADMA Descritor Error: This error occurs when invalid descriptor fetched by ADMA: 1 Error 0 No Error
2 ADMALME	ADMA Length Mismatch Error: This error occurs in the following 2 cases: • While the Block Count Enable is being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length • Total data length can not be divided by the block length

Table continues on the next page...

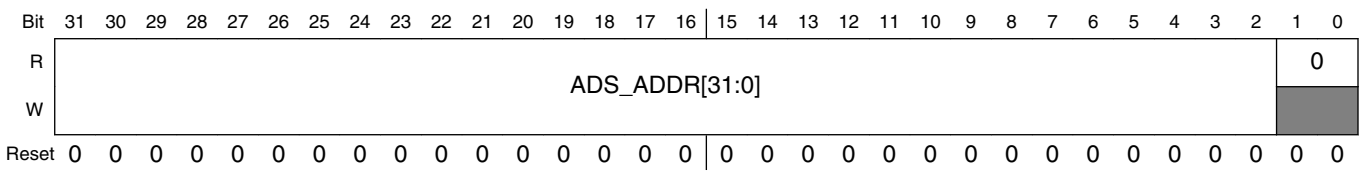
ESDHCV3x_ADMAES field descriptions (continued)

Field	Description
	1 Error 0 No Error
1–0 ADMAES	ADMA Error State (when ADMA Error is occurred.): This field indicates the state of the ADMA when an error has occurred during an ADMA data transfer. Refer to ADMA Error Status Register (ESDHCV3_ADMAES) for more details.

30.7.18 ADMA System Address (ESDHCV3x_ADSADDR)

This register contains the physical system memory address used for ADMA transfers.

Addresses: ESDHCV3-3_ADSADDR is 5002_0000h base + 58h offset = 5002_0058h



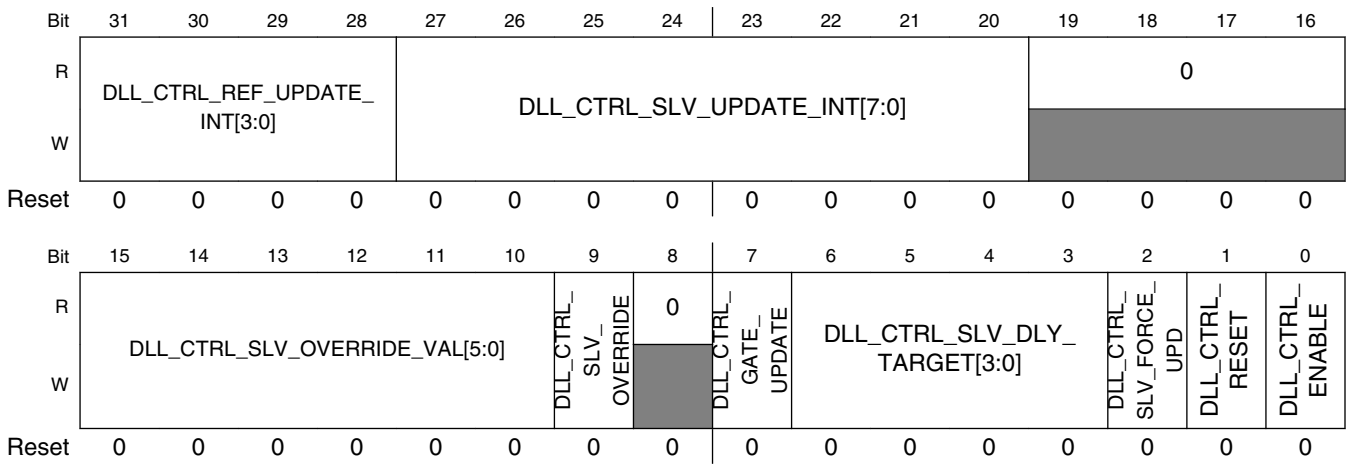
ESDHCV3x_ADSADDR field descriptions

Field	Description
31–2 ADS_ ADDR[31:0]	ADMA System Address: This register holds the word address of the executing command in the Descriptor table. At the start of ADMA, the Host Driver will set the start address of the Descriptor table. The ADMA engine increments this register address whenever fetching a Descriptor command. When the ADMA is stopped at the Block Gap, this register indicates the address of the next executable Descriptor command. When the ADMA Error Interrupt is generated, this register will hold the valid Descriptor address depending on the ADMA state. The lower 2 bits of this register is tied to '0' so the ADMA address is always word aligned. As this register supports dynamic address reflecting, when TC bit is set, it automatically alters the value of internal address counter, so this register value cannot be changed when the TC bit is set. Such restriction is also listed in on page 30-142.
1–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

30.7.19 DLL (Delay Line) Control (ESDHCV3x_DLLCTRL)

This register contains control bits for DLL.

Addresses: ESDHCV3-3_DLLCTRL is 5002_0000h base + 60h offset = 5002_0060h



ESDHCV3x_DLLCTRL field descriptions

Field	Description
31–28 DLL_CTRL_REF_UPDATE_INT[3:0]	DLL control loop update interval. The interval cycle is (2 + REF_UPDATE_INT) * ref_clock. By default, the DLL control loop will update every two ref_clock cycles. It should be noted that increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that may effect the delay (such as voltage and temperature)
27–20 DLL_CTRL_SLV_UPDATE_INT[7:0]	Slave delay line update interval. If default 0 is used, it means 256 cycles of ref_clock. A value of 0x0f results in 15 cycles and so on. **NOTE:** The software can always cause an update of the slave-delay line using the SLV_FORCE_UPDATE register. **NOTE:** The slave delay line will also update automatically when the reference DLL transitions to a locked state (from an un-locked state).
19–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–10 DLL_CTRL_SLV_OVERRIDE_VAL[5:0]	When SLV_OVERRIDE=1 This field is used to select 1 of 64 physical taps manually. A value of 0 selects tap 1, and a value of 0x3f selects tap 64.
9 DLL_CTRL_SLV_OVERRIDE	Set this bit to 1 to Enable manual override for slave delay chain using SLV_OVERRIDE_VAL; to set 0 to disable manual override. This feature does not require the DLL to tbe enabled using the ENABLE bit. In fact to reduce power, if SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE=0
8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7 DLL_CTRL_GATE_UPDATE	Set this bit to 1 to force DLL to not update from now on. Beacue, when clock_in exists, glitches might appear during update. This bit is used by software if we met such kind of condition. Set it to 0 to allow DLL to update automatically

Table continues on the next page...

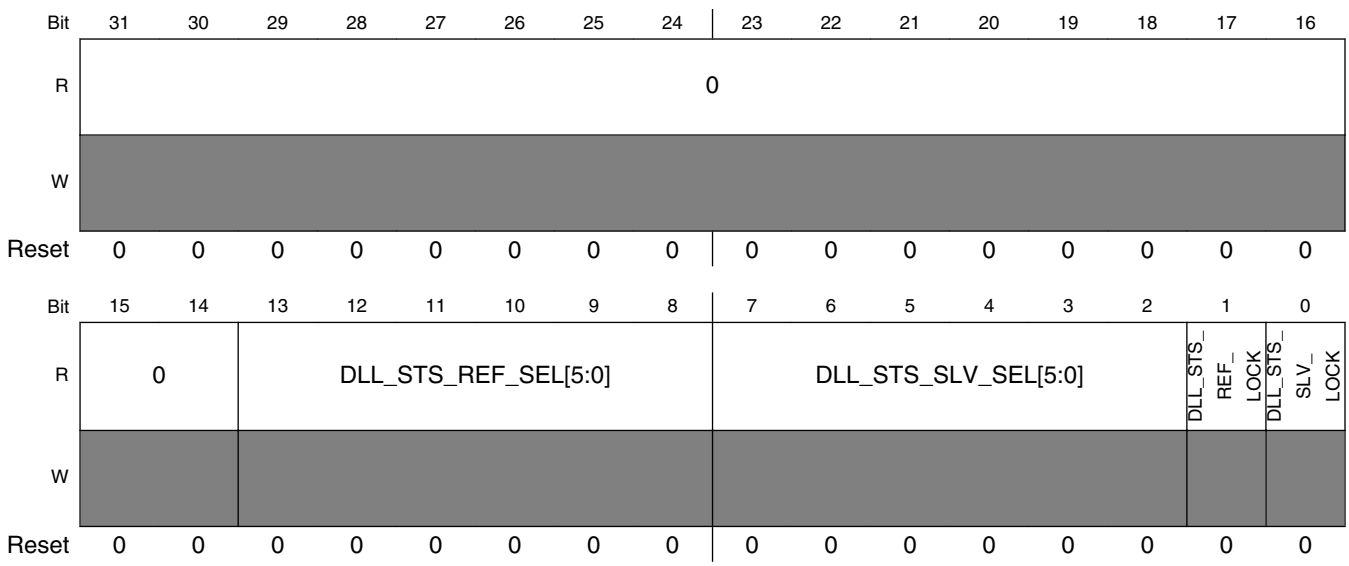
ESDHCV3x_DLLCTRL field descriptions (continued)

Field	Description
6–3 DLL_CTRL_ SLV_DLY_ TARGET[3:0]	The delay target for the ESDHC loopback read clock can be programmed in 1/16th increments of an ref_clock half-period. So the input read-clock can be delayed relative input data from (ref_clock/2)/16 to ref_clock/2
2 DLL_CTRL_ SLV_FORCE_ UPD	Setting this bit to 1, forces the slave delay line to update to the DLL calibrated value immediately. The slave delay line will update automatically based on the SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the slave-line update can only occur if SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered). Be sure to use it when ESDHC is idle. This function may not work when ESDHC is working on data/cmd/response.
1 DLL_CTRL_ RESET	Setting this bit to 1 force a reset on DLL. This will cause the DLL to lose lock and re-calibrate to detect an ref_clock half period phase shift. This signal is used by the DLL as edge-sensitive, so in order to create a subsequent reset, RESET must be taken low and then asserted again
0 DLL_CTRL_ ENABLE	Set this bit to 1 to enable the DLL and delay chain; otherwise; set to 0 to bypasses DLL. **NOTE:** Using the slave delay line override feature with SLV_OVERRIDE and SLV_OVERRIDE VAL, the DLL does not need to be enabled

30.7.20 DLL Status (ESDHCV3x_DLLSTS)

This register contains the DLL status information. All bits are read only and will read the same as the power-reset value.

Addresses: ESDHCV3-3_DLLSTS is 5002_0000h base + 64h offset = 5002_0064h



ESDHCV3x_DLLSTS field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

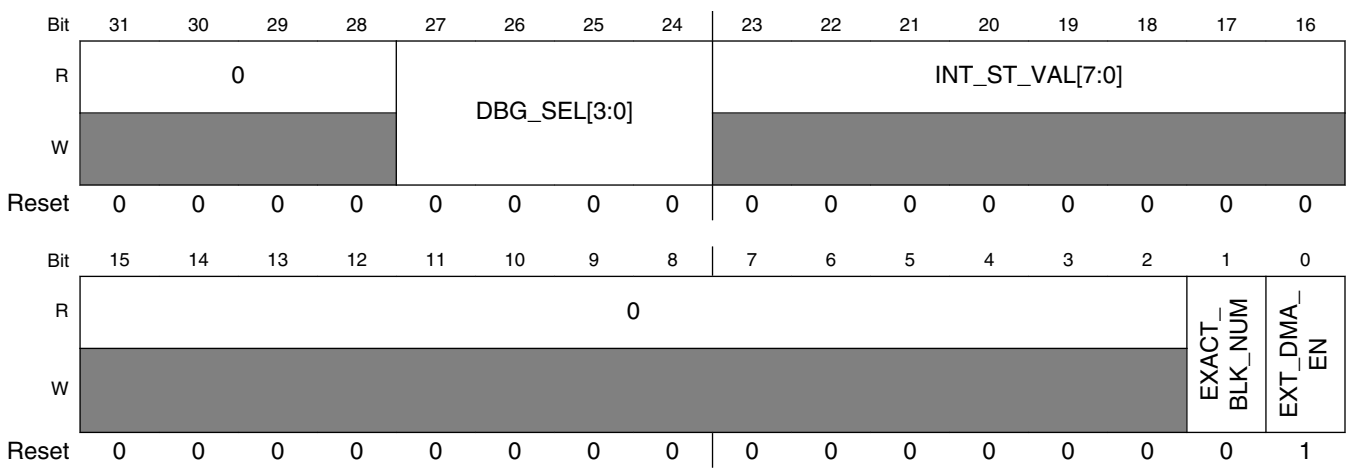
ESDHCV3x_DLLSTS field descriptions (continued)

Field	Description
13–8 DLL_STS_REF_SEL[5:0]	Reference delay line select taps. **NOTE:** this is encoded by 6 bits for 64 taps.
7–2 DLL_STS_SLV_SEL[5:0]	Slave delay line select status. This is the instant value generated from the reference chain. Because only when ref_lock is detected can the reference chain get updated, this value should be the right value to be updated next to the slave line when reference is locked.
1 DLL_STS_REF_LOCK	Reference DLL lock status. This signifies that the DLL has detected and locked to a half-phase ref_clock shift, allowing the slave delay-line to perform programmed clock delays
0 DLL_STS_SLV_LOCK	Slave delay-line lock status. This signifies that a valid calibration has been set to the slave-delay line and that the slave-delay line is implementing the programmed delay value

30.7.21 Vendor Specific Register (ESDHCV3x_VENDOR)

This register contains the vendor specific control/status register.

Addresses: ESDHCV3-3_VENDOR is 5002_0000h base + C0h offset = 5002_00C0h



ESDHCV3x_VENDOR field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved
27–24 DBG_SEL[3:0]	Debug Select Select the internal sub-block to show its internal state value.
23–16 INT_ST_VAL[7:0]	Internal State Value Internal state value, reflecting the corresponding state value selected by Debug Select field. This field is read-only and write to this field does not have effect.

Table continues on the next page...

ESDHCV3x_VENDOR field descriptions (continued)

Field	Description
15–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1 EXACT_BLK_NUM	Exact block number block read enable for SDIO CMD53 This bit should be set before S/W issues CMD53 multi-block read with exact block number. This bit should not be set if the CMD53 multi-block read is not exact block number. 0 none exact block read 1 exact block read for SDIO CMD53
0 EXT_DMA_EN	External DMA Request Enable Enable the request to external DMA. When the internal DMA (either Simple DMA or Advanced DMA) is not in use, and this bit is set, ESDHC will send out DMA request when the internal buffer is ready. This bit is particularly useful when transferring data by ARM platform polling mode, and it is not allowed to send out the external DMA request. By default, this bit is set. 0 In any scenario, ESDHC does not send out external DMA request 1 When internal DMA is not active, the external DMA request will be sent out

30.7.22 MMC Boot Register (ESDHCV3x_MMCBOOT)

This register contains the MMC Fast Boot control register.

Addresses: ESDHCV3-3_MMCBOOT is 5002_0000h base + C4h offset = 5002_00C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BOOT_BLK_CNT[15:0]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								AUTO_SABG_EN	BOOT_EN	MMC_BOOT_MODE	BOOT_ACK	DTOCV_ACK[3:0]			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESDHCV3x_MMCBOOT field descriptions

Field	Description
31–16 BOOT_BLK_CNT[15:0]	The value defines the Stop At Block Gap value of automatic mode. When received card block cnt is equal to BOOT_BLK_CNT and AUTO_SABG_EN is 1, then Stop At Block Gap.
15–8 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

ESDHCV3x_MMCBOOT field descriptions (continued)

Field	Description
7 AUTO_SABG_EN	When boot, enable auto stop at block gap function. This function will be triggered, and host will stop at block gap when received card block cnt is equal to BOOT_BLK_CNT.
6 BOOT_EN	Boot mode enable 0: fast boot disable 1: fast boot enable
5 MMC_BOOT_MODE	Boot mode select 0: normal boot 1: alternative boot
4 BOOT_ACK	Boot ack mode select 0: no ack 1: ack
3–0 DTOCV_ACK[3:0]	Boot ACK time out counter value. 0000 SDCLK x 2^8 0001 SDCLK x 2^9 0010 SDCLK x 2^10 0011 SDCLK x 2^11 0100 SDCLK x 2^12 0101 SDCLK x 2^13 0110 SDCLK x 2^14 0111 SDCLK x 2^15 1110 SDCLK x 2^22 1111 Reserved 0000 SDCLK x 2^7 (when in DDR mode) 0001 SDCLK x 2^8 (when in DDR mode) 0010 SDCLK x 2^9 (when in DDR mode) 0011 SDCLK x 2^10 (when in DDR mode) 0100 SDCLK x 2^11 (when in DDR mode) 0101 SDCLK x 2^12 (when in DDR mode) 0110 SDCLK x 2^13 (when in DDR mode) 0111 SDCLK x 2^14 (when in DDR mode) 1110 SDCLK x 2^21 (when in DDR mode) 1111 Reserved (when in DDR mode)

30.7.23 Host Controller Version (ESDHCV3x_HOSTVER)

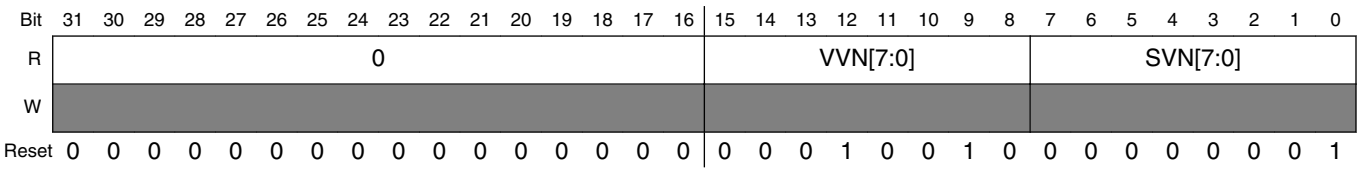
This register contains the vendor Host Controller version information. All bits are read only and will read the same as the power-reset value.

NOTE

The VVN field is 0x12 for both ESDHCV2 and ESDHCV3. Todetermine whether the controller is ESDHCV2 or ESDHCV3 do the following:

Set DLLCTRL [10]. This bit only exists in ESDHCV3. If it is set to 1 successfully, then the controller is ESDHCV3, otherwise it is ESDHCV2.

Addresses: ESDHCV3-3_HOSTVER is 5002_0000h base + FCh offset = 5002_00FCh



ESDHCV3x_HOSTVER field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–8 VVN[7:0]	Vendor Version Number: These status bits are reserved for the vendor version number. The Host Driver will not use this status. Settings not shown are reserved. 0x00 Freescale ESDHC Version 1.0 0x10 Freescale ESDHC Version 2.0 0x11 Freescale ESDHC Version 2.1 0x12 Freescale ESDHC Version 2.2
7–0 SVN[7:0]	Specification Version Number: These status bits indicate the Host Controller Specification Version. Settings not shown are reserved. — 0x01 SD Host Specification Version 2.0, supports Test Event Register and ADMA.

Chapter 31
External Memory Controller (EXTMC)

31.1 General Introduction

The EXTMC is a memory controller of all memory devices in the system, both internal and external. It provides a capability to the system to control the external memory device, by performing several type of access like read, write, program and erase as well as internal memories of the system. All access from the system to a memory device is arbitrated inside EXTMC by the Multi Master Multi Memory Interface (M4IF) sub-block to the right memory controller to perform the access.

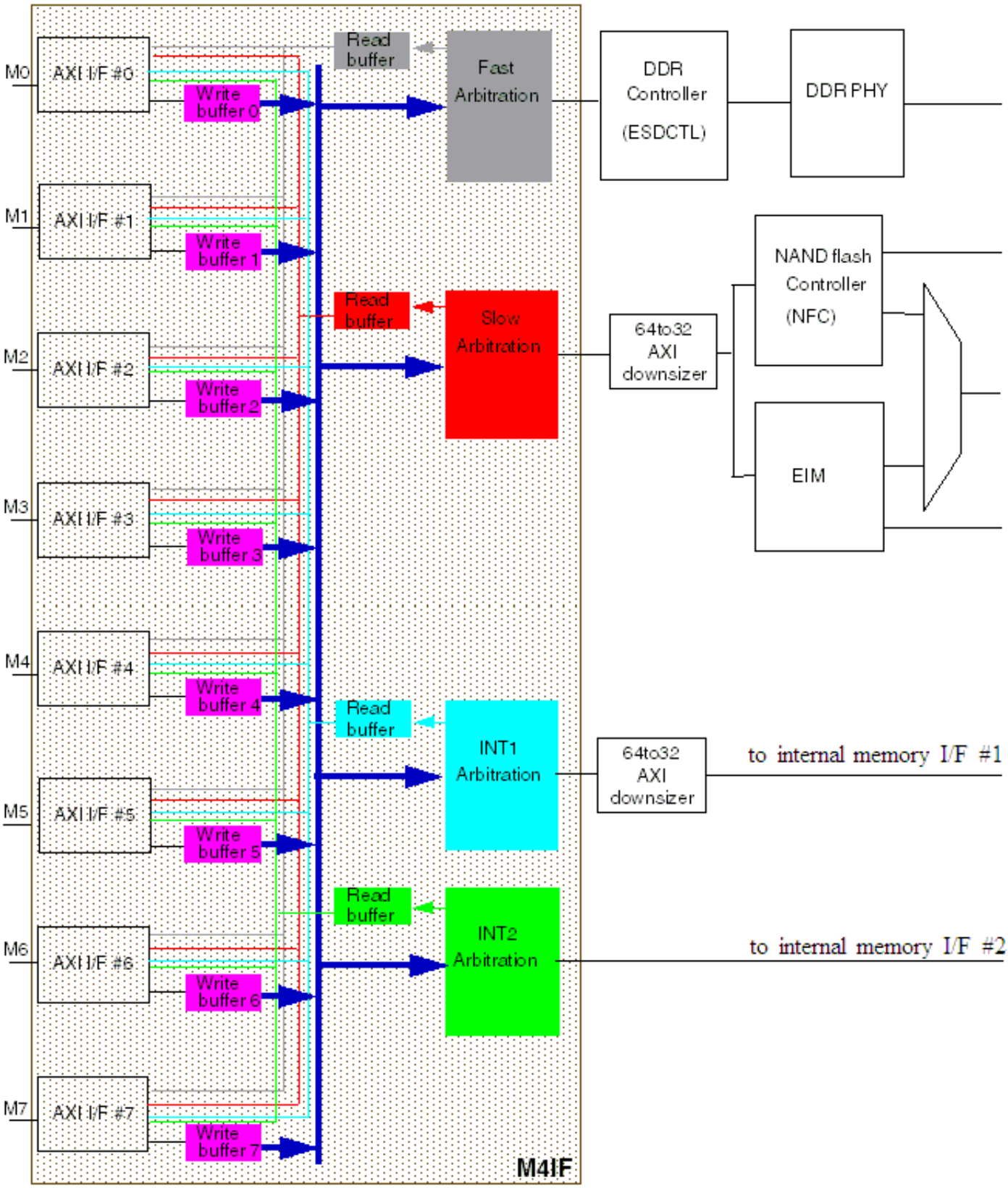


Figure 31-1. EXTMC high level block diagram

31.1.1 Overview

The EXTMC is an External Memory Controller with arbitration logic between multi AXI masters to multi memory controllers, divided into four major channels: fast memories channel (DDR2/DDR3), slow memories channel (NOR-FLASH/PSRAM/NAND-FLASH etc.) and two Internal Memory channels. The internal channels can be used for any AXI compliant slave. They are mostly used for accessing the internal memories of the chip (RAM and ROM).

In order to increase performance, the EXTMC separates the buffering and arbitration between accesses to fast channel slow channel and Internal Memory channels. Therefore, parallel accesses can occur. By separating these four channels, slow accesses do not interfere with fast accesses. EXTMC is being designed such that accesses by a slow master with a narrow bus does not block the access by a fast master with a wide bus.

The EXTMC contains the following sub-blocks:

31.1.1.1 AXI Port Gasket

This is the EXTMC interface to AXI masters and includes 8 AXI port gaskets. This domain sample the data into the buffers, samples the access (address and controls) into FIFO and sort the access to the right channel - fast, slow or Internal Memory channels (Intr1 or Intr2).

31.1.1.2 Dedicated Write Buffers Module

Each AXI port gasket (Master) has a Dedicated Write buffer Module. Write data from masters is sampled in the Write Buffers Module with master clock domain and being transferred to the Memory controller upon request in memory controller clock domain. The Write Buffers' size varies from master to master. Please refer to the M4IF spec chapter.

31.1.1.3 Read Shared Buffers Module

The Read Shared buffer Module serves all 8 AXI ports, one read buffer per each arbitration, slow, fast and internal memory channels, a total of four read buffers. Read data from memory controllers is sampled in the Read buffers with memory controller clock domain, and being transferred to the AXI port gasket upon request. The data from

the memory controller is written to the read buffer in one clock domain, while the gasket provides the data back to the master with master clock domain. See the M4IF chapter for more information.

31.1.1.4 Fast Arbitration Module

The Fast Arbitration Module arbitrates accesses to the Enhanced SDRAM Controller (ESDCTL), using most comprehensive information to optimize accesses to the ESDCTL, such as Page hit/miss read/write access etc. If needed this additional information can be disabled and the arbitration will function in fixed priority mode, based on the user defined priority, more details on the dynamic priority arbitration can be found in M4IF Chapter.

31.1.1.5 Slow and Intr Arbitration Modules

The Slow Arbitration Module uses a reduced model of the fast arbitration which arbitrates accesses to Slow Memories such as External Interface Module (EIM) and NAND Flash Controller (NFC). The slow arbitration block is based on the fast arbitration block design. The algorithm of the slow channel is much simpler than the fast channel, and its latency is smaller in comparison. For a detailed description please refer to M4IF spec.

31.1.1.6 Memory Controller Modules

These blocks includes all memory controllers, ESDCTL, and NFC. The memory controllers receives the access request from the arbitration block and transfer the access to the relevant memory. It is important to mention that although EXTMC arbitrates the internal memory access in the system, it does not contain any internal memory controller like RAM/ROM controller.

31.1.1.7 IPS Interface unit

IPS Interface unit is the EXTMC interface to IP Bus. EXTMC registers would all be IP Bus based, this unit would route block access to the relevant sub-block above based on addr and ips_module_en signal. In EXTMC the following sub-blocks include block registers: M4IF, ESDCTL, EIM, NFC.

31.1.1.8 Debug Unit

Debug unit privides the ability to monitor several important internal EXTMC signals, improving the debug capability of the system.

In addition, the debug unit gives the ability to profile a master connected to EXTMC and get the bus performance of the arbitration.

Please refer to M4IF spec for a detailed description of its registers and functionality.

31.1.2 Features

The EXTMC includes the following features:

- Multi Master Multi Memory Interface (M4IF)
  - Supports multiple accesses from 8 masters through different input ports interfaces. Each port can support either of the following two data width options:
    - x32 AXI port.
    - x64 AXI port.
  - Support different clock domain for each AXI port master.
  - Configureable memory watermark protection per CS.
  - Enhanced arbitration scheme for fast channel, consider page hit/miss, last access details (read/write) and fixed priority configuration.
- Enhanced SDRAM Controller (ESDRAMC)
  - Up to 2 chip selects support up to 1GByte each.
  - Support x16/x32 DDR2 memories up to 400MHz (800MHz data rate).
  - Support x16/x32 DDR3 memories up to 400MHz (800MHz data rate).
  - Support x16/x32 LPDDR2 1ch memories up to 400MHz (800MHz data rate).
  - Support latency hiding logic.
- NandFlash Controller - (NFC)
  - x8/x16 NAND interface.
  - 4.5K RAM Internal Buffer.
  - MLC and SLC memory support.
  - Configureable operation mode - symmetric and asymmetric.
  - Configureable page mode, 1/2K,2K or 4K.
  - Pages per blocks supported are 32, 64, 128, 256.
  - Up to 6 address cycles.
  - ECC support up to 16bit.
  - Support for up to 8 mutually exclusive, yet interleaved nand devices.
  - Automatic Common Nand Operations.
- External Interface Memory Controller - (EIM)
  - Up to 6 chip selects.

- Support x32/x16 PSRAM (up to 133 MHz).
- support x32/x16 muxed mode PSRAM / NOR (up to 133 MHz).
- Support x32/x16 NOR (up to 133 MHz).

- Support Dynamic Voltage and Frequency Scaling Core (DVFSC).
- Support watermark configuration.
- Support of automatic shut-down (power saving features).
- Enhanced debug capabilities (trace mode).

31.1.3 Modes of Operation

The following sub sections describes several operation modes of EXTMC. In order to have a better knowledge on operation modes of each sub-block in EXTMC, please refer to each sub-block Chapter for "Modes of Operation" section.

31.1.3.1 Low Power Modes

SoC system control provides an LPM request signal to EXTMC in order to let EXTMC enter into low power mode. At the end of low power mode sequence, EXTMC sends back an acknowledge signal to inform the system it entered into low power mode.

In this mode, all internal clocks in EXTMC and its sub-blocks would be disabled by the system clock controller. Low power mode definition is described in more details in each sub-block separately.

31.1.3.2 Debug Mode

Several internal signals would be routed out by EXTMC debug unit allowing tracking the internal logic mainly for the arbitration mechanism and memory served accesses.

The debug unit is described in detail in M4IF Chapter.

31.1.3.3 Dynamic Voltage and Frequency Scaling (DVFS) support

EXTMC provides real-time support of voltage and/or frequency change to its memory controllers. There is an handshake mechanism between EXTMC and the system so that EXTMC is informed of any frequency change that is going to be made in the system. When this handshake signal is sent to EXTMC suspends all access to the external/internal

memories controllers and finishes any in-process access in the memory controller. Accesses coming from the masters will be accepted but not forwarded to the memory controllers.

After that the frequency change logic inside EXTMC would be active so that at the end of the routine an acknowledge would be sent to inform a change clock frequency is allowed. At the moment the frequency was changed, the system should lower the handshake signal (request) to EXTMC which means the frequency change is done. EXTMC would force a new measure in the delay line and move back to idle after the measure unit is updated with the new value (ready signals of the memory controllers will return to HIGH).

EXTMC does not allow accesses to the memory during a measure.

More details of DVFS support, Delay Line and measurement unit can be found in ESDCTL Chapter.

31.1.3.4 Power Saving Mode

EXTMC supports few power saving modes other than LPMD. M4IF provides information on whether there are pending requests to a certain arbitration, when was the last request in this arbitration etc. Based on that info, both the specific arbitration and its corresponding memory controller, would be able to enter power saving modes like Self refresh and Auto Self refresh in ESDCTL memory controller. In this mode, each arbitration path is considered as a separate domain so that in a certain time internal memory path can be in power saving mode while DDR memory path will continue to work. It is important to mention that each power saving mode would leave an active logic to be able to get out of this mode whenever a new access arrives. In this case it would take few more cycles to serve the request since getting out of power saving mode routine is taking place first.

A quiet similar mechanism exists in each of M4IF gaskets. M4IF can automatically shut off the clock of a specific gasket based on internal timer that counts a certain number of idle cycle.

Please refer to M4IF Chapter for more details on Power Saving mode.

31.2 Sharing of I/O Pins

EXTMC does not contain any I/O pins sharing or muxing between different arbitration paths. The only I/O sharing done inside EXTMC is EIM and NFC I/O pin sharing.

31.2.1 EIM and NFC IO pin sharing

EXTMC controls the I/O pin sharing between NFC and EIM.

The EIM in i.MX53 supports the following muxing modes:

Table 31-1. EIM multiplexing

Setup	Non Multiplexed Address/Data Mode						Multiplexed Address/ Data mode	
	8 Bit			16 Bit		32 Bit	16 Bit	32 Bit
	MUM = 0, DSZ = 100	MUM = 0, DSZ = 101	MUM = 0, DSZ = 111	MUM = 0, DSZ = 001	MUM = 0, DSZ = 010	MUM = 0, DSZ = 011	MUM = 1, DSZ = 001	MUM = 1, DSZ = 011
A[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]	EIM_DA[15:0]
A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[25:16]	EIM_A[24:16][1]	EIM_A[25:16]	NANDF_D[8:0]
D[7:0], EIM_EB0	NANDF_D[7:0]	-	-	NANDF_D[7:0][2]	-	NANDF_D[7:0]	EIM_DA[7:0]	EIM_DA[7:0]
D[15:8], EIM_EB1	-	NANDF_D[15:8]	-	NANDF_D[15:8][3]	-	NANDF_D[15:8]	EIM_DA[15:8]	EIM_DA[15:8]
D[23:16], EIM_EB2	-	-	-	-	EIM_D[23:16]	EIM_D[23:16]	-	NANDF_D[7:0]
D[31:24], EIM_EB3	-	-	EIM_D[31:24]	-	EIM_D[31:24]	EIM_D[31:24]	-	NANDF_D[15:8]

1. For 32-bit mode, the address range is A[24:0], due to address space allocation in memory map.
2. NANDF_D[7:0] multiplexed on ALT3 mode of PATA_DATA[7:0]
3. NANDF_D[15:8] multiplexed on ALT3 mode of PATA_DATA[15:8]

Muxing between data pads of EIM and NFC exist on the output path only, based on which memory controller own the bus for the specific transaction.

The sharing control logic is based on the fix priority arbitrating between EIM and NFC. EIM has the priority on the external bus whenever there is a conflict between EIM and NFC. However, when NFC is using the bus and EIM access is issued in parallel, NFC will free up the bus after number of clock cycles that depends on the specific command/ access type which is currently served in NFC. No specific time period is guaranteed for the NFC to free up the bus for EIM pending access.

31.3 Memory Map and Register Definition

Table 31-2. External/Internal EXTMC Memory Map

Address	Ext memory CS	Access	Reset Value
0x{int_mem_base[3:0],000_0000} - 0x{int_mem_end[3:0],FFF_FFFF}[1]	Internal Memory region (2GB)	R/W[2]	Defined by SoC via
0x{esdctl_base[3:0],000_0000} - 0x{esdctl_base[3:0],FFF_FFFF}	CSD0 SDRAM/DDR memory region (256 MB)	R/W	Defined by SoC via
0x{(esdctl_base[3:0]+1),000_0000} - 0x{(esdctl_base[3:0]+1),FFF_FFFF}	CSD1 SDRAM/DDR memory region (256 MB)	R/W	Defined by SoC via
0x{weim_nfc_base[3:0][3] ,000_0000} - 0x{(weim_nfc_base[3:0]+1),FFE_FFFF}	EIM CS memory region (configure-able, up to 512MB total) EXTMC	R/W	Defined by SoC via
0x{(weim_nfc_base[3:0]+1),FFF_0000} - 0x{(weim_nfc_base[3:0]+1),FFF_FFFF}	NFC memory region (4K, NandFlash) - support up to 4 CS	R/W	Defined by SoC via

1. Internal memory space is configured by SoC vias at system level. Please refer to int_mem_base[3:0] and int_mem_end[3:0] signal description for more details and for SoC configuration Chapter for specific chip configuration.
2. Note that Internal memory space may contain some read-only memory region.
3. EIM and NFC base address must not be equal to 0xF, since wrap around addressing is not supported in EIM.

31.3.1 IP Bus Memory Map

EXTMC registers are mapped to IP bus memory area and contains 16Kbyte space for all EXTMC registers. For more details on the block registers, please refer to each EXTMC sub-block's spec.

Table 31-3. EXTMC Block memory map

Start ADDR	End ADDR	Block name
0x63FD_8000	0x63FD_8FFF	M4IF
0x63FD_9000	0x63FD_9FFF	ESDCTL
0x63FD_A000	0x63FD_AFFF	EIM
0x63FD_B000	0x63FD_BEFF	NFC
0x63FD_BF00	0x63FD_BFFF	EXTMC

31.4 Functional Description

EXTMC is the external memory arbitration and controller, and the internal memory arbitration of the system. Each master access to the internal or external memory region would be routed out through EXTMC. The EXTMC is based on AXI pipeline separate read and write buses and therefore it can reorder the access of the masters inside the arbitration to provide the highest throughput on the external memory bus.

EXTMC has separate arbitration control blocks for fast and slow external/internal interfaces. An access to the slow interface can be served in parallel to an access of the fast interface without any interference between them.

In addition, EXTMC interface supports asynchronous master interface. Therefore, a slow master would be able to read/write data in parallel to other fast master accesses. The arbitration logic would optimize the access between the masters to the memory controller to provide maximum throughput on the external memory bus and in the same time a minimum latency for pending access in the arbitration.

Special care is taken in order to reduce power in idle and low power mode. Power saving modes exist to provide automatic capability of reducing the power consumption without any special user configuration.

DVFS is supported for both frequency and voltage changes based on handshake mechanism between EXTMC and the system.

A detailed functional description on the sub-block functionality can be found in each block chapter.

31.4.1 Clocks

EXTMC contains the following clock domains:

- ESDCTL main clock (up to 400MHz)
- Slow arbitration clock. (up to 133MHz)
- Internal memory arbitration 1 & 2 clocks. (up to 200MHz)
- 8 x masters clocks, can be asynchronous to EXTMC arbitration clocks (up to 200MHz).
- NFC main clock - should be integer divided from slow arbitration clock.
- SDCLK - DRAM clock.
- BCLK - NOR Flash/PSRAM clock EIM in synchronous mode.
- EXTMC IP Bus clock.

- This clock should be at least 2X slower than the fast, slow, and internal arbitration clocks.
- This clock should be at least 2X slower than any master's clock

The EXTMC arbitrator (M4IF) assumes all of its 4 main clocks: ESDCTL main clock (aclk_fast), Slow arbitration clock (aclk_slow) and Internal memory arbitrations clock (aclk_int1,aclk_int2), are on by default (after reset). If the system wishes to turn one of the clocks off, it must do so by going through the specific lpmd sequence. For example, if the system wishes to turn off "aclk_fast", it will have to wake up with this clock on, send an "lpmd_fast" request, wait for "lpack_fast" acknowledge, and only then turn off "aclk_fast". "lpmd_fast" will have to be asserted for the whole duration of "aclk_fast" being off.

31.4.2 Reset

EXTMC supports cold reset and warm reset.

31.4.2.1 Cold Reset

Cold Reset is the general ARESET signal that enters EXTMC and generates an HW reset to all sub-blocks of EXTMC. All logic is being reset when cold reset is asserted besides the POR reset logic.

31.4.2.2 Warm Reset

Warm reset is supported for ESDCTL only. In this mode the data outside in the external device will be kept during warm reset and will be ready to be used at the moment the device is getting out of warm reset. See the ESDCTL Chapter for more details.

31.4.3 Interrupts

The following interrupt are supported in EXTMC.

- Watermark violation ARM platform interrupt.
- Watermark violation BP interrupt.
- NFC interrupt.
- EIM int for OneNand and MDOC support.
- Configure-able AP_ORed interrupt of all EXTMC interrupts.

- Configure-able BP_ORed interrupt of all EXTMC interrupts.
- LEN interrupt from M4IF

31.4.4 Endianness

EXTMC supports Little endian only. Detailed description on EXTMC endianess may be found on EXTMC Endianness .

31.4.5 IP Bus Interface

All EXTMC registers are IP Bus memory mapped. The registers will be configureable read/write via IP Bus interface only.

The IP Bus interface unit inside EXTMC combines all IP Bus interfaces from EXTMC sub-blocks as appears in Table 31-3. The IP Bus IF unit routes the access to the right sub-block based on ips_module_en signal and address decoding.

EXTMC registers use an accessibility policy as described below.

EXTMC decodes ips_master_id of each block access. The privilege master has the capability to lock or unlock any register from being configured by any other master but the BP Watermark registers. Therefore, the system can ensure that when there is a lock on the registers, no register will be changed by a master other than the privilege master. In a similar way, the BP master is the only master that can lock and configure BP Watermark registers so that even the privilege master can not changed its values. In a case of a non-privilege master that tries to access a lock register, EXTMC does not respond in error; no error signal is sent back on the bus.

See the Register Definition section in this document for more details on the lock mechanism.

All EXTMC registers are 32-bit width. Only block accesses of 32-bit data width are supported.

NOTE

There is no special mechanism inside EXTMC to prevent block register configuration while other operations on AXI are taking place. Therefore, system should control block configuration and prevent a conflict in that case.

When the access is done to a valid address location but no register is mapped to that location, an ips_xfr_err will be generated as long as the XFR_ERR_EN bit in the IPLCK register is high. If XFR_ERR_EN bit is low the ips_xfr_err signal is disabled and remains low.

31.5 Application Information

31.5.1 Buffers Size

See the M4IF chapter for exact buffer sizes for each master and slave

31.5.2 Master to Slave Paths - Restrictions

EXTMC can send any access from any master to any slave. However there are few restrictions.

- Some paths must be synchronized
- Some paths are disconnected

Please refer to the M4IF chapter for exact details.

31.5.3 EXTMC Endianness

31.5.3.1 Introduction to Endianness

EXTMC supports Little Endian mode only. The chapter describes the data arrangement on the data bus in all stages inside EXTMC starting at the AXI interface at M4IF toward the external memory bus at the various memory controllers.

31.5.3.2 AXI Endianness Support

EXTMC support AXI masters with bus width of x64 and x32. Therefore, the sections below are related to these bus width only.

31.5.3.3 AXI master x64

This section shows the data location on AXI bus as it entered into EXTMC on M4IF AXI gaskets. In the next section we will show the data byte locations inside EXTMC toward the external bus and the storage at external memory devices.

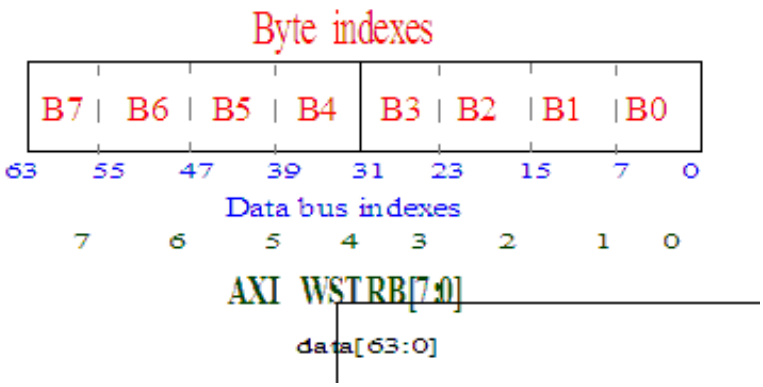


Figure 31-2. Little Endian Bytes Placement on AXI x64

It can be seen from this illustration that the data byte with index 0 uses byte lane 0 (data[7:0]) and data byte with index 7 uses byte lane 7 (data[63:56]).

31.5.3.4 AXI master x32

This section shows the data location on AXI bus as it entered into EXTMC on M4IF AXI gaskets. The following section demonstrates the data byte locations inside EXTMC toward the external bus, and the storage at external memory devices.

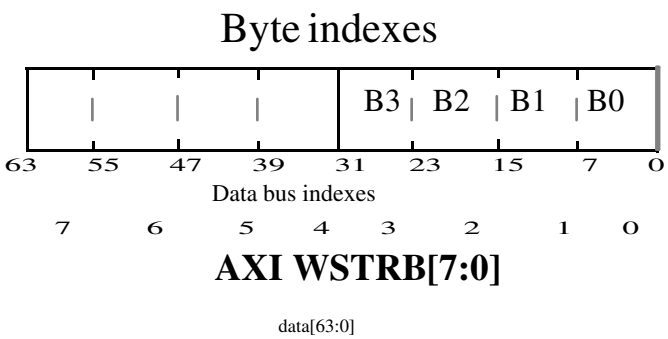


Figure 31-3. Little Endian Bytes Placement on AXI x32

As shown in this illustration, the data byte with index 0 uses byte lane 0 (data[7:0]) and data byte with index 3 uses byte lane 3 (data[31:24]).

31.5.3.5 EXTMC Endianness support

The following sections will describe the Endianness support in each of EXTMC sub-blocks. In general, EXTMC support LE only.

31.5.3.6 M4IF Endianess support

The following figures shows the way M4IF handle little Endian mode. In general M4IF has two different operation modes regardless to Endian mode, it would change its operation logic based on fast, slow, and internal. Since fast and internal arbitration path are x64 while slow arbitration is x32, the way M4IF should provide the data to the memory controller is different.

31.5.3.6.1 M4IF behavior for x64 arbitration

In this case M4IF would provide the data on the bus the way it was provided to it from the masters in the system without any change.

31.5.3.6.2 M4IF behavior for x32 arbitration

In this case M4IF would provide the data on the x32 bus according to the byte lanes. If the access is between B0 to B3, it provides lower 32 word (byte0-3), while when the access is between B4 to B7 it provides higher 32 word (byte4-7).

The following figures shows some examples in LE mode for x32 arbitration:

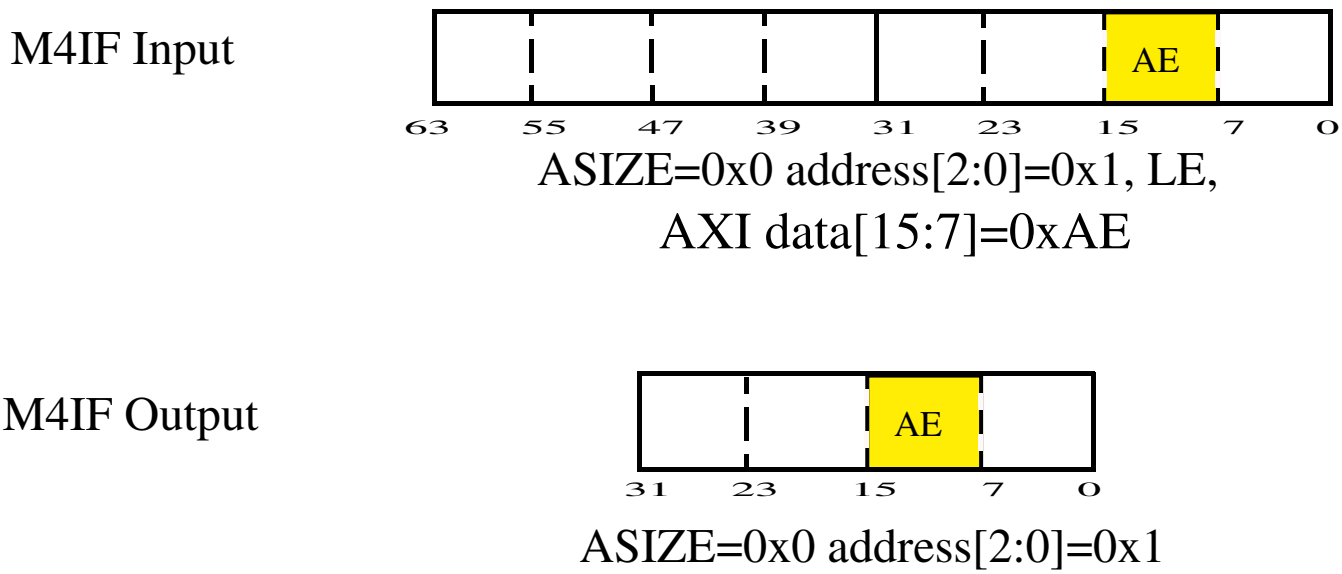


Figure 31-4. Byte access for AXI x64 master and x32 Arbitration slave

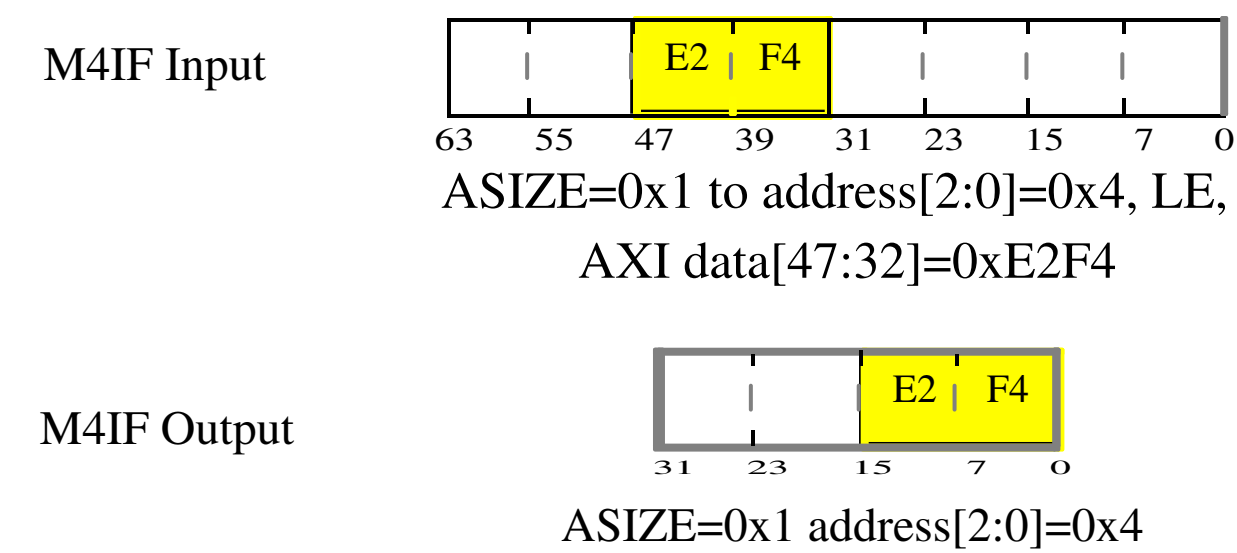


Figure 31-5. Half-Word access for AXI x64 master and x32 Arbitration slave

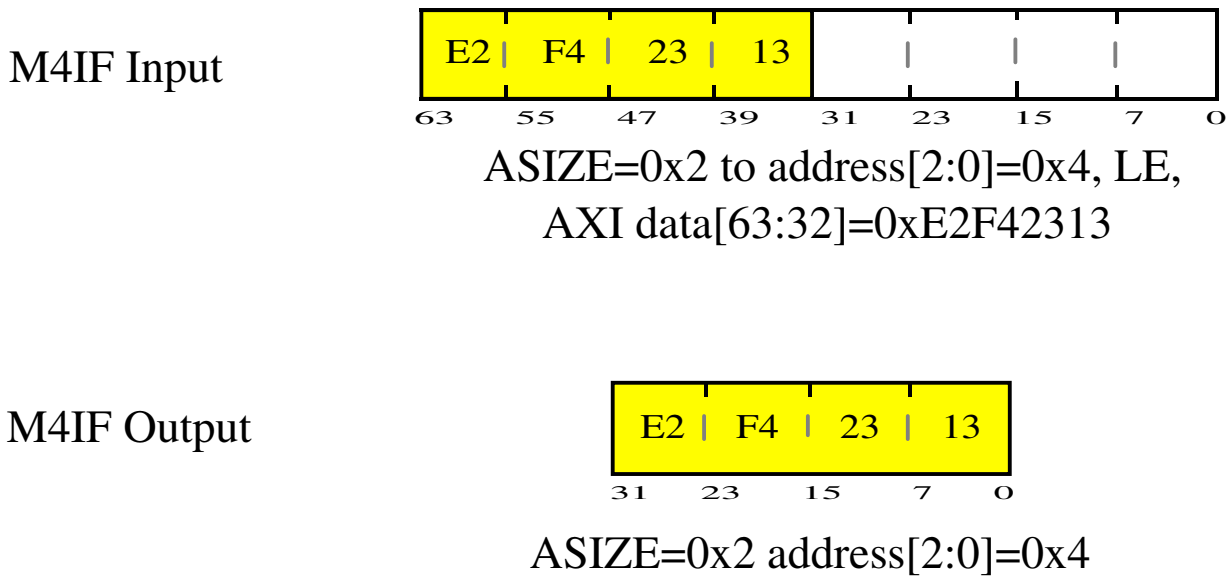


Figure 31-6. Word access for AXI x64 master and x32 Arbitration slave

31.5.4 Data Storage Arrangement

This Section describes the way various memory controllers in EXTMC arrange the data in the external memory device. The section describes only x32 and x16 external memory devices, because x8 devices are not supported in EXTMC.

The following figures describe the data storage location in the external memory devices.

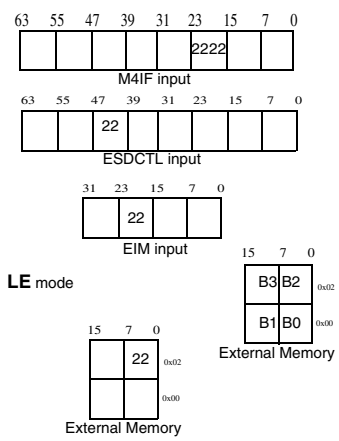


Figure 31-7. Byte access, ASIZE=0x0, ext. memory x16

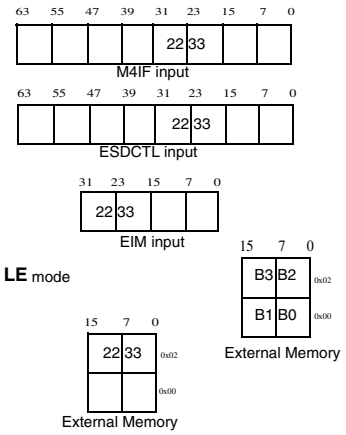


Figure 31-8. Half-word access, ASIZE=0x1, ext. memory x16

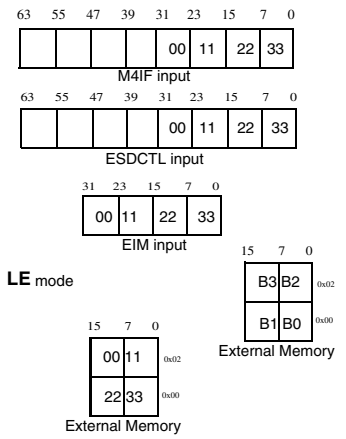


Figure 31-9. Word access, ASIZE=0x2, ext. memory x16

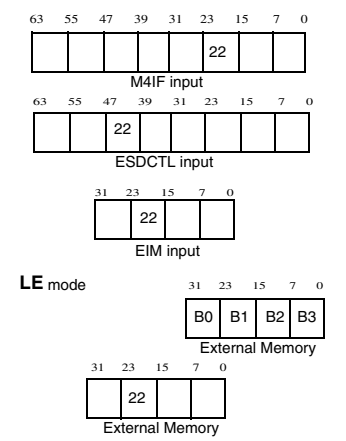


Figure 31-10. Byte access, ASIZE=0x0, ext. memory x32

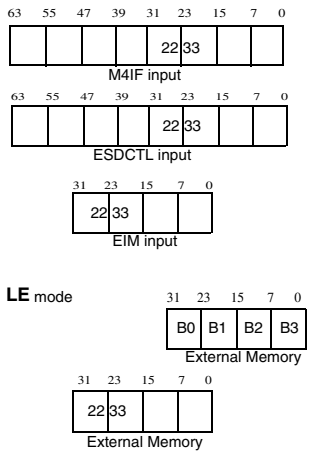


Figure 31-11. Half-word access, ASIZE=0x1, ext. memory x32

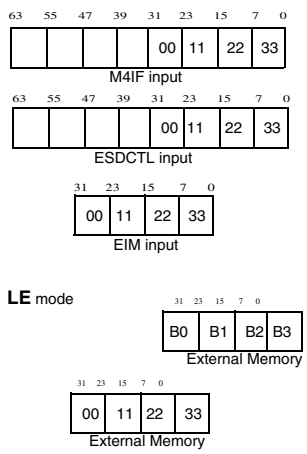


Figure 31-12. Word access, ASIZE=0x2, ext. memory x32

31.6 Programmable Registers

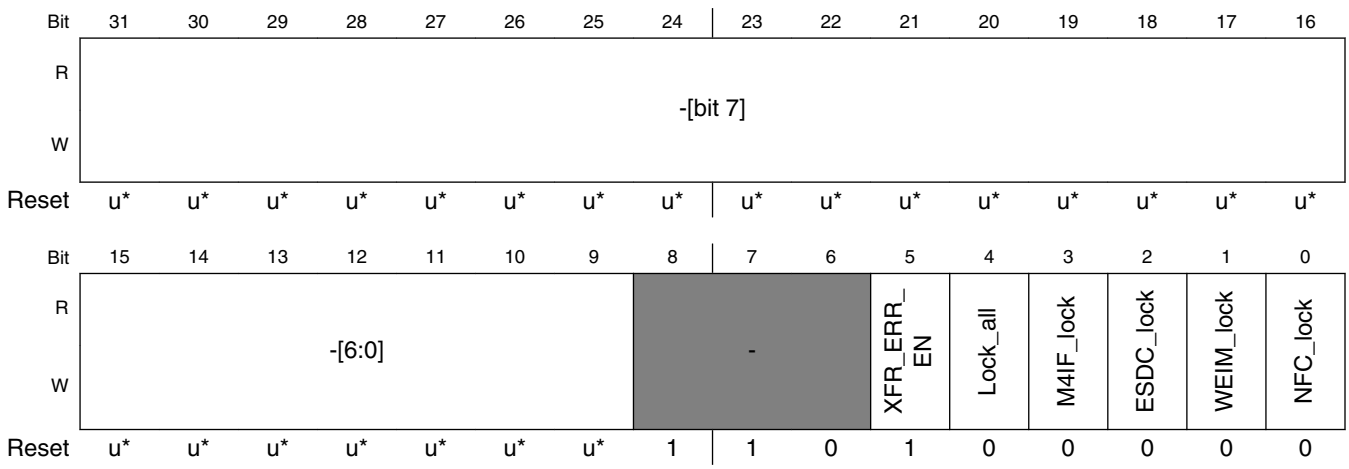
EXTMC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FD_BF00	IP Lock register (EXTMC_EXTMC_IPLCK)	32	R/W	See section	31.6.1/ 1556
63FD_BF04	Interrupt control and Status register (EXTMC_EXTMC_EICS)	32	R/W	See section	31.6.2/ 1557

31.6.1 IP Lock register (EXTMC_EXTMC_IPLCK)

EXTMC_IPLCK register contains the locking configuration bits of each block registers region in EXTMC sub-blocks. This register is accessible by the ARM platform privilege master only as defined at SoC level. Other non privilege masters can only read the content of this register.

Address: EXTMC_EXTMC_IPLCK is 63FD_BF00h base + 0h offset = 63FD_BF00h



* Notes:
- u = Unaffected by reset.

EXTMC_EXTMC_IPLCK field descriptions

Field	Description
31–9 -	N/A.
8–6 -	Reserved.
5 XFR_ERR_EN	EXTMC's ips_xfr_err enable bit. 0 ips_xfr_err is disabled (always equal to 0). 1 ips_xfr_err is enabled.
4 Lock_all	Lock all EXTMC registers from being configured by non privilege masters. 0 register is unlocked. 1 register is locked.
3 M4IF_lock	Lock M4IF registers from being configured by non privilege masters (except BP privilege registers). 0 register is unlocked. 1 register is locked.
2 ESDC_lock	Lock ESDCTL registers from being configured by non privilege masters.

Table continues on the next page...

EXTMC_EXTMC_IPLCK field descriptions (continued)

Field	Description
	0 register is unlocked. 1 register is locked.
1 WEIM_lock	Lock EIM registers from being configured by non privilege masters. 0 register is unlocked. 1 register is locked.
0 NFC_lock	Lock NFC register from being configured by non privilege masters. 0 register is unlocked. 1 register is locked.

31.6.2 Interrupt control and Status register (EXTMC_EXTMC_EICS)

EXTMC_EICS register configured the ORed interrupt scheme. It contains a masking bit for the ORed interrupt as well. EXTMC_EICS is a general type register, when Lock_all bit in IPLCK register is high, only ARM platform master can to write to EXTMC_EICS register, when Lock_all bit is low, all other masters as well as ARM platform can write to it.

Address: EXTMC_EXTMC_EICS is 63FD_BF00h base + 4h offset = 63FD_BF04h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EAOIE	EBOIE	-													
W																
Reset	1	1	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AUTO_PROG_DONE_BP_int	AUTO_PROG_DONE_BP_int	BLEN_int	ALEN_int	BM4IF_AP_int	BM4IF_BP_int	BWEIM_int	BNFC_int	MAIS	MBIS	WIS	NIS	AM4IF_AP_int	AM4IF_BP_int	AWEIM_int	ANFC_int
W																
Reset	1	1	1	1	0	1	1	1	0	0	0	0	1	0	1	1

* Notes:
- u = Unaffected by reset.

EXTMC_EXTMC_EICS field descriptions

Field	Description
31 EAOIE	EXTMC AP_ORed Interrupt Enable. This bit enables the AP_ORed interrupt generation of EXTMC. For example, if EAOIE bit is set and one of EXTMC sub-block interrupt is generated, AP_ORed interrupt based on its configuration is generated as well. 0 EXTMCv2.1 AP_ORed interrupt is disabled.

Table continues on the next page...

EXTMC_EXTMC_EICS field descriptions (continued)

Field	Description
	1 EXTMCv2.1 AP_ORed interrupt is enabled. (default)
30 EBOIE	EXTMC BP_ORed Interrupt Enable. This bit enables the BP_ORed interrupt generation of EXTMC. For example, if EBOIE bit is set and one of EXTMC sub-block interrupt is generated, BP_ORed interrupt based on its configuration is generated as well. 0 EXTMC BP_ORed interrupt is disabled. 1 EXTMC BP_ORed interrupt is enabled. (default)
29–16 -	Reserved.
15 AUTO_PROG_ DONE_BP_int	BP AUTO_PROG_DONE_int - this bit defines whether AUTO_PROG_DONE BP interrupt is included in the EXTMC BP_ORed interrupt or not. 0 interrupt is not included in BP_ORed.(default) 1 interrupt is included in BP_ORed.
14 AUTO_PROG_ DONE_BP_int	ARM platform AUTO_PROG_DONE_int - this bit defines whether AUTO_PROG_DONE ARM platform interrupt is included in the EXTMC AP_ORed interrupt or not. 0 interrupt is not included in AP_ORed. 1 interrupt is included in AP_ORed. (default)
13 BLEN_int	BP LEN_int - this bit defines whether LEN BP interrupt is included in the EXTMC BP_ORed interrupt or not. 0 interrupt is not included in BP_ORed.(default) 1 interrupt is included in BP_ORed.
12 ALEN_int	ARM platform LEN_int - this bit defines whether LEN ARM platform interrupt is included in the EXTMC AP_ORed interrupt or not. 0 interrupt is not included in AP_ORed. 1 interrupt is included in AP_ORed. (default)
11 BM4IF_AP_int	BP M4IF_AP_int - this bit defines whether M4IF ARM platform interrupt is included in the EXTMC BP_ORed interrupt or not. 0 interrupt is not included in BP_ORed.(default) 1 interrupt is included in BP_ORed.
10 BM4IF_BP_int	BP M4IF_BP_int - this bit defines whether M4IF BP interrupt is included in the EXTMC BP_ORed interrupt or not. 0 interrupt is not included in BP_ORed. 1 interrupt is included in BP_ORed. (default)
9 BWEIM_int	BP WEIM_int - this bit defines whether EIM interrupt is included in the EXTMC BP_ORed interrupt or not. 0 interrupt is not included in BP_ORed. 1 interrupt is included in BP_ORed. (default)
8 BNFC_int	BP NFC_int - this bit defines whether NFC interrupt is included in the EXTMC BP_ORed interrupt or not. 0 interrupt is not included in BP_ORed. 1 interrupt is included in BP_ORed. (default)

Table continues on the next page...

EXTMC_EXTMC_EICS field descriptions (continued)

Field	Description
7 MAIS	M4IF ARM platform interrupt status. This bit indicates if M4IF ARM platform interrupt had occurred. MAIS is updated in the first positive edge of ipg_clk_s in a block read access to EICS. 0 M4IF ARM platform interrupt did not occur. 1 M4IF ARM platform interrupt had occurred.
6 MBIS	M4IF BP interrupt status. This bit indicates if M4IF BP interrupt had occurred. MBIS is updated in the first positive edge of ipg_clk_s in a block read access to EICS. 0 M4IF BP interrupt did not occur. 1 M4IF BP interrupt had occurred.
5 WIS	EIM interrupt status. This bit indicates if EIM interrupt had occurred. WIS is updated in the first positive edge of ipg_clk_s in a block read access to EICS. 0 EIM interrupt did not occur. 1 EIM interrupt had occurred.
4 NIS	NFC interrupt status. This bit indicates if NFC interrupt had occurred. NIS is updated in the first positive edge of ipg_clk_s in a block read access to EICS. 0 NFC interrupt did not occur. 1 NFC interrupt had occurred.
3 AM4IF_AP_int	ARM platform M4IF_AP_int - this bit defines whether M4IF ARM platform interrupt is included in the EXTMC AP_ORed interrupt or not. 0 interrupt is not included in AP_ORed. 1 interrupt is included in AP_ORed. (default)
2 AM4IF_BP_int	ARM platform M4IF_BP_int - this bit defines whether M4IF BP interrupt is included in the EXTMC AP_ORed interrupt or not. 0 interrupt is not included in AP_ORed.(default) 1 interrupt is included in AP_ORed.
1 AWEIM_int	ARM platform WEIM_int - this bit defines whether EIM interrupt is included in the EXTMC AP_ORed interrupt or not. 0 interrupt is not included in AP_ORed. 1 interrupt is included in AP_ORed. (default)
0 ANFC_int	ARM platform NFC_int - this bit defines whether NFC interrupt is included in the EXTMC AP_ORed interrupt or not. 0 interrupt is not included in AP_ORed. 1 interrupt is included in AP_ORed. (default)

Chapter 32
Fast Ethernet Controller (FEC)

32.1 Overview

This chapter provides a feature-set overview, a functional block diagram, and transceiver connection information for the 10- and 100-Mbps media independent interfaces (MII), 10- and 100-Mbps reduced media independent interfaces (RMII), and the 7-wire serial interface. Detailed functional descriptions and application/initialization information are included.

The Fast Ethernet controller (FEC) is designed to support both 10- and 100-Mbps Ethernet/IEEE 802.3 networks. An external transceiver interface and transceiver function are required to complete the interface to the media. The FEC supports three different standard physical interfaces (MAC-PHY) for connection to an external Ethernet transceiver.

The FEC is implemented with a combination of hardware and microcode. The following figure is a block diagram of the FEC.

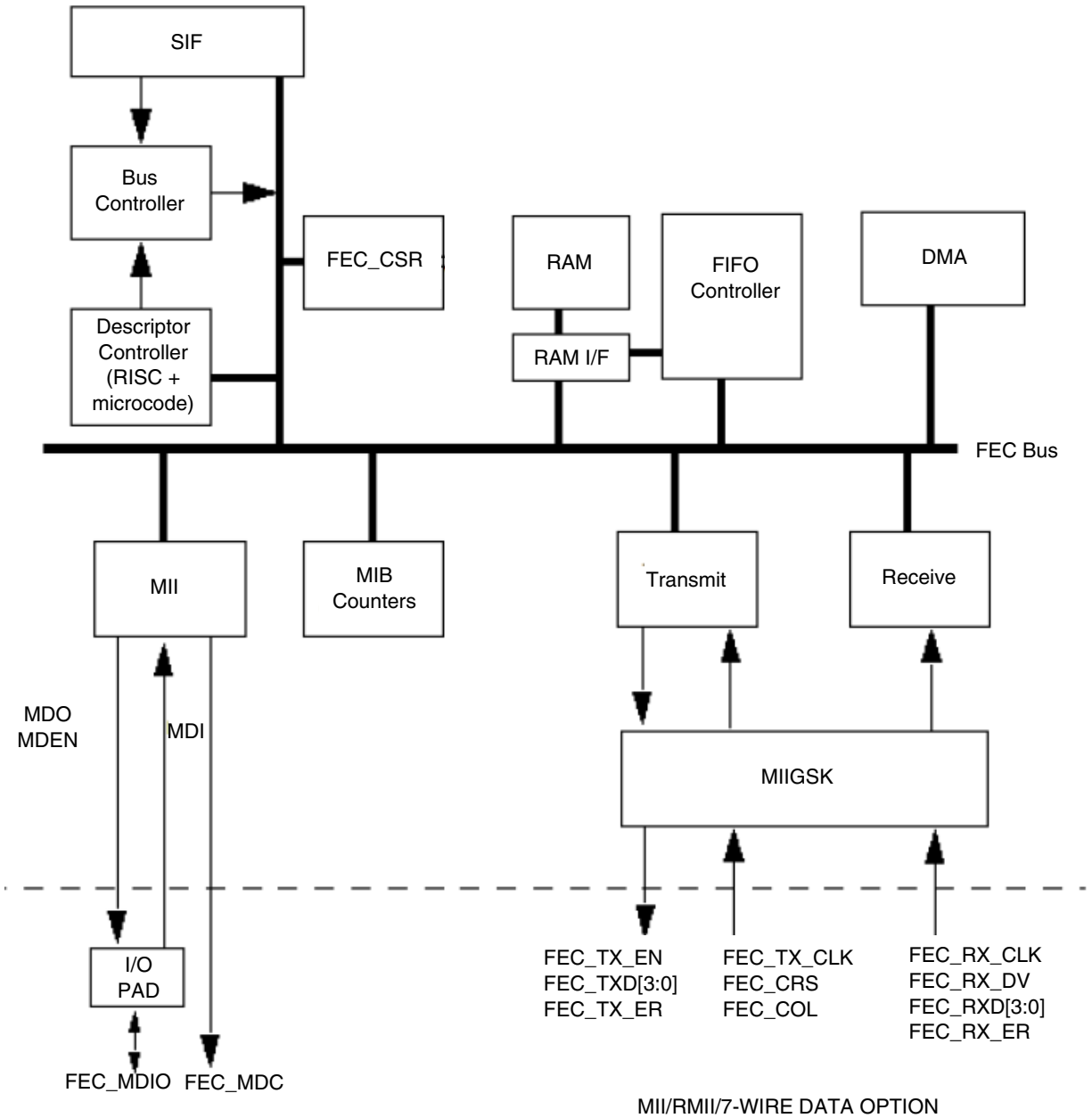


Figure 32-1. FEC Block Diagram

The following section provides information regarding the FEC submodules.

- The descriptor controller is a RISC-based controller that provides the following functions in the FEC:
  - Initialization (those internal registers not initialized by the user or hardware)
  - High-level control of the DMA channels (initiating DMA transfers)
  - Interpreting buffer descriptors
  - Address recognition for receive frames
  - Random number generation for transmit collision backoff timer

NOTE

This DMA engine is for the transfer of FEC data only and is not related to the system DMA controller.

- The RAM is the focal point of all data flow in the FEC and is divided into transmit and receive FIFOs. The FIFO boundaries are programmable using the FEC_FRSR register. User data flows to and from the DMA block, from and to the receive/transmit FIFOs. Transmit data flows from the transmit FIFO to the transmit block, and receive data flows from the receive block to the receive FIFO.
- The user controls the FEC by writing, through the slave interface (SIF) submodule, to control registers located in each block. The control and status register (FEC_CSR) provides global control (for example, Ethernet reset and enable) and interrupt handling registers.
- The MII provides a serial channel for control/status communication with the external physical layer device (transceiver). This serial channel consists of the management data clock and management data input/output lines of the MII interface (FEC_MDC and FEC_MDIO, respectively).
- The DMA provides multiple channels allowing transmit data, transmit descriptor, receive data and receive descriptor accesses to run independently.
- The transmit and receive blocks provide the Ethernet MAC functionality (with some assistance from microcode).
- The message information block (MIB) maintains counters for a variety of network events and statistics. It is not necessary for operation of the FEC but provides valuable counters for network management. The counters supported are the RMON (RFC 1757) Ethernet statistics group and some of the IEEE 802.3 counters. See Message Information Block (MIB) Counters Memory Map, for more information.
- The MII gasket block (MIIGSK) provides an interface between the MII (Media Independent Interface specified by IEEE 802.3 standard) I/F and RMII (low pin count Reduced Media Independent Interface as specified by the RMII Consortium™ standard) I/F.

32.1.1 Features

The FEC incorporates the following features:

- Support for three different Ethernet physical interfaces:
  - 10-Mbps and 100-Mbps IEEE Std. 802.3™ MII
  - 10-Mbps and 100-Mbps RMII
  - 10-Mbps 7-wire interface (industry standard)
- IEEE 802.3 full-duplex flow control
- Programmable max frame length supports IEEE 802.1 VLAN tags and priority

- Support for full-duplex operation (200Mbps throughput) with a minimum system clock rate of 50 MHz
- Support for half-duplex operation (100Mbps throughput) with a minimum system clock rate of 25 MHz
- Retransmission from transmit FIFO following a collision (no processor bus utilization)
- Automatic internal flushing of the receive FIFO for runts (collision fragments) and address recognition rejects (no processor bus utilization)
- Address recognition
  - Frames with broadcast address can be always accepted or always rejected
  - Exact match for single 48-bit individual (unicast) address
  - Hash (64-bit hash) check of individual (unicast) addresses
  - Hash (64-bit hash) check of group (multicast) addresses
  - Promiscuous mode

32.2 Modes of Operation

The primary operational modes are described in this section.

32.2.1 Full- and Half-Duplex Operation

Full-duplex mode is intended for use on point to point links between switches or end node to switch. Half-duplex mode is used in connections between an end node and a repeater or between repeaters. Selection of the duplex mode is controlled by the FEC_TCR[FDEN] bit. When configured for full-duplex mode, flow control can be enabled, which is effected by the FEC_TCR[RFC_PAUSE], FEC_TCR[TFC_PAUSE], and FEC_RCR[FCE] bits. See Full-Duplex Flow Control, for more details.

32.2.2 Interface Options

The following interface options are supported. A detailed discussion of the interface configurations is provided in Network Interface Options.

32.2.2.1 10-Mbps and 100-Mbps Media Independent Interface (MII)

The MII is defined by the IEEE 802.3 standard for 10/100-Mbps operation. The MAC-PHY interface can be configured to operate in MII mode by asserting FEC_RCR[MII_MODE].

The speed of operation is determined by the FEC_TX_CLK and FEC_RX_CLK signals which are driven by the external transceiver. The transceiver either autonegotiates the speed or control by software via the serial management interface (FEC_MDC/FEC_MDIO) signals to the transceiver. See the descriptions in MII management frame register (FEC_MMFR) and MII speed control register (FEC_MSCR) respectively, as well as MII documentation for a description of how to read and write registers in the transceiver through this interface.

32.2.2.2 10 Mbps and 100 Mbps RMII Interface

RMII is a reduced MII interface specified by the RMII Consortium™ standard. The purpose of this interface is to provide a low cost alternative to the MII, with 8 pin interface instead of 16 (Not including MDC, MDIO pins).

The FEC uses a gasket (MIIGSK) to support RMII. It will operate in RMII mode by setting FEC_MIIGSK_CFGR[I/F_MODE] = 01, while setting FEC_MIIGSK_CFGR[I/F_MODE] = 00 will disable RMII mode. The speed selection in RMII Mode is supported by configuring the FEC_MIIGSK_CFGR[FRCONT]. The serial management interface is still available in RMII mode.

32.2.2.3 10-Mbps 7-Wire Interface Operation

The FEC supports a 7-wire interface as used by many 10 Mbps ethernet transceivers. The FEC_RCR[MII_MODE] bit controls this functionality. If this bit is cleared, the MII mode is disabled and the 10 Mbps, 7-wire mode is enabled.

32.2.3 Address Recognition Options

The address options supported are promiscuous, broadcast reject, individual address (hash or exact match), and multicast hash match. Address recognition options are discussed in detail in Ethernet Address Recognition.

32.2.4 Internal Loopback

Two levels of internal loopback modes are supported. The first level, which will loop data back before they enter MIIGSK, is selected by FEC_RCR[LOOP]. The second level, which will loop data back inside MIIGSK RMII domain, is selected by setting FEC_MIIGSK_CFGR[LBMODE] and FEC_MIIGSK_CFGR[I/F_MODE] = 01. Loopback mode is discussed in detail in Internal and External Loopback.

32.3 Functional Description

This section provides a detailed description of the functions of the FEC including:

- Network interface options
- Frame transmission and reception
- Full-duplex flow control
- Internal and external loopback

32.3.1 Network Interface Options

The FEC supports an MII interface, an RMII interface for 10/100 Mbps Ethernet and a 7-wire serial interface for 10 Mbps Ethernet.

The interface mode is selected by FEC_MIIGSK_CFGR[I/F_MODE] together with the FEC_RCR[MII_MODE] bit as shown in the table below.

Table 32-1. Interface Mode Selection

FEC_MIIGSK_CFGR[I/F_MODE]	FEC_RCR[MII_MODE]	Interface Mode Selected
00	0	7-Wire
00	1	MII
01	1	RMII

In MII mode, there are 18 signals defined by the IEEE 802.3 standard and supported by the FEC_ These signals are shown in the table below.

Table 32-2. MII Mode Signal Configuration

Signal Description	FEC Signal Name	Direction
Transmit clock	FEC_TX_CLK	In
Transmit enable	FEC_TX_EN	Out
Transmit data	FEC_TXD[3:0]	Out

Table continues on the next page...

Table 32-2. MII Mode Signal Configuration (continued)

Signal Description	FEC Signal Name	Direction
Transmit error	FEC_TX_ER	Out
Collision	FEC_COL	In
Carrier sense	FEC_CRS	In
Receive clock	FEC_RX_CLK	In
Receive data valid	FEC_RX_DV	In
Receive data	FEC_RXD[3:0]	In
Receive error	FEC_RX_ER	In
Management data clock	FEC_MDC	Out
Management data input/output	FEC_MDIO	I/O

In RMII mode, a reduced set of 10 signals are used. These signals are shown in Table 32-3.

Table 32-3. RMII Mode Signal Configuration

Signal Description	FEC Signal Name	Direction
Synchronous clock reference (REF_CLK)	FEC_TX_CLK	In
Transmit Enable	FEC_TX_EN	Out
Transmit Data	FEC_TXD[1:0]	Out
Carrier Sense/Receive Data Valid (CRS_DV)	FEC_RX_DV	In
Receive Data	FEC_RXD[1:0]	In
Receive Error	FEC_RX_ER	In
Management Data Clock	FEC_MDC	Out
Management Data Input/Output	FEC_MDIO	I/O

The 7-wire serial interface operates in what is generally referred to as AMD mode. 7-wire mode connections to the external transceiver are shown in Table 32-4.

Table 32-4. 7-Wire Mode Signal Configuration

Signal Description	FEC Signal Name	Direction
Transmit clock	FEC_TX_CLK	In
Transmit enable	FEC_TX_EN	Out
Transmit data	FEC_TXD[0]	Out
Collision	FEC_COL	In
Receive clock	FEC_RX_CLK	In
Receive data valid	FEC_RX_DV	In
Receive data	FEC_RXD[0]	In

32.3.2 FEC Frame Transmission

The Ethernet transmitter is designed to work with almost no intervention from software. After FEC_ECR[ETHER_EN] is set to 1 and data appears in the transmit FIFO, the FEC is able to transmit on to the network.

When the transmit FIFO fills to the watermark (defined by the FEC_TFWR register), the FEC transmit logic asserts FEC_TX_EN and starts transmitting the preamble (PA) sequence, the start frame delimiter (SFD), and then the frame information from the FIFO. However, the controller postpones transmission if the network is busy (that is, the carrier sense signal FEC_CRS is asserted). Before transmitting, the controller waits for carrier sense to become inactive, then determines if carrier sense remains inactive for 60 bit periods. If so, the transmission begins after waiting an additional 36 bit periods (96 bit periods after carrier sense originally became inactive). See Transmission Error Handling for more details.

If a collision occurs during transmission of a frame in half-duplex mode, the Ethernet controller follows the specified backoff procedures (see Collision Handling) and attempts to retransmit the frame until the retry limit is reached. The transmit FIFO stores at least the first 64 bytes of the transmit frame, so that they do not have to be retrieved from system memory in case of a collision. This improves bus utilization and latency in case immediate retransmission is necessary.

When all the frame data has been transmitted, if the TC bit is set in the transmit frame control word then a 32-bit cyclic redundancy check (CRC) known as the frame check sequence (FCS) is appended. If the ABC bit is set in the transmit frame control word, a bad CRC is appended to the frame data regardless of the TC bit value. Following the transmission of the CRC, the Ethernet controller writes the frame status information to the MIB block. Short frames are automatically padded by the transmit logic (if the TC bit in the transmit buffer descriptor for the end of frame buffer is set to 1).

Both buffer (TXB) and frame (TXF) interrupts can be generated as determined by the settings in the FEC_EIMR.

The transmit error interrupts are HBERR, BABT, LATE_COL, COL_RETRY_LIM, and XFIFO_UN. If the transmit frame length exceeds MAX_FL bytes the BABT interrupt is asserted: however, the entire frame is transmitted (no truncation).

Transmission is paused by setting the graceful transmit stop (GTS) bit in the FEC_TCR register. When the FEC_TCR[GTS] is set to 1, the FEC transmitter stops immediately if transmission is not in progress; otherwise, it continues transmission until the current

frame either finishes or terminates with a collision. After the transmitter has stopped the GRA (graceful stop complete) interrupt is asserted. If FEC_TCR[GTS] is cleared, the FEC resumes transmission with the next frame.

The Ethernet controller transmits bytes LSB first.

32.3.2.1 Transmit Inter-Packet Gap (IPG) Time

The minimum inter-packet gap (IPG) time for back-to-back transmission is 96 bit periods. After completing a transmission or after the backoff algorithm completes, the transmitter waits for carrier sense to be negated before starting its 96-bit-period IPG counter. Frame transmission begins 96 bit periods after carrier sense is negated if it stays negated for at least 60 bit periods. If carrier sense is asserted during the last 36 bit periods, it is ignored and a collision occurs.

32.3.2.2 Collision Handling

If a collision occurs during frame transmission, the Ethernet controller continues the transmission for at least 32 bit periods, transmitting a jam pattern consisting of 32 ones. If the collision occurs during the preamble sequence, the jam pattern is sent after the end of the preamble sequence.

If a collision occurs within 512 bit periods, the retry process is initiated. The transmitter waits a random number of slot periods, where one slot period is 512 bit periods. If a collision occurs after 512 bit periods, then no retransmission is performed and the end of frame buffer is closed with a late collision (LC) error indication.

32.3.2.3 Transmission Error Handling

The Ethernet controller reports frame transmission error conditions using the FEC RxBDs, the FEC_EIR register, and the MIB block counters.

There are four types of transmission errors:

- Transmitter underrun
- Retransmission attempts limit expired
- Late collision
- Heartbeat

The FEC's procedures for handling these errors are described in the following subsections.

32.3.2.3.1 Transmitter Underrun

If this error occurs, the FEC sends 32 bits that ensure a CRC error, then stops transmitting. All remaining buffers for that frame are then flushed and closed. The UN bit is set in the FEC_EIR and the UN interrupt is asserted (if enabled in the FEC_EIMR register). The FEC then continues to the next transmit buffer descriptor and begin transmitting the next frame.

32.3.2.3.2 Retransmission Attempts Limit Expired

When this error occurs, the FEC terminates transmission. All remaining buffers for that frame are flushed and closed, and the RL bit is set in the FEC_EIR. The FEC then continues to the next transmit buffer descriptor and begin transmitting the next frame.

The "RL" interrupt is asserted if enabled in the FEC_EIMR register.

32.3.2.3.3 Late Collision

When a collision occurs after the slot time (512 bits starting at the preamble), the FEC terminates transmission. All remaining buffers for that frame are flushed and closed, and the LC bit is set in the FEC_EIR register. The FEC then continues to the next transmit buffer descriptor and begin transmitting the next frame.

The LC interrupt is asserted if enabled in the FEC_EIMR register.

32.3.2.3.4 Heartbeat

Some transceivers have a self-test feature called "heartbeat" or "signal quality error." To signify a good self-test, the transceiver indicates a collision to the FEC within 4 microseconds after completion of a frame transmitted by the Ethernet controller. This indication of a collision does not imply a real collision error on the network, but is rather an indication that the transceiver still seems to be functioning properly. This is called the heartbeat condition.

If the HBC bit is set in the FEC_TCR register and the heartbeat condition is not detected by the **FEC** after a frame transmission, then a heartbeat error occurs. When this error occurs, the FEC closes the buffer, sets the HB bit in the FEC_EIR register, and generates the HBERR interrupt if it is enabled.

32.3.3 FEC Frame Reception

The FEC receiver is designed to work with almost no intervention from the host and can perform address recognition, CRC checking, short frame checking, and maximum frame length checking.

When the driver enables the FEC receiver by setting FEC_ECR[ETHER_EN] to 1, it immediately starts processing receive frames. When FEC_RX_DV asserts, the receiver first checks for a valid PA/SFD header. If the PA/SFD is valid, it is stripped and the frame is processed by the receiver. If a valid PA/SFD is not found, the frame is ignored.

In serial mode, the first 16 bit periods of RX_D0 following assertion of FEC_RX_DV are ignored. Following the first 16 bit periods the data sequence is checked for alternating 1/0s. If a 0b11 or 0b00 data sequence is detected during bit periods 17 to 21, the remainder of the frame is ignored. After bit period 21, the data sequence is monitored for a valid SFD (11). If a 0b00 is detected, the frame is rejected. When a 0b11 is detected, the PA/SFD sequence is complete.

In MII mode, the receiver checks for at least one byte matching the SFD. Zero or more PA bytes can occur, but if a 0b00 bit sequence is detected prior to the SFD byte, the frame is ignored.

After the first 6 bytes of the frame have been received, the FEC performs address recognition on the frame.

After a collision window (64 bytes) of data has been received, and if address recognition has not rejected the frame, the receive FIFO is signaled that the frame is accepted and can be passed on to the DMA. If the frame is a runt (due to collision) or is rejected by address recognition, the receive FIFO is notified to reject the frame. Thus no collision fragments are presented to the user except late collisions, which indicate serious LAN problems.

During reception, the Ethernet controller checks for various error conditions and after the entire frame is written into the FIFO, a 32-bit frame status word is written into the FIFO. This status word contains the M, BC, MC, LG, NO, CR, OV and TR status bits, and the frame length. See Reception Error Handling for more details.

Receive Buffer (RXB) and Frame Interrupts (RXF) can be generated if enabled by the FEC_EIMR register. The only receive error interrupt is babbling receiver error (BABR). Receive frames are not truncated if they exceed the max frame length (MAX_FL); however, the BABR interrupt occurs and the LG bit in the receive buffer descriptor (RxBD) is set. See Ethernet Receive Buffer Descriptor (RxBD) for more details.

When the receive frame is complete, the FEC sets the L bit in the RxBD, writes the other frame status bits into the RxBD, and clears the E bit. The Ethernet controller next generates a maskable interrupt (RXF bit in FEC_EIR, maskable by RXF bit in FEC_EIMR), indicating that a frame has been received and is in memory. The Ethernet controller then waits for a new frame.

The Ethernet controller receives serial data LSB first.

32.3.3.1 Receive Inter-Packet Gap (IPG) Time

The receiver receives back-to-back frames with a minimum spacing of 28-bit periods. If an inter-packet gap between receive frames is less than 28 bit periods, the second frame may be discarded by the receiver.

32.3.3.2 Ethernet Address Recognition

Address recognition is accomplished through the use of the receive block and microcode running on the microcontroller. The flowchart shown in Figure 32-2 illustrates the address recognition decisions made by the receive block, while Figure 32-3 illustrates the decisions made by the microcontroller.

The FEC filters the received frames based on destination address (DA) type - individual (unicast), group (multicast), or broadcast (all-ones group address). The difference between an individual address and a group address is determined by the I/G bit in the DA field.

If the DA is a broadcast address, and broadcast reject (FEC_RCR[BC_REJ]) is cleared, then the frame is accepted unconditionally, as shown in Figure 32-2. Otherwise, if the DA is not a broadcast address, then the microcontroller runs the address recognition subroutine, as shown in Figure 32-3.

If the DA is a group (multicast) address and flow control is disabled, then the microcontroller performs a group hash table lookup using the 64-entry hash table programmed in FEC_GAUR and FEC_GALR. If a hash match occurs, the receiver accepts the frame. The hash algorithm is described in Hash Algorithm.

If flow control is enabled, the microcontroller does an exact address match check between the DA and the designated PAUSE DA (01:80:C2:00:00:01). If the receive block determines that the received frame is a valid pause frame, then the frame is rejected.

NOTE

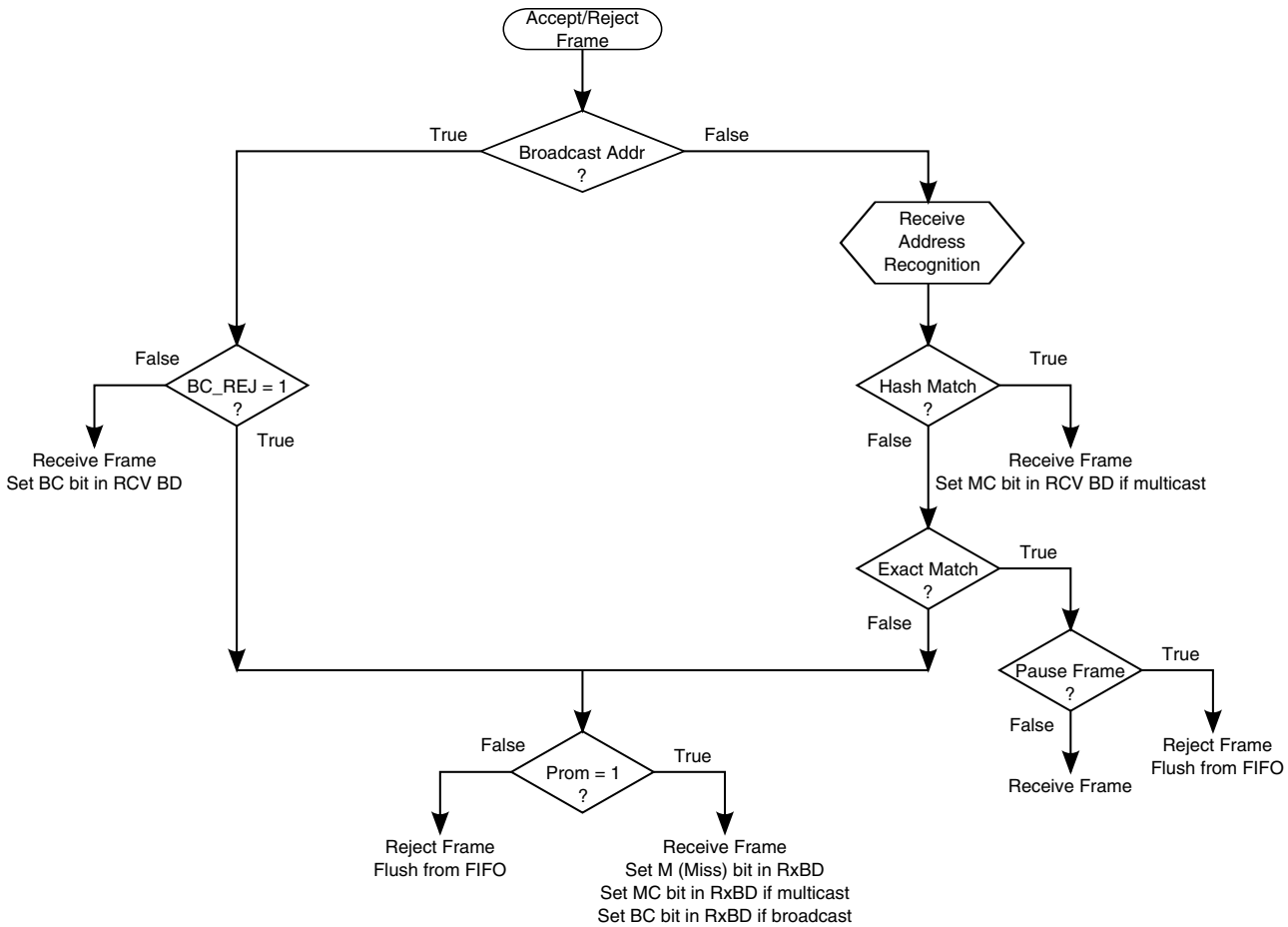
The receiver detects a pause frame with the DA field set to either the designated PAUSE DA or the unicast physical address.

If the DA is the individual (unicast) address, the microcontroller performs an individual exact match comparison between the DA and 48-bit physical address that the user programs in the FEC_PALR and FEC_PAUR registers. If an exact match occurs, the frame is accepted; otherwise, the microcontroller does an individual hash table lookup using the 64-entry hash table programmed in registers, FEC_IAUR and FEC_IALR (the hash algorithm is described in Hash Algorithm). In the case of an individual hash match, the frame is accepted. Again, the receiver accepts or rejects the frame based on pause frame detection, shown in Figure 32-2.

If neither a hash match (group or individual), nor an exact match occur, then if promiscuous mode is enabled (FEC_RCR[PROM] = 1), the frame is accepted and the MISS bit in the receive buffer descriptor is set. Otherwise, the frame is rejected.

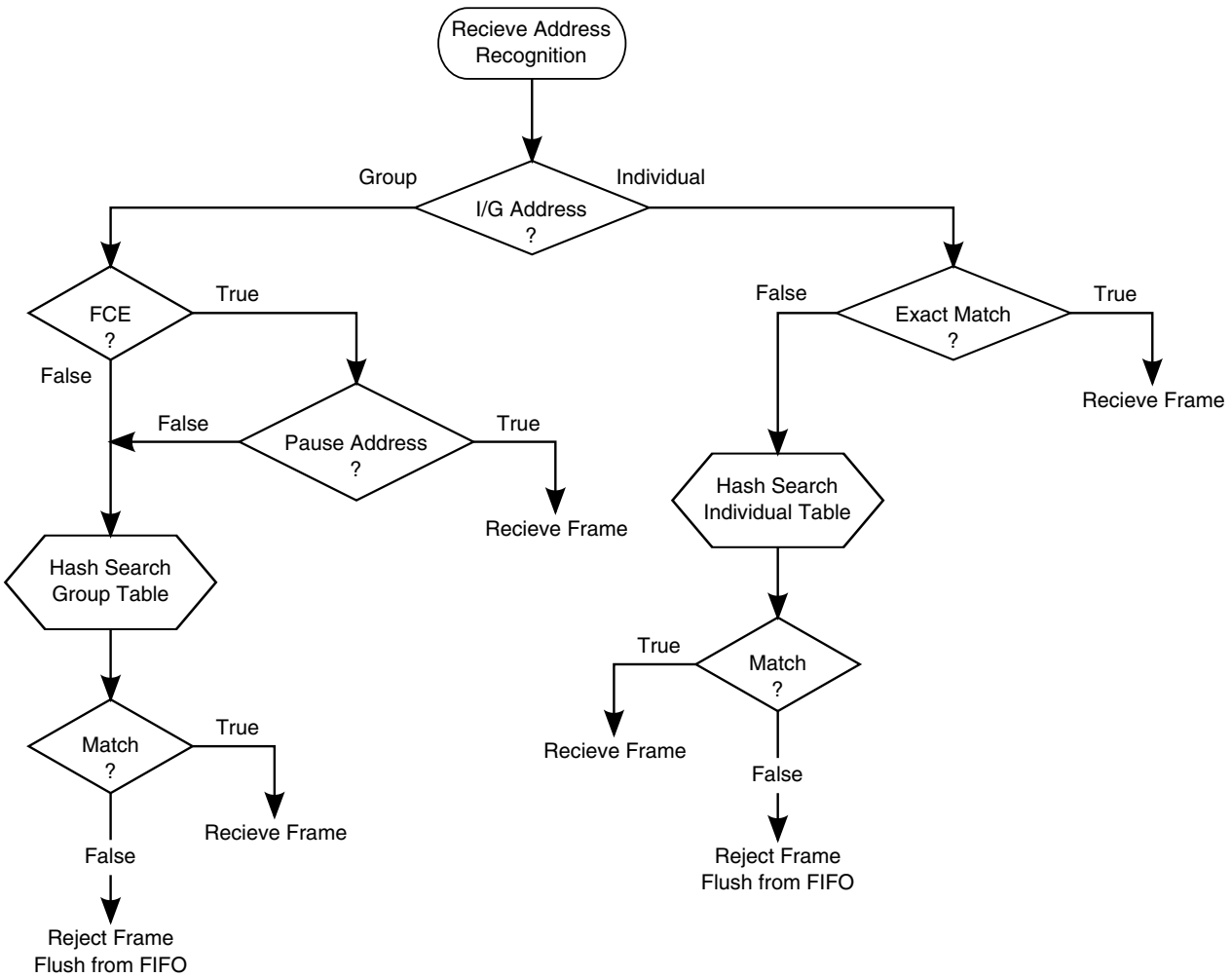
Similarly, if the DA is a broadcast address, broadcast reject (FEC_RCR[BC_REJ]) is asserted, and promiscuous mode is enabled, then the frame is accepted and the MISS bit in the receive buffer descriptor is set. Otherwise, the frame is rejected.

In general, when a frame is rejected, it is flushed from the FIFO.



NOTES:
BC_REJ - Field in RCR register (BroadCast REJect)
PROM-Field in RCR register (PROMiscuous mode)
Pause Frame - Valid PAUSE frame received

Figure 32-2. Ethernet Address Recognition-Receive Block Decisions



NOTES:
FCE - field in RCR register (Flow Control Enable)
I/G - Individual/Group bit in Destination Address (least signifigant bit in first byte recieved in MAC frame)

Figure 32-3. Ethernet Address Recognition-Microcode Decisions

32.3.3.2.1 Hash Algorithm

The hash table algorithm used in the group and individual hash filtering operates as follows. The 48-bit destination address is mapped into one of 64 bits, which are represented by 64 bits stored in FEC_GAUR, FEC_GALR (group address hash match) or FEC_IAUR, FEC_IALR (individual address hash match). This mapping is performed by passing the 48-bit address through the on-chip 32-bit CRC generator and selecting the six most significant bits of the CRC-encoded result to generate a number between 0 and 63. The MSB of the CRC result selects FEC_GAUR (MSB = 1) or FEC_GALR (MSB = 0). The least significant five bits of the hash result select the bit within the selected register. If the CRC generator selects a bit that is set in the hash table, the frame is accepted; otherwise, it is rejected. For example, if eight group addresses are stored in the hash table

and random group addresses are received, the hash table prevents roughly 56/64 (87.5%) of the group address frames from reaching memory. Those that do reach memory must be further filtered by the processor to determine if they truly contain one of the eight desired addresses.

The effectiveness of the hash table declines as the number of addresses increases.

The hash table registers must be initialized by the user. The CRC32 polynomial to use in computing the hash is shown in the following equation:

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$$

Table 32-5 shows example destination addresses and corresponding hash values is included below for reference.

Table 32-5. Destination Address to 6-Bit Hash

48-bit DA (in hex)	6-bit Hash (in hex)	Hash Decimal Value
65:FF:FF:FF:FF:FF	0x0	0
55:FF:FF:FF:FF:FF	0x1	1
15:FF:FF:FF:FF:FF	0x2	2
35:FF:FF:FF:FF:FF	0x3	3
B5:FF:FF:FF:FF:FF	0x4	4
95:FF:FF:FF:FF:FF	0x5	5
D5:FF:FF:FF:FF:FF	0x6	6
F5:FF:FF:FF:FF:FF	0x7	7
DB:FF:FF:FF:FF:FF	0x8	8
FB:FF:FF:FF:FF:FF	0x9	9
BB:FF:FF:FF:FF:FF	0xA	10
8B:FF:FF:FF:FF:FF	0xB	11
0B:FF:FF:FF:FF:FF	0xC	12
3B:FF:FF:FF:FF:FF	0xD	13
7B:FF:FF:FF:FF:FF	0xE	14
5B:FF:FF:FF:FF:FF	0xF	15
27:FF:FF:FF:FF:FF	0x10	16
07:FF:FF:FF:FF:FF	0x11	17

Table continues on the next page...

Table 32-5. Destination Address to 6-Bit Hash (continued)

48-bit DA (in hex)	6-bit Hash (in hex)	Hash Decimal Value
57:FF:FF:FF:FF:FF	0x12	18
77:FF:FF:FF:FF:FF	0x13	19
F7:FF:FF:FF:FF:FF	0x14	20
C7:FF:FF:FF:FF:FF	0x15	21
97:FF:FF:FF:FF:FF	0x16	22
A7:FF:FF:FF:FF:FF	0x17	23
99:FF:FF:FF:FF:FF	0x18	24
B9:FF:FF:FF:FF:FF	0x19	25
F9:FF:FF:FF:FF:FF	0x1A	26
C9:FF:FF:FF:FF:FF	0x1B	27
59:FF:FF:FF:FF:FF	0x1C	28
79:FF:FF:FF:FF:FF	0x1D	29
29:FF:FF:FF:FF:FF	0x1E	30
19:FF:FF:FF:FF:FF	0x1F	31
D1:FF:FF:FF:FF:FF	0x20	32
F1:FF:FF:FF:FF:FF	0x21	33
B1:FF:FF:FF:FF:FF	0x22	34
91:FF:FF:FF:FF:FF	0x23	35
11:FF:FF:FF:FF:FF	0x24	36
31:FF:FF:FF:FF:FF	0x25	37
71:FF:FF:FF:FF:FF	0x26	38
51:FF:FF:FF:FF:FF	0x27	39
7F:FF:FF:FF:FF:FF	0x28	40
4F:FF:FF:FF:FF:FF	0x29	41
1F:FF:FF:FF:FF:FF	0x2A	42
3F:FF:FF:FF:FF:FF	0x2B	43
BF:FF:FF:FF:FF:FF	0x2C	44
9F:FF:FF:FF:FF:FF	0x2D	45
DF:FF:FF:FF:FF:FF	0x2E	46
EF:FF:FF:FF:FF:FF	0x2F	47
93:FF:FF:FF:FF:FF	0x30	48
B3:FF:FF:FF:FF:FF	0x31	49
F3:FF:FF:FF:FF:FF	0x32	50
D3:FF:FF:FF:FF:FF	0x33	51
53:FF:FF:FF:FF:FF	0x34	52

Table continues on the next page...

Table 32-5. Destination Address to 6-Bit Hash (continued)

48-bit DA (in hex)	6-bit Hash (in hex)	Hash Decimal Value
73:FF:FF:FF:FF:FF	0x35	53
23:FF:FF:FF:FF:FF	0x36	54
13:FF:FF:FF:FF:FF	0x37	55
3D:FF:FF:FF:FF:FF	0x38	56
0D:FF:FF:FF:FF:FF	0x39	57
5D:FF:FF:FF:FF:FF	0x3A	58
7D:FF:FF:FF:FF:FF	0x3B	59
FD:FF:FF:FF:FF:FF	0x3C	60
DD:FF:FF:FF:FF:FF	0x3D	61
9D:FF:FF:FF:FF:FF	0x3E	62
BD:FF:FF:FF:FF:FF	0x3F	63

32.3.3.3 Reception Error Handling

The Ethernet controller reports frame reception error conditions using the FEC RxBDs, the FEC_EIR register, and the MIB counters.

There are five types of reception errors:

- Overrun
- Non-octet (dribbling bits)
- CRC
- Frame-length violation
- Truncation

These are described in the following subsections.

32.3.3.3.1 Overrun

If the receive block has data to put into the receive FIFO and the receive FIFO is full, the FEC sets the OV bit in the RxBD. All subsequent data in the frame is discarded, and subsequent frames can also be discarded until the receive FIFO is serviced by the DMA and space is made available. At this point the receive frame/status word is written into the FIFO with the OV bit set. This frame must be discarded by the driver.

32.3.3.3.2 Non-Octet (Dribbling Bits)

The Ethernet controller handles up to seven dribbling bits when the receive frame terminates past an non-octet aligned boundary. Dribbling bits are not used in the CRC calculation. If there is a CRC error, then the frame non-octet aligned (NO) error is reported in the RxBD. If there is no CRC error, then no error is reported.

32.3.3.3.3 CRC

When a CRC error occurs with no dribble bits, the FEC closes the buffer and sets the CR bit in the RxBD. CRC checking cannot be disabled, but the CRC error can be ignored if checking is not required.

32.3.3.3.4 Frame Length Violation

When the receive frame length exceeds MAX_FL bytes the BABR interrupt is generated, and the LG bit in the end of frame RxBD is set. The frame is not truncated unless the frame length exceeds 2047 bytes).

32.3.3.3.5 Truncation

When the receive frame length exceeds 2047 bytes, the frame is truncated and the TR bit is set in the RxBD.

32.3.4 Full-Duplex Flow Control

Full-duplex flow control allows the user to transmit pause frames and to detect received pause frames. Upon detection of a pause frame, FEC data frame transmission stops for a given pause duration.

To enable pause frame detection, the FEC must operate in full-duplex mode (FEC_TCR[FDEN] asserted) and flow control enable (FEC_RCR[FCE]) must be asserted. The FEC detects a pause frame when the fields of the incoming frame match the pause frame specifications, as shown in Table 32-6. In addition, the receive status associated with the frame indicates that the frame is valid.

Table 32-6. Pause Frame Field Specification

48-bit destination address	0x0180_C200_0001 or physical address
48-bit source address	Any
16-bit type	0x8808
16-bit opcode	0x0001

Table continues on the next page...

Table 32-6. Pause Frame Field Specification (continued)

48-bit destination address	0x0180_C200_0001 or physical address
16-bit pause duration	0x0000-0xFFFF

Pause frame detection is performed by the receiver and microcontroller modules. The microcontroller runs an address recognition subroutine to detect the specified pause frame destination address, while the receiver detects the type and opcode pause frame fields. On detection of a pause frame, FEC_TCR[GTS] is asserted by the FEC internally. When transmission has paused, the FEC_EIR[GRA] interrupt is asserted and the pause timer begins to increment.

NOTE

The pause timer makes use of the transmit backoff timer hardware, which is used for tracking the appropriate collision backoff time in half-duplex mode. The pause timer increments after every slot time, until FEC_OPD[PAUSE_DUR] slot times have expired.

On FEC_OPD[PAUSE_DUR] expiration, FEC_TCR[GTS] is cleared allowing FEC data frame transmission to resume.

NOTE

The receive flow control pause (FEC_TCR[RFC_PAUSE]) status bit is asserted while the transmitter is paused due to reception of a pause frame.

To transmit a pause frame, the FEC must operate in full-duplex mode and the user must assert flow control pause (FEC_TCR[TFC_PAUSE]). On assertion of transmit flow control pause (FEC_TCR[TFC_PAUSE]), the transmitter asserts FEC_TCR[GTS] internally. When the transmission of data frames stops, the FEC_EIR[GRA] (graceful stop complete) interrupt asserts. Following FEC_EIR[GRA] assertion, the pause frame is transmitted. On completion of pause frame transmission, flow control pause (FEC_TCR[TFC_PAUSE]) and FEC_TCR[GTS] are cleared internally.

The user must specify the desired pause duration in the FEC_OPD register.

NOTE

When the transmitter is paused due to receiver/microcontroller pause frame detection, transmit flow control pause (FEC_TCR[TFC_PAUSE]) still can be asserted and causes the transmission of a single pause frame. In this case, the FEC_EIR[GRA] interrupt is not asserted.

32.3.5 Internal and External Loopback

Both internal and external loopback are supported by the Ethernet controller. In loopback mode, both of the FIFOs are used and the FEC actually operates in a full-duplex fashion. Both internal and external loopback are configured using combinations of the LOOP and DRT bits in the FEC_RCR register, the FDEN bit in the FEC_TCR register and the LBMODE bit in FEC_MIIGSK_CFGR register. Furthermore internal loopback has two levels which loops data back in MII domain and RMII domain respectively.

For both internal and external loopback set FDEN = 1 and FEC_RCR[DRT] = 0.

For the MII domain internal loopback set FEC_RCR[LOOP] = 1 and FEC_MIIGSK_CFGR[LBMODE] = 0. FEC_TX_EN and FEC_TX_ER will not assert during internal loopback. During the MII domain internal loopback, the transmit/receive data rate is higher than in normal operation because the internal system clock is used by the transmit and receive blocks instead of the clocks from the external transceiver. This will cause an increase in the required system bus bandwidth for transmit and receive data being DMA'd to/from external memory. It may be necessary to pace the frames on the transmit side and/or limit the size of the frames to prevent transmit FIFO underrun and receive FIFO overflow.

For the RMII domain internal loopback, set FEC_RCR[LOOP] = 0 and FEC_MIIGSK_CFGR[LBMODE] = 1. The data is sent from the MII domain into MIIGSK, converted to RMII format by a RMII transmitter inside the MIIGSK, then looped back through a RMII receiver of the MIIGSK, finally gets back to the MII domain.

For external loopback set FEC_RCR[LOOP] = 0, FEC_MIIGSK_CFGR[LBMODE] = 0, and configure the external transceiver for loopback.

32.4 Initialization/Application Information

32.4.1 Initialization Sequence

This section describes which registers are reset due to hardware reset, which are reset by the FEC RISC, and what locations the user must initialize prior to enabling the FEC.

32.4.1.1 Hardware Controlled Initialization

In the FEC, registers and control logic that generate interrupts are reset by hardware. A hardware reset negates output signals and resets general configuration bits.

Other registers are reset when the FEC_ECR[ETHER_EN] bit is cleared, as indicated in Table 32-7. FEC_ECR[ETHER_EN] is cleared by a hard reset or can be cleared by software to halt operation. By clearing FEC_ECR[ETHER_EN], the configuration control registers such as the FEC_TCR and FEC_RCR do not reset, but the entire data path resets.

Table 32-7. Effect on FEC of Clearing FEC_ECR[ETHER_EN]

Register/Machine	Reset Value
XMIT block	Transmission is aborted (bad CRC appended)
RECV block	Receive activity is aborted
DMA block	All DMA activity is terminated
FEC_RDAR	Cleared
FEC_TDAR	Cleared
Descriptor Controller block	Halt operation
MIIGSK	Interface to transceiver is disabled

32.4.1.2 User Initialization (Prior to Asserting FEC_ECR[ETHER_EN])

The user must initialize portions of the FEC prior to setting the FEC_ECR[ETHER_EN] bit. The exact values depend on the particular application. The order of initializations is not important except those that are explicitly mentioned.

The FEC and FEC FIFO/DMA registers which require initialization are defined in Table 32-8.

Table 32-8. Registers Requiring Initialization before Setting FEC_ECR[ETHER_EN]

Register	Location (FEC or FIFO/DMA)	Comments
FEC_EIMR	FEC	Requires initialization
FEC_EIR	FEC	Must be cleared by writing 0xFFFF_FFFF
FEC_TFWR	FEC	Optional
FEC_IALR / FEC_IAUR	FEC	-
FEC_GAUR / FEC_GALR	FEC	-

Table continues on the next page...

Table 32-8. Registers Requiring Initialization before Setting FEC_ECR[ETHER_EN] (continued)

Register	Location (FEC or FIFO/ DMA)	Comments
FEC_PALR / FEC_PAUR	FEC	-
FEC_OPD	FEC	Only needed for full-duplex flow control
FEC_RCR	FEC	-
FEC_TCR	FEC	-
FEC_MSCR	FEC	Optional
FEC_MIIGSK_CFGR	FEC	Must be configured before setting FEC_MIIGSK_ENR[EN]
FEC_MIIGSK_ENR	FEC	EN bit must be set
MIB counters	FEC	Must be cleared
FEC_FRSR	FIFO/DMA	Initialization is optional
FEC_EMRBR	FIFO/DMA	-
FEC_ERDSR	FIFO/DMA	-
FEC_ETDSR	FIFO/DMA	-
Transmit Descriptor ring	FIFO/DMA	Must be emptied
Receive Descriptor ring	FIFO/DMA	Must be emptied

32.4.1.3 Microcontroller Initialization

In the FEC, the descriptor control RISC initializes some registers after FEC_ECR[ETHER_EN] is asserted. After the microcontroller initialization sequence is complete, the hardware is ready for operation.

The microcontroller initialization sequence follows:

1. Initialize BackOff Random Number Seed
2. Activate Receiver
3. Activate Transmitter
4. Clear Transmit FIFO
5. Clear Receive FIFO
6. Initialize Transmit Ring Pointer
7. Initialize Receive Ring Pointer
8. Initialize FIFO Count Register

32.4.1.4 User Initialization (after asserting FEC_ECR[ETHER_EN])

After asserting FEC_ECR[ETHER_EN], the user can set up the buffer/frame descriptors and write to the FEC_TDAR and FEC_RDAR. See Buffer Descriptors for more details.

32.4.2 Buffer Descriptors

This section provides a description of the operation of the driver/DMA through the buffer descriptors (BD). It is followed by a detailed description of the receive and transmit descriptor fields.

32.4.2.1 Driver/DMA Operation with Buffer Descriptors

The data for the FEC frames must reside in memory external to the FEC. The data for a frame is placed in one or more buffers. Associated with each buffer is a buffer descriptor (BD) which contains a starting address (pointer), data length, and status/control information (which contains the current state for the buffer). To permit maximum user flexibility, the BDs are also located in external memory and are read in by the FEC DMA engine.

Software "produces" buffers by allocating/initializing memory and initializing buffer descriptors. Setting the RxBD[E] or TxBD[R] bit "produces" the buffer. Software writing to either the FEC_TDAR or FEC_RDAR tells the FEC that a buffer has been placed in external memory for the transmit or receive data traffic, respectively. The hardware reads the BDs and "consumes" the buffers after they have been produced. After the data DMA is complete and the buffer descriptor status bits have been written by the DMA engine, the RxBD[E] or TxBD[R] bit is cleared by hardware to signal the buffer has been "consumed." Software can poll the BDs to detect when the buffers have been consumed or can rely on the buffer/frame interrupts. These buffers can then be processed by the driver and returned to the free list.

The FEC_ECR[ETHER_EN] signal operates as a reset to the BD/DMA logic. When FEC_ECR[ETHER_EN] is cleared the DMA engine BD pointers are reset to point to the starting transmit and receive BDs. The buffer descriptors are not initialized by hardware during reset. At least one transmit and receive buffer descriptor must be initialized by software before the FEC_ECR[ETHER_EN] bit is set.

The buffer descriptors operate as two separate rings. FEC_ERDSR defines the starting address for receive BDs and FEC_ETDSR defines the starting address for transmit BDs. The last buffer descriptor in each ring is defined by the Wrap (W) bit. When set, W indicates that the next descriptor in the ring is at the location pointed to by FEC_ERDSR and FEC_ETDSR for the receive and transmit rings, respectively.

NOTE

Buffer descriptor rings must start on a 128-bit boundary.

32.4.2.2 Ethernet Transmit Buffer Descriptor (TxBD)

Table 32-9 shows the transmit buffer descriptor format. Table 32-10 describes the TxBD fields.

Status bits for the buffer/frame are not included in the transmit buffer descriptors. Transmit frame status is indicated by individual interrupt bits (error conditions) and in statistic counters in the MIB block. See Message Information Block (MIB) Counters Memory Map, for more details.

Table 32-9. Transmit Buffer Descriptor (TxBD)

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	R	TO1	W	TO2	L	TC	ABC	PTP								
	Data Length															
0x0004	Tx Data Buffer Pointer A[31:16]															
	Tx Data Buffer Pointer A[15:0]															

Table 32-10. Transmit Buffer Descriptor Field Definitions

Offset	Field	Description
0x0000	31 R	Ready. Written by the FEC and the user. 0 The data buffer associated with this BD is not ready for transmission. The user is free to manipulate this BD or its associated data buffer. The FEC clears this bit after the buffer has been transmitted or after an error condition is encountered. 1 The data buffer, which has been prepared for transmission by the user, has not been transmitted or is currently being transmitted. No fields of this BD can be written by the user after this bit is set.
0x0000	30 TO1	Transmit software ownership. This field is reserved for software use. This read/write bit is not modified by hardware, nor does its value affect hardware.
0x0000	29 W	Wrap. Written by user. 0 The next buffer descriptor is found in the consecutive location 1 The next buffer descriptor is found at the location defined in FEC_ETDSR.
0x0000	28 TO2	Transmit software ownership. This field is reserved for use by software. This read/write bit is not modified by hardware, nor does its value affect hardware.
0x0000	27 L	Last in frame. Written by user. 0 The buffer is not the last in the transmit frame. 1 The buffer is the last in the transmit frame.

Table continues on the next page...

Table 32-10. Transmit Buffer Descriptor Field Definitions (continued)

Offset	Field	Description
0x0000	26 TC	Tx CRC. Written by user (only valid if L = 1). 0 End transmission immediately after the last data byte. 1 Transmit the CRC sequence after the last data byte.
0x0000	25 ABC	Append bad CRC. Written by user (only valid if L = 1). 0 No effect 1 Transmit the CRC sequence inverted after the last data byte (regardless of TC value).
0x0000	24 PTP	PTP. Written by user (only valid if L = 1). 0 No effect 1 The frame is an IEEE1588 PTP frame. FEC will inform IPTP Assist modue to capture the timestamp for this frame in case of successful transmission. If unmasked, an interrupt will be issued by the IPTP Assist block after the timestamp is available.
0x0000	23-16	Reserved.
0x0000	15-0 Data Length	Data Length, written by user. Data length is the number of octets the FEC transmits from this BD's data buffer. It is never modified by the FEC. Bits [10:0] are used by the DMA engine, bits[15:11] are ignored.
0x0004	31-0 A	Tx data buffer pointer[1]

1. The transmit buffer pointer, which contains the address of the associated data buffer, must always be divisible by 16. The buffer must reside in memory external to the FEC. This value is never modified by the Ethernet controller.

32.4.2.2.1 Driver/DMA Operation with Transmit Buffer Descriptors

Data is presented to the FEC for transmission by arranging it in buffers referenced by the channel's TxBDs. After the software driver has set up the buffers for a frame, it sets up the corresponding BDs. In the TxBD the user initializes the R, W, L, and TC bits and the length (in bytes) in the first longword, and the buffer pointer in the second longword.The last step in setting up the BDs for a transmit frame is to set the R bit in the first BD for the frame. The driver follows that with a write to FEC_TDAR which triggers the FEC to poll the next BD in the ring.

The Ethernet controller confirms transmission by clearing the ready bit (R bit) when DMA of the buffer is complete.

32.4.2.2.1.1 Transmit Frame in Multiple Buffers

Typically a transmit frame is divided between multiple buffers. For example, it is possible to have an application payload in one buffer, TCP header in a 2nd buffer, IP header in a 3rd buffer, and Ethernet/IEEE 802.3 header in a 4th buffer. The FEC does not prepend the Ethernet header (Destination Address, Source Address, Length/Type

field(s)), so this must be provided by the driver in one of the transmit buffers. The FEC or the driver can append the Ethernet CRC to the frame. The driver must set the TC bit in the transmit BD to determine whether the CRC is appended by the FEC or by the driver.

The driver (TxBD software producer) sets up Tx BDs in such a way that a complete transmit frame is given to the hardware at once. If a transmit frame consists of three buffers, first the BD's are initialized with pointer, length and control (W, L, TC, ABC) and then the TxBD[R] bits are set to 1 in *reverse order* (3rd, 2nd, 1st BD) to insure that the complete frame is ready in memory before the DMA begins. If the TxBDs are set up in order, the DMA controller could DMA the first BD before the 2nd was made available, potentially causing a transmit FIFO underrun.

In the FEC, the DMA is notified by the driver that new transmit frame(s) are available by writing to the FEC_TDAR register. When this register is written to (data value is not significant) the FEC RISC tells the DMA to read the next transmit BD in the ring. After started, the RISC + DMA continues to read and interpret transmit BDs in order and DMA the associated buffers, until a transmit BD is encountered with the R bit = 0. At this point the FEC polls this BD one more time. If the R bit = 0 the second time, then the RISC stops the transmit descriptor read process until software sets up another transmit frame and writes to FEC_TDAR.

When the DMA of each transmit buffer is complete, the DMA writes back to the BD to clear the R bit, indicating that the hardware consumer is finished with the buffer.

32.4.2.3 Ethernet Receive Buffer Descriptor (RxBD)

Table 32-11. Receive Buffer Descriptor (RxBD)

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	E	RO1	W	RO2	L	PTP		M	B	MC	LG	NO		CR	OV	TR
	Data Length															
0x0004	Rx Data Buffer Pointer A[31:16]															
	Rx Data Buffer Pointer A[15:0]															

Table 32-12. Receive Buffer Descriptor Field Definitions

Offset	Field	Description
0x0000	31 E	Empty. Written by the FEC (=0) and user (=1). 0 The data buffer associated with this BD has been filled with received data, or data reception has been aborted due to an error condition. The status and length fields have been updated as required. 1 The data buffer associated with this BD is empty, or reception is currently in progress.

Table continues on the next page...

Table 32-12. Receive Buffer Descriptor Field Definitions (continued)

Offset	Field	Description
0x0000	30 RO1	Receive software ownership. This field is reserved for use by software. This read/write bit is not modified by hardware, nor does its value affect hardware.
0x0000	29 W	Wrap. Written by user. 0 The next buffer descriptor is found in the consecutive location 1 The next buffer descriptor is found at the location defined in FEC_ERDSR.
0x0000	28 RO2	Receive software ownership. This field is reserved for use by software. This read/write bit is not modified by hardware, nor does its value affect hardware.
0x0000	27 L	Last in frame. Written by the FEC. 0 The buffer is not the last in a frame. 1 The buffer is the last in a frame.
0x0000	26 PTP	PTP frame received. Written by the FEC. This bit is valid only if the L-bit is set. 0 Received frame is not an IEEE1588 PTP frame. 1 Received frame is an IEEE1588 PTP frame. FEC has informed IPTP Assist block to capture the timestamp for this frame. If unmasked, an interrupt will be issued by the IPTP Assist block after the timestamp is available.
0x0000	25	Reserved.
0x0000	24 M	Miss. Written by the FEC. This bit is set by the FEC for frames that were accepted in promiscuous mode, but were flagged as a "miss" by the internal address recognition. Thus, while in promiscuous mode, the user can use the M-bit to quickly determine whether the frame was destined to this station. This bit is valid only if the L-bit is set and the PROM bit is set. 0 The frame was received because of an address recognition hit. 1 The frame was received because of promiscuous mode.
0x0000	23 BC	Set if the DA is broadcast (FF:FF:FF:FF:FF:FF).
0x0000	22 MC	Set if the DA is multicast and not BC.
0x0000	21 LG	Rx frame length violation. Written by the FEC. A frame length greater than FEC_RCR[MAX_FL] was recognized. This bit is valid only if the L-bit is set. The receive data is not altered in any way unless the length exceeds 2047 bytes.
0x0000	20 NO	Receive non-octet aligned frame. Written by the FEC. A frame that contained a number of bits not divisible by 8 was received, and the CRC check that occurred at the preceding byte boundary generated an error. This bit is valid only if the L-bit is set. If this bit is set the CR bit is not set.
0x0000	19	Reserved
0x0000	18 CR	Receive CRC error. Written by the FEC. This frame contains a CRC error and is an integral number of octets in length. This bit is valid only if the L-bit is set.

Table continues on the next page...

Table 32-12. Receive Buffer Descriptor Field Definitions (continued)

Offset	Field	Description
0x0000	17 OV	Overrun. Written by the FEC. A receive FIFO overrun occurred during frame reception. If this bit is set, the other status bits, M, LG, NO, CR, and CL lose their normal meaning and is zero. This bit is valid only if the L-bit is set.
0x0000	16 TR	Set if the receive frame is truncated (frame length > 2047 bytes). If the TR bit is set the frame is discarded and the other error bits ignored as they can be incorrect.
0x0000	15-0 Data Length	Data length. Written by the FEC. Data length is the number of octets written by the FEC into this BD's data buffer if L = 0 (the value is equal to FEC_EMRBR), or the length of the frame including CRC if L = 1. It is written by the FEC once as the BD is closed.
0x0004	31-0	RX data buffer pointer[1]

1. The receive buffer pointer, which contains the address of the associated data buffer, must always be divisible by 16. The buffer must reside in memory external to the FEC. This value is never modified by the Ethernet controller.

32.4.2.3.1 Driver/DMA Operation with Receive Buffer Descriptors

Unlike the transmit case, the length of the receive frame is unknown by the driver ahead of time. Therefore the driver must set a variable to define the length of all receive buffers. In the FEC, this variable is written to the FEC_EMRBR register.

The driver (RxBD software producer) sets up some number of "empty" buffers for the Ethernet by initializing the address field and the E and W bits of the associated receive BDs. The hardware (receive DMA) consumes these buffers by filling them with data as frames are received, and clearing the E bit and writing to the L bit (1 indicates last buffer in frame), the frame status bits (if L = 1) and the length field.

If a receive frame spans multiple receive buffers, the L bit is only set for the last buffer in the frame. For non-last buffers, the length field in the receive BD is written by the DMA (at the same time the E bit is cleared) with the default receive buffer length value. For end of frame buffers, the L=1 bit is set in the receive BD, and information written to the status bits (M, BC, MC, LG, NO, CR, OV, TR). Some of the status bits are error indicators which, if set, indicate the receive frame is discarded and not given to higher layers. The frame status/length information is written into the receive FIFO following the end of the frame (as a single 32-bit word) by the receive logic. The length field for the end of frame buffer is written with the length of the entire frame, not just the length of the last buffer.

For simplicity the driver can assign the default receive buffer length to be large enough to contain an entire frame, keeping in mind that a malfunction on the network or out-of-specification implementation could result in giant frames. Frames of 2 Kbytes (2048 bytes) or larger are truncated by the FEC at 2047 bytes, so software never sees a receive frame larger than 2047 bytes.

As in the transmit case, the FEC polls the receive descriptor ring after the driver sets up receive BDs and writes to the FEC_RDAR register. As frames are received the FEC fills receive buffers and update the associated BDs, then reads the next BD in the receive descriptor ring. If the FEC reads a receive BD and finds the E bit is cleared, it polls this BD once more. If E is still cleared, then the FEC stops reading receive BDs until the driver writes to FEC_RDAR.

NOTE

Whenever the software driver sets an E bit in one or more receive descriptors, the driver follows that with a write to FEC_RDAR.

32.5 Programmable Registers

32.5.1 Top Level Block Memory Map

The FEC implementation requires a 1 Kbyte memory space. This space is divided into 2 sections of 512 bytes each. The first is used for control/status registers, and the second contains event/statistics counters held in the MIB block.

Table 32-13 defines the top level memory map. For the base address of a particular block instantiation, see the system memory map.

Table 32-13. Block Memory Map

Base Address Offset	Function
0x0000-01FF	Control/Status Registers
0x0200-02FF	MIB Block Counters
0x0300-03FF	Extension Registers (MIIGSK)

32.5.2 Message Information Block (MIB) Counters Memory Map

Table 32-14 shows the MIB counters memory map, which defines the locations in the MIB RAM space where hardware maintained counters reside. It is the responsibility of software to poll the counters often enough to ensure that rollover is detected. For example, on a 100 Mbps channel an octets counter could roll over every 5.7 minutes.

These counters fall in the 0x0200-0x03FF address offset range, and are divided into RMON counters and IEEE counters, as follows:

- RMON counters are included which cover the Ethernet statistics counters defined in RFC 1757. In addition to the counters defined in the Ethernet statistics group, a counter is included to count truncated frames as the FEC only supports frame lengths up to 2047 bytes. The RMON counters are implemented independently for transmit and receive to insure accurate network statistics when operating in full-duplex mode.
- IEEE counters are included which support the mandatory and recommended counter packages defined in section 5 of ANSI/IEEE Std. 802.3 (1998 edition). The IEEE basic package objects are supported by the FEC but do not require counters in the MIB block. In addition, some of the recommended package objects which are supported do not require MIB counters. Counters for transmit and receive full-duplex flow control frames are included as well.

Table 32-14. MIB Counters Memory Map

Base Address Offset	Mnemonic	Description
0x0200	RMON_T_DROP	Count of frames not counted correctly
0x0204	RMON_T_PACKETS	RMON Tx packet count
0x0208	RMON_T_BC_PKT	RMON Tx broadcast packets
0x020C	RMON_T_MC_PKT	RMON Tx multicast packets
0x0210	RMON_T_CRC_ALIGN	RMON Tx packets w CRC/Align error
0x0214	RMON_T_UNDERSIZE	RMON Tx packets < 64 bytes, good crc
0x0218	RMON_T_OVERSIZE	RMON Tx packets > MAX_FL bytes, good crc
0x021C	RMON_T_FRAG	RMON Tx packets < 64 bytes, bad crc
0x0220	RMON_T_JAB	RMON Tx packets > MAX_FL bytes, bad crc
0x0224	RMON_T_COL	RMON Tx collision count
0x0228	RMON_T_P64	RMON Tx 64 byte packets
0x022C	RMON_T_P65TO127	RMON Tx 65 to 127 byte packets
0x0230	RMON_T_P128TO255	RMON Tx 128 to 255 byte packets
0x0234	RMON_T_P256TO511	RMON Tx 256 to 511 byte packets
0x0238	RMON_T_P512TO1023	RMON Tx 512 to 1023 byte packets
0x023C	RMON_T_P1024TO2047	RMON Tx 1024 to 2047 byte packets
0x0240	RMON_T_P_GTE2048	RMON Tx packets w > 2048 bytes
0x0244	RMON_T_OCTETS	RMON Tx octets
0x0248	IEEE_T_DROP	Count of frames not counted correctly
0x024C	IEEE_T_FRAME_OK	Frames transmitted OK
0x0250	IEEE_T_1COL	Frames transmitted with single collision
0x0254	IEEE_T_MCOL	Frames transmitted with multiple collisions
0x0258	IEEE_T_DEF	Frames transmitted after deferral delay
0x025C	IEEE_T_LCOL	Frames transmitted with late collision

Table continues on the next page...

Table 32-14. MIB Counters Memory Map (continued)

Base Address Offset	Mnemonic	Description
0x0260	IEEE_T_EXCOL	Frames transmitted with excessive collisions
0x0264	IEEE_T_MACERR	Frames transmitted with Tx FIFO underrun
0x0268	IEEE_T_CSERR	Frames transmitted with carrier sense error
0x026C	IEEE_T_SQE	Frames transmitted with SQE error
0x0270	IEEE_T_FDXFC	Flow control pause frames transmitted
0x0274	IEEE_T_OCTETS_OK	Octet count for frames transmitted w/o error
0x0284	RMON_R_PACKETS	RMON Rx packet count
0x0288	RMON_R_BC_PKT	RMON Rx broadcast packets
0x028C	RMON_R_MC_PKT	RMON Rx multicast packets
0x0290	RMON_R_CRC_ALIGN	RMON Rx packets w CRC/Align error
0x0294	RMON_R_UNDERSIZE	RMON Rx packets < 64 bytes, good crc
0x0298	RMON_R_OVERSIZE	RMON Rx packets > MAX_FL bytes, good crc
0x029C	RMON_R_FRAG	RMON Rx packets < 64 bytes, bad crc
0x02A0	RMON_R_JAB	RMON Rx packets > MAX_FL bytes, bad crc
0x02A4	RMON_R_RESVD_0	-
0x02A8	RMON_R_P64	RMON Rx 64 byte packets
0x02AC	RMON_R_P65TO127	RMON Rx 65 to 127 byte packets
0x02B0	RMON_R_P128TO255	RMON Rx 128 to 255 byte packets
0x02B4	RMON_R_P256TO511	RMON Rx 256 to 511 byte packets
0x02B8	RMON_R_P512TO1023	RMON Rx 512 to 1023 byte packets
0x02BC	RMON_R_P1024TO2047	RMON Rx 1024 to 2047 byte packets
0x02C0	RMON_R_P_GTE2048	RMON Rx packets w > 2048 bytes
0x02C4	RMON_R_OCTETS	RMON Rx octets
0x02C8	IEEE_R_DROP	Count of frames not counted correctly
0x02CC	IEEE_R_FRAME_OK	Frames received OK
0x02D0	IEEE_R_CRC	Frames received with CRC error
0x02D4	IEEE_R_ALIGN	Frames received with alignment error
0x02D8	IEEE_R_MACERR	Receive FIFO overflow count
0x02DC	IEEE_R_FDXFC	Flow control pause frames received
0x02E0	IEEE_R_OCTETS_OK	Octet count for frames received w/o error

32.5.3 MIIGSK Registers Memory Map

Table 32-15. MIIGSK Registers Memory Map

Offset	Mnemonic	Description
0x0300	FEC_MIIGSK_CFGR	MIIGSK Configuration Register
0x0308	FEC_MIIGSK_ENR	MIIGSK Enable Register

This section includes the FEC memory map and detailed descriptions of all the registers, followed by a description of the buffers.

The FEC is programmed by a combination of control and status registers (FEC_CSRs) and buffer descriptors. The FEC_CSRs are used for mode control and to extract global status information. The descriptors are used to pass data buffers and related buffer information between the hardware and software.

The FEC implementation requires a 1 Kbyte memory space. This space is divided into 2 sections of 512 bytes each. The first is used for control/status registers, and the second contains event/statistics counters held in the MIB block.

Table defines the top level memory map. For the base address of a particular block instantiation, see the system memory map.

Table 32-17. Block Memory Map

Base Address Offset	Function
0x0000-01FF	Control/Status Registers
0x0200-02FF	MIB Block Counters
0x0300-03FF	Extension Registers (MIIGSK)

The following table shows the FEC register memory map. For the base address of a particular block instantiation, see the system memory map.

FEC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FE_C004	Ethernet interrupt event register (FEC_EIR)	32	R/W	0000_0000h	32.5.4/ 1595
63FE_C008	Ethernet interrupt mask register (FEC_EIMR)	32	R/W	0000_0000h	32.5.5/ 1597

Table continues on the next page...

FEC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FE_C010	Receive descriptor active register (FEC_RDAR)	32	R/W	0000_0000h	32.5.6/ 1599
63FE_C014	Transmit descriptor active register (FEC_TDAR)	32	R/W	0000_0000h	32.5.7/ 1600
63FE_C024	Ethernet control register (FEC_ECR)	32	R/W	F000_0000h	32.5.8/ 1601
63FE_C040	MII management frame register (FEC_MMFR)	32	R/W	Unaffected	32.5.9/ 1602
63FE_C044	MII speed control register (FEC_MSCR)	32	R/W	0000_0000h	32.5.10/ 1604
63FE_C064	MIB control register (FEC_MIBC)	32	R/W	C000_0000h	32.5.11/ 1606
63FE_C084	Receive control register (FEC_RCR)	32	R/W	05EE_0001h	32.5.12/ 1607
63FE_C0C4	Transmit control register (FEC_TCR)	32	R/W	0000_0000h	32.5.13/ 1608
63FE_C0E4	Physical address low register (FEC_PALR)	32	R/W	0000_0000h	32.5.14/ 1609
63FE_C0E8	Physical address upper register (FEC_PAUR)	32	R/W	0000_8808h	32.5.15/ 1610
63FE_C0EC	Opcode and pause duration register (FEC_OPDR)	32	R/W	0001_0000h	32.5.16/ 1611
63FE_C118	Descriptor individual address upper register (FEC_IAUR)	32	R/W	0000_0000h	32.5.17/ 1611
63FE_C11C	Descriptor individual address lower register (FEC_IALR)	32	R/W	0000_0000h	32.5.18/ 1612
63FE_C120	Descriptor group address upper register (FEC_GAUR)	32	R/W	0000_0000h	32.5.19/ 1612
63FE_C124	Descriptor group address lower register (FEC_GALR)	32	R/W	0000_0000h	32.5.20/ 1613
63FE_C144	Transmit FIFO watermark register (FEC_TFWR)	32	R/W	0000_0000h	32.5.21/ 1613
63FE_C14C	FIFO receive bound register (FEC_FRBR)	32	R	0000_0600h	32.5.22/ 1614
63FE_C150	FIFO receive FIFO start registers (FEC_FRSR)	32	R/W	0000_0500h	32.5.23/ 1614
63FE_C180	Receive buffer descriptor ring start register (FEC_ERDSR)	32	R/W	Unaffected	32.5.24/ 1615
63FE_C184	Transmit buffer descriptor ring start register (FEC_ETDSR)	32	R/W	0000_0000h	32.5.25/ 1616

Table continues on the next page...

FEC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FE_C188	Maximum receive buffer size register (FEC_EMRBR)	32	R/W	0000_0000h	32.5.26/ 1616

32.5.4 Ethernet interrupt event register (FEC_EIR)

The FEC_EIR bit assignments are shown below. When an event occurs that sets a bit in the FEC_EIR, an interrupt is generated if the corresponding bit in the interrupt mask register (FEC_EIMR) is also set. The bit in the FEC_EIR is cleared if a one is written to that bit position; writing zero has no effect. This register is cleared upon hardware reset.

Interrupts can be divided into three classes as follows:

- Interrupts that occur in normal operation: these include GRA, TXF, TXB, RXF, RXB, and MII.
- Interrupts that result from errors or problems detected in the network or transceiver: these include HBERR, BABR, BABT, LC, and RL.
- Interrupts that result from internal errors are EBERR and UN.

Some of the error interrupts are independently counted in the MIB block counters. The correspondence between interrupts and counters is shown in the following table. Software can choose to mask off the interrupts because these errors are visible to network management through the MIB counters.

Table 32-19. Error Interrupts and Block Counters

Interrupt	Counter(s)
HBERR	IEEE_T_SQE
BABR	RMON_R_OVERSIZE (good CRC), RMON_R_JAB (bad CRC)
BABT	RMON_T_OVERSIZE (good CRC), RMON_T_JAB (bad CRC)
LATE_COL	IEEE_T_LCOL
COL_RETRY_LIM	IEEE_T_EXCOL
XFIFO_UN	IEEE_T_MACERR

Address: FEC_EIR is 63FE_C000h base + 4h offset = 63FE_C004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HBERR	BABR	BABT	GRA	TXF	TXB	RXF	RXB	MII	EBERR	LC	RL	UN	-																		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FEC_EIR field descriptions

Field	Description
31 HBERR	Heartbeat error. This interrupt indicates that HBC is set in the FEC_TCR register and that the COL input was not asserted within the Heartbeat window following a transmission.
30 BABR	Babbling receive error. This bit indicates a frame was received with length in excess of FEC_RCR[MAX_FL] bytes.
29 BABT	Babbling transmit error. This bit indicates that the transmitted frame length has exceeded FEC_RCR[MAX_FL] bytes. This condition is usually caused by a frame that is too long being placed into the transmit data buffer(s). Truncation does not occur.
28 GRA	Graceful stop complete. This interrupt is asserted for one of three reasons. Graceful stop means that the transmitter is put into a pause state after completion of the frame currently being transmitted. 1) A graceful stop, which was initiated by the setting of the FEC_TCR[GTS] bit is now complete. 2) A graceful stop, which was initiated by the setting of the FEC_TCR[TFC_PAUSE] bit is now complete. 3) A graceful stop, which was initiated by the reception of a valid full-duplex flow control "pause" frame is now complete. See the "Full-Duplex Flow Control" section of the Functional Description chapter.
27 TXF	Transmit frame interrupt. This bit indicates that a frame has been transmitted and that the last corresponding buffer descriptor has been updated.
26 TXB	Transmit buffer interrupt. This bit indicates that a transmit buffer descriptor has been updated.
25 RXF	Receive frame interrupt. This bit indicates that a frame has been received and that the last corresponding buffer descriptor has been updated.
24 RXB	Receive buffer interrupt. This bit indicates that a receive buffer descriptor has been updated that was not the last in the frame.
23 MII	MII interrupt. This bit indicates that the MII has completed the data transfer requested.
22 EBERR	Ethernet bus error. This bit indicates that a system bus error occurred when a DMA transaction was underway. When the EBERR bit is set, FEC_ECR[ETHER_EN] is cleared, halting frame processing by the FEC_ When this occurs software needs to insure that the FIFO controller and DMA are also soft reset.
21 LC	Late collision. This bit indicates that a collision occurred beyond the collision window (slot time) in half-duplex mode. The frame is truncated with a bad CRC and the remainder of the frame is discarded.
20 RL	Collision retry limit. This bit indicates that a collision occurred on each of 16 successive attempts to transmit the frame. The frame is discarded without being transmitted and transmission of the next frame commences. Can only occur in half-duplex mode.
19 UN	Transmit FIFO underrun. This bit indicates that the transmit FIFO became empty before the complete frame was transmitted. A bad CRC is appended to the frame fragment and the remainder of the frame is discarded.
18–0 -	Reserved, read as 0

32.5.5 Ethernet interrupt mask register (FEC_EIMR)

The FEC_EIMR controls which of the interrupt events flagged in the FEC_EIR are allowed to generate actual interrupts. If the corresponding bits in both the FEC_EIR and FEC_EIMR are set, the interrupt is signaled to the ARM platform. The interrupt signal remains asserted until a 1 is written to the FEC_EIR bit (write 1 to clear) or a 0 is written to the FEC_EIMR bit. This register is cleared upon a hardware reset.

The FEC_EIMR bit assignments are the same as for the FEC_EIR.

Address: FEC_EIMR is 63FE_C000h base + 8h offset = 63FE_C008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HBERR	BABR	BABT	GRA	TXF	TXB	RXF	RXB	MII	EBERR	LC	RL	UN	0																		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FEC_EIMR field descriptions

Field	Description
31 HBERR	Interrupt mask. Each bit corresponds to an interrupt source defined by the FEC_EIR register. The corresponding FEC_EIMR bit determines whether an interrupt condition can generate an interrupt. At every system clock, the FEC_EIR samples the signal generated by the interrupting source. The corresponding FEC_EIR bit reflects the state of the interrupt signal even if the corresponding FEC_EIMR bit is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
30 BABR	Interrupt mask. Each bit corresponds to an interrupt source defined by the FEC_EIR register. The corresponding FEC_EIMR bit determines whether an interrupt condition can generate an interrupt. At every system clock, the FEC_EIR samples the signal generated by the interrupting source. The corresponding FEC_EIR bit reflects the state of the interrupt signal even if the corresponding FEC_EIMR bit is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
29 BABT	Interrupt mask. Each bit corresponds to an interrupt source defined by the FEC_EIR register. The corresponding FEC_EIMR bit determines whether an interrupt condition can generate an interrupt. At every system clock, the FEC_EIR samples the signal generated by the interrupting source. The corresponding FEC_EIR bit reflects the state of the interrupt signal even if the corresponding FEC_EIMR bit is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
28 GRA	Interrupt mask. Each bit corresponds to an interrupt source defined by the FEC_EIR register. The corresponding FEC_EIMR bit determines whether an interrupt condition can generate an interrupt. At every system clock, the FEC_EIR samples the signal generated by the interrupting source. The corresponding FEC_EIR bit reflects the state of the interrupt signal even if the corresponding FEC_EIMR bit is cleared.

Table continues on the next page...

FEC_EIMR field descriptions (continued)

Field	Description
	0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
27 TXF	Interrupt mask. Each bit corresponds to an interrupt source defined by the FEC_EIR register. The corresponding FEC_EIMR bit determines whether an interrupt condition can generate an interrupt. At every system clock, the FEC_EIR samples the signal generated by the interrupting source. The corresponding FEC_EIR bit reflects the state of the interrupt signal even if the corresponding FEC_EIMR bit is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
26 TXB	Interrupt mask. Each bit corresponds to an interrupt source defined by the FEC_EIR register. The corresponding FEC_EIMR bit determines whether an interrupt condition can generate an interrupt. At every system clock, the FEC_EIR samples the signal generated by the interrupting source. The corresponding FEC_EIR bit reflects the state of the interrupt signal even if the corresponding FEC_EIMR bit is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
25 RXF	Interrupt mask. Each bit corresponds to an interrupt source defined by the FEC_EIR register. The corresponding FEC_EIMR bit determines whether an interrupt condition can generate an interrupt. At every system clock, the FEC_EIR samples the signal generated by the interrupting source. The corresponding FEC_EIR bit reflects the state of the interrupt signal even if the corresponding FEC_EIMR bit is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
24 RXB	Interrupt mask. Each bit corresponds to an interrupt source defined by the FEC_EIR register. The corresponding FEC_EIMR bit determines whether an interrupt condition can generate an interrupt. At every system clock, the FEC_EIR samples the signal generated by the interrupting source. The corresponding FEC_EIR bit reflects the state of the interrupt signal even if the corresponding FEC_EIMR bit is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
23 MII	Interrupt mask. Each bit corresponds to an interrupt source defined by the FEC_EIR register. The corresponding FEC_EIMR bit determines whether an interrupt condition can generate an interrupt. At every system clock, the FEC_EIR samples the signal generated by the interrupting source. The corresponding FEC_EIR bit reflects the state of the interrupt signal even if the corresponding FEC_EIMR bit is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
22 EBERR	Interrupt mask. Each bit corresponds to an interrupt source defined by the FEC_EIR register. The corresponding FEC_EIMR bit determines whether an interrupt condition can generate an interrupt. At every system clock, the FEC_EIR samples the signal generated by the interrupting source. The corresponding FEC_EIR bit reflects the state of the interrupt signal even if the corresponding FEC_EIMR bit is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.

Table continues on the next page...

FEC_EIMR field descriptions (continued)

Field	Description
21 LC	Interrupt mask. Each bit corresponds to an interrupt source defined by the FEC_EIR register. The corresponding FEC_EIMR bit determines whether an interrupt condition can generate an interrupt. At every system clock, the FEC_EIR samples the signal generated by the interrupting source. The corresponding FEC_EIR bit reflects the state of the interrupt signal even if the corresponding FEC_EIMR bit is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
20 RL	Interrupt mask. Each bit corresponds to an interrupt source defined by the FEC_EIR register. The corresponding FEC_EIMR bit determines whether an interrupt condition can generate an interrupt. At every system clock, the FEC_EIR samples the signal generated by the interrupting source. The corresponding FEC_EIR bit reflects the state of the interrupt signal even if the corresponding FEC_EIMR bit is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
19 UN	Interrupt mask. Each bit corresponds to an interrupt source defined by the FEC_EIR register. The corresponding FEC_EIMR bit determines whether an interrupt condition can generate an interrupt. At every system clock, the FEC_EIR samples the signal generated by the interrupting source. The corresponding FEC_EIR bit reflects the state of the interrupt signal even if the corresponding FEC_EIMR bit is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
18–0 Reserved	This read-only field is reserved and always has the value zero. Reserved, read as 0

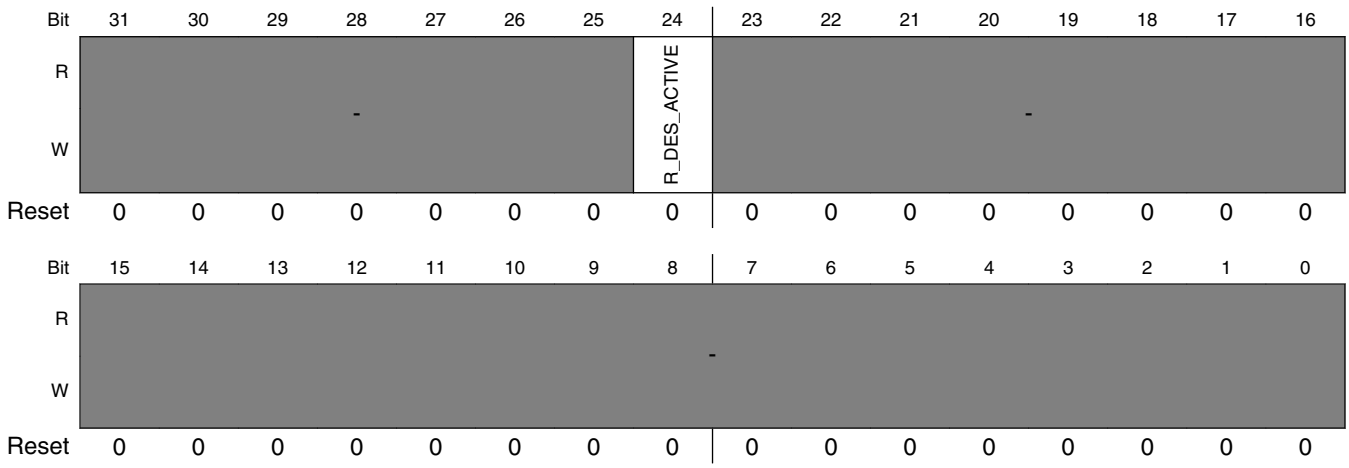
32.5.6 Receive descriptor active register (FEC_RDAR)

FEC_RDAR is a user-writeable command register which indicates that the receive descriptor ring has been updated, and that empty receive buffers have been produced by the driver with the empty bit set.

The FEC_RDAR[R_DES_ACTIVE] bit is set whenever the register is written, independent of the data actually written by the user. When set, the FEC polls the receive descriptor ring and processes receive frames (provided FEC_ECR[ETHER_EN] is also set to 1). After the FEC polls a receive descriptor whose empty bit is not set, then the FEC clears the FEC_RDAR[R_DES_ACTIVE] bit and ceases receive descriptor ring polling until the user sets the bit again, signifying that additional descriptors have been placed into the receive descriptor ring.

The FEC_RDAR is cleared at reset, and when FEC_ECR[ETHER_EN] is cleared.

Address: FEC_RDAR is 63FE_C000h base + 10h offset = 63FE_C010h



FEC_RDAR field descriptions

Field	Description
31–25 -	Reserved, read as 0
24 R_DES_ACTIVE	Set to one when this register is written, regardless of the value written. Cleared by the FEC device when the FEC polls a receive descriptor whose empty bit is not set. Also cleared when FEC_ECR[ETHER_EN] is cleared.
23–0 -	Reserved, read as 0

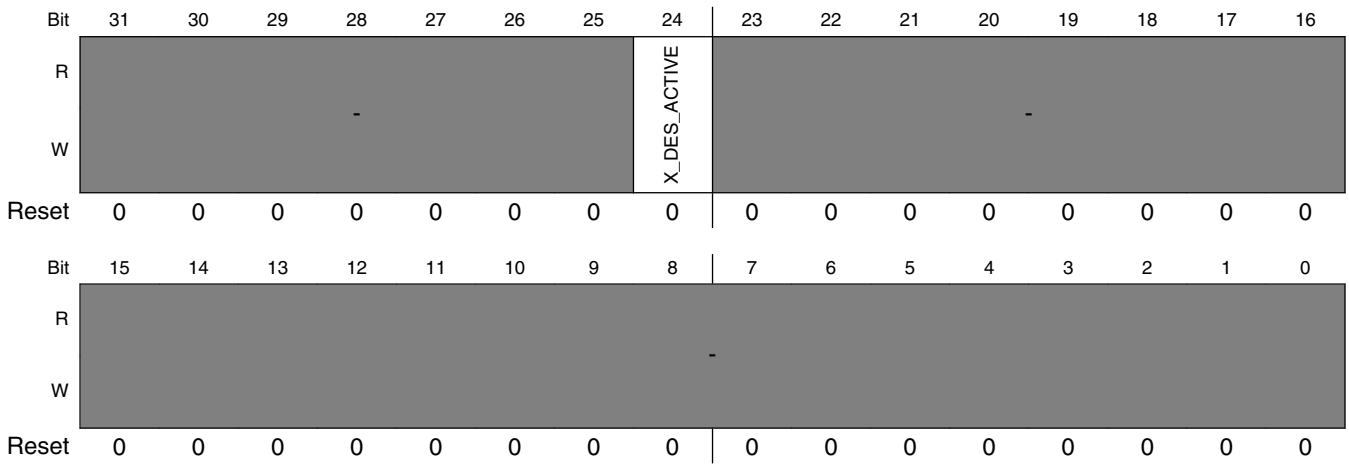
32.5.7 Transmit descriptor active register (FEC_TDAR)

The FEC_TDAR is a command register, written to by the user, to indicate that the transmit descriptor ring has been updated (transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor).

Whenever the register is written the FEC_TDAR[X_DES_ACTIVE] bit is set to 1, independent of the data actually written by the user. When set, the FEC polls the transmit descriptor ring and process transmit frames (provided FEC_ECR[ETHER_EN] is also set to 1). After the FEC polls a transmit descriptor whose ready bit is not set, then the FEC clears the FEC_TDAR[X_DES_ACTIVE] bit and ceases transmit descriptor ring polling until the user sets the bit again, signifying additional descriptors have been placed into the transmit descriptor ring.

The FEC_TDAR is cleared at reset, when FEC_ECR[ETHER_EN] is cleared, or when FEC_ECR[RESET] is set to 1.

Address: FEC_TDAR is 63FE_C000h base + 14h offset = 63FE_C014h



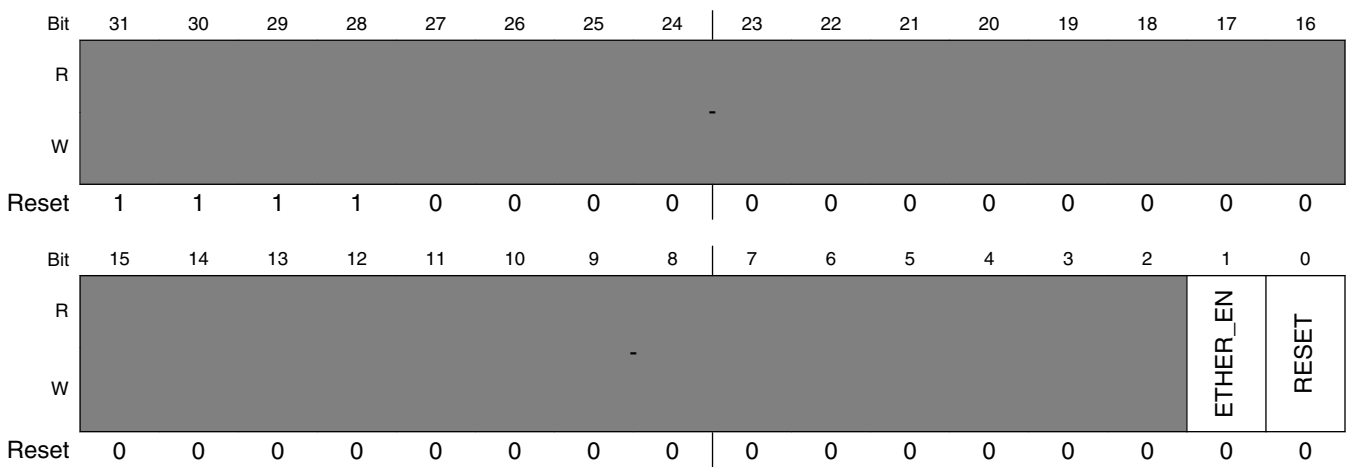
FEC_TDAR field descriptions

Field	Description
31–25 -	Reserved, read as 0
24 X_DES_ACTIVE	Set to one when this register is written, regardless of the value written. Cleared by the FEC device when the FEC polls a transmit descriptor whose ready bit is not set. Also cleared when FEC_ECR[ETHER_EN] is cleared.
23–0 -	Reserved, read as 0

32.5.8 Ethernet control register (FEC_ECR)

The ECR is used to enable/disable the FEC_ ECR is a read/write user register, though both fields in this register can also be altered by hardware.

Address: FEC_ECR is 63FE_C000h base + 24h offset = 63FE_C024h



FEC_ECR field descriptions

Field	Description
31–2 -	Reserved.
1 ETHER_EN	When this bit is set, the FEC is enabled, and reception and transmission are possible. When this bit is cleared, reception is immediately stopped and transmission is stopped after a bad CRC is appended to any currently transmitted frame. The buffer descriptor(s) for an aborted transmit frame are not updated after clearing this bit. When ETHER_EN is cleared, the DMA, buffer descriptor, and FIFO control logic are reset, including the buffer descriptor and FIFO pointers. The ETHER_EN bit is altered by hardware under the following conditions: FEC_ECR[RESET] is set by software, in which case ETHER_EN is cleared an error condition causes the FEC_EIR[EBERR] bit to set, in which case ETHER_EN is cleared
0 RESET	When this bit is set, the equivalent of a hardware reset is performed but it is local to the FEC_ETHER_EN is cleared and all other FEC registers take their reset values. Also, any transmission/reception currently in progress is abruptly aborted. This bit is automatically cleared by hardware during the reset sequence. The reset sequence takes approximately 8 system clock cycles after RESET is written with a 1.

32.5.9 MII management frame register (FEC_MMFR)

The FEC_MMFR is used to communicate with the attached MII compatible PHY device(s), providing read/write access to their MII registers. Performing a write to FEC_MMFR causes a management frame to be generated unless the MII-SPEED field of the FEC_MSCR has been set to 0, in which case FEC_MSCR is set to a non-zero value and an MII frame is generated with the data previously written to FEC_MMFR. This allows FEC_MMFR and FEC_MSCR to be programmed in either order if the MII-SPEED field of FEC_MSCR is zero (for further details see MII speed control register (FEC_MSCR)).

The FEC_MMFR does not reset to a definite value.

To perform a read or write operation on the MII Management Interface, the FEC_MMFR must be written to by the user. To generate a valid read or write management frame, the ST field must be written with a 01 pattern, and the TA field must be written with a 10. If other patterns are written to these fields, a frame is generated but does not comply with the IEEE 802.3 MII definition.

To generate an IEEE 802.3-compliant MII management interface write frame (write to a PHY register), the user must write {01 01 PHYAD REGAD 10 DATA} to the bit fields of the FEC_MMFR, as shown in MII management frame register (FEC_MMFR). Writing this pattern causes the control logic to shift out the data in the FEC_MMFR following a preamble generated by the control state machine. During this time the contents of the FEC_MMFR is altered as the contents are serially shifted and is unpredictable if read by

the user. After the write management frame operation has completed, the MII interrupt is generated. At this time the contents of the FEC_MMFR matches the original value written.

To generate an MII management interface read frame (read a PHY register) the user must write {01 10 PHYAD REGAD 10 XXXX} to the bit fields of the FEC_MMFR shown in MII management frame register (FEC_MMFR) (the contents of the 4-bit DATA field are arbitrary). Writing this pattern causes the control logic to shift out the data in the FEC_MMFR following a preamble generated by the control state machine. During this time the contents of the FEC_MMFR is altered as the contents are serially shifted, and is unpredictable if read by the user. After the read management frame operation has completed, the MII interrupt is generated. At this time the contents of the FEC_MMFR matches the original value written except for the DATA field whose contents have been replaced by the value read from the PHY register.

If the FEC_MMFR is written to while frame generation is in progress, the frame contents is altered. Software uses the MII interrupt to avoid writing to the FEC_MMFR while frame generation is in progress.

Address: FEC_MMFR is 63FE_C000h base + 40h offset = 63FE_C040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ST		OP		PA					RA					TA		DATA															
Reset	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*

* Notes:
- u = Unaffected by reset.

FEC_MMFR field descriptions

Field	Description
31–30 ST	Start of frame delimiter. These bits must be programmed to 01 for a valid MII management frame.
29–28 OP	Operation code. This field must be programmed to 10 (read) or 01 (write) to generate a valid MII management frame. A value of 11 produces "read" frame operation while a value of 00 produces "write" frame operation, but these frames is not MII compliant.
27–23 PA	PHY address. This field specifies one of up to 32 attached PHY devices.
22–18 RA	Register address. This field specifies one of up to 32 registers within the specified PHY device.
17–16 TA	Turn around. This field must be programmed to 10 to generate a valid MII management frame.
15–0 DATA	Management frame data. This is the field for data to be written to or read from the PHY register.

32.5.10 MII speed control register (FEC_MSCR)

The FEC_MSCR provides control of the frequency of the MII clock (FEC_MDC signal), and allows a preamble drop on the MII management frame.

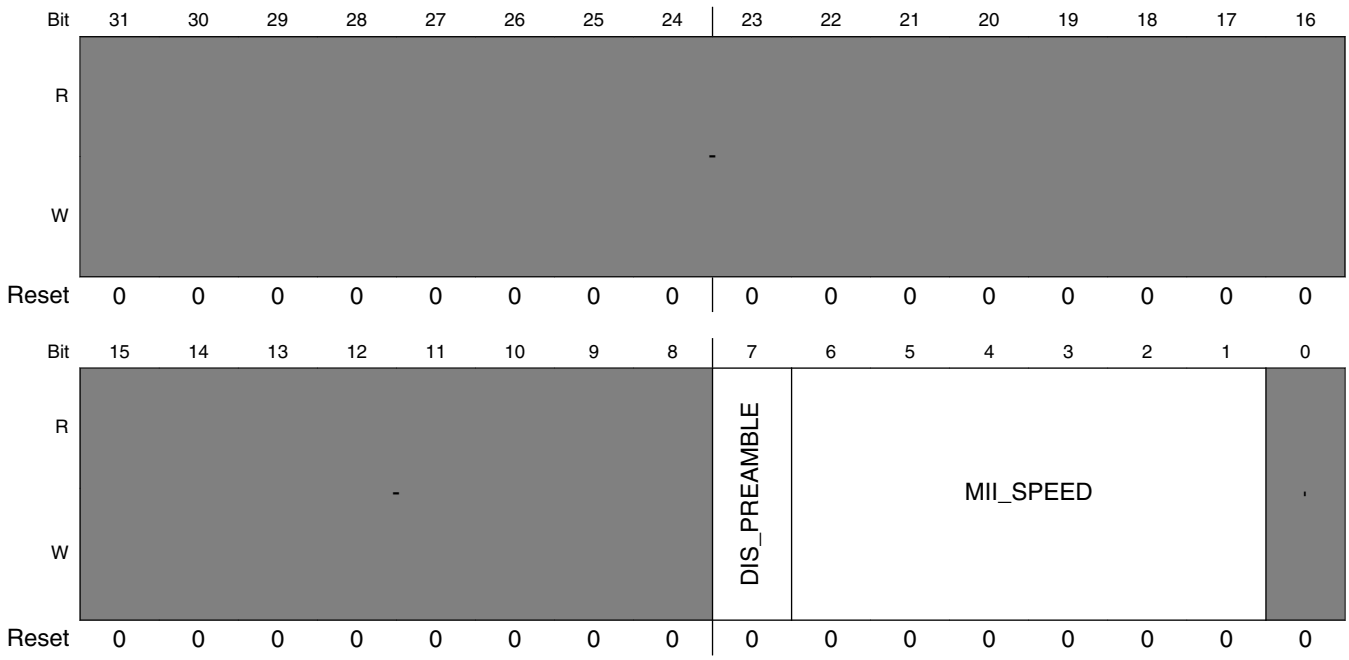
The MII_SPEED field must be programmed with a value to provide an FEC_MDC frequency of less than or equal to 2.5 MHz to be compliant with the IEEE 802.3 MII specification. The MII_SPEED must be set to a non-zero value in order to generate a read or write management frame. After the management frame is complete the FEC_MSCR can optionally be set to zero to turn off the FEC_MDC. The FEC_MDC generated has a 50% duty cycle except when MII_SPEED is changed during operation (change takes effect following either a rising or falling edge of FEC_MDC).

The FEC_MDC frequency depends on both the system clock frequency and the MII_SPEED register. If the system clock is 25 MHz, programming the MII_SPEED register to 0x0000_0005 results in an FEC_MDC frequency of 25 MHz * 1/10 = 2.5 MHz. A table showing optimum values for MII_SPEED for different system clock frequencies is provided below.

Table 32-26. Programming Examples for FEC_MSCR

System Clock Frequency	MII_SPEED (field in reg)	FEC_MDC frequency
25 MHz	0x5	2.5 MHz
33 MHz	0x7	2.36 MHz
40 MHz	0x8	2.5 MHz
50 MHz	0xA	2.5 MHz
66 MHz	0xD	2.54 MHz

Address: FEC_MSCR is 63FE_C000h base + 44h offset = 63FE_C044h



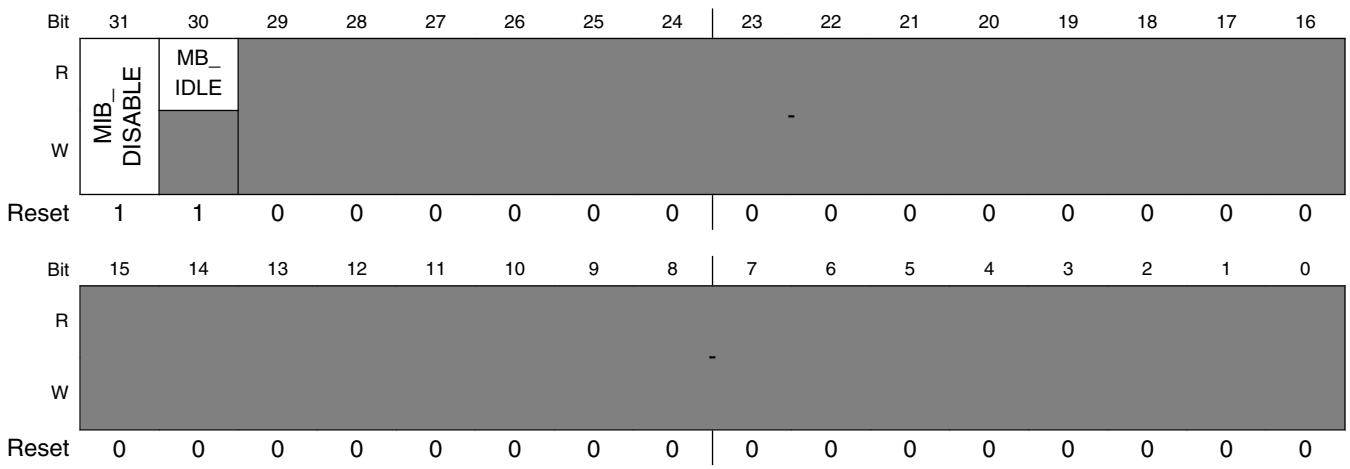
FEC_MSCR field descriptions

Field	Description
31–8 -	Reserved, read as 0
7 DIS_PREAMBLE	Asserting this bit causes preamble (0xFFFF_FFFF) not to be prepended to the MII management frame. The MII standard allows the preamble to be dropped if the attached PHY device(s) does not require it.
6–1 MII_SPEED	MII_SPEED controls the frequency of the MII management interface clock (FEC_MDC) relative to the system clock. A value of 0 in this field "turns off" the FEC_MDC and leave it in low voltage state. Any non-zero value results in the FEC_MDC frequency of 1/(MII_SPEED*2) of the system clock frequency.
0 -	Reserved, read as 0

32.5.11 MIB control register (FEC_MIBC)

The MIB control register is a read/write register used to provide control of and to observe the state of the Message Information Block (MIB). This register is accessed by user software if there is a need to disable the MIB operation. For example, in order to clear all MIB counters in RAM the user disables the MIB, then clears all the MIB RAM locations, then enables the MIB. The MIB_DISABLE bit is reset to 1. See MIB Counters Memory Map for the locations of the MIB counters.

Address: FEC_MIBC is 63FE_C000h base + 64h offset = 63FE_C064h



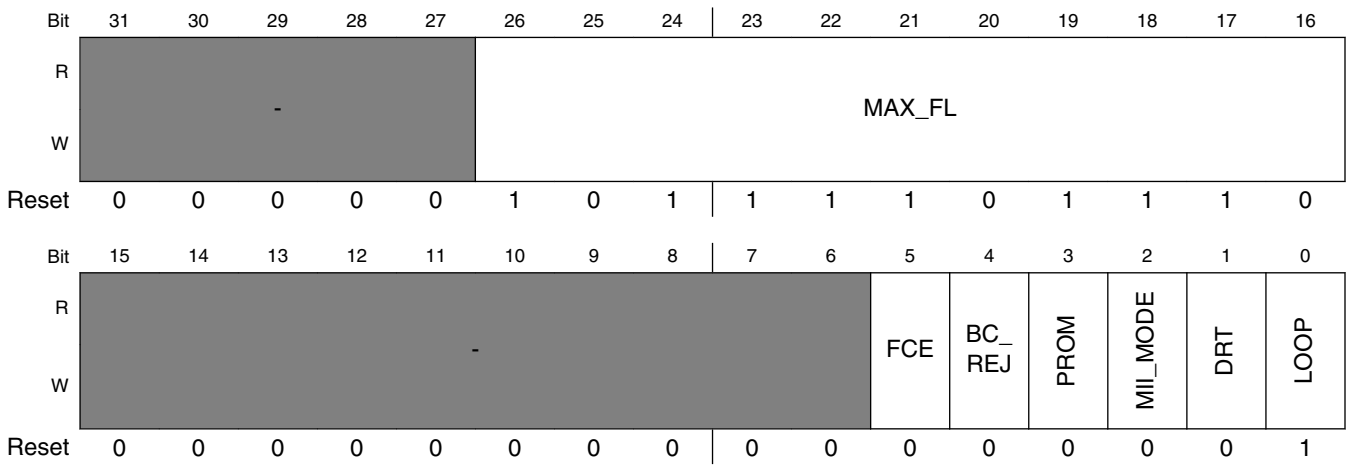
FEC_MIBC field descriptions

Field	Description
31 MIB_DISABLE	A read/write control bit. If set, the MIB logic halts and not update any MIB counters.
30 MB_IDLE	A read-only status bit. If set the MIB block is not currently updating any MIB counters.
29–0 -	Reserved.

32.5.12 Receive control register (FEC_RCR)

The FEC_RCR is programmed by the user, and controls the operational mode of the receive block. It can only be written to when FEC_ECR[ETHER_EN] = 0 (that is, during initialization).

Address: FEC_RCR is 63FE_C000h base + 84h offset = 63FE_C084h



FEC_RCR field descriptions

Field	Description
31–27 -	Reserved, read as 0
26–16 MAX_FL	Maximum frame length. Resets to decimal 1518. Length is measured starting at DA and includes the CRC at the end of the frame. Transmit frames longer than MAX_FL causes the BABT interrupt to occur. Receive Frames longer than MAX_FL causes the BABR interrupt to occur and sets the LG bit in the end of frame receive buffer descriptor. The recommended default value to be programmed by the user is 1518 or 1522 (if VLAN Tags are supported).
15–6 -	Reserved, read as 0
5 FCE	Flow control enable. When FCE is set to 1, the receiver detects pause frames. Upon pause frame detection, the transmitter stops transmitting data frames for a given duration.
4 BC_REJ	Broadcast frame reject. When BC_REJ is set to 1, frames with DA (destination address) = 0xFF_FF_FF_FF_FF_FF are rejected unless the PROM bit is set to 1. If both BC_REJ and PROM are set to 1, then frames with broadcast DA is accepted and the M (MISS) bit is set in the receive buffer descriptor.
3 PROM	Promiscuous mode. All frames are accepted regardless of address matching.
2 MII_MODE	Media independent interface mode. Selects external interface mode. Setting this bit to one selects MII mode, setting this bit equal to zero selects 7-wire mode (used only for serial 10 Mbps). This bit controls the interface mode for both transmit and receive blocks.
1 DRT	Disable receive on transmit.

Table continues on the next page...

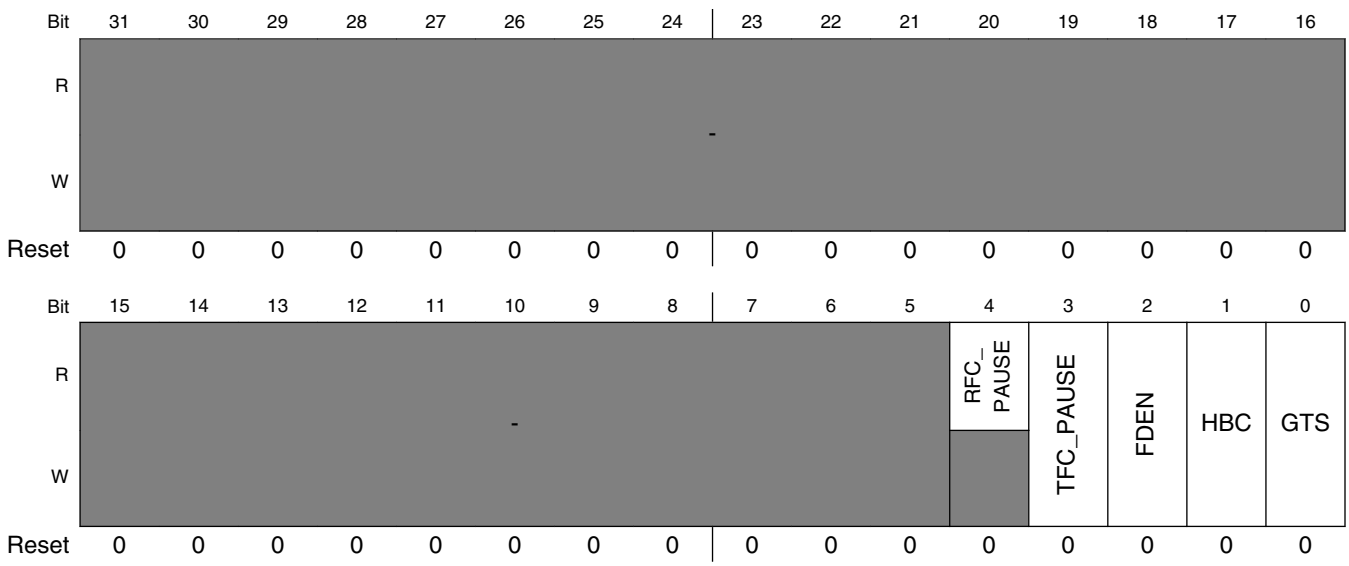
FEC_RCR field descriptions (continued)

Field	Description
	0 Receive path operates independently of transmit (use for full-duplex or to monitor transmit activity in half-duplex mode). 1 Disable reception of frames while transmitting (normally used for half-duplex mode).
0 LOOP	Internal loopback.When LOOP is set to 1, transmitted frames are looped back internal to the device and the transmit output signals are not asserted. The system clock is substituted for the FEC_TX_CLK when LOOP is set to 1. DRT must be set to zero when setting LOOP to 1.

32.5.13 Transmit control register (FEC_TCR)

This register is read/write, and is written by the user to configure the transmit block. Bits [2:1] must only be modified when FEC_ECR[ETHER_EN] = 0 (that is, during initialization). This register is cleared at system reset.

Address: FEC_TCR is 63FE_C000h base + C4h offset = 63FE_C0C4h



FEC_TCR field descriptions

Field	Description
31–5 -	Reserved read as 0
4 RFC_PAUSE	Receive frame control pause. This read-only status bit is set to 1 when a full-duplex flow control pause frame has been received and the transmitter is paused for the duration defined in this pause frame. This bit automatically clears when the pause duration is complete. 0 Transmitter is not paused 1 Transmitter is paused after reception of full-duplex flow control pause frame
3 TFC_PAUSE	Transmit frame control pause. When this bit is set to 1, a pause frame is transmitted according to the following steps:

Table continues on the next page...

FEC_TCR field descriptions (continued)

Field	Description
	1. FEC stops transmission of data frames after the current transmission is complete. 2. The GRA interrupt in the FEC_EIR register is asserted. 3. With transmission of data frames stopped, the FEC transmits a MAC control pause frame. 4. The FEC clears the TFC_PAUSE bit and resume transmitting data frames. The FEC can still transmit a MAC control pause frame when the transmitter is paused due to user assertion of GTS or reception of a pause frame. 0 No pause frame is transmitted 1 Pause frame is transmitted
2 FDEN	Full duplex enable. When FDEN is set to 1, frames are transmitted independent of carrier sense and collision inputs. This bit must only be modified when ETHER_EN is cleared. 0 Full duplex is not enabled 1 Full duplex is enabled
1 HBC	Heartbeat control. When HBC is set to 1, the heartbeat check is performed after end of transmission and the HB bit in the status register is set if the collision input does not assert within the heartbeat window. This bit must only be modified when ETHER_EN is cleared. 0 Heartbeat check is not performed after end of transmission 1 Heartbeat check is performed after end of transmission
0 GTS	Graceful transmit stop. When GTS is set to 1, the FEC stops transmission after any frame that is currently being transmitted is complete and the GRA interrupt in the FEC_EIR register is asserted. If frame transmission is not currently underway, the GRA interrupt is asserted immediately. After transmission has completed, a "restart" can be accomplished by clearing the GTS bit. The next frame in the transmit FIFO is then transmitted. If an early collision occurs during transmission when GTS = 1, transmission stops after the collision. The frame is transmitted again after GTS is cleared. There can be old frames in the transmit FIFO that is transmitted when GTS is reasserted. To avoid this, clear FEC_ECR[ETHER_EN] following the GRA interrupt. 0 Graceful transmit stop is not enabled 1 Graceful transmit stop is enabled.

32.5.14 Physical address low register (FEC_PALR)

The FEC_PALR is written by the user, and contains the lower 32 bits (bytes 0,1,2,3) of the 48-bit address used in the address recognition process to check for possible match between the DA field of receive frames and an individual DA. This register is also used for bytes 0 through 3 of the 6-byte source address field when transmitting pause frames. This register is unaffected by reset and must be initialized by the user.

Address: FEC_PALR is 63FE_C000h base + E4h offset = 63FE_C0E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PADDR1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						

FEC_PALR field descriptions

Field	Description
31–0 PADDR1	Bytes 0 (bits 31:24), 1 (bits 23:16), 2 (bits 15:8) and 3 (bits 7:0) of the 6-byte individual address to be used for exact match, and the source address field in pause frames.

32.5.15 Physical address upper register (FEC_PAUR)

The FEC_PAUR is written by the user, and contains the upper 16 bits (bytes 4 and 5) of the 48-bit address used in the address recognition process to check for possible match between the DA field of receive frames and an individual DA. In addition, this register is used in bytes 4 and 5 of the 6-byte source address field when transmitting pause frames. Bits 15:0 of FEC_PAUR contain a constant type field (0x8808) used for transmission of pause frames. This register is unaffected by reset, and bits 31:16 must be initialized by the user.

Address: FEC_PAUR is 63FE_C000h base + E8h offset = 63FE_C0E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PADDR2																TYPE															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

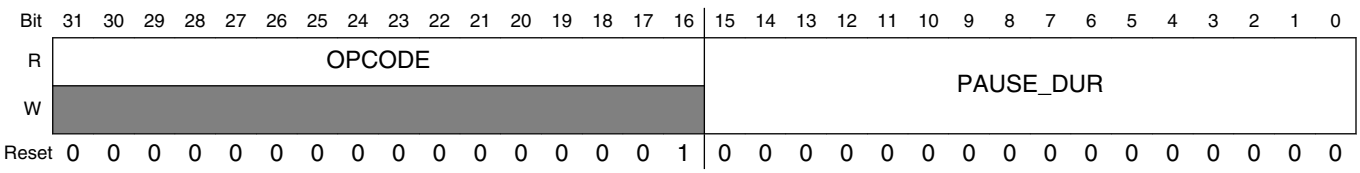
FEC_PAUR field descriptions

Field	Description
31–16 PADDR2	Bytes 4 (bits 31:24) and 5 (bits 23:16) of the 6-byte individual address to be used for exact match, and the source address field in pause frames.
15–0 TYPE	Type field in pause frames. This field has a constant value of 0x8808.

32.5.16 Opcode and pause duration register (FEC_OPDR)

FEC_OPDR contains the 16-bit Opcode, and 16-bit pause duration fields used in transmission of a pause frame. The Opcode field is a constant value, 0x0001. When another node detects a pause frame, that node pauses transmission for the duration specified in the pause duration field. This register is not reset and must be initialized by the user.

Address: FEC_OPDR is 63FE_C000h base + ECh offset = 63FE_C0ECh



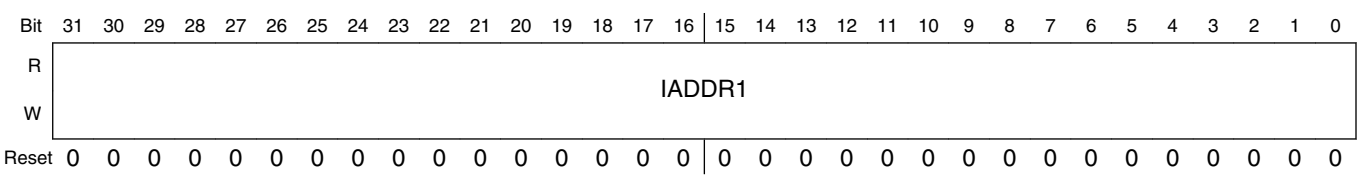
FEC_OPDR field descriptions

Field	Description
31–16 OPCODE	Opcode field used in pause frames. These bits are a constant, 0x0001.
15–0 PAUSE_DUR	Pause duration field used in pause frames.

32.5.17 Descriptor individual address upper register (FEC_IAUR)

The FEC_IAUR is written by the user, and contains the upper 32 bits of the 64-bit individual address hash table used in the address recognition process to check for possible match between the DA field of receive frames and an individual DA. This register is unaffected by reset, and must be initialized by the user.

Address: FEC_IAUR is 63FE_C000h base + 118h offset = 63FE_C118h



FEC_IAUR field descriptions

Field	Description
31–0 IADDR1	The upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR1 contains hash index bit 63. Bit 0 of IADDR1 contains hash index bit 32.

32.5.18 Descriptor individual address lower register (FEC_IALR)

The FEC_IALR is written by the user, and contains the lower 32 bits of the 64-bit individual address hash table used in the address recognition process to check for possible match with the DA field of receive frames with an individual DA. This register is unaffected by reset, and must be initialized by the user.

Address: FEC_IALR is 63FE_C000h base + 11Ch offset = 63FE_C11Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IADDR2																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FEC_IALR field descriptions

Field	Description
31–0 IADDR2	The lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR2 contains hash index bit 31. Bit 0 of IADDR2 contains hash index bit 0.

32.5.19 Descriptor group address upper register (FEC_GAUR)

The FEC_GAUR is written by the user, and contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. This register must be initialized by the user.

Address: FEC_GAUR is 63FE_C000h base + 120h offset = 63FE_C120h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GADDR1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

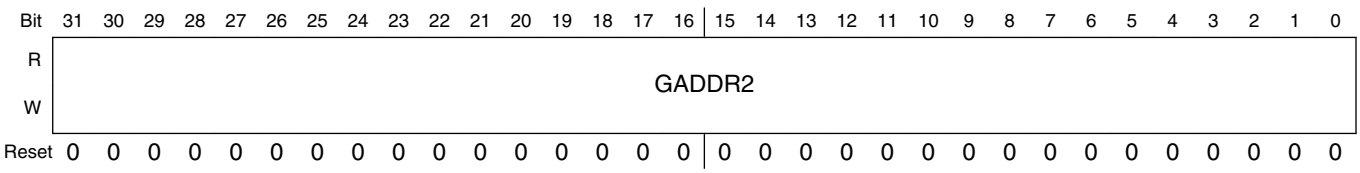
FEC_GAUR field descriptions

Field	Description
31–0 GADDR1	The GADDR1 register contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR1 contains hash index bit 63. Bit 0 of GADDR1 contains hash index bit 32.

32.5.20 Descriptor group address lower register (FEC_GALR)

The FEC_GALR is written by the user, and contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. This register must be initialized by the user.

Address: FEC_GALR is 63FE_C000h base + 124h offset = 63FE_C124h



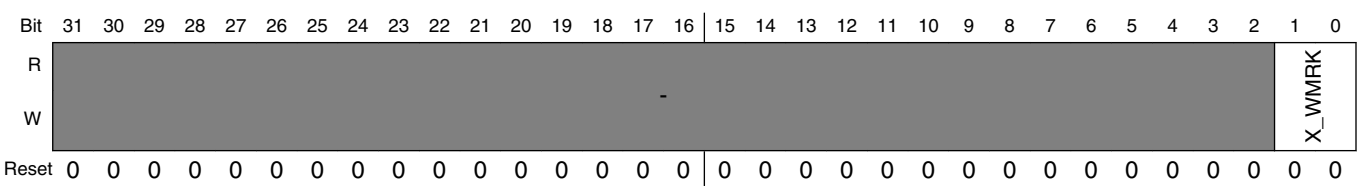
FEC_GALR field descriptions

Field	Description
31–0 GADDR2	The GADDR2 register contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR2 contains hash index bit 31. Bit 0 of GADDR2 contains hash index bit 0.

32.5.21 Transmit FIFO watermark register (FEC_TFWR)

The FEC_TFWR is programmed by the user to control the amount of data required in the transmit FIFO before transmission of a frame can begin. This allows the user to minimize transmit latency (FEC_TFWR[1:0] = 0n) or allow for larger bus access latency (FEC_TFWR[1:0] = 11) due to contention for the system bus. Setting the watermark to a high value minimizes the risk of transmit FIFO underrun due to contention for the system bus. In some use cases the byte counts associated with the FEC_TFWR field need to be modified to match system requirements, such as the worst-case bus access latency by the transmit data DMA channel.

Address: FEC_TFWR is 63FE_C000h base + 144h offset = 63FE_C144h



FEC_TFWR field descriptions

Field	Description
31–2 -	Reserved, read as 0

Table continues on the next page...

FEC_TFWR field descriptions (continued)

Field	Description
1–0 X_WMRK	Number of bytes written to transmit FIFO before transmission of a frame begins 0x 64 bytes written 10 128 bytes written 11 192 bytes written

32.5.22 FIFO receive bound register (FEC_FRBR)

The FEC_FRBR register can be read to determine the upper address bound of the FIFO RAM. Drivers can use this value, along with the FEC_FRSR to appropriately divide the available FIFO RAM between the transmit and receive data paths.

Address: FEC_FRBR is 63FE_C000h base + 14Ch offset = 63FE_C14Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																						R_BOUND								0	
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0

FEC_FRBR field descriptions

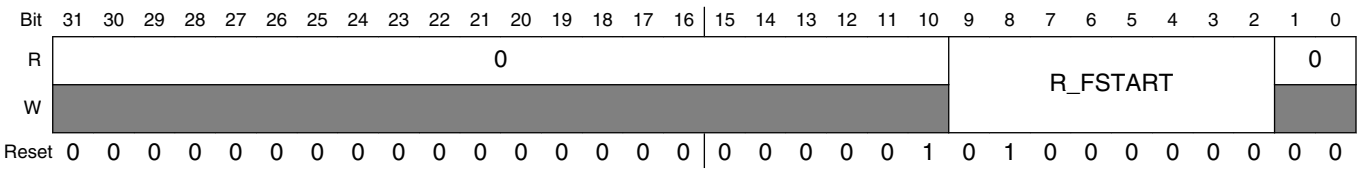
Field	Description
31–10 Reserved	This read-only field is reserved and always has the value zero. Reserved, read as 0 (except bit 10, which is read as 1).
9–2 R_BOUND	Read-only. Highest valid FIFO RAM address.
1–0 Reserved	This read-only field is reserved and always has the value zero. Reserved, read as 0.

32.5.23 FIFO receive FIFO start registers (FEC_FRSR)

FEC_FRSR is an 8-bit register programmed by the user to indicate the starting address of the receive FIFO. FEC_FRSR marks the boundary between the transmit and receive FIFOs. The transmit FIFO uses addresses from the start of the FIFO to the location four bytes before the address programmed into the FEC_FRSR. The receive FIFO uses the addresses from FEC_FRSR to FEC_FRBR inclusive.

The default value of the receive FIFO starting address is 0x40. This is the value assigned by hardware at reset.

Address: FEC_FRSR is 63FE_C000h base + 150h offset = 63FE_C150h



FEC_FRSR field descriptions

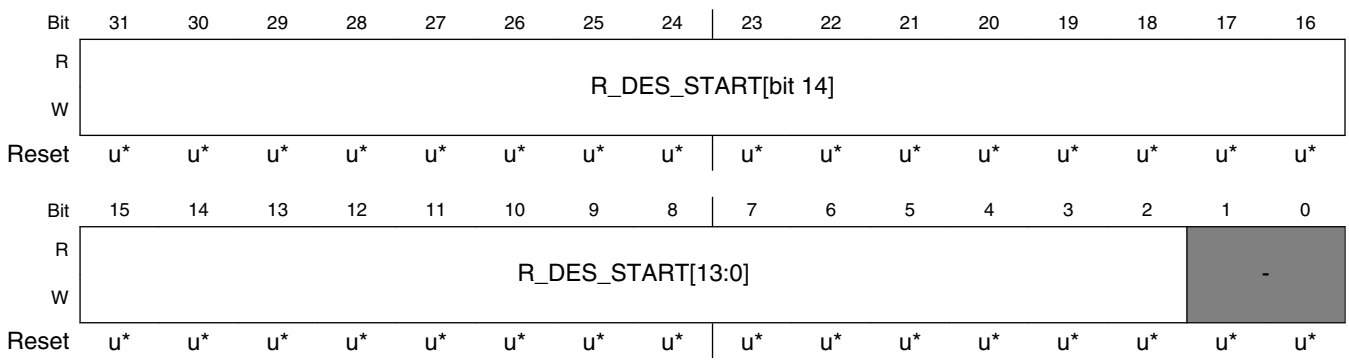
Field	Description
31–10 Reserved	This read-only field is reserved and always has the value zero. Reserved, read as 0 (except bit 10, which is read as 1).
9–2 R_FSTART	Address of first receive FIFO location. Acts as delimiter between receive and transmit FIFOs.
1–0 Reserved	This read-only field is reserved and always has the value zero. Reserved, read as 0.

32.5.24 Receive buffer descriptor ring start register (FEC_ERDSR)

The register is written by the user, and provides a pointer to the start of the circular receive buffer descriptor queue in external memory. This pointer must be 128-bit aligned (that is, evenly divisible by 16).

This register is unaffected by reset and must be initialized by the user prior to operation.

Address: FEC_ERDSR is 63FE_C000h base + 180h offset = 63FE_C180h



* Notes:
- u = Unaffected by reset.

FEC_ERDSR field descriptions

Field	Description
31–2 R_DES_START	Pointer to start of receive buffer descriptor queue.
1–0 -	Reserved, read as 0

32.5.25 Transmit buffer descriptor ring start register (FEC_ETDSR)

The register is written by the user, and provides a pointer to the start of the circular transmit buffer descriptor queue in external memory. This pointer must be 128-bit aligned (that is, evenly divisible by 16).

This register is unaffected by reset and must be initialized by the user prior to operation.

Address: FEC_ETDSR is 63FE_C000h base + 184h offset = 63FE_C184h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	X_DES_START[bit 14]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	X_DES_START[13:0]														-	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FEC_ETDSR field descriptions

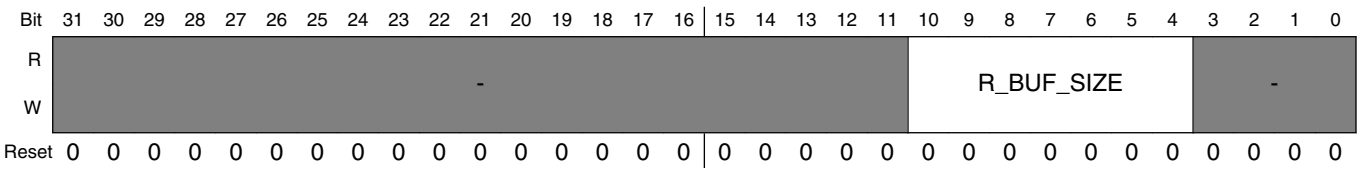
Field	Description
31–2 X_DES_START	Pointer to start of transmit buffer descriptor queue.
1–0 -	Reserved, read as 0

32.5.26 Maximum receive buffer size register (FEC_EMRBR)

The FEC_EMRBR is a user-programmable register which dictates the maximum size of all receive buffers. Note that because receive frames is truncated at 2k-1(2047) bytes, bits 31-11 are not used. The programmed value accounts for the fact that the receive CRC is always written into the last receive buffer. To allow one maximum size frame per buffer, FEC_EMRBR must be set to FEC_RCR[MAX_FL] or larger. The FEC_EMRBR must be evenly divisible by 16. To ensure this, bits 3-0 are forced low, and hence only bits 10-4 are actually used. To minimize bus utilization (descriptor fetches) it is recommended that FEC_EMRBR be greater than or equal to 256 bytes.

The FEC_EMRBR is unaffected by reset, and must be initialized by the user.

Address: FEC_EMRBR is 63FE_C000h base + 188h offset = 63FE_C188h



FEC_EMRBR field descriptions

Field	Description
31–11 -	Reserved, is written to 0 by the host processor.
10–4 R_BUF_SIZE	Receive buffer size.
3–0 -	Reserved, is written to 0 by the host processor.

Chapter 33
Fast Infrared Interface (FIRI)

33.1 Overview

The Fast Infrared Interface block (FIRI) is capable of establishing any of the following links:

- 0.576 Mbit/s
- 1.152 Mbit/s
- 4 Mbit/s half duplex (using a LED and IR detector)

The FIRI supports these links:

- 0.576 Mbit/s
- 1.152 Mbit/s Medium Infrared (MIR) physical layer protocol
- 4 Mbit/s Fast Infrared (FIR) physical layer protocol (defined by IrDA, version 1.4)

FIRI interface signals are multiplexed with UART counterpart signals using a GPIO configuration for a complete infrared interface that supports SIR, MIR, and FIR modes, as illustrated in Figure 33-1. In addition, the Serial Infrared (SIR) protocol, which supports data rate 115.2 kbps or lower, is implemented in the UART block.

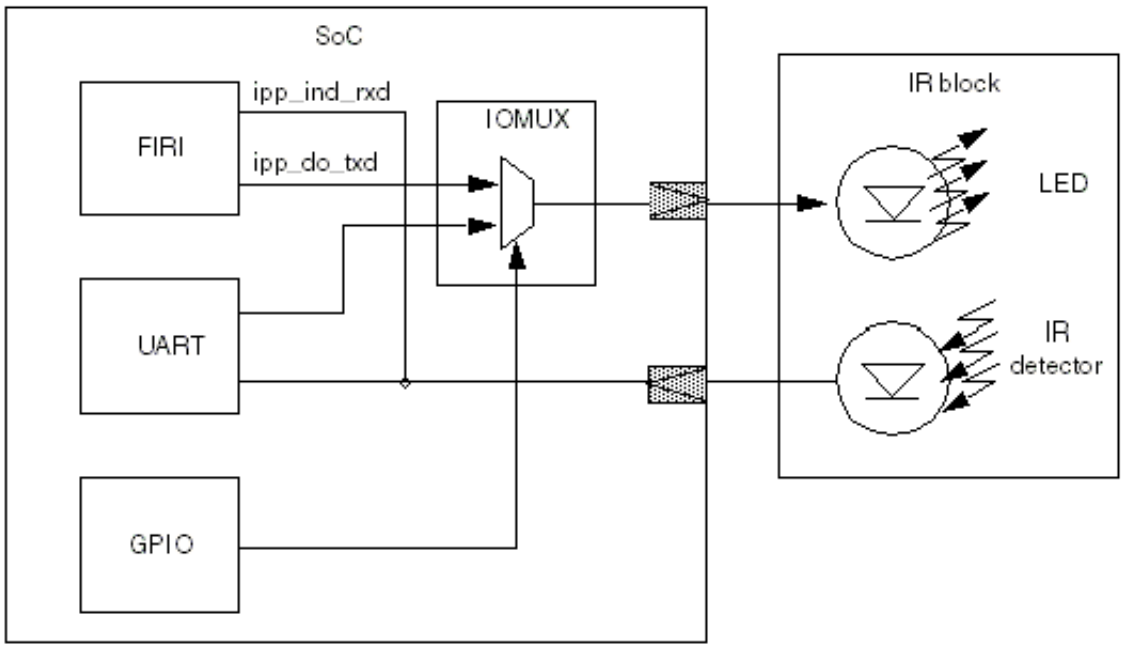


Figure 33-1. FIRI Integration Diagram

See Figure 33-2 for an illustration of the hardware components that comprise the FIRI controller.

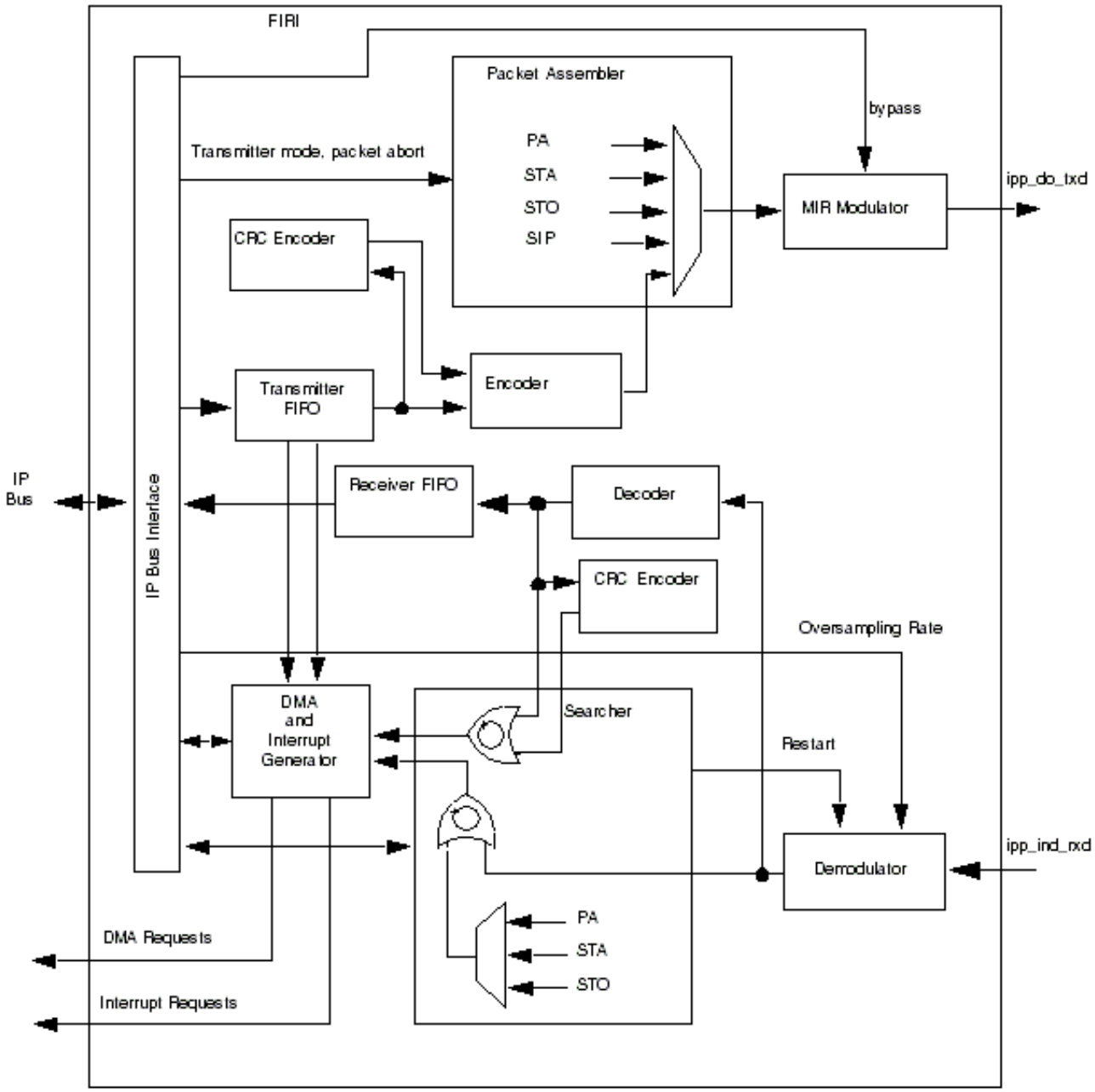


Figure 33-2. FIRI Block Diagram

The FIRI is divided into these four major functional components:

- Transmitter
- Receiver
- FIFO
- IP Bus interface

Each of the four components that make up the FIRI is further made up of individual blocks. The transmitter consists of these sub-blocks:

- Packet Assembler
- CRC Encoder
- MIR Modulator

The receiver consists of these sub-blocks:

- Searcher
- CRC Encoder
- Decoder
- Demodulator

33.1.1 Overview of IrDA Medium Infrared and Fast Infrared Standards

The FIRI supports:

- MIR (0.576 Mbit/s, 1.152 Mbit/s)
- FIR (4 Mbit/s) physical layer protocols

This sub-section provides a brief overview of the MIR and FIR standards.

33.1.1.1 MIR Packet Structure

The MIR packet format follows the standard High-Level Data Link Control (HDLC) format, except that it requires two beginning flags.

The MIR packet consists of:

- Two beginning flags (STA)
- An address
- Control fields
- Data fields
- A frame check sequence (CRC) field
- A minimum of one ending flag (STO)

See the table below for an illustration of the MIR packet structure, where STA (start flag) and STO (stop flag) are equal, predefined sequences (the left symbol is transmitted/ received first).

Table 33-1. MIR Packet Structure

STA	STA	Address	Control & Data	CRC	STO

The fields in the table above are defined as follows:

- STA, STO

  The MIR links use the same physical layer flag, b'0111,1110, for both STA and STO.

- 8-bit Address Field
- 8-bit Control Field plus up to 2045 bytes in the information field
- CRC field

  The MIR links use a 16-bit CRC-CCITT to check received frames for errors. The CRC is computed from the ADDR and Data fields.

The address, control, data, and CRC fields are not transmitted in original form: They are first converted according to the MIR standards described in the next sections.

33.1.1.2 FIR Packet Structure

See the table below for an illustration of the 4 Mbit/s FIR data packet format. The chip patterns and symbols for PA, STA, CRC field, and STO are listed in the following table, where PA (preamble), STA, and STO are a predefined sequence (left symbol transmitted/ received first).

Table 33-2. FIR Packet Structure

PA	STA	Address & Control & Data	CRC32	STO

The fields in the table above are defined as follows:

- PA-The preamble field is used by the receiver to establish phase lock. The preamble field consists of exactly sixteen repeated transmissions of the following stream of symbols:

  b'1000,0000,1010,1000

- STA-The STA consists of exactly one transmission of the following stream of symbols:

  b'0000,1100,0000,1100,0110,0000,0110,0000

- STO-The STO consists of exactly one transmission of a stream of symbols:

  b'0000,1100,0000,1100,0000,0110,0000,0110

- ACD-The payload data is encoded as described in the 4 PPM encoding in the preceding table. The encoded symbols reside in the ACD field and can be up to 2048 bytes long.
- CRC32-The CRC field consists of the 4 PPM encoded data, resulting from the IEEE 802 CRC32 algorithm for cyclic redundancy check as applied to the payload data contained in the packet.

33.1.1.3 MIR CRC

The CRC field is 16 bits long and generated according to the CRC-CCITT algorithm. The CRC polynomial is defined as follows:

$$CRC(x) = x^{16} + x^{12} + x^5 + 1$$

The CRC(x) value is inverted prior to transmission.

33.1.1.4 FIR CRC

The CRC32 field is 32 bits long and generated according to IEEE 802 CRC32 algorithm. The CRC32 polynomial is defined as follows:

$$CRC32(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

The CRC(x) value is inverted prior to transmission.

33.1.1.5 MIR Modulation

For MIR data rates, the NZR modulation scheme is used. A "0" is represented by a light pulse. The optical pulse duration is nominally 1/4 of a bit duration. The LED is off when a "1" is transmitted.

Table 33-3. MIR Modulation

Data Bit	Data Symbol (Address, Control, and Data)
0	1000
1	0000

NOTE

Tolerance of bit rate according to the standard must not exceed +0.1%.

33.1.1.6 FIR Modulation

For 4.0 Mbit/s, the modulation scheme is 4 PPM. In the modulation scheme, a pair of bits is taken together and called a data symbol. A data symbol is divided into four chips, only one of which contains an optical pulse. The nominal pulse duration is 125 ns. A "1" is represented by a light pulse.

Table 33-4. 4 PPM Mapping

Data Bit Pair	4 PPM Data Symbol
00	1000
01	0100
10	0010
11	0001

Verify that the tolerance of chip rate according to the standard does not exceed +0.01%. Pulse width tolerance of an electrical signal driven by an IR detector (photodiode) can be greater than the tolerance of the optical signal defined by IrDA, and can depend on the LED and IR devices used with the FIRI.

33.1.2 Features

The FIRI includes the following distinctive features:

- 0.576 Mbit/s, MIR protocol
- 1.152 Mbit/s, MIR protocol
- 4 Mbit/s, 4 PPM, FIR protocol
- Device Destination Detection Hardware Support
- Interrupt generation
- DMA capability
- SIP generation for collision avoidance

33.1.3 Modes of Operation

The FIRI supports the following modes:

- Hardware packet assembly:
  - FIR mode
  - 0.576 Mbps MIR mode
  - 1.152 Mbps MIR mode
  - Serial Infrared Interaction Pulse (SIP) generation
- Hardware packet search:
  - FIR mode
  - 0.576 Mbps MIR mode
  - 1.152 Mbps MIR mode
- Software packet assembling
- Software packet search

33.2 External Signal Description

See the following table for the list of signals used to control the FIRI.

Table 33-5. Signal Properties

Name	Direction	Port	Function	Reset State	Pull up
Signals Connecting To IC IO					
LED and IR Detector Interface					
IPP_DO_TXD	Output	ipp_do_txd	Data transmit signal. Active high.	1	-
IPP_IND_RXD	Input	ipp_ind_rxd	Data receive signal. Active high.	NA	at top level IO ring, if necessary

33.2.1 Detailed Signal Descriptions

33.2.1.1 IPP_DO_TXD

The Data Transmit Signal is the output signal from the Packet Assembler sub-block. It must be connected to an LED through the IOMUX, as shown in Figure 33-1. Add a DC-level translator off-chip, if necessary.

33.2.1.2 IPP_IND_RXD

The Data Receive Signal is driven by IR detector. Add a DC-level translator off-chip, if necessary. The signal is input to the Demodulator sub-block.

33.3 Programmable Registers

The FIRI includes six 32-bit registers, a transmit FIFO, and a receive FIFO. For information about the FIFOs, see Transmitter FIFO and Receiver FIFO.

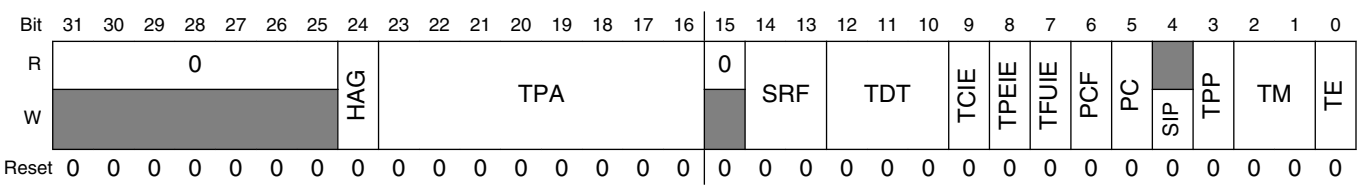
FIRI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FA_8000	FIRI Transmit Control Register (FIRI_TCR)	32	R/W	0000_0000h	33.3.1/ 1627
63FA_8004	FIRI Transmit Count Register (FIRI_TCTR)	32	R/W	0000_0000h	33.3.2/ 1629
63FA_8008	FIRI Receive Control Register (FIRI_RCR)	32	R/W	0000_0000h	33.3.3/ 1630
63FA_800C	FIRI Transmit Status Register (FIRI_TSR)	32	w1c	0000_0000h	33.3.4/ 1632
63FA_8010	FIRI Receive Status Register (FIRI_RSR)	32	w1c	0000_0000h	33.3.5/ 1633
63FA_801C	FIRI Control Register (FIRI_CR)	32	R/W	0000_0000h	33.3.6/ 1634

33.3.1 FIRI Transmit Control Register (FIRI_TCR)

The Transmitter Control Register is a 32-bit, read-write register that controls transmitter operation. The table below shows the register; The figure below shows its field descriptions.

Address: FIRI_TCR is 63FA_8000h base + 0h offset = 63FA_8000h



FIRI_TCR field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved
24 HAG	Hardware Address Generator. When this bit is set, the content of the TPA bits is transmitted as a packet address. When the bit is cleared, the packet address is read from TX FIFO. 0 Read packet address from TX FIFO 1 Use TPA bits as packet address
23–16 TPA	Transmit Packet Address. This field contains the 8-bit Transmit Packet Address. If the HAG bit is cleared, the TPA bits have no effect.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–13 SRF	Start Field Repeat Factor. This field contains the number of PA or STA fields transmitted in the beginning of the packet for FIR and MIR mode, respectively. In FIR mode, only the 00 value should be used. 00 16 PA or 2 STA fields is transmitted. 01 32 PA or 4 STA fields is transmitted. 10 64 PA or 8 STA fields is transmitted. 11 128 PA or 16 STA fields is transmitted.
12–10 TDT	This field controls the TX FIFO depth that triggers a DMA request. Refer to Transmitter FIFO, for details. **NOTE:** To avoid TX FIFO overflow, verify that the sum of TDT level and burst length (set by BL bits in the FIR_ICR register) does not exceed TX FIFO size, which is 128 bytes. 0 DMA request generated when TX FIFO is empty 1 DMA request generated when TX FIFO contain 16 bytes of data 2 DMA request generated when TX FIFO contain 32 bytes of data 3 DMA request generated when TX FIFO contain 48 bytes of data - ... - ... 7 DMA request generated when TX FIFO contains 112 bytes of data
9 TCIE	Transmit Complete Interrupt Enable. This bit enables the TC status bits of the FIRI_TSR register to generate a Transmit Complete Interrupt. 0 Transmit Complete Interrupt is not triggered by TC bit. 1 Transmit Complete Interrupt is triggered by TC bit.
8 TPEIE	Transmitter Packet End Interrupt Enable. This bit enables the TPE and SIPE status bits of the FIRI_TSR register to generate a Packet End Interrupt. 0 PEI interrupt is not triggered by TPE or SIPE bits. 1 PEI interrupt is triggered by TPE or SIPE bits.
7 TFUIE	Transmitter FIFO Underrun Interrupt Enable. This bit enables the TFU status bit of the FIRI_TSR register to generate a TFU interrupt. 0 TFU interrupt disabled 1 TFU interrupt enabled
6 PCF	Packet Complete by FIFO. This bit determines how a packet is completed if a TX FIFO underrun event occurs. Do not write software intentionally to cause underrun events. However, if due to erroneous

Table continues on the next page...

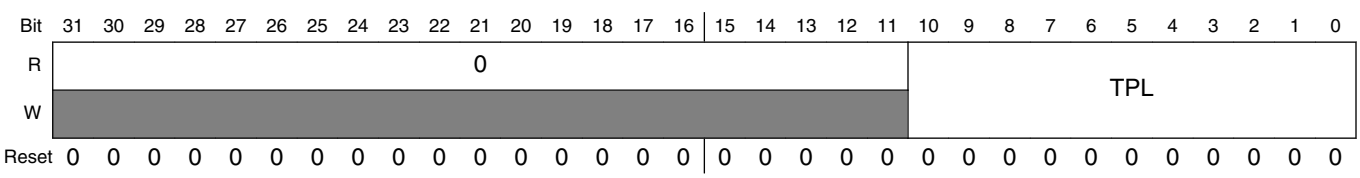
FIRI_TCR field descriptions (continued)

Field	Description
	conditions, the value of this bit selects between two recovery modes. Set the PCF based on system and upper layer IrDA protocol requirements. 0 Send CRC and STO fields 1 Send packet abort symbol
5 PC	Packet Complete. This bit determines how a packet is completed when the TE bit is cleared or when the SIP bit is set in the middle of the transfer. Do not write software intentionally to clear the TE bit, or to set the SIP bit in the middle of the transfer. However, if it is done due to erroneous conditions, the value of this PC bit selects between two recovery modes. Set PC based on system and upper layer IrDA protocols requirements. 0 Send CRC and STO fields 1 Send packet abort symbol
4 SIP	Transmit Enable of SIP. Writing "1" to this bit produces a "Serial Infrared Interaction Pulse" transmission. Writing a "0" to this bit is ignored. This bit is always read as "0". If this bit is set while in the middle of the transfer, the packet will be completed according to the setting of the PC bit.
3 TPP	Transmitter Pulse Polarity bit. 0 Transmitted pulse is not inverted. 1 Transmitted pulse is inverted.
2–1 TM	Transmitter Mode. This bits controls the transmission mode. 00 4 Mbps FIR Mode 01 0.576 Mbps MIR Mode 10 1.152 Mbps MIR Mode 11 Software Packet Assembling
0 TE	Transmitter Enable. This bit controls the FIRI transmitter. If this bit is cleared in the middle of the transfer, the packet will be completed according to the setting of the PC bit. When the packet is completed, the transmitter clocks are then gated OFF. 0 Transmitter Disabled 1 Transmitter Enabled

33.3.2 FIRI Transmit Count Register (FIRI_TCTR)

The Transmitter Count Register is 32-bit, read-write register that controls the packet size. The figure below shows the register; The table below shows its field descriptions.

Address: FIRI_TCTR is 63FA_8000h base + 4h offset = 63FA_8004h



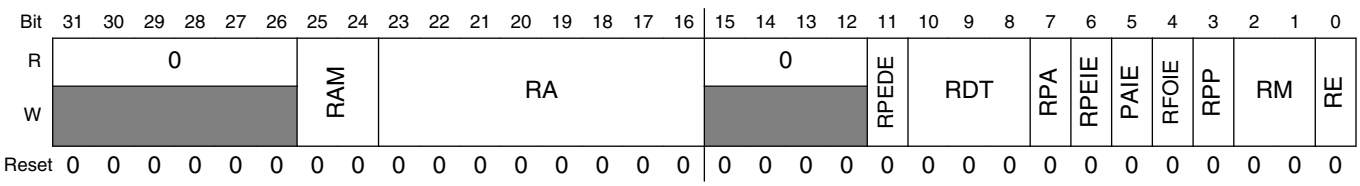
FIRI_TCTR field descriptions

Field	Description
31–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–0 TPL	Transmit Packet Length bits. The length of the address, control, and data fields (see Table 33-1 and FIR Packet Structure). 0 Send 1 byte 1 Send 2 bytes - ... - ... 2047 Send 2048 bytes

33.3.3 FIRI Receive Control Register (FIRI_RCR)

The Receiver Control Register is 32-bit, read-write register that controls receiver operation. The figure below shows the register; The table below shows its field descriptions.

Address: FIRI_RCR is 63FA_8000h base + 8h offset = 63FA_8008h



FIRI_RCR field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value zero. Reserved
25–24 RAM	Address Match bit. The value of this bit can be changed when the RE bit is cleared. 00 Does not match address 01 Match packet address to RA bits 10 Match packet address to Broadcast address 11 Match packet address to RA bits and to broadcast address
23–16 RA	Receiver Address. Determines Receiver Packet Address. If the RAM bit value is 00 or 10, the RA bit has no effect. The value of this bit can be changed when the RE bit is cleared.
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11 RPEDE	Receiver Packet End DMA Request Enable bit; Enable DMA request generation at the end of the packet. If this bit is set, verify that the BL value in the FIRICR register is not greater than the sum of the address, control, data, and CRC field length.

Table continues on the next page...

FIRI_RCR field descriptions (continued)

Field	Description
	0 DMA request is not affected by the end of packet. 1 DMA request is generated at the end of packet.
10–8 RDT	Receiver DMA Request Trigger level. Sets minimum RX FIFO depth, which triggers a DMA request. For more details, refer to Receiver FIFO. **NOTE:** To avoid RX FIFO underflow, set the RDT level to less than or equal value of the burst length (set by the BL bit in the FIRI_CR register). 0 Reserved 1 DMA request generated when RX FIFO contains 16 bytes of data 2 DMA request generated when RX FIFO contains 32 bytes of data 3 DMA request generated when RX FIFO contains 48 bytes of data - ... - ... 7 DMA request generated when RX FIFO contains 112 bytes of data
7 RPA	Receiver Packet Abort bit. Determines behavior of the RX FIFO upon detection of an illegal symbol. When an illegal symbol is detected, the DDE or CRCE bit in the FIRI_RSR register is set. If the RPA bit is set, the RX FIFO pointers are cleared and the receiver starts to search for the PA or STA fields for FIR and MIR mode, respectively. If RPA is cleared, the receiver continues to write to the RX FIFO. 0 Does not clear the RX FIFO upon detection of an illegal symbol 1 Clears the RX FIFO upon detection of illegal symbol
6 RPEIE	Receiver Packet End Interrupt Enable bit. Enables the RPE status bit in the FIRI_RSR register to assert the Packet End Interrupt. 0 PEI interrupt is not triggered by RPE bit. 1 PEI interrupt is triggered by RPE bit.
5 PAIE	Packet Abort Interrupt Enable bit. Enables the DDE, CRCE status bits in the FIRI_RSR register to cause a PAI interrupt. 0 PAI interrupt is disabled. 1 PAI interrupt is enabled.
4 RFOIE	Receiver FIFO Overrun Interrupt Enable bit. Enables the RFO status bit in the FIRI_RSR register to cause a RFOI interrupt. 0 RFOI interrupt is disabled. 1 RFOI interrupt is enabled.
3 RPP	Receiver Pulse Polarity bit. 0 Receiver signal is not inverted. 1 Received signal is inverted.
2–1 RM	Receiver Mode bits. 00 FIR Mode 01 0.576 Mbps MIR Mode 10 1.152 Mbps MIR Mode 11 Software Packet Assembling
0 RE	Receiver enable bit. When this bit is cleared, the receiver clocks are gated OFF.

Table continues on the next page...

FIRI_RCR field descriptions (continued)

Field	Description
	0 Receiver is disabled. 1 Receiver is enabled.

33.3.4 FIRI Transmit Status Register (FIRI_TSR)

The Transmit Status Register is 32-bit register that reflects the status of the transmitter and the TX FIFO. The FIRI_TSR contains read-only and write-one-to-clear bits. The figure below shows the register; The table below shows its field descriptions.

Address: FIRI_TSR is 63FA_8000h base + Ch offset = 63FA_800Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TFP								0				TC	SIPE	TPE	TXU
W													w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIRI_TSR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–8 TFP	Transmitter FIFO Pointer bits. The value of the TFP bits represent the number of available bytes in TX FIFO (read-only bits).
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved
3 TC	Transmit Complete bit. Indicates that the transmission is completed (including the CRC and STO fields), and the transmitter FIFO is empty. This bit is cleared by writing a "1". 0 Transmitting is not completed. 1 Transmitting is completed.
2 SIPE	SIP End bit. Indicates that "Serial Infrared Interaction Pulse" has been transmitted. This bit is cleared by writing a "1". 0 SIP transmission is not completed. 1 SIP had been transmitted.
1 TPE	Transmitter Packet End bit. Indicates that the TPS bytes of data (address, control, and data fields) have been transmitted. This bit is cleared by writing a "1".

Table continues on the next page...

FIRI_TSR field descriptions (continued)

Field	Description
	0 Transmissions of address, control, and data fields are not completed. 1 Address, control, and data fields had been transmitted.
0 TXU	Transmitter FIFO Underrun bit. Indicates the occurrence of a TX FIFO underrun. A TX FIFO underrun occurred if the transmitter attempted to read from the TX FIFO when the TX FIFO is empty. This bit is cleared by writing a "1". 0 No transmitter FIFO underrun 1 Transmitter FIFO is underrun

33.3.5 FIRI Receive Status Register (FIRI_RSR)

The Receive Status Register is 32-bit register that reflects the status of the receiver and the FIFO. It contains read-only and write-one-to-clear bits. The figure below shows the register; The table below shows its field descriptions.

Address: FIRI_RSR is 63FA_8000h base + 10h offset = 63FA_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RFP								0		PAS	RPE	RFO	BAM	CRCE	DDE
W												w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIRI_RSR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–8 RFP	Receiver FIFO Pointer bits. The value of the RFP bits represent the number of available bytes in the RX FIFO (read-only bits).
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5 PAS	Preamble Search bit. Indicates that the receiver is searching for PA+STA, or STA fields for FIR and MIR modes, respectively (read-only bit). 0 Receiver does not search for the PA or STA field. 1 Receiver searches for the PA or STA field.
4 RPE	Receiver Packet End bit. Indicates that the STO field or packet abort symbol (b'0000,000 and b'0000,0000 for MIR and FIR, respectively) is detected. This bit is cleared by writing a "1".

Table continues on the next page...

FIRI_RSR field descriptions (continued)

Field	Description
	0 STO was not detected. 1 STO field is detected.
3 RFO	Receiver FIFO Overrun bit. Indicates that the receiver attempted to write to the RX FIFO when the RX FIFO is full. This bit is cleared by writing a "1". 0 Receiver FIFO is not overrun. 1 Receiver FIFO is overrun.
2 BAM	Broadcast Address Match bit. Indicates that the address field of the packet matches the Broadcast Address 0xFF. This bit is cleared by writing a "1". 0 No Broadcast 1 Broadcast packet
1 CRCE	CRC Error bit. Indicates that the CRC check failed. This bit is cleared by writing a "1". 0 No CRC check failure 1 CRC check failure
0 DDE	Address, Control, or Data Field Error bit. Indicates that an illegal symbol has been detected in the address, control, data, or CRC32/CRC fields. This bit is cleared by writing a "1". 0 No illegal symbols in address, control, data, or CRC field 1 Illegal symbol in address, control, data, or CRC field

33.3.6 FIRI Control Register (FIRI_CR)

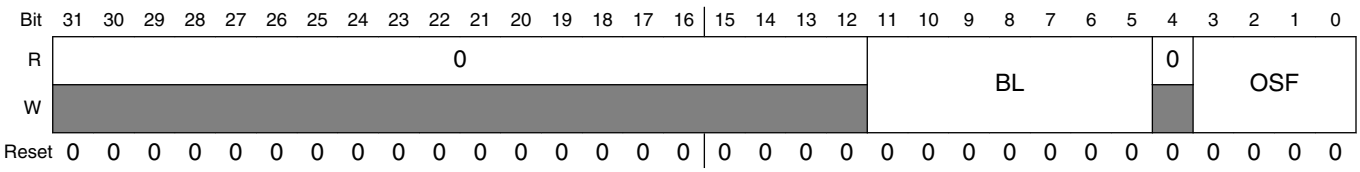
The Control Register is a 32-bit, read-write register that is shared by receiver and transmitter. The figure below shows the register; The table below shows its field descriptions.

To enable proper receive operations, the period of the divided clock must be four or more times shorter than the pulse width of the IPP_IND_RXD signal, and one or more times shorter than the gap between two consecutive pulses.

NOTE

The pulse width of the IPP_IND_RXD signal is dependent upon the characteristics of the LED and the Photo Diode used with the FIRI, and is different from the pulse width defined by IrDA for the optical signal. The value of the OSF bits must be adjusted accordingly. A greater OSF value leads to a better SNR on the expense of an increase in the power consumption.

Address: FIRI_CR is 63FA_8000h base + 1Ch offset = 63FA_801Ch



FIRI_CR field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11–5 BL	Burst Length. Used to prevent an overrun of the TX FIFO and an underrun of the RX FIFO. The DMA request of the transmitter deasserts when the number of vacant bytes in the TX FIFO is less than the burst length settings specified by the BL bits. The DMA request of the receiver deasserts when the number of available bytes in the RX FIFO is less than the burst length settings specified by the BL bits. 1 1 byte 2 2 bytes 3 3 bytes - ... 127 127 bytes 0 128 bytes
4 Reserved	This read-only field is reserved and always has the value zero. Reserved
3–0 OSF	Over Sampling Factor. This field controls the oversampling factor of the "chip" on the IPP_IND_RXD signal if the RE bit in the FIRI_RCR register is set, or the prescaling factor of the IPG_CLK_FIRI_BAUD signal if the TE bit in the FIRI_TCR register is set. **NOTE:** The "chip" rate is 4 and 2 times greater than the bit rate for MIR and FIR, respectively. 0 Does not oversample 1 Oversamples by 2 2 Oversamples by 3 - ... - ... 15 Oversamples by 16

33.4 Functional Description

This section details the FIRI functional description.

33.4.1 Transmitter Overview

The transmitter has three modes of operation:

- MIR

- FIR
- Software Packet Assembly

33.4.1.1 MIR Mode-Transmitter

In MIR mode, the following sequence occurs:

1. The Packet Assembler block sends the STA sequences (two or more times) to the MIR Modulator sub-block.
2. The Encoder block accesses the TX FIFO, aligns the data, and sends the ACD (Address, Control, and Data) fields to the Packet Assembler sub-block.
3. The CRC field calculated by CRC block follows the ACD field; after which, the STO bits are sent to the Modulator.

33.4.1.2 FIR Mode-Transmitter

In FIR mode, the following sequence occurs:

1. The Packet Assembler block sends the predefined PA (16 or more times) and STA sequences to the LED.
2. The Encoder block accesses the TX FIFO, encodes the data (4 PPM), and sends the ACD (Address, Control and Data) to the Packet Assembler sub-block.
3. The CRC32 field calculated by CRC block follows the ACD field; after which, the STO bits are sent to the Modulator.

NOTE

The MIR Modulator block is not operable in FIR mode.

If the DMA request is not served during MIR or FIR modes, and there is no valid data to transmit, the Assembler sub-block terminates the packet using the packet-abort symbol (with CRC/CRC32 and STO fields [see the PCF bit description]), and a TFUI interrupt is generated. The packet can also be aborted by the packet length counter or by the software (see PC bit description).

33.4.1.3 Serial Infrared Interaction Pulse

The transmitter must send the "Serial Infrared Interaction Pulse" (SIP) at 500 ms to guarantee that low speed IR devices will not interfere with its pulses. The pulse must be 8.7 µs wide; and positive phase must be 1.6 µs wide.

33.4.1.4 Software Packet Assembly Mode

In Software Packet Assembly mode, the entire packet is read from the TX FIFO, including the PA, STA, and STO bits. The MIR modulator and Encoder sub-blocks are disabled. This mode is used for debug purposes and can be used for testing compatibility with future IrDA protocols, if appropriate.

33.4.2 Transmitter FIFO

The transmitter FIFO is a 128-byte FIFO used for holding transmit data. The TX FIFO gets filled by the accesses to the Transmitter FIFO address (by core or DMA) and is emptied by the transmitter. When the DMA is used, a pre-programmed FIFO trigger level controls the triggering of the DMA request. When the TX FIFO is emptied below the trigger-level, a DMA request signal is asserted. The DMA request gets deasserted when the number of vacant bytes in the TX FIFO is less than the burst length settings specified by the BL bits in the FIRI_CR register. If the DMA request was not serviced and, as a result, there are no more data in the TX FIFO, the TFU bit in the FIRI_TSR register sets, and the $\overline{\text{IPI\_INT\_FIFO}}$ interrupt signal asserts depending on TFOIE bit in the FIRI_TCR register. During the transmit operation, the TFP bit value in the FIRI_TSR register reflects the amount of available bytes in the TX FIFO.

33.4.3 Receiver Overview

The receiver has three modes of operation:

- MIR
- FIR
- Software Packet Disassembly Modes

In MIR and FIR modes, each incoming pulse is oversampled by the programmable factor in the Demodulator sub-block. Next, the Demodulator detects the edges of the pulses and synchronizes to the phase of the transmitted pulse. Phase correction is performed until the end of the packet because the bit rate tolerance accumulates to very large values for long packets.

33.4.3.1 MIR Mode-Receiver

In MIR mode, the Searcher sub-block searches for the STA field. If the sequence is matched, the data stream following the STA field is sent to the Decoder block. The Decoder block searches for five consecutive "1"s and skips next the "zero" bit inserted by the transceiver for phase synchronization. The data is then written to the RX FIFO. In parallel, the Searcher sub-block searches for the STO field. When the STO field is detected, a PEI interrupt is generated together with a DMA request, and the corresponding flag is set in the status register. The Searcher sub-block continuously searches for illegal symbols (seven or more consecutive "1"s). If an illegal symbol is detected, the PAI interrupt is generated and the corresponding flag is set in the status register.

33.4.3.2 FIR Mode-Receiver

In FIR mode, the Searcher sub-block searches for the PA field and then for the STA field. If found, the "chip" stream is converted to a data stream by the Decoder sub-block and then it is written to the RX FIFO. In a parallel operation, the Searcher sub-block searches for the STO field. When the STO field is detected, a PEI interrupt can be generated together with a DMA request. While receiving, the Searcher sub-block continuously searches the PA, STA, ACD, CRC32, and STO fields for illegal symbols. In addition, the Searcher sub-block looks for a packet abort symbol. On detection of an illegal symbol, a PAI interrupt is generated and the corresponding flag is set in the status register.

In MIR and FIR modes, the "Address" bits can be compared to a predefined value or/and to the broadcast value 0xFF (see the description of RAM bits). If the CRC field value does not match an expected value calculated by the CRC Encoder sub-block, a PAI interrupt is generated. The CRC field sends the RX FIFO, irrespective of a comparison result.

33.4.3.3 Software Packet Disassembly Mode

In Software Packet Disassembly mode, the entire packet is sent to the RX FIFO, even if illegal symbols have been detected.

33.4.4 Receiver FIFO

The receiver FIFO is a 128-byte FIFO, used for holding received data. RX FIFO gets filled by incoming data and emptied by excess RX FIFO (by the core or DMA). When DMA is used, a pre-programmed FIFO trigger level controls the triggering of the DMA

request. When RX FIFO is filled beyond the trigger level, a DMA request signal is asserted. The DMA request gets deasserted when the number of available bytes in RX FIFO is less than the burst length settings specified by the BL bits in the FIRI_CR register. If the DMA request was not serviced and, as a result, there are no vacant bytes in the RX FIFO for receive data, the RFO bit in the FIRI_RSR register is set and the IPI_INT_FIFO interrupt signal is asserted depending on RFOIE bit in the FIRI_RCR register. During a receive operation, the RFP bits' value in the FIRI_RSR register reflects the amount of available bytes in the RX FIFO.

33.5 Initialization/Application Information

33.5.1 Examples of FIRI Programming

33.5.1.1 Transmitter Programming Scenario

The purpose of this scenario is to transmit 1024 bytes in FIR mode using an infrared link. Verify that the packet length is 512 bytes.

- Configure Clock Controller. For this example, the frequency of the IPG_CLK_FIRI_BAUD clock is 48 MHz.
- Configure the DMA for DMA-service FIRI transactions. A DMA transaction will be requested by the negative edge of DMA_REQ[1]. The DMA access size is 4 bytes. For this example, the burst length is 4 accesses.
- Configure the FIRI to work with a 32-byte burst (FIRICR.BL bits).

NOTE

Observe the burst length of the DMA, and the burst length selected by FIRI_CR. The BL bits are not necessarily equal.

- Set the FIRI Oversampling Factor (FIRI_CR.OSF bits) to 6 to achieve a 4-Mbit rate with the IPG_CLK_FIRI_BAUD clock.
- Configure the FIRI to transmit 512-byte long packets (FIRI_TCTR.TPL bits).
- Set the DMA trigger level to 16 (FIRI_TCR.TDT bits). A DMA request will be generated when there are only 16 bytes remaining in the TX FIFO. Select FIR mode (FIRI_TCR.TM bits).
- Finally, enable the transmitter (FIRI_TCT.TE bit). The FIRI immediately asserts a DMA request because the TX FIFO is empty. The DMA controller delivers a first burst to the FIRI. The FIRI starts the transmission.

33.5.1.2 Receiver Programming Scenario

Fort the receiver programming scenario, the data is transmitted in FIR mode. An interrupt should be generated at the end of each packet.

- Configure Clock Controller. For this example, the frequency of the IPG_CLK_FIRI_BAUD clock is 48 MHz.
- Configure the DMA for the FIRI to DMA transactions. The DMA transaction will be requested by the negative edge of $\overline{\text{DMA\_REQ}}$[0]. DMA access size is 4 bytes. For this example, burst length is 4 accesses.
- Configure the FIRI to work with 32-byte burst (FIRI_CR.BL bits). Set the FIRI Oversampling Factor (FIRI_CR.OSF bits) to 6 to achieve a 4-Mbit rate with the IPG_CLK_FIRI_BAUD clock.
- Set the DMA trigger level to 16 (FIRI_RCR.RDT bits). A DMA request is generated when there are 16 bytes in RX FIFO. Select FIR mode (FIRI_RCR.RM bits). Enable Packet End Interrupt.
- Finally, enable the receiver (FIRI_RCR.RE bit). The receiver searches for the PA and STA fields. When a PEI interrupt is generated, verify that the software clears the FIRI_RSR.RPE bit

Chapter 34
Flexible Controller Area Network (FLEXCAN)

34.1 Introduction

The Flexible Controller Area Network (FLEXCAN) block is a communication controller implementing the CAN protocol according to the CAN 2.0B protocol specification [Ref. 1]. A general block diagram is shown in Figure 34-1, and describes the main sub-blocks implemented in the FLEXCAN block, including two embedded memories, one for storing Message Buffers and another one for storing Rx Individual Mask Registers. Support for 64 Message Buffers is provided. The functions of the sub-blocks are described in subsequent sections.

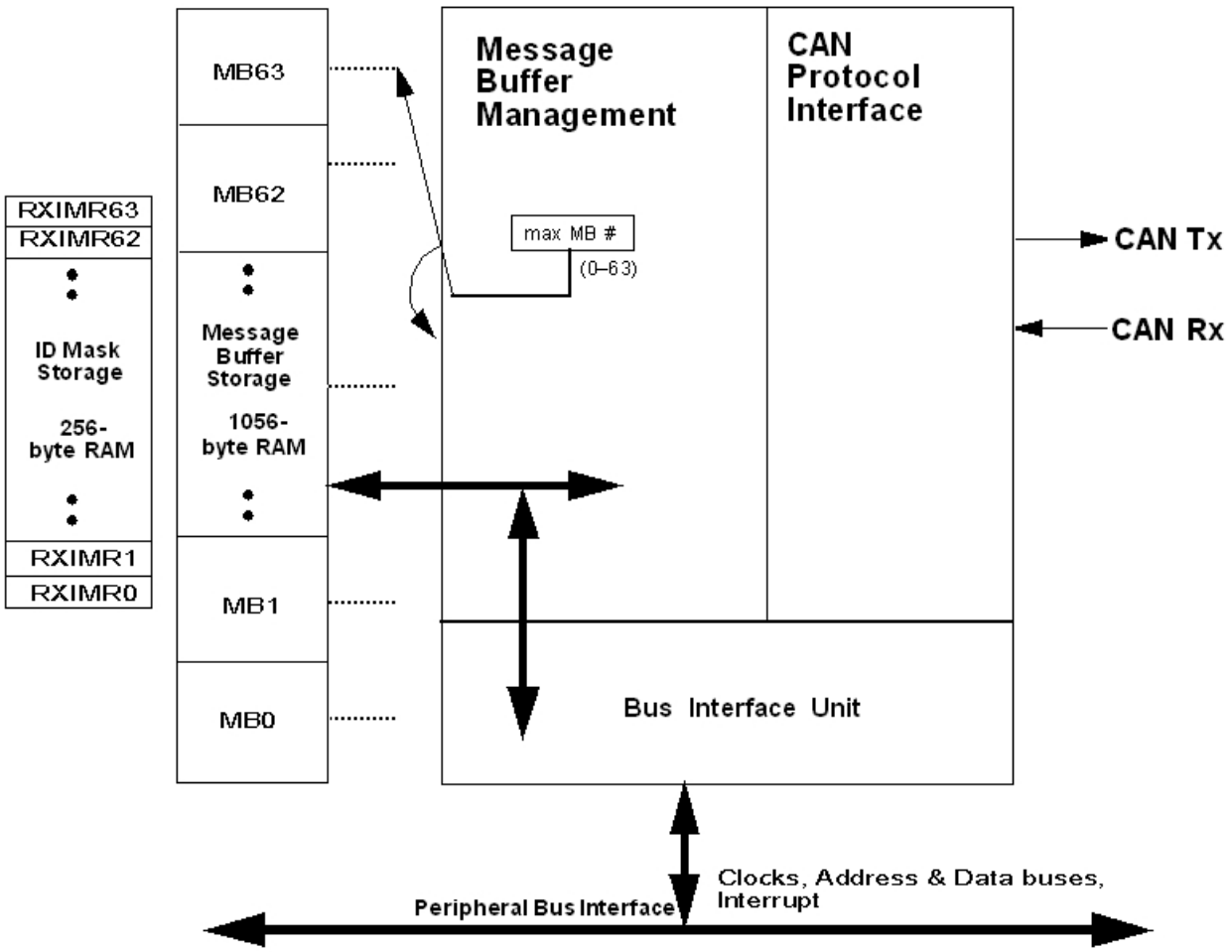


Figure 34-1. FLEXCAN Block Diagram

34.1.1 General Overview

The CAN protocol was designed primarily to be used as a vehicle serial data bus, meeting the specific requirements of this field: real-time processing, reliable operation in the EMI environment of a vehicle, cost-effectiveness and required bandwidth. The FLEXCAN block is a full implementation of the CAN protocol specification, which supports both standard and extended message frames. Sixty-four Message Buffers are supported. The Message Buffers are stored in an embedded RAM dedicated to the FLEXCAN.

The CAN Protocol Interface (CPI) sub-block manages the serial communication on the CAN bus, requesting RAM access for receiving and transmitting message frames, validating received messages and performing error handling. The Message Buffer Management (MBM) sub-block handles Message Buffer selection for reception and transmission, taking care of arbitration and ID matching algorithms. The Bus Interface

Unit (BIU) sub-block controls the access to and from the internal interface bus, in order to establish connection to the ARM and to other blocks. Clocks, address and data buses, interrupt outputs and test signals are accessed through the Bus Interface Unit

34.1.2 FLEXCAN Block Features

The FLEXCAN includes these distinctive features:

- Full Implementation of the CAN protocol specification
  - Standard data and remote frames
  - Extended data and remote frames
  - Zero to eight bytes data length
  - Programmable bit rate up to 1 Mb/sec
  - Content-related addressing
- Flexible Message Buffers of zero to eight bytes data length
- Each Message Buffer configurable as Rx or Tx, all supporting standard and extended messages
- Individual Rx Mask Registers per Message Buffer
- Includes 1056 bytes (64 Mbytes) of RAM used for Message Buffer storage
- Includes 256 bytes (64 Mbytes) of RAM used for individual Rx Mask Registers
- Full featured Rx FIFO with storage capacity for 6 frames and internal pointer handling
- Powerful Rx FIFO ID filtering, capable of matching incoming IDs against either 8 extended, 16 standard or 32 partial (8 bits) IDs, with individual masking capability
- Selectable backwards compatibility with previous CAN version
- Programmable clock source to the CAN Protocol Interface, either bus clock or crystal oscillator
- Unused message buffer and Rx Mask Register space can be used as general purpose RAM space
- Listen only mode capability
- Programmable loop-back mode supporting self-test operation
- Programmable transmission priority scheme: lowest ID, lowest buffer number or highest priority
- Time Stamp based on 16-bit free-running timer
- Global network time, synchronized by a specific message
- Maskable interrupts
- Independent of the transmission medium (an external transceiver is assumed)
- Short latency time due to an arbitration scheme for high-priority messages
- Low power modes, with programmable wake up on bus activity
- Configurable Glitch filter width to filter the noise on CAN bus when waking up

34.1.3 Modes of Operation

The FLEXCAN has four functional modes: Normal Mode (User and Supervisor), Freeze Mode, Listen-Only Mode and Loop-Back Mode. There are also two low power modes: Disable Mode and Stop Mode.

- Normal Mode (User or Supervisor):

In Normal Mode, the block operates receiving and/or transmitting message frames, errors are handled normally and all the CAN Protocol functions are enabled. User and Supervisor Modes differ in the access to some restricted control registers.

- Freeze Mode:

It is enabled when the FRZ bit in the MCR Register is asserted. If enabled, Freeze Mode is entered when the HALT bit in MCR is set or when Debug Mode is requested at ARM platform level . In this mode, no transmission or reception of frames is done and synchronicity to the CAN bus is lost. See Freeze Mode.

- Listen-Only Mode:

The block enters this mode when the LOM bit in the Control Register is asserted. In this mode, transmission is disabled, all error counters are frozen and the block operates in a CAN Error Passive mode [Ref. 1]. Only messages acknowledged by another CAN station will be received. If FLEXCAN detects a message that has not been acknowledged, it will flag a BIT0 error (without changing the REC), as if it was trying to acknowledge the message.

- Loop-Back Mode:

The block enters this mode when the LPB bit in the Control Register is asserted. In this mode, FLEXCAN performs an internal loop back that can be used for self test operation. The bit stream output of the transmitter is internally fed back to the receiver input. The Rx CAN input pin is ignored and the Tx CAN output goes to the recessive state (logic '1'). FLEXCAN behaves as it normally does when transmitting and treats its own transmitted message as a message received from a remote node. In this mode, FLEXCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field to ensure proper reception of its own message. Both transmit and receive interrupts are generated.

- Block Disable Mode:

This low power mode is entered when the MDIS bit in the MCR Register is asserted. When disabled, the block shuts down the clocks to the CAN Protocol Interface and Message Buffer Management sub-blocks. Exit from this mode is done by negating the MDIS bit in the MCR Register. See Block Disable Mode.

- Stop Mode:

This low power mode is entered when Stop Mode is requested at the ARM platform level . When in Stop Mode, the block puts itself in an inactive state and then informs the ARM that the clocks can be shut down globally. Exit from this mode happens when the Stop Mode request is removed or when activity is detected on the CAN bus and the Self Wake Up mechanism is enabled. See Stop Mode.

34.2 External Signal Description

34.2.1 Signals Overview

The FLEXCAN has two I/O signals, both of which are summarized in Table 34-1 and described in more detail in the following sub-sections.

Table 34-1. FLEXCAN Signals

Signal Name	Direction	Description
CAN Rx	Input	CAN Receive Pin
CAN Tx	Output	CAN Transmit Pin

34.2.2 Signal Descriptions

34.2.2.1 CAN Rx

This pin is the receive pin from the CAN bus transceiver. Dominant state is represented by logic level. Recessive state is represented by logic level.

34.2.2.2 CAN Tx

This pin is the transmit pin to the CAN bus transceiver. Dominant state is represented by logic level. Recessive state is represented by logic level.

34.3 Buffers

34.3.1 Standard/Extended Message Buffer (MB0) Memory Map

The FLEXCAN stores CAN messages for transmission and reception using a Message Buffer structure. Each individual Message Buffer is formed by 16 bytes mapped on memory as described in Table 34-2.

Table 34-2 shows a Standard/Extended Message Buffer (MB0) memory map, using 16 bytes total (0x80,0x8F space).

Table 34-2. Message Buffer MB0 Memory Mapping

Address Offset	Message Buffer Field
0x80	Control and Status (C/S)
0x84	Identifier Field
0x88,0x8F	Data Field 0, Data Field 7 (1 byte each)

34.3.2 Message Buffer Structure

The Message Buffer structure used by the FLEXCAN is represented below. Both the Extended and Standard Frames (29-bit Identifier and 11-bit Identifier, respectively) used in the CAN specification are represented.

Table 34-3. Message Buffer Structure

	31			28	27			24		22	21	20	19			16	15														0
	0			3	4			7		9	10	11	12			15	16														31
0x0	U / R				CODE				U / R	SR	ID	RT	LENGTH				TIME STAMP															
0x4		ID (Standard/Extended)												ID (Extended)																		
0x8	Data Byte 0								Data Byte 1								Data Byte 2														Data Byte 3	
0xC	Data Byte 4								Data Byte 5								Data Byte 6														Data Byte 7	
	U / R = Unimplemented or Reserved																															

CODE Message Buffer Code

This 4-bit field can be accessed (read or write) by the ARM core and by the FLEXCAN itself, as part of the message buffer matching and arbitration process. The encoding is shown in Table 34-4 and Table 34-5.

Table 34-4. Message Buffer Code for Rx buffers

Rx Code BEFORE Rx New Frame	Description	Rx Code AFTER Rx New Frame	Comment
0000	INACTIVE: Message Buffer is not active.		Message buffer does not participate in the matching process.
0100	EMPTY: Message buffer is active and empty.	0010	Message buffer participates in the matching process. When a frame is received successfully, the code is automatically updated to FULL.
0010	FULL: Message buffer is full.	0010	The act of reading the C/S word followed by unlocking the Message buffer does not make the code return to EMPTY. It remains FULL. If a new frame is written to the message buffer after the C/S word was read and the message buffer was unlocked, the code still remains FULL.
		0110	If the message buffer is FULL and a new frame is overwritten to this message buffer before the ARM had time to read it, the code is automatically updated to OVERRUN. Refer to for details about overrun behavior.
0110	OVERRUN: a frame was overwritten into a full buffer.	0010	If the code indicates OVERRUN but the ARM reads the C/S word and then unlocks the message buffer, when a new frame is written to the message buffer the code returns to FULL.
		0110	If the code already indicates OVERRUN, and yet another new frame must be written, the message buffer will be overwritten again, and the code will remain OVERRUN. Refer to for details about overrun behavior.
0XY1[1, 1]	BUSY: FLEXCAN is updating the contents of the message buffer. The ARM must not access the Message Buffer.	0010	An EMPTY buffer was written with a new frame (XY was 01).
		0110	A FULL/OVERRUN buffer was overwritten (XY was 11).

1. Note that for Tx message buffers (see Table 34-5), the BUSY bit should be ignored upon read, except when AEN bit is set in the MCR register.

Table 34-5. Message Buffer Code for Tx buffers

RTR	Initial Tx code	Code after successful transmission	Description
X	1000		INACTIVE: Message Buffer does not participate in the arbitration process.

Table continues on the next page...

Table 34-5. Message Buffer Code for Tx buffers (continued)

RTR	Initial Tx code	Code after successful transmission	Description
X	1001	-	ABORT: Message Buffer was configured as Tx and ARM core aborted the transmission. This code is only valid when AEN bit in MCR is asserted. Message Buffer does not participate in the arbitration process.
0	1100	1000	Transmit data frame unconditionally once. After transmission, the message buffer automatically returns to the INACTIVE state.
1	1100	0100	Transmit remote frame unconditionally once. After transmission, the message buffer automatically becomes an Rx message buffer with the same ID.
0	1010	1010	Transmit a data frame whenever a remote request frame with the same ID is received. This message buffer participates simultaneously in both the matching and arbitration processes. The matching process compares the ID of the incoming remote request frame with the ID of the message buffer. If a match occurs this message buffer is allowed to participate in the current arbitration process and the Code field is automatically updated to '1110' to allow the message buffer to participate in future arbitration runs. When the frame is eventually transmitted successfully, the Code automatically returns to '1010' to restart the process again.
0	1110	1010	This is an intermediate code that is automatically written to the message buffer by the MBM as a result of match to a remote request frame. The data frame will be transmitted unconditionally once and then the code will automatically return to 1010. The ARM can also write this code with the same effect.

SRR Substitute Remote Request

Fixed recessive bit, used only in extended format. It must be set to '1' by the user for transmission (Tx Buffers) and will be stored with the value received on the CAN bus for Rx receiving buffers. It can be received as either recessive or dominant. If FLEXCAN receives this bit as dominant, then it is interpreted as arbitration loss.

1 = Recessive value is compulsory for transmission in Extended Format frames

0 = Dominant is not a valid value for transmission in Extended Format frames

IDE ID Extended Bit

This bit identifies whether the frame format is standard or extended.

1 = Frame format is extended

0 = Frame format is standard

RTR Remote Transmission Request

This bit is used for requesting transmissions of a data frame. If FLEXCAN transmits this bit as '1' (recessive) and receives it as '0' (dominant), it is interpreted as arbitration loss. If this bit is transmitted as '0' (dominant), then if it is received as '1' (recessive), the FLEXCAN treats it as bit error. If the value received matches the value transmitted, it is considered as a successful bit transmission.

1 = Indicates the current message buffer has a Remote Frame to be transmitted

0 = Indicates the current message buffer has a Data Frame to be transmitted

LENGTH Length of Data in Bytes

This 4-bit field is the length (in bytes) of the Rx or Tx data, which is located in offset 0x8 through 0xF of the message buffer space (see Table 34-3). In reception, this field is written by the FLEXCAN, copied from the DLC (Data Length Code) field of the received frame. In transmission, this field is written by the ARM and corresponds to the DLC field value of the frame to be transmitted. When RTR=1, the Frame to be transmitted is a Remote Frame and does not include the data field, regardless of the Length field.

TIME STAMP Free-Running Counter Time Stamp

This 16-bit field is a copy of the Free-Running Timer, captured for Tx and Rx frames at the time when the beginning of the Identifier field appears on the CAN bus.

PRIO Local priority

This 3-bit field is only used when LPRIO_EN bit is set in MCR and it only makes sense for Tx buffers. These bits are not transmitted. They are appended to the regular ID to define the transmission priority.

ID Frame Identifier

In Standard Frame format, only the 11 most significant bits (28 to 18) are used for frame identification in both receive and transmit cases. The 18 least significant bits are ignored. In Extended Frame format, all bits are used for frame identification in both receive and transmit cases.

DATA Data Field

Up to eight bytes can be used for a data frame. For Rx frames, the data is stored as it is received from the CAN bus. For Tx frames, the ARM core prepares the data field to be transmitted within the frame.

34.3.3 Rx FIFO Structure

When the FEN bit is set in the MCR, the memory area from 0x80 to 0xFF (which is normally occupied by MB0 to MB7) is used by the reception FIFO engine. Table 34-6 shows the Rx FIFO data structure. The region 0x80-0x8F contains an message buffer structure which is the port through which the ARM core reads data from the FIFO (the oldest frame received and not read yet). The region 0x90-0xDF is reserved for internal use of the FIFO engine. The region 0xE0-0xFF contains an 8-entry ID table that specifies filtering criteria for accepting frames into the FIFO. Table 34-7 shows the three different formats that the elements of the ID table can assume, depending on the IDAM field of the MCR. Note that all elements of the table must have the same format.

Table 34-6. Rx FIFO Structure

	31			28	27			24		22	21	20	19			16	15						8	7							0
0x80										SRR	IDE	RTR	LENGTH				TIME STAMP														
0x84		ID (Standard/Extended)															ID (Extended)														
0x88	Data Byte 0									Data Byte 1							Data Byte 2														
0x8C	Data Byte 4									Data Byte 5							Data Byte 6														
0x90	Reserved																														
to																															
0xDF																															
0xE0	ID Table 0																														
0xE4	ID Table 1																														
0xE8	ID Table 2																														
0xEC	ID Table 3																														
0xF0	ID Table 4																														
0xF4	ID Table 5																														
0xF8	ID Table 6																														
0xFC	ID Table 7																														
			= Unimplemented or Reserved																												

Table 34-7. ID Table 0 - 7

Format	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table continues on the next page...

Table 34-7. ID Table 0 - 7 (continued)

A	REM	EXT	RXIDA (Standard = 29-19, Extended = 29-1)			
B	REM	EXT	RXIDB_0 (Standard = 29-19, Extended = 29-16)	REM	EXT	RXIDB_1 (Standard = 13-3, Extended = 13-0)
C	RXIDC_0 (Std/Ext = 31-24)	RXIDC_1 (Std/Ext = 23-16)	RXIDC_2 (Std/Ext = 15-8)	RXIDC_3 (Std/Ext = 7-0)		
			= Unimplemented or Reserved			

REM Remote Frame

This bit specifies if Remote Frames are accepted into the FIFO if they match the target ID.

1 = Remote Frames can be accepted and data frames are rejected

0 = Remote Frames are rejected and data frames can be accepted

EXT Extended Frame

Specifies whether extended or standard frames are accepted into the FIFO if they match the target ID.

1 = Extended frames can be accepted and standard frames are rejected

0 = Extended frames are rejected and standard frames can be accepted

RXIDA Rx Frame Identifier (Format A)

Specifies an ID to be used as acceptance criteria for the FIFO. In the standard frame format, only the 11 most significant bits are used for frame identification. In the extended frame format, all bits are used.

RXIDB_0, RXIDB_1 Rx Frame Identifier (Format B)

Specifies an ID to be used as acceptance criteria for the FIFO. In the standard frame format, the 11 most significant bits (a full standard ID) are used for frame identification. In the extended frame format, all 14 bits of the field are compared to the 14 most significant bits of the received ID.

RXIDC_0, RXIDC_1, RXIDC_2, RXIDC_3 Rx Frame Identifier (Format C)

Specifies an ID to be used as acceptance criteria for the FIFO. In both standard and extended frame formats, all 8 bits of the field are compared to the 8 most significant bits of the received ID.

34.4 Functional Description

34.4.1 Overview

The FLEXCAN is a CAN protocol engine with a very flexible mailbox system for transmitting and receiving CAN frames. The mailbox system is composed by a set of 64 Message Buffers that store configuration and control data, time stamp, message ID and data. The memory corresponding to the first 8 message buffers can be configured to support a FIFO reception scheme with a powerful ID filtering mechanism, capable of checking incoming frames against a table of IDs (up to 8 extended IDs or 16 standard IDs or 32 8-bit ID slices), each one with its own individual mask register. Simultaneous reception through FIFO and mailbox is supported. For mailbox reception, a matching algorithm makes it possible to store received frames only into message buffers that have the same ID programmed on its ID field. A masking scheme makes it possible to match the ID programmed on the message buffer with a range of IDs on received CAN frames. For transmission, an arbitration algorithm decides the prioritization of message buffers to be transmitted based on the message ID (optionally augmented by 3 local priority bits) or the message buffer ordering.

Before proceeding with the functional description, an important concept must be explained. A Message Buffer is said to be 'active' at a given time if it can participate in the matching and arbitration algorithms that are happening at that time. An Rx message buffer with a '0000' code is inactive (refer to Table 34-4). Similarly, a Tx message buffer with a '1000' or '1001' code is also inactive (refer to Table 34-5). An message buffer not programmed with '0000','1000' or '1001' will be temporarily deactivated (will not participate in the current arbitration or matching run) when the ARM writes to the C/S field of that message buffer.

The FLEXCAN also provide a glitch filter which can filter the noises on CAN bus when the FLEXCAN is in the STOP mode. The glitch filter width is configurable by the register.

34.4.2 Transmit Process

In order to transmit a CAN frame, the ARM must prepare a Message Buffer for transmission by executing the following procedure:

- If the message buffer is active (transmission pending), write an ABORT code ('1001') to the Code field of the Control and Status word to request an abortion of the

transmission, then read back the Code field and the IFLAG register to check if the transmission was aborted. If backwards compatibility is desired (AEN in MCR negated), just write ,'1000', to the Code field to inactivate the message buffer but then the pending frame may be transmitted without notification (see Table 34-5).
- Write the ID word.
- Write the data bytes.
- Write the Length, Control and Code fields of the Control and Status word to activate the message buffer.

Once the message buffer is activated in the fourth step, it will participate into the arbitration process and eventually be transmitted according to its priority. At the end of the successful transmission, the value of the Free Running Timer is written into the Time Stamp field, the Code field in the Control and Status word is updated, a status flag is set in the Interrupt Flag Register and an interrupt is generated if allowed by the corresponding Interrupt Mask Register bit. The new Code field after transmission depends on the code that was used to activate the message buffer in step four (see Table 34-4 and Table 34-5). When the Abort feature is enabled (AEN in MCR is asserted), after the Interrupt Flag is asserted for a message buffer configured as transmit buffer, the message buffer is blocked, therefore the ARM is not able to update it until the Interrupt Flag is negated by ARM. It means that the ARM must clear the corresponding IFLAG before starting to prepare this message buffer for a new transmission or reception.

34.4.3 Arbitration process

The arbitration process is an algorithm executed by the MBM that scans the whole message buffer memory looking for the highest priority message to be transmitted. All message buffers programmed as transmit buffers will be scanned to find the lowest ID[1] or the lowest message buffer number or the highest priority, depending on the LBUF and LPRIO_EN bits on the Control Register. The arbitration process is triggered in the following events:

- During the CRC field of the CAN frame
- During the error delimiter field of the CAN frame
- During Intermission, if the winner message buffer defined in a previous arbitration was deactivated, or if there was no message buffer to transmit, but the ARM wrote to the C/S word of any message buffer after the previous arbitration finished
- When MBM is in Idle or Bus Off state and the ARM writes to the C/S word of any message buffer
- Upon leaving Freeze Mode

1. Actually, if LBUF is negated, the arbitration considers not only the ID, but also the RTR and IDE bits placed inside the ID at the same positions they are transmitted in the CAN frame.

When LBUF is asserted, the LPRIO_EN bit has no effect and the lowest number buffer is transmitted first. When LBUF and LPRIO_EN are both negated, the message buffer with the lowest ID is transmitted first but. If LBUF is negated and LPRIO_EN is asserted, the PRIO bits augment the ID used during the arbitration process. With this extended ID concept, arbitration is done based on the full 32-bit ID and the PRIO bits define which message buffer should be transmitted first, therefore message buffers with PRIO = 000 have higher priority. If two or more message buffers have the same priority, the regular ID will determine the priority of transmission. If two or more message buffers have the same priority (3 extra bits) and the same regular ID, the lowest message buffer will be transmitted first.

Once the highest priority message buffer is selected, it is transferred to a temporary storage space called Serial Message Buffer (SMB), which has the same structure as a normal message buffer but is not user accessible. This operation is called 'move-out' and after it is done, write access to the corresponding message buffer is blocked (if the AEN bit in MCR is asserted). The write access is released in the following events:

- After the message buffer is transmitted
- FLEXCAN enters in HALT or BUS OFF
- FLEXCAN loses the bus arbitration or there is an error during the transmission

At the first opportunity window on the CAN bus, the message on the SMB is transmitted according to the CAN protocol rules. FLEXCAN transmits up to eight data bytes, even if the DLC (Data Length Code) value is bigger.

34.4.4 Receive Process

To be able to receive CAN frames into the mailbox message buffers, the ARM must prepare one or more Message Buffers for reception by executing the following steps:

- If the message buffer has a pending transmission, write an ABORT code ('1001') to the Code field of the Control and Status word to request an abortion of the transmission, then read back the Code field and the IFLAG register to check if the transmission was aborted. If backwards compatibility is desired (AEN in MCR negated), just write '1000' to the Code field to inactivate the message buffer, but then the pending frame may be transmitted without notification. If the message buffer already programmed as a receiver, just write '0000' to the Code field of the Control and Status word to keep the message buffer inactive. (See Table 34-3)
- Write the ID word
- Write '0100' to the Code field of the Control and Status word to activate the message buffer

Once the message buffer is activated in the third step, it will be able to receive frames that match the programmed ID. At the end of a successful reception, the message buffer is updated by the MBM as follows:

- The value of the Free Running Timer is written into the Time Stamp field
- The received ID, Data (8 bytes at most) and Length fields are stored
- The Code field in the Control and Status word is updated (see Table 34-4 and Table 34-5)
- A status flag is set in the Interrupt Flag Register and an interrupt is generated if allowed by the corresponding Interrupt Mask Register bit

Upon receiving the message buffer interrupt, the ARM should service the received frame using the following procedure:

- Read the Control and Status word (mandatory - activates an internal lock for this buffer)
- Read the ID field (optional - needed only if a mask was used)
- Read the Data field
- Read the Free Running Timer (optional - releases the internal lock)

Upon reading the Control and Status word, if the BUSY bit is set in the Code field, then the ARM should defer the access to the message buffer until this bit is negated. Reading the Free Running Timer is not mandatory. If not executed the message buffer remains locked, unless the ARM reads the C/S word of another message buffer. Note that only a single message buffer is locked at a time. The only mandatory ARM read operation is the one on the Control and Status word to assure data coherency (see Table 34-4).

The ARM should synchronize to frame reception by the status flag bit for the specific message buffer in one of the IFLAG Registers and not by the Code field of that message buffer. Polling the Code field does not work because once a frame was received and the ARM services the message buffer (by reading the C/S word followed by unlocking the message buffer), the Code field will not return to EMPTY. It will remain FULL, as explained in Table 34-2 . If the ARM tries to workaround this behavior by writing to the C/S word to force an EMPTY code after reading the message buffer, the message buffer is actually deactivated from any currently ongoing matching process. As a result, a newly received frame matching the ID of that message buffer may be lost. In summary: never do polling by reading directly the C/S word of the message buffers. Instead, read the IFLAG registers.

Note that the received ID field is always stored in the matching message buffer, thus the contents of the ID field in an message buffer may change if the match was due to masking. Note also that FLEXCAN does receive frames transmitted by itself if there exists an Rx matching message buffer, provided the SRX_DIS bit in the MCR is not

asserted. If SRX_DIS is asserted, FLEXCAN will not store frames transmitted by itself in any message buffer, even if it contains a matching message buffer, and no interrupt flag or interrupt signal will be generated due to the frame reception.

To be able to receive CAN frames through the FIFO, the ARM must enable and configure the FIFO during Freeze Mode (see Freeze Mode). Upon receiving the frames available interrupt from FIFO, the ARM should service the received frame using the following procedure:

- Read the Control and Status word (optional - needed only if a mask was used for IDE and RTR bits)
- Read the ID field (optional - needed only if a mask was used)
- Read the Data field
- Clear the frames available interrupt (mandatory - release the buffer and allow the ARM to read the next FIFO entry)

34.4.5 Matching Process

The matching process is an algorithm executed by the MBM that scans the message buffer memory looking for Rx message buffers programmed with the same ID as the one received from the CAN bus. If the FIFO is enabled, the 8-entry ID table from FIFO is scanned first and then, if a match is not found within the FIFO table, the other message buffers are scanned. In the event that the FIFO is full, the matching algorithm will always look for a matching message buffer outside the FIFO region.

When the frame is received, it is temporarily stored in a hidden auxiliary message buffer called Serial Message Buffer (SMB). The matching process takes place during the CRC field of the received frame. If a matching ID is found in the FIFO table or in one of the regular message buffers, the contents of the SMB will be transferred to the FIFO or to the matched message buffer during the 6th bit of the End-Of-Frame field of the CAN protocol. This operation is called "move-in". If any protocol error (CRC, ACK, etc.) is detected, than the move-in operation does not happen.

For the regular mailbox message buffers, an message buffer is said to be "free to receive" a new frame if the following conditions are satisfied:

- The message buffer is not locked
- The Code field is either EMPTY or else it is FULL or OVERRUN but the ARM has already serviced the message buffer (read the C/S word and then unlocked the message buffer)

If the first message buffer with a matching ID is not "free to receive" the new frame, then the matching algorithm keeps looking for another free message buffer until it finds one. If it can not find one that is free, then it will overwrite the last matching message buffer (unless it is locked) and set the Code field to OVERRUN (refer to Table 34-4and Table 34-5). If the last matching message buffer is locked, then the new message remains in the SMB, waiting for the message buffer to be unlocked.

Suppose, for example, that the FIFO is disabled and there are two message buffers with the same ID, and FLEXCAN starts receiving messages with that ID. Let us say that these message buffers are the second and the fifth in the array. When the first message arrives, the matching algorithm will find the first match in message buffer number 2. The code of this message buffer is EMPTY, so the message is stored there. When the second message arrives, the matching algorithm will find message buffer number 2 again, but it is not "free to receive" so it will keep looking and find message buffer number 5 and store the message there. If yet another message with the same ID arrives, the matching algorithm finds out that there are no matching message buffers that are "free to receive" so it decides to overwrite the last matched message buffer, which is number 5. In doing so, it sets the Code field of the message buffer to indicate OVERRUN.

The ability to match the same ID in more than one message buffer can be exploited to implement a reception queue (in addition to the full featured FIFO) to allow more time for the ARM to service the message buffers. By programming more than one message buffer with the same ID, received messages will be queued into the message buffers. The ARM can examine the Time Stamp field of the message buffers to determine the order in which the messages arrived.

The matching algorithm described above can be changed to be the same one used in previous versions of the FLEXCAN. When the BCC bit in MCR is negated, the matching algorithm stops at the first message buffer with a matching ID that it finds, whether this message buffer is free or not. As a result, the message queueing feature does not work if the BCC bit is negated.

Matching to a range of IDs is possible by using ID Acceptance Masks. FLEXCAN supports individual masking per message buffer. During the matching algorithm, if a mask bit is asserted, then the corresponding ID bit is compared. If the mask bit is negated, the corresponding ID bit is "don't care". Please note that the Individual Mask Registers are implemented in RAM, so they are not initialized out of reset. Also, they can only be programmed if the BCC bit is asserted and while the block is in Freeze Mode.

FLEXCAN also supports an alternate masking scheme with only three mask registers (RGXMASK, RX14MASK and RX15MASK) for backwards compatibility. This alternate masking scheme is enabled when the BCC bit in the MCR Register is negated.

34.4.6 Data Coherence

In order to maintain data coherency and proper FLEXCAN operation, the ARM software must obey the rules described in Transmit Process and Receive Process when accessing a message buffer structure within the block. Violating these rules may cause FLEXCAN to behave in an unpredictable way.

34.4.6.1 Transmission Abort Mechanism

The abort mechanism provides a safe way to request an abort of a pending transmission. A feedback mechanism is provided to inform the ARM if the transmission was aborted or if the frame could not be aborted and was transmitted instead. In order to maintain backwards compatibility, the abort mechanism must be explicitly enabled by asserting the AEN bit in the MCR.

In order to abort a transmission, the ARM must write a specific abort code (1001) to the Code field of the Control and Status word. When the abort mechanism is enabled, the active message buffers configured as transmission must be aborted first and then they may be updated. If the abort code is written to an message buffer that is currently being transmitted, or to an message buffer that was already loaded into the SMB for transmission, the write operation is blocked and the message buffer is not deactivated, but the abort request is captured and kept pending until one of the following conditions are satisfied:

- The block loses the bus arbitration
- There is an error during the transmission
- The block is put into Freeze Mode

If none of conditions above are reached, the message buffer is transmitted correctly, the interrupt flag is set in the IFLAG register and an interrupt to the ARM is generated (if enabled). The abort request is automatically cleared when the interrupt flag is set. On the other hand, if one of the above conditions is reached, the frame is not transmitted, therefore the abort code is written into the Code field, the interrupt flag is set in the IFLAG and an interrupt is (optionally) generated to the ARM.

If the ARM writes the abort code before the transmission begins internally, then the write operation is not blocked, therefore the message buffer is updated and no interrupt flag is set. In this way the ARM just needs to read the abort code to make sure the active message buffer was deactivated. Although the AEN bit is asserted and the ARM wrote

the abort code, in this case the message buffer is deactivated and not aborted, because the transmission did not start yet. One message buffer is only aborted when the abort request is captured and kept pending until one of the previous conditions are satisfied.

The abort procedure can be summarized as follows:

- ARM writes 1001 into the code field of the C/S word
- ARM reads the Code field and compares it to the value that was written
- If the Code field that was read is different from the value that was written, the ARM must read the corresponding IFLAG to check if the frame was transmitted or it is being currently transmitted. If the corresponding IFLAG is set, the frame was transmitted. If the corresponding IFLAG is reset, the ARM must wait for it to be set, and then the ARM must read the Code field to check if the message buffer was aborted (CODE=1001) or it was transmitted (CODE=1000).

34.4.6.2 Message Buffer Deactivation

Deactivation is a mechanism provided to maintain data coherence when the ARM writes to the Control and Status word of active message buffers out of Freeze Mode. Any ARM write access to the Control and Status word of an message buffer causes that message buffer to be excluded from the transmit or receive processes during the current matching or arbitration round. The deactivation is temporary, affecting only for the current match/arbitration round.

The purpose of deactivation is data coherency. The match/arbitration process scans the message buffers to decide which message buffer to transmit or receive. If the ARM updates the message buffer in the middle of a match or arbitration process, the data of that message buffer may no longer be coherent, therefore deactivation of that message buffer is done.

Even with the coherence mechanism described above, writing to the Control and Status word of active message buffers when not in Freeze Mode may produce undesirable results. Examples are:

- Matching and arbitration are one-pass processes. If message buffers are deactivated after they are scanned, no re-evaluation is done to determine a new match/winner. If an Rx message buffer with a matching ID is deactivated during the matching process after it was scanned, then this message buffer is marked as invalid to receive the frame, and FLEXCAN will keep looking for another matching message buffer within the ones it has not scanned yet. If it can not find one, then the message will be lost. Suppose, for example, that two message buffers have a matching ID to a received frame, and the user deactivated the first matching message buffer after FLEXCAN

has scanned the second. The received frame will be lost even if the second matching message buffer was "free to receive".

- If a Tx message buffer containing the lowest ID is deactivated after FLEXCAN has scanned it, then FLEXCAN will look for another winner within the message buffers that it has not scanned yet. Therefore, it may transmit an message buffer with ID that may not be the lowest at the time because a lower ID might be present in one of the message buffers that it had already scanned before the deactivation.
- There is a point in time until which the deactivation of a Tx message buffer causes it not to be transmitted (end of move-out). After this point, it is transmitted but no interrupt is issued and the Code field is not updated. In order to avoid this situation, the abort procedures described in should be used.

34.4.6.3 Message Buffer Lock Mechanism

Besides message buffer deactivation, FLEXCAN has another data coherence mechanism for the receive process. When the ARM reads the Control and Status word of an "active not empty" Rx message buffer, FLEXCAN assumes that the ARM wants to read the whole message buffer in an atomic operation, and thus it sets an internal lock flag for that message buffer. The lock is released when the ARM reads the Free Running Timer (global unlock operation), or when it reads the Control and Status word of another message buffer. The message buffer locking is done to prevent a new frame to be written into the message buffer while the ARM is reading it.

NOTE

The locking mechanism only applies to Rx message buffers which have a code different than INACTIVE (0000) or EMPTY[2] (0100). Also, Tx message buffers can not be locked.

Suppose, for example, that the FIFO is disabled and the second and the fifth message buffers of the array are programmed with the same ID, and FLEXCAN has already received and stored messages into these two message buffers. Suppose now that the ARM decides to read message buffer number 5 and at the same time another message with the same ID is arriving. When the ARM reads the Control and Status word of message buffer number 5, this message buffer is locked. The new message arrives and the matching algorithm finds out that there are no "free to receive" message buffers, so it decides to override message buffer number 5. However, this message buffer is locked, so the new message can not be written there. It will remain in the SMB waiting for the message buffer to be unlocked, and only then will be written to the message buffer. If the message buffer is not unlocked in time and yet another new message with the same ID arrives,

2. In previous FLEXCAN versions, reading the C/S word locked the message buffer even if it was EMPTY. This behavior will be honoured when the BCC bit is ne

then the new message overwrites the one on the SMB and there will be no indication of lost messages either in the Code field of the message buffer or in the Error and Status Register.

While the message is being moved-in from the SMB to the message buffer, the BUSY bit on the Code field is asserted. If the ARM reads the Control and Status word and finds out that the BUSY bit is set, it should defer accessing the message buffer until the BUSY bit is negated.

NOTE

If the BUSY bit is asserted or if the message buffer is empty, then reading the Control and Status word does not lock the message buffer.

Deactivation takes precedence over locking. If the ARM deactivates a locked Rx message buffer, then its lock status is negated and the message buffer is marked as invalid for the current matching round. Any pending message on the SMB will not be transferred anymore to the message buffer.

34.4.7 Rx FIFO

The receive-only FIFO is enabled by asserting the FEN bit in the MCR. The reset value of this bit is zero to maintain software backwards compatibility with previous versions of the block that did not have the FIFO feature. When the FIFO is enabled, the memory region normally occupied by the first 8 Message Buffers (0x80-0xFF) is now reserved for use of the FIFO engine. Management of read and write pointers is done internally by the FIFO engine. The ARM can read the received frames sequentially, in the order they were received, by repeatedly accessing a Message Buffer structure at the beginning of the memory.

The FIFO can store up to 6 frames pending service by the ARM. An interrupt is sent to the ARM when new frames are available in the FIFO. Upon receiving the interrupt, the ARM must read the frame (accessing an message buffer in the 0x80 address) and then clear the interrupt. The act of clearing the interrupt triggers the FIFO engine to replace the message buffer in 0x80 with the next frame in the queue, and then issue another interrupt to the ARM. If the FIFO is full and more frames continue to be received, an OVERFLOW interrupt is issued to the ARM and subsequent frames are not accepted until the ARM creates space in the FIFO by reading one or more frames. A warning interrupt is also generated when 4frames are accumulated in the FIFO.

A powerful filtering scheme is provided to accept only frames intended for the target application, thus reducing the interrupt servicing work load. The filtering criteria is specified by programming a table of 8 32-bit registers that can be configured to one of the following formats:

- Format A: 8 extended or standard IDs (including IDE and RTR)
- Format B: 16 standard IDs or 16 extended 14-bit ID slices (including IDE and RTR)
- Format C: 32 standard or extended 8-bit ID slices

NOTE

A chosen format is applied to all 8 registers of the filter table. It is not possible to mix formats within the table.

The eight elements of the filter table are individually affected by the first eight Individual Mask Registers (RXIMR0 - RXIMR7), allowing very powerful filtering criteria to be defined. The rest of the RXIMR, starting from RXIM8, continue to affect the regular message buffers, starting from MB8. If the BCC bit is negated, then the FIFO filter table is affected by the legacy mask registers as follows: element 6 is affected by RX14MASK, element 7 is affected by RX15MASK and the other elements (0 to 5) are affected by RXGMASK.

34.4.8 CAN Protocol Related Features

34.4.8.1 Overload Frames

FLEXCAN does transmit overload frames due to detection of the following conditions on CAN bus:

- Detection of a dominant bit in the first/second bit of Intermission
- Detection of a dominant bit at the 7th bit (last) of End of Frame field (Rx frames)
- Detection of a dominant bit at the 8th bit (last) of Error Frame Delimiter or Overload Frame Delimiter

34.4.8.2 Time Stamp

The value of the Free Running Timer is sampled at the beginning of the Identifier field on the CAN bus, and is stored at the end of "move-in" in the TIME STAMP field, providing network behavior with respect to time.

Note that the Free Running Timer can be reset upon a specific frame reception, enabling network time synchronization. Refer to TSYN description in Control Register (FLEXCAN_CTRL).

34.4.8.3 Protocol Timing

The FLEXCAN supports a variety of means to setup bit timing parameters that are required by the CAN protocol. The Control Register has various fields used to control bit timing parameters: PRESDIV, PROPSEG, PSEG1, PSEG2 and RJW. See Control Register (FLEXCAN_CTRL) .

The PRESDIV field controls a prescaler that generates the Serial Clock (Sclock), whose period defines the "time quantum" used to compose the CAN waveform. A time quantum is the atomic unit of time handled by the CAN engine.

$$f_{Tq} = \frac{f_{CANCLK}}{(Prescaler\ \ value)}$$

A bit time is subdivided into three segments[3]

- SYNC_SEG: This segment has a fixed length of one time quantum. Signal edges are expected to happen within this section
- Time Segment 1: This segment includes the Propagation Segment and the Phase Segment 1 of the CAN standard. It can be programmed by setting the PROPSEG and the PSEG1 fields of the CTRL Register so that their sum (plus 2) is in the range of 4 to 16 time quanta
- Time Segment 2: This segment represents the Phase Segment 2 of the CAN standard. It can be programmed by setting the PSEG2 field of the CTRL Register (plus 1) to be 2 to 8 time quanta long

$$Bit\ \ Rate = \frac{f_{Tq}}{(number\ \ of\ \ Time\ \ Quanta)}$$

3. For further explanation of the underlying concepts please refer to ISO/DIS 11519-1, Section 10.3. Reference also the Bosch CAN 2.0A/B protocol specification dated September 1991 for bit timing.

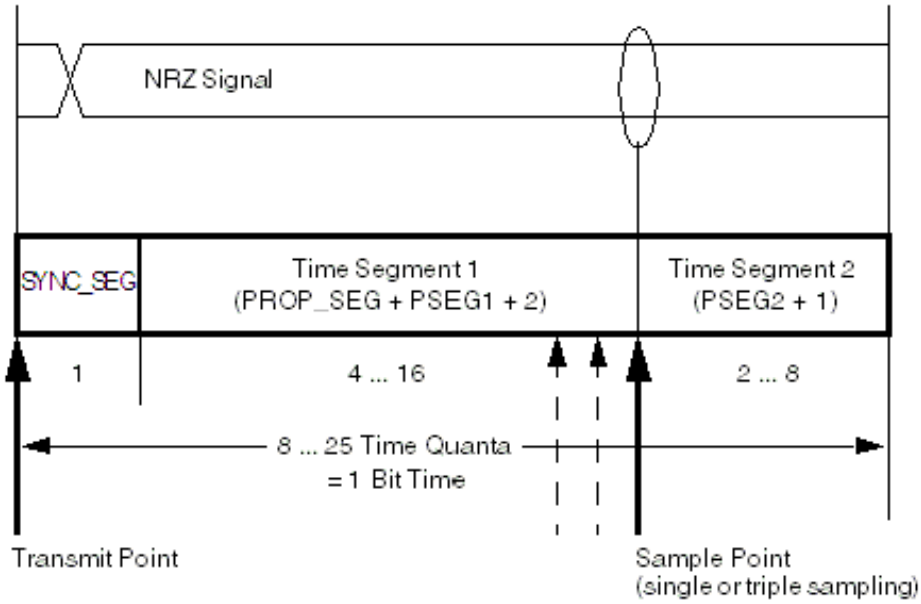


Figure 34-2. Segments within the Bit Time

Table 34-8. Time Segment Syntax

Syntax	Description
SYNC_SEG	System expects transitions to occur on the bus during this period.
Transmit Point	A node in transmit mode transfers a new value to the CAN bus at this point.
Sample Point	A node samples the bus at this point. If the three samples per bit option is selected, then this point marks the position of the third sample.

Table 34-9 gives an overview of the CAN compliant segment settings and the related parameter values.

Table 34-9. CAN Standard Compliant Bit Time Segment Settings

Time Segment 1	Time Segment 2	Re-synchronization Jump Width
5 .. 10	2	1 .. 2
4 .. 11	3	1 .. 3
5 .. 12	4	1 .. 4
6 .. 13	5	1 .. 4
7 .. 14	6	1 .. 4
8 .. 15	7	1 .. 4
9 .. 16	8	1 .. 4

NOTE

It is the user's responsibility to ensure the bit time settings are in compliance with the CAN standard. For bit time calculations,

use an IPT (Information Processing Time) of 2, which is the value implemented in the FLEXCAN block.

34.4.8.4 Arbitration and Matching Timing

During normal transmission or reception of frames, the arbitration, matching, move-in and move-out processes are executed during certain time windows inside the CAN frame, as shown in Figure 34-3.

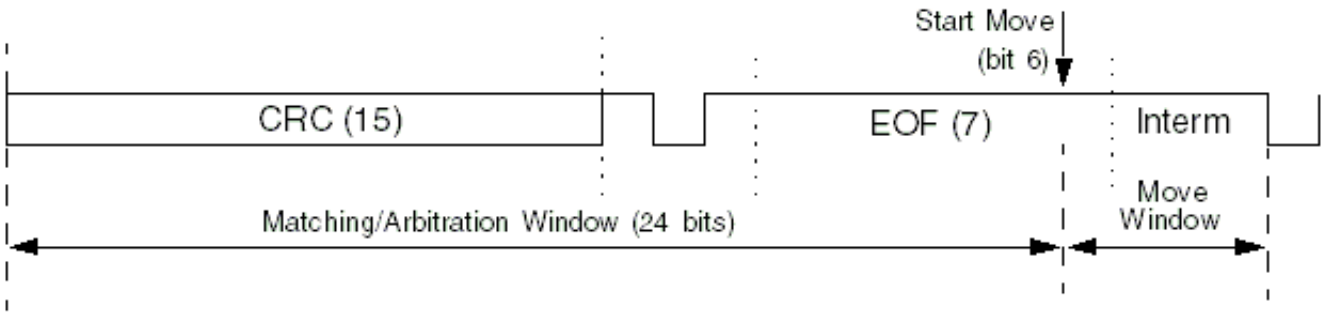


Figure 34-3. Arbitration, Match and Move Time Windows

When doing matching and arbitration, FLEXCAN needs to scan the whole Message Buffer memory during the available time slot. In order to have sufficient time to do that, the following requirements must be observed:

- A valid CAN bit timing must be programmed, as indicated in Table 34-9
- The peripheral clock frequency can not be smaller than the oscillator clock frequency, that is, the PLL can not be programmed to divide down the oscillator clock
- There must be a minimum ratio between the peripheral clock frequency and the CAN bit rate, as specified in Table 34-10.

Table 34-10. Minimum Ratio Between Peripheral Clock Frequency and CAN Bit Rate

Number of Message Buffers	Minimum Ratio
16	8
32	8
64	16

A direct consequence of the first requirement is that the minimum number of time quanta per CAN bit must be 8, so the oscillator clock frequency should be at least 8 times the CAN bit rate. The minimum frequency ratio specified in Table 34-10. can be achieved by choosing a high enough peripheral clock frequency when compared to the oscillator clock

frequency, or by adjusting one or more of the bit timing parameters (PRESDIV, PROPSEG, PSEG1, PSEG2). As an example, taking the case of 64 Mbytes, if the oscillator and peripheral clock frequencies are equal and the CAN bit timing is programmed to have 8 time quanta per bit, then the prescaler factor (PRESDIV + 1) should be at least 2. For prescaler factor equal to one and CAN bit timing with 8 time quanta per bit, the ratio between peripheral and oscillator clock frequencies should be at least 2.

34.4.9 Modes of Operation Details

34.4.9.1 Freeze Mode

This mode is entered by asserting the HALT bit in the MCR Register or when the ARM platform is put into Debug Mode. In both cases it is also necessary that the FRZ bit is asserted in the MCR Register and the block is not in any of the low power modes (Disable, Stop). When Freeze Mode is requested during transmission or reception, FLEXCAN does the following:

- Waits to be in either Intermission, Passive Error, Bus Off or Idle state
- Waits for all internal activities like arbitration, matching, move-in and move-out to finish
- Ignores the Rx input pin and drives the Tx pin as recessive
- Stops the prescaler, thus halting all CAN protocol activities
- Grants write access to the Error Counters Register, which is read-only in other modes
- Sets the NOT_RDY and FRZ_ACK bits in MCR

After requesting Freeze Mode, the user must wait for the FRZ_ACK bit to be asserted in MCR before executing any other action, otherwise FLEXCAN may operate in an unpredictable way. In Freeze mode, all memory mapped registers are accessible.

Exiting Freeze Mode is done in one of the following ways:

- ARM negates the FRZ bit in the MCR Register
- The ARM platform is removed from Debug Mode and/or the HALT bit is negated

Once out of Freeze Mode, FLEXCAN tries to re-synchronize to the CAN bus by waiting for 11 consecutive recessive bits.

34.4.9.2 Block Disable Mode

This low power mode is entered when the MDIS bit in the MCR Register is asserted. If the block is disabled during Freeze Mode, it shuts down the clocks to the CPI and MBM sub-blocks, sets the LPM_ACK bit and negates the FRZ_ACK bit. If the block is disabled during transmission or reception, FLEXCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and then checks it to be recessive
- Waits for all internal activities like arbitration, matching, move-in and move-out to finish
- Ignores its Rx input pin and drives its Tx pin as recessive
- Shuts down the clocks to the CPI and MBM sub-blocks
- Sets the NOT_RDY and LPM_ACK bits in MCR

The Bus Interface Unit continues to operate, enabling the ARM to access memory mapped registers, except the Free Running Timer, the Error Counter Register and the Message Buffers, which cannot be accessed when the block is in Disable Mode. Exiting from this mode is done by negating the MDIS bit, which will resume the clocks and negate the LPM_ACK bit.

34.4.9.3 Stop Mode

This is a system low power mode in which all ARM platform clocks are stopped for maximum power savings. If FLEXCAN receives the global Stop Mode request during Freeze Mode, it sets the LPM_ACK bit, negates the FRZ_ACK bit and then sends a Stop Acknowledge signal to the ARM, in order to shut down the clocks globally. If Stop Mode is requested during transmission or reception, FLEXCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and checks it to be recessive
- Waits for all internal activities like arbitration, matching, move-in and move-out to finish
- Ignores its Rx input pin and drives its Tx pin as recessive
- Sets the NOT_RDY and LPM_ACK bits in MCR
- Sends a Stop Acknowledge signal to the ARM, so that it can shut down the clocks globally

Exiting Stop Mode is done in one of the following ways:

- ARM resuming the clocks and removing the Stop Mode request
- ARM resuming the clocks and Stop Mode request as a result of the Self Wake mechanism

In the Self Wake mechanism, if the SLF_WAK bit in MCR Register was set at the time FLEXCAN entered Stop Mode, then upon detection of a recessive to dominant transition on the CAN bus, FLEXCAN sets the WAK_INT bit in the ESR Register and, if enabled by the WAK_MSK bit in MCR, generates a Wake Up interrupt to the ARM. Upon receiving the interrupt, the ARM should resume the clocks and remove the Stop Mode request. FLEXCAN will then wait for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, it will not receive the frame that woke it up. Table 34-11 details the effect of SLF_WAK and WAK_MSK upon wake-up from Stop Mode. Note that wake-up from Stop Mode only works when both bits are asserted.

Table 34-11. Wake-up from Stop Mode

SLF_WAK	WAK_MSK	ARM Platform Clocks Enabled	Wake-up Interrupt Generated
0	0	No	No
0	1	No	No
1	0	No	No
1	1	Yes	Yes

The sensitivity to CAN bus activity can be modified by applying a low-pass filter function to the Rx CAN input line while in Stop Mode. See the WAK_SRC bit in. This feature can be used to protect FLEXCAN from waking up due to short glitches on the CAN bus lines. Such glitches can result from electromagnetic interference within noisy environments.

34.4.10 Interrupts

The block can generate 70 interrupt sources (64 interrupts due to message buffers and 6 interrupts due to ORed interrupts from message buffers, Bus Off, Error, Tx Warning, Rx Warning and Wake Up).

Each one of the message buffers can be an interrupt source, if its corresponding IMASK bit is set. There is no distinction between Tx and Rx interrupts for a particular buffer, under the assumption that the buffer is initialized for either transmission or reception. Each of the buffers has assigned a flag bit in the IFLAG Registers. The bit is set when the corresponding buffer completes a successful transmission/reception and is cleared when the ARM writes it to '1' (unless another interrupt is generated at the same time).

NOTE

It must be guaranteed that the ARM only clears the bit causing the current interrupt. For this reason, bit manipulation instructions (BSET) must not be used to clear interrupt flags.

These instructions may cause accidental clearing of interrupt flags which are set after entering the current interrupt service routine.

If the Rx FIFO is enabled (bit FEN on MCR set), the interrupts corresponding to message buffers 0 to 7 have a different behavior. Bit 7 of the IFLAG1 becomes the "FIFO Overflow" flag; bit 6 becomes the FIFO Warning flag, bit 5 becomes the "Frames Available in FIFO flag" and bits 4-0 are unused. See Control Register (FLEXCAN_CTRL).

A combined interrupt for all message buffers is generated by an Or of all the interrupt sources from message buffers. This interrupt gets generated when any of the message buffers generates an interrupt. In this case the ARM must read the IFLAG Registers to determine which message buffer caused the interrupt.

The other 5 interrupt sources (Bus Off, Error, Tx Warning, Rx Warning and Wake Up) generate interrupts like the message buffer ones, and can be read from the Error and Status Register. The Bus Off, Error, Tx Warning and Rx Warning interrupt mask bits are located in the Control Register, and the Wake-Up interrupt mask bit is located in the MCR.

34.5 Initialization/Application Information

This section provides instructions for initializing the FLEXCAN.

34.5.1 FLEXCAN Initialization Sequence

The FLEXCAN may be reset in two ways:

- ARM platform level hard reset which resets all memory mapped registers asynchronously
- SOFT_RST bit in MCR, which resets some of the memory mapped registers synchronously (refer to the Memory Map table to see what registers are affected by soft reset)

Soft reset is synchronous and has to follow an internal request/acknowledge procedure across clock domains. Therefore, it may take some time to fully propagate its effects. The SOFT_RST bit remains asserted while soft reset is pending, so software can poll this bit to know when the reset has completed. Also, soft reset can not be applied while clocks are shut down in any of the low power modes. The low power mode should be exited and the clocks resumed before applying soft reset.

The clock source (CLK_SRC bit) should be selected while the block is in Disable Mode. After the clock source is selected and the block is enabled (MDIS bit negated), FLEXCAN automatically goes to Freeze Mode. In Freeze Mode, FLEXCAN is un-synchronized to the CAN bus, the HALT and FRZ bits in MCR Register are set, the internal state machines are disabled and the FRZ_ACK and NOT_RDY bits in the MCR Register are set. The Tx pin is in recessive state and FLEXCAN does not initiate any transmission or reception of CAN frames. Note that the Message Buffers and the Rx Individual Mask Registers are not affected by reset, so they are not automatically initialized.

For any configuration change/initialization it is required that FLEXCAN is put into Freeze Mode (see). The following is a generic initialization sequence applicable to the FLEXCAN:

- Initialize the Module Configuration Register
  - Enable the individual filtering per message buffer and reception queue features by setting the BCC bit
  - Enable the warning interrupts by setting the WRN_EN bit
  - If required, disable frame self reception by setting the SRX_DIS bit
  - Enable the FIFO by setting the FEN bit
  - Enable the abort mechanism by setting the AEN bit
  - Enable the local priority feature by setting the LPRIO_EN bit
- Initialize the Control Register
  - Determine the bit timing parameters: PROPSEG, PSEG1, PSEG2, RJW
  - Determine the bit rate by programming the PRESDIV field
  - Determine the internal arbitration mode (LBUF bit)
- Initialize the Message Buffers
  - The Control and Status word of all Message Buffers must be initialized
  - If FIFO was enabled, the 8-entry ID table must be initialized
  - Other entries in each Message Buffer should be initialized as required
- Initialize the Rx Individual Mask Registers
- Set required interrupt mask bits in the IMASK Registers (for all message buffer interrupts), in CTRL Register (for Bus Off and Error interrupts) and in MCR Register for Wake-Up interrupt
- Negate the HALT bit in MCR

Starting with the last event, FLEXCAN attempts to synchronize to the CAN bus.

34.6 Programmable Registers

This section describes the registers and data structures in the FLEXCAN. The base address of the FLEXCAN-1 is 0x53FE_8000, the base of the FLEXCAN-2 is 0x53FE_C000. The addresses presented here are relative to the base address.

The address space occupied by FLEXCAN has 96 bytes for registers starting at the block base address, followed by message buffer storage space in embedded RAM starting at address 0x080, and an extra ID Mask storage space in a separate embedded RAM starting at address 0x880.

The complete memory map for a FLEXCAN with 64 Mbytes capability is shown below. Each individual register is identified by its complete name and the corresponding mnemonic. The access type can be Supervisor (S) or Unrestricted (U). Most of the registers can be configured to have either Supervisor or Unrestricted access by programming the SUPV bit in the MCR Register. These registers are identified as S/U in the Access column shown below.

The Rx Global Mask (RXGMASK), Rx Buffer 14 Mask (RX14MASK) and the Rx Buffer 15 Mask (RX15MASK) registers are provided for backwards compatibility, and are not used when the BCC bit in MCR is asserted.

The address ranges 0x080, 0x47F and 0x880, 0x97F are occupied by two separate embedded memories. These two ranges are completely occupied by RAM (1056 and 256 bytes, respectively).

FLEXCAN memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FC_8000	Module Configuration Register (FLEXCAN-1_MCR)	32	R/W	5980_000Fh	34.6.1/ 1673
53FC_8004	Control Register (FLEXCAN-1_CTRL)	32	R/W	0000_0000h	34.6.2/ 1677
53FC_8008	Free Running Timer (FLEXCAN-1_TIMER)	32	R/W	0000_0000h	34.6.3/ 1679
53FC_8010	Rx Global Mask (FLEXCAN-1_RXGMASK)	32	R/W	FFFF_ FFFFh	34.6.4/ 1680
53FC_8014	Rx 14 Mask (FLEXCAN-1_RX14MASK)	32	R/W	FFFF_ FFFFh	34.6.5/ 1681
53FC_8018	Rx 15 Mask (FLEXCAN-1_RX15MASK)	32	R/W	FFFF_ FFFFh	34.6.6/ 1681
53FC_801C	Error Counter Register (FLEXCAN-1_ECR)	32	R/W	0000_0000h	34.6.7/ 1682
53FC_8020	Error and Status Register (FLEXCAN-1_ESR)	32	R/W	0000_0000h	34.6.8/ 1684

Table continues on the next page...

FLEXCAN memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FC_8024	Interrupt Masks 2 Register (FLEXCAN-1_IMASK2)	32	R/W	0000_0000h	34.6.9/ 1686
53FC_8028	Interrupt Masks 1 Register (FLEXCAN-1_IMASK1)	32	R/W	0000_0000h	34.6.10/ 1687
53FC_802C	Interrupt Flags 2 Register (FLEXCAN-1_IFLAG2)	32	R/W	0000_0000h	34.6.11/ 1688
53FC_8030	Interrupt Flags 1 Register (FLEXCAN-1_IFLAG1)	32	R/W	0000_0000h	34.6.12/ 1688
53FC_8034	Glitch Filter Width Register (FLEXCAN-1_GFWR)	32	R/W	0000_007Fh	34.6.13/ 1689
53FC_8880	Rx Individual Mask Registers (FLEXCAN-1_RX0IMR)	32	R/W	0000_0000h	34.6.14/ 1690
53FC_897C	Rx Individual Mask Registers (FLEXCAN-1_RX63IMR)	32	R/W	0000_0000h	34.6.14/ 1690
53FC_C000	Module Configuration Register (FLEXCAN-2_MCR)	32	R/W	5980_000Fh	34.6.1/ 1673
53FC_C004	Control Register (FLEXCAN-2_CTRL)	32	R/W	0000_0000h	34.6.2/ 1677
53FC_C008	Free Running Timer (FLEXCAN-2_TIMER)	32	R/W	0000_0000h	34.6.3/ 1679
53FC_C010	Rx Global Mask (FLEXCAN-2_RXGMASK)	32	R/W	FFFF_ FFFFh	34.6.4/ 1680
53FC_C014	Rx 14 Mask (FLEXCAN-2_RX14MASK)	32	R/W	FFFF_ FFFFh	34.6.5/ 1681
53FC_C018	Rx 15 Mask (FLEXCAN-2_RX15MASK)	32	R/W	FFFF_ FFFFh	34.6.6/ 1681
53FC_C01C	Error Counter Register (FLEXCAN-2_ECR)	32	R/W	0000_0000h	34.6.7/ 1682
53FC_C020	Error and Status Register (FLEXCAN-2_ESR)	32	R/W	0000_0000h	34.6.8/ 1684
53FC_C024	Interrupt Masks 2 Register (FLEXCAN-2_IMASK2)	32	R/W	0000_0000h	34.6.9/ 1686
53FC_C028	Interrupt Masks 1 Register (FLEXCAN-2_IMASK1)	32	R/W	0000_0000h	34.6.10/ 1687
53FC_C02C	Interrupt Flags 2 Register (FLEXCAN-2_IFLAG2)	32	R/W	0000_0000h	34.6.11/ 1688
53FC_C030	Interrupt Flags 1 Register (FLEXCAN-2_IFLAG1)	32	R/W	0000_0000h	34.6.12/ 1688
53FC_C034	Glitch Filter Width Register (FLEXCAN-2_GFWR)	32	R/W	0000_007Fh	34.6.13/ 1689

Table continues on the next page...

FLEXCAN memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FC_C880	Rx Individual Mask Registers (FLEXCAN-2_RX0IMR)	32	R/W	0000_0000h	34.6.14/ 1690
53FC_C97C	Rx Individual Mask Registers (FLEXCAN-2_RX63IMR)	32	R/W	0000_0000h	34.6.14/ 1690

34.6.1 Module Configuration Register (FLEXCANx_MCR)

This register defines global system configurations, such as the block operation mode (low power, for example) and maximum message buffer configuration. Most of the fields in this register can be accessed at any time, except the MAXMB field, which should only be changed while the block is in Freeze Mode.

Addresses: FLEXCAN-1_MCR is 53FC_8000h base + 0h offset = 53FC_8000h

FLEXCAN-2_MCR is 53FC_C000h base + 0h offset = 53FC_C000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MDIS	FRZ	FEN	HALT	NOT_ RDY	WAK_MSK	SOFT_RST	FRZ_ ACK	SUPV	SLF_WAK	WRN_EN	LPM_ ACK	WAK_SRC	DOZE	SRX_DIS	BCC
Reset	0	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-		LPRIO_EN	AEN	0		IDAM		0		MAXMB					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

FLEXCANx_MCR field descriptions

Field	Description
31 MDIS	This bit controls whether FLEXCAN is enabled or not. When disabled, FLEXCAN shuts down the clocks to the CAN Protocol Interface and Message Buffer Management sub-blocks. This is the only bit in MCR not affected by soft reset. 1 Disable the FLEXCAN block 0 Enable the FLEXCAN block
30 FRZ	The FRZ bit specifies the FLEXCAN behavior when the HALT bit in the MCR Register is set or when Debug Mode is requested at ARM platform level. When FRZ is asserted, FLEXCAN is enabled to enter Freeze Mode. Negation of this bit field causes FLEXCAN to exit from Freeze Mode. 1 Enabled to enter Freeze Mode 0 Not enabled to enter Freeze Mode

Table continues on the next page...

FLEXCANx_MCR field descriptions (continued)

Field	Description
29 FEN	This bit controls whether the FIFO feature is enabled or not. When FEN is set, message buffers 0 to 7 cannot be used for normal reception and transmission because the corresponding memory region (0x80-0xFF) is used by the FIFO engine. 1 FIFO enabled 0 FIFO not enabled
28 HALT	Assertion of this bit puts the FLEXCAN block into Freeze Mode. The ARM should clear it after initializing the Message Buffers and Control Register. No reception or transmission is performed by FLEXCAN before this bit is cleared. While in Freeze Mode, the ARM has write access to the Error Counter Register, that is otherwise read-only. Freeze Mode can not be entered while FLEXCAN is in any of the low power modes. 1 Enters Freeze Mode if the FRZ bit is asserted. 0 No Freeze Mode request.
27 NOT_RDY	This read-only bit indicates that FLEXCAN is either in Disable Mode, Stop Mode or Freeze Mode. It is negated once FLEXCAN has exited these modes. 1 FLEXCAN block is either in Disable Mode, Stop Mode or Freeze Mode 0 FLEXCAN block is either in Normal Mode, Listen-Only Mode or Loop-Back Mode
26 WAK_MSK	This bit enables the Wake Up Interrupt generation. 1 Wake Up Interrupt is enabled 0 Wake Up Interrupt is disabled
25 SOFT_RST	When this bit is asserted, FLEXCAN resets its internal state machines and some of the memory mapped registers. The following registers are reset: MCR (except the MDIS bit), TIMER, ECR, ESR, IMASK1, IMASK2, IFLAG1, IFLAG2. Configuration registers that control the interface to the CAN bus are not affected by soft reset. The following registers are unaffected: CTRL RXIMR0,RXIMR63 RXGMASK, RX14MASK, RX15MASK all Message Buffers The SOFT_RST bit can be asserted directly by the ARM when it writes to the MCR Register. Because soft reset is synchronous and has to follow a request/acknowledge procedure across clock domains, it may take some time to fully propagate its effect. The SOFT_RST bit remains asserted while reset is pending, and is automatically negated when reset completes. Therefore, software can poll this bit to know when the soft reset has completed. Soft reset cannot be applied while clocks are shut down in any of the low power modes. The block should be first removed from low power mode, and then soft reset can be applied. 1 Resets the registers marked as affected by soft reset shown in the memory map 0 No reset request
24 FRZ_ACK	This read-only bit indicates that FLEXCAN is in Freeze Mode and its prescaler is stopped. The Freeze Mode request cannot be granted until current transmission or reception processes have finished. Therefore the software can poll the FRZ_ACK bit to know when FLEXCAN has actually entered Freeze Mode. If Freeze Mode request is negated, then this bit is negated once the FLEXCAN prescaler is running again. If Freeze Mode is requested while FLEXCAN is in any of the low power modes, then the FRZ_ACK bit will only be set when the low power mode is exited.

Table continues on the next page...

FLEXCANx_MCR field descriptions (continued)

Field	Description
	1 FLEXCAN in Freeze Mode, prescaler stopped 0 FLEXCAN not in Freeze Mode, prescaler running
23 SUPV	This bit configures some of the FLEXCAN registers to be either in Supervisor or Unrestricted memory space. The registers affected by this bit are marked as S/U in the Access Type column of the Memory Map. Reset value of this bit is 1, so the affected registers start with Supervisor access restrictions. 1 Affected registers are in Supervisor memory space. Any access without supervisor permission behaves as though the access was done to an unimplemented register location 0 Affected registers are in Unrestricted memory space
22 SLF_WAK	This bit enables the Self Wake Up feature when FLEXCAN is in Stop Mode. If this bit had been asserted by the time FLEXCAN entered Stop Mode, then FLEXCAN will look for a recessive to dominant transition on the bus during these modes. If a transition from recessive to dominant is detected during Stop Mode, then FLEXCAN generates, if enabled to do so, a Wake Up interrupt to the ARM so that it can resume the clocks globally. This bit can not be written while the block is in Stop Mode. 1 FLEXCAN Self Wake Up feature is enabled 0 FLEXCAN Self Wake Up feature is disabled
21 WRN_EN	When asserted, this bit enables the generation of the TWRN_INT and RWRN_INT flags in the Error and Status Register. If WRN_EN is negated, the TWRN_INT and RWRN_INT flags will always be zero, independent of the values of the error counters, and no warning interrupt will ever be generated. 1 TWRN_INT and RWRN_INT bits are set when the respective error counter transition from <96 to ℜ≥ 96. 0 TWRN_INT and RWRN_INT bits are zero, independent of the values in the error counters.
20 LPM_ACK	This read-only bit indicates that FLEXCAN is either in Disable Mode and Stop Mode. Either of these low power modes can not be entered until all current transmission or reception processes have finished, so the ARM can poll the LPM_ACK bit to know when FLEXCAN has actually entered low power mode. 1 FLEXCAN is either in Disable Mode, or Stop mode 0 FLEXCAN not in any of the low power modes
19 WAK_SRC	This bit defines whether the integrated low-pass filter is applied to protect the Rx CAN input. 1 FLEXCAN us
18 DOZE	This bit defines whether FLEXCAN is allowed to enter low power mode when Doze Mode is requested at ARM platform level. This bit is automatically reset when FLEXCAN wakes up from Doze Mode upon detecting activity on the CAN bus (self wake-up enabled). Doze Mode is not supported in i.MX53 1 FLEXCAN is enabled to enter low power mode when Doze Mode is requested 0 FLEXCAN is not enabled to enter low power mode when Doze Mode is requested
17 SRX_DIS	This bit defines whether FLEXCAN is allowed to receive frames transmitted by itself. If this bit is asserted, frames transmitted by the block will not be stored in any message buffers, regardless if the message buffers is programmed with an ID that matches the transmitted frame, and no interrupt flag or interrupt signal will be generated due to the frame reception. 1 Self reception disabled 0 Self reception enabled
16 BCC	This bit is provided to support Backwards Compatibility with previous FLEXCAN versions. When this bit is negated, the following configuration is applied:

Table continues on the next page...

FLEXCANx_MCR field descriptions (continued)

Field	Description
	For ARM platforms supporting individual Rx ID masking, this feature is disabled. Instead of individual ID masking per message buffer, FLEXCAN uses its previous masking scheme with RXGMASK, RX14MASK and RX15MASK. The reception queue feature is disabled. Upon receiving a message, if the first message buffer with a matching ID that is found is still occupied by a previous unread message, FLEXCAN will not look for another matching message buffer. It will override this message buffer with the new message and set the CODE field to 0110 (overrun). Upon reset this bit is negated, allowing legacy software to work without modification. 1 Individual Rx masking and queue feature are enabled. 0 Individual Rx masking and queue feature are disabled.
15–14 -	Reserved
13 LPRIO_EN	This bit is provided for backwards compatibility reasons. It controls whether the local priority feature is enabled or not. It is used to extend the ID used during the arbitration process. With this extended ID concept, the arbitration process is done based on the full 32-bit word, but the actual transmitted ID still has 11-bit for standard frames and 29-bit for extended frames. 1 Local Priority enabled 0 Local Priority disabled
12 AEN	This bit is supplied for backwards compatibility reasons. When asserted, it enables the Tx abort feature. This feature guarantees a safe procedure for aborting a pending transmission, so that no frame is sent in the CAN bus without notification. 1 Abort enabled 0 Abort disabled
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–8 IDAM	This 2-bit field identifies the format of the elements of the Rx FIFO filter table, as shown below. Note that all elements of the table are configured at the same time by this field (they are all the same format). See Rx FIFO Structure. 00 Format A One full ID (standard or extended) per filter element. 01 Format B Two full standard IDs or two partial 14-bit extended IDs per filter element 10 Format C Four partial 8-bit IDs (standard or extended) per filter element. 11 Format D All frames rejected.
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5–0 MAXMB	This 6-bit field defines the maximum number of message buffers that will take part in the matching and arbitration processes. The reset value (0x0F) is equivalent to 16 Mbyte configuration. This field should be changed only while the block is in Freeze Mode. Maximum message buffers in use = MAXMB + 1. MAXMB has to be programmed with a value smaller or equal to the number of available Message Buffers, otherwise FLEXCAN will not transmit or receive frames.

34.6.2 Control Register (FLEXCANx_CTRL)

This register is defined for specific FLEXCAN control features related to the CAN bus, such as bit-rate, programmable sampling point within an Rx bit, Loop Back Mode, Listen Only Mode, Bus Off recovery behavior and interrupt enabling (Bus-Off, Error, Warning). It also determines the Division Factor for the clock prescaler. Most of the fields in this register should only be changed while the block is in Disable Mode or in Freeze Mode. Exceptions are the BOFF_MSK, ERR_MSK, TWRN_MSK, RWRN_MSK and BOFF_REC bits, that can be accessed at any time.

Addresses: FLEXCAN-1_CTRL is 53FC_8000h base + 4h offset = 53FC_8004h

FLEXCAN-2_CTRL is 53FC_C000h base + 4h offset = 53FC_C004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PRESDIV								RJW		PSEG1			PSEG2		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BOFF_MSK	ERR_MSK	-	LPB	TWRN_MSK	RWRN_MSK	0		SMP	BOFF_REC	TSYN	LBUF	LOM	PROP_SEG		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXCANx_CTRL field descriptions

Field	Description
31–24 PRESDIV	This 8-bit field defines the ratio between the CPI clock frequency and the Serial Clock (Sclock) frequency. The Sclock period defines the time quantum of the CAN protocol. For the reset value, the Sclock frequency is equal to the CPI clock frequency. The Maximum value of this register is 0xFF, that gives a minimum Sclock frequency equal to the CPI clock frequency divided by 256. For more information refer to Protocol Timing. Sclock frequency = CPI clock frequency / (PRESDIV + 1)
23–22 RJW	This 2-bit field defines the maximum number of time quanta[1] that a bit time can be changed by one re-synchronization. The valid programmable values are 0,3. Resync Jump Width = RJW + 1.
21–19 PSEG1	This 3-bit field defines the length of Phase Buffer Segment 1 in the bit time. The valid programmable values are 0,7. Phase Buffer Segment 1 = (PSEG1 + 1) x Time-Quanta.
18–16 PSEG2	This 3-bit field defines the length of Phase Buffer Segment 2 in the bit time. The valid programmable values are 1,7. Phase Buffer Segment 2 = (PSEG2 + 1) x Time-Quanta.

Table continues on the next page...

FLEXCANx_CTRL field descriptions (continued)

Field	Description
15 BOFF_MSK	This bit provides a mask for the Bus Off Interrupt. 1 Bus Off interrupt enabled 0 Bus Off interrupt disabled
14 ERR_MSK	This bit provides a mask for the Error Interrupt. 1 Error interrupt enabled 0 Error interrupt disabled
13 -	Reserved 1 The CAN engine clock source is the bus clock, 66.5MHz 0 The CAN engine clock source is the oscillator clock, 24.576MHz
12 LPB	This bit configures FLEXCAN to operate in Loop-Back Mode. In this mode, FLEXCAN performs an internal loop back that can be used for self test operation. The bit stream output of the transmitter is fed back internally to the receiver input. The Rx CAN input pin is ignored and the Tx CAN output goes to the recessive state (logic 1). FLEXCAN behaves as it normally does when transmitting, and treats its own transmitted message as a message received from a remote node. In this mode, FLEXCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field, generating an internal acknowledge bit to ensure proper reception of its own message. Both transmit and receive interrupts are generated. 1 Loop Back enabled 0 Loop Back disabled
11 TWRN_MSK	This bit provides a mask for the Tx Warning Interrupt associated with the TWRN_INT flag in the Error and Status Register. This bit has no effect if the WRN_EN bit in MCR is negated and it is read as zero when WRN_EN is negated. 1 Tx Warning Interrupt enabled 0 Tx Warning Interrupt disabled
10 RWRN_MSK	This bit provides a mask for the Rx Warning Interrupt associated with the RWRN_INT flag in the Error and Status Register. This bit has no effect if the WRN_EN bit in MCR is negated and it is read as zero when WRN_EN is negated. 1 Rx Warning Interrupt enabled 0 Rx Warning Interrupt disabled
9–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7 SMP	This bit defines the sampling mode of CAN bits at the Rx input. 1 Three samples are used to determine the value of the received bit: the regular one (sample point) and 2 preceding samples, a majority rule is used 0 Just one sample is used to determine the bit value
6 BOFF_REC	This bit defines how FLEXCAN recovers from Bus Off state. If this bit is negated, automatic recovering from Bus Off state occurs according to the CAN Specification 2.0B. If the bit is asserted, automatic recovering from Bus Off is disabled and the block remains in Bus Off state until the bit is negated by the user. If the negation occurs before 128 sequences of 11 recessive bits are detected on the CAN bus, then Bus Off recovery happens as if the BOFF_REC bit had never been asserted. If the negation occurs after 128 sequences of 11 recessive bits occurred, then FLEXCAN will re-synchronize to the bus by waiting for 11 recessive bits before joining the bus. After negation, the BOFF_REC bit can be re-asserted again during Bus Off, but it will only be effective the next time the block enters Bus Off. If BOFF_REC was

Table continues on the next page...

FLEXCANx_CTRL field descriptions (continued)

Field	Description
	negated when the block entered Bus Off, asserting it during Bus Off will not be effective for the current Bus Off recovery. 1 Automatic recovering from Bus Off state disabled 0 Automatic recovering from Bus Off state enabled, according to CAN Spec 2.0 part B
5 TSYN	This bit enables a mechanism that resets the free-running timer each time a message is received in Message Buffer 0. This feature provides a means to synchronize multiple FLEXCAN stations with a special SYNC message (for example, global network time). If the FEN bit in MCR is set (FIFO enabled), MB8 is used for timer synchronization instead of MB0. 1 Timer Sync feature enabled 0 Timer Sync feature disabled
4 LBUF	This bit defines the ordering mechanism for Message Buffer transmission. When asserted, the LPRIO_EN bit does not affect the priority arbitration. 1 Lowest number buffer is transmitted first 0 Buffer with highest priority is transmitted first
3 LOM	This bit configures FLEXCAN to operate in Listen Only Mode. In this mode, transmission is disabled, all error counters are frozen and the block operates in a CAN Error Passive mode [Ref. 1]. Only messages acknowledged by another CAN station will be received. If FLEXCAN detects a message that has not been acknowledged, it will flag a BIT0 error (without changing the REC), as if it was trying to acknowledge the message. 1 FLEXCAN block operates in Listen Only Mode 0 Listen Only Mode is deactivated
2–0 PROP_SEG	This 3-bit field defines the length of the Propagation Segment in the bit time. The valid programmable values are 0,7. Propagation Segment Time = (PROPSEG + 1) * Time-Quanta. Time-Quantum = one Sclock period.

1. One time quantum is equal to the Sclock period.

34.6.3 Free Running Timer (FLEXCANx_TIMER)

This register represents a 16-bit free running counter that can be read and written by the ARM. The timer starts from 0x0000 after Reset, counts linearly to 0xFFFF, and wraps around.

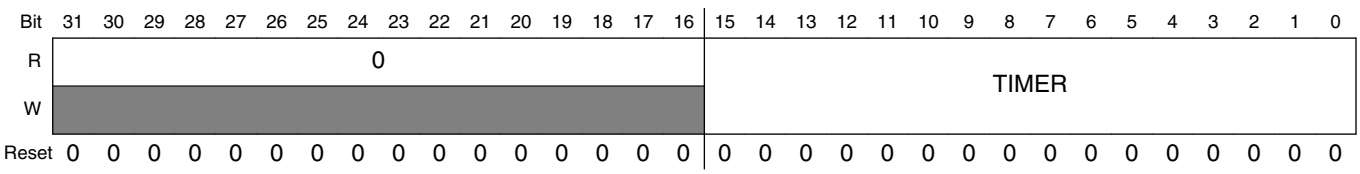
The timer is clocked by the FLEXCAN bit-clock (which defines the baud rate on the CAN bus). During a message transmission/reception, it increments by one for each bit that is received or transmitted. When there is no message on the bus, it counts using the previously programmed baud rate. During Freeze Mode, the timer is not incremented.

The timer value is captured at the beginning of the identifier field of any frame on the CAN bus. This captured value is written into the Time Stamp entry in a message buffer after a successful reception or transmission of a message.

Writing to the timer is an indirect operation. The data is first written to an auxiliary register and then an internal request/acknowledge procedure across clock domains is executed. All this is transparent to the user, except for the fact that the data will take some time to be actually written to the register. If desired, software can poll the register to discover when the data was actually written.

Addresses: FLEXCAN-1_TIMER is 53FC_8000h base + 8h offset = 53FC_8008h

FLEXCAN-2_TIMER is 53FC_C000h base + 8h offset = 53FC_C008h



FLEXCANx_TIMER field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. This read-only bitfield is reserved and always has the value zero.
15–0 TIMER	Timer value Contains the free-running counter value.

34.6.4 Rx Global Mask (FLEXCANx_RXGMASK)

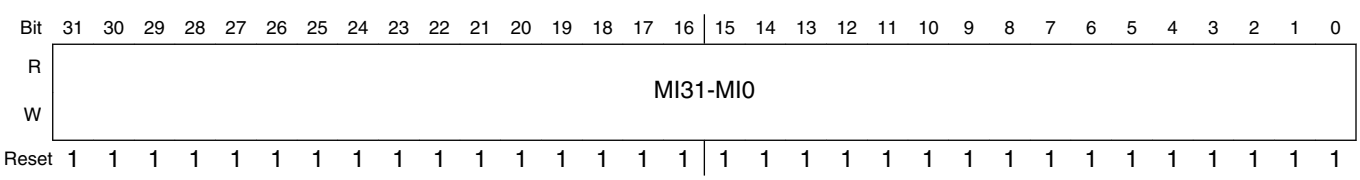
Supports individual masks per message buffer, setting the BCC bit in MCR causes the RXGMASK Register to have no effect on the block operation.

RXGMASK is used as acceptance mask for all Rx message buffers, excluding message buffers 14, 15, which have individual mask registers. When the FEN bit in MCR is set (FIFO enabled), the RXGMASK also applies to all elements of the ID filter table, except elements 6-7, which have individual masks.

The contents of this register must be programmed while the block is in Freeze Mode, and must not be modified when the block is transmitting or receiving frames.

Addresses: FLEXCAN-1_RXGMASK is 53FC_8000h base + 10h offset = 53FC_8010h

FLEXCAN-2_RXGMASK is 53FC_C000h base + 10h offset = 53FC_C010h



FLEXCANx_RXGMASK field descriptions

Field	Description
31–0 MI31-MI0	For normal Rx message buffers, the mask bits affect the ID filter programmed on the message buffer. For the Rx FIFO, the mask bits affect all bits programmed in the filter table (ID, IDE, RTR). 1 The corresponding bit in the filter is checked against the one received 0 the corresponding bit in the filter is dont care

34.6.5 Rx 14 Mask (FLEXCANx_RX14MASK)

Setting the BCC bit in MCR causes the RX14MASK Register to have no effect on the block operation.

RX14MASK is used as acceptance mask for the Identifier in Message Buffer 14. When the FEN bit in MCR is set (FIFO enabled), the RXG14MASK also applies to element 6 of the ID filter table. This register has the same structure as the Rx Global Mask Register. It must be programmed while the block is in Freeze Mode, and must not be modified when the block is transmitting or receiving frames.

- Address Offset: 0x14
- Reset Value: 0xFFFF_FFFF

Addresses: FLEXCAN-1_RX14MASK is 53FC_8000h base + 14h offset = 53FC_8014h
FLEXCAN-2_RX14MASK is 53FC_C000h base + 14h offset = 53FC_C014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MI31-MI0																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

FLEXCANx_RX14MASK field descriptions

Field	Description
31–0 MI31-MI0	Acceptance mask for the Identifier in Message Buffer 14

34.6.6 Rx 15 Mask (FLEXCANx_RX15MASK)

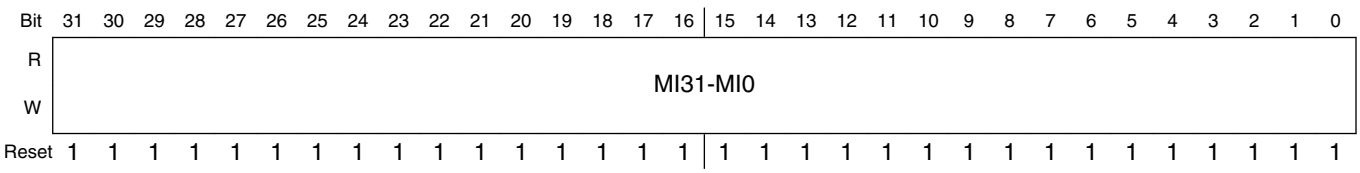
Setting the BCC bit in MCR causes the RX15MASK Register to have no effect on the block operation.

When the BCC bit is negated, RX15MASK is used as acceptance mask for the Identifier in Message Buffer 15. When the FEN bit in MCR is set (FIFO enabled), the RXG14MASK also applies to element 7 of the ID filter table. This reg

structure as the Rx Global Mask Register. It must be programmed while the block is in Freeze Mode, and must not be modified when the block is transmitting or receiving frames.

- Address Offset: 0x18
- Reset Value: 0xFFFF_FFFF

Addresses: FLEXCAN-1_RX15MASK is 53FC_8000h base + 18h offset = 53FC_8018h
FLEXCAN-2_RX15MASK is 53FC_C000h base + 18h offset = 53FC_C018h



FLEXCANx_RX15MASK field descriptions

Field	Description
31–0 MI31-MI0	Acceptance mask for the Identifier in Message Buffer 15

34.6.7 Error Counter Register (FLEXCANx_ECR)

This register has 2 8-bit fields reflecting the value of two FLEXCAN error counters: Transmit Error Counter (Tx_Err_Counter field) and Receive Error Counter (Rx_Err_Counter field). The rules for increasing and decreasing these counters are described in the CAN protocol and are completely implemented in the FLEXCAN block. Both counters are read only except in Freeze Mode, where they can be written by the ARM.

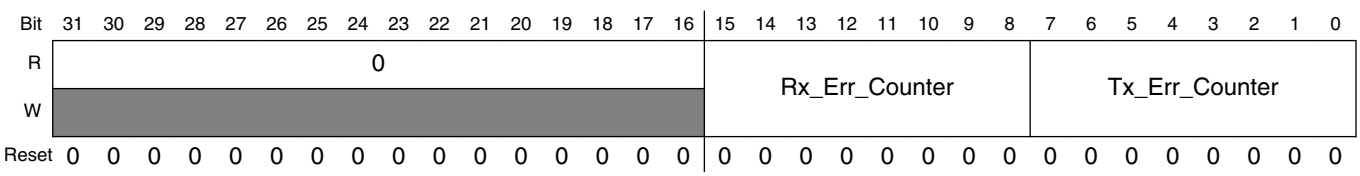
Writing to the Error Counter Register while in Freeze Mode is an indirect operation. The data is first written to an auxiliary register and then an internal request/acknowledge procedure across clock domains is executed. All this is transparent to the user, except for the fact that the data will take some time to be actually written to the register. If desired, software can poll the register to discover when the data was actually written.

FLEXCAN responds to any bus state as described in the protocolthat is transmit Error Active or Error Passive flag, delay its transmission start time (Error Passive) and avoid any influence on the bus when in , Bus Off state. The following are the basic rules for FLEXCAN bus state transitions.

- If the value of Tx_Err_Counter or Rx_Err_Counter increases to be greater than or equal to 128, the FLT_CONF field in the Error and Status Register is updated to reflect Error Passive state.

- If the FLEXCAN state is Error Passive and either Tx_Err_Counter or Rx_Err_Counter decrements to a value less than or equal to 127. While the other already satisfies this condition, the FLT_CONF field in the Error and Status Register is updated to reflect ErrorError Active state.
- If the value of Tx_Err_Counter increases to be greater than 255, the FLT_CONF field in the Error and Status Register is updated to reflect Bus Off state and an interrupt may be issued. The value of Tx_Err_Counter is then reset to zero.
- If FLEXCAN is in Bus Off state then Tx_Err_Counter is cascaded together with another internal counter to count the 128th occurrences of 11 consecutive recessive bits on the bus. Hence, Tx_Err_Counter is reset to zero and counts in a manner where the internal counter counts 11 such bits and then wraps around while incrementing the Tx_Err_Counter. When Tx_Err_Counter reaches the value of 128, the FLT_CONF field in the Error and Status Register is updated to be Error Active and both error counters are reset to zero. At any instance of dominant bit following a stream of less than 11 consecutive recessive bits, the internal counter resets itself to zero without affecting the Tx_Err_Counter value.
- If during system start-up only one node is operating, then its Tx_Err_Counter increases in each message it is trying to transmit as a result of acknowledge errors (indicated by the ACK_ERR bit in the Error and Status Register). After the transition to Error Passive state, the Tx_Err_Counter does not increment anymore by acknowledge errors. Therefore the device never goes to the Bus Off state.
- If the Rx_Err_Counter increases to a value greater than 127, it is not incremented further even if more errors are detected while being a receiver. At the next successful message reception, the counter is set to a value between 119 and 127 to resume to Error Active state.

Addresses: FLEXCAN-1_ECR is 53FC_8000h base + 1Ch offset = 53FC_801Ch
FLEXCAN-2_ECR is 53FC_C000h base + 1Ch offset = 53FC_C01Ch



FLEXCANx_ECR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. This read-only bitfield is reserved and always has the value zero
15–8 Rx_Err_Counter	Receive Error Counter
7–0 Tx_Err_Counter	Transmit Error Counter

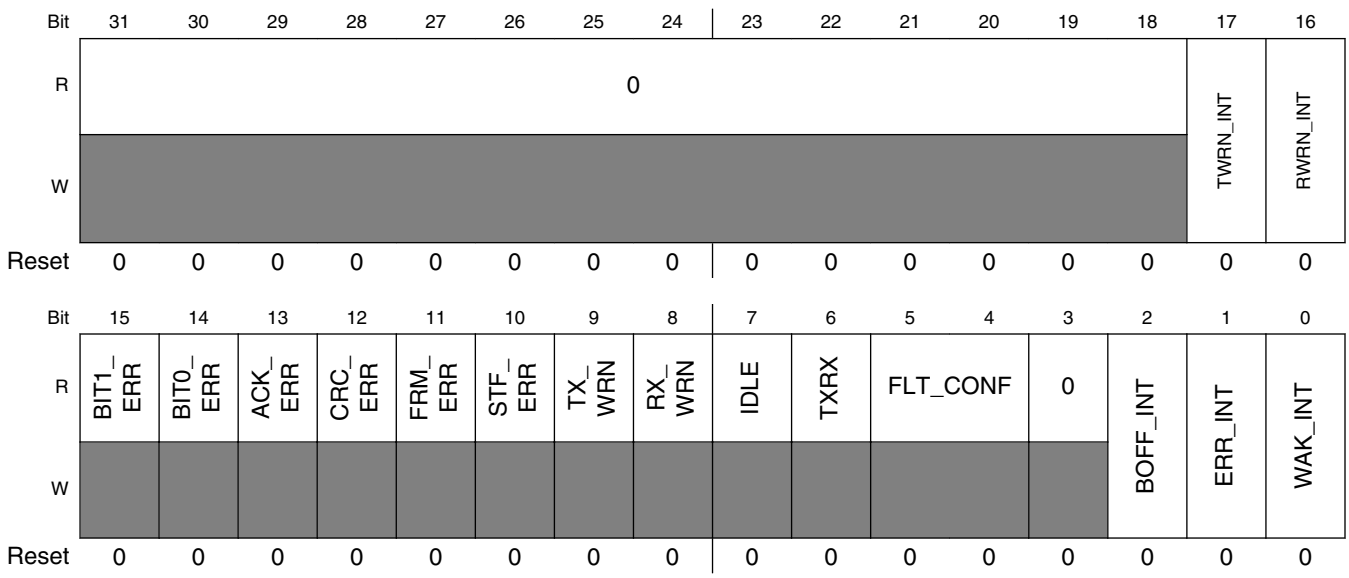
34.6.8 Error and Status Register (FLEXCANx_ESR)

This register reflects various error conditions. Four of these error conditions may be programed to signal an interrupt to the ARM. The reported error conditions (bits 15, 10) are those that occurred since the last time the ARM read this register. The ARM read action clears (bits 15,10)(bits. (Bits 9, 4) (bits are status bits.

Most bits in this register are read only, except TWRN_INT, RWRN_INT, BOFF_INT, WAK_INT and ERR_INT, that are interrupt flags that can be cleared by writing 1 to them (writing 0 has no effect).

Addresses: FLEXCAN-1_ESR is 53FC_8000h base + 20h offset = 53FC_8020h

FLEXCAN-2_ESR is 53FC_C000h base + 20h offset = 53FC_C020h



FLEXCANx_ESR field descriptions

Field	Description
31–18 Reserved	This read-only field is reserved and always has the value zero. Reserved
17 TWRN_INT	If the WRN_EN bit in MCR is asserted, the TWRN_INT bit is set when the TX_WRN flag transitions from 0 to 1, meaning that the Tx error counter reached 96. If the corresponding mask bit in the Control Register (TWRN_MSK) is set, an interrupt is generated to the ARM. This bit is cleared by writing it to 1. Writing 0 has no effect. 1 The Tx error counter transition from < 96 to ℜ≥ 96 0 No such occurrence
16 RWRN_INT	If the WRN_EN bit in MCR is asserted, the RWRN_INT bit is set when the RX_WRN flag transitions from 0 to 1, meaning that the Rx error counters reached 96. If the corresponding mask bit in the Control Register (RWRN_MSK) is set, an interrupt is generated to the ARM. This bit is cleared by writing it to 1. Writing 0 has no effect.

Table continues on the next page...

FLEXCANx_ESR field descriptions (continued)

Field	Description
	1 The Rx error counter transition from < 96 to ℜ≥ 96 0 No such occurrence
15 BIT1_ERR	This bit indicates when an inconsistency occurs between the transmitted and the received bit in a message. This bit is not set by a transmitter in case of arbitration field or ACK slot, or in case of a node sending a passive error flag that detects dominant bits. 1 At least one bit sent as recessive is received as dominant 0 No such occurrence
14 BIT0_ERR	This bit indicates when an inconsistency occurs between the transmitted and the received bit in a message. 1 At least one bit sent as dominant is received as recessive 0 No such occurrence
13 ACK_ERR	This bit indicates that an Acknowledge Error has been detected by the transmitter node, that is, a dominant bit has not been detected during the ACK SLOT. 1 An ACK error occurred since last read of this register 0 No such occurrence
12 CRC_ERR	This bit indicates that a CRC Error has been detected by the receiver node, that is, the calculated CRC is different from the received. 1 A CRC error occurred since last read of this register. 0 No such occurrence
11 FRM_ERR	This bit indicates that a Form Error has been detected by the receiver node, that is, a fixed-form bit field contains at least one illegal bit. 1 A Form Error occurred since last read of this register 0 No such occurrence
10 STF_ERR	This bit indicates that a Stuffing Error has been detected. 1 A Stuffing Error occurred since last read of this register. 0 No such occurrence.
9 TX_WRN	This bit indicates when repetitive errors are occurring during message transmission. 1 TX_Err_Counter ℜ≥ 96 0 No such occurrence
8 RX_WRN	This bit indicates when repetitive errors are occurring during message reception. 1 Rx_Err_Counter ℜ≥ 96 0 No such occurrence
7 IDLE	This bit indicates when CAN bus is in IDLE state. 1 CAN bus is now IDLE 0 No such occurrence
6 TXRX	This bit indicates if FLEXCAN is transmitting or receiving a message when the CAN bus is not in IDLE state. This bit has no meaning when IDLE is asserted.

Table continues on the next page...

FLEXCANx_ESR field descriptions (continued)

Field	Description
	1 FLEXCAN is transmitting a message (IDLE=0) 0 FLEXCAN is receiving a message (IDLE=0)
5–4 FLT_CONF	This 2-bit field indicates the Confinement State of the FLEXCAN block, as shown in below: If the LOM bit in the Control Register is asserted, the FLT_CONF field will indicate Error Passive. Because the Control Register is not affected by soft reset, the FLT_CONF field will not be affected by soft reset if the LOM bit is asserted 01 Error Passive 1x Bus off
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2 BOFF_INT	This bit is set when FLEXCAN enters Bus Off state. If the corresponding mask bit in the Control Register (BOFF_MSK) is set, an interrupt is generated to the ARM. This bit is cleared by writing it to 1. Writing 0 has no effect. 1 FLEXCAN block entered Bus Off state 0 No such occurrence
1 ERR_INT	This bit indicates that at least one of the Error Bits (bits 15-10) is set. If the corresponding mask bit in the Control Register (ERR_MSK) is set, an interrupt is generated to the ARM. This bit is cleared by writing it to 1. Writing 0 has no effect. 1 Indicates setting of any Error Bit in the Error and Status Register 0 No such occurrence
0 WAK_INT	When FLEXCAN is Stop Mode and a recessive to dominant transition is detected on the CAN bus and if the WAK_MSK bit in the MCR Register is set, an interrupt is generated to the ARM. This bit is cleared by writing it to 1. Writing 0 has no effect. 1 Indicates a recessive to dominant transition received on the CAN bus when the FLEXCAN is in Stop Mode 0 No such occurrence

34.6.9 Interrupt Masks 2 Register (FLEXCANx_IMASK2)

This register allows any number of a range of 32 Message Buffer Interrupts to be enabled or disabled. It contains one interrupt mask bit per buffer enabling the ARM to determine which buffer generates an interrupt after a successful transmission or reception (that is, when the corresponding IFLAG2 bit is set).

NOTE

Setting or clearing a bit in the IMASK2 Register can assert or negate an interrupt request if the corresponding IFLAG2 bit is set.

Addresses: FLEXCAN-1_IMASK2 is 53FC_8000h base + 24h offset = 53FC_8024h
FLEXCAN-2_IMASK2 is 53FC_C000h base + 24h offset = 53FC_C024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUF63M,BUF32M																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXCANx_IMASK2 field descriptions

Field	Description
31–0 BUF63M,BUF32M	Each bit enables or disables the respective FLEXCAN Message Buffer (MB32 to MB63) Interrupt. 1 The corresponding buffer Interrupt is enabled 0 The corresponding buffer Interrupt is disabled

34.6.10 Interrupt Masks 1 Register (FLEXCANx_IMASK1)

This register allows the enabling or disabling of any number of a range of 32 Message Buffer Interrupts. It contains one interrupt mask bit per buffer enabling the ARM to determine which buffer generates an interrupt after a successful transmission or reception (that is, when the corresponding IFLAG1 bit is set).

NOTE

Setting or clearing a bit in the IMASK1 Register can assert or negate an interrupt request if the corresponding IFLAG1 bit is set.

Addresses: FLEXCAN-1_IMASK1 is 53FC_8000h base + 28h offset = 53FC_8028h
FLEXCAN-2_IMASK1 is 53FC_C000h base + 28h offset = 53FC_C028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUF31M-BUF0M																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXCANx_IMASK1 field descriptions

Field	Description
31–0 BUF31M-BUF0M	Each bit enables or disables the respective FLEXCAN Message Buffer (MB0 to MB31) Interrupt. 1 The corresponding buffer Interrupt is enabled 0 The corresponding buffer Interrupt is disabled

34.6.11 Interrupt Flags 2 Register (FLEXCANx_IFLAG2)

This register defines the flags for 32 Message Buffer interrupts. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the corresponding IFLAG2 bit. If the corresponding IMASK2 bit is set, an interrupt will be generated. The interrupt flag must be cleared by writing it to 1. Writing 0 has no effect.

When the AEN bit in the MCR is set (Abort enabled), while the IFLAG2 bit is set for a message buffer configured as Tx, the writing access done by ARM into the corresponding message buffer will be blocked.

Addresses: FLEXCAN-1_IFLAG2 is 53FC_8000h base + 2Ch offset = 53FC_802Ch
FLEXCAN-2_IFLAG2 is 53FC_C000h base + 2Ch offset = 53FC_C02Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUF63I-BUF32I																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXCANx_IFLAG2 field descriptions

Field	Description
31–0 BUF63I-BUF32I	Each bit flags the respective FLEXCAN Message Buffer (MB32 to MB63) interrupt. 1 The corresponding buffer has successfully completed transmission or reception 0 No such occurrence

34.6.12 Interrupt Flags 1 Register (FLEXCANx_IFLAG1)

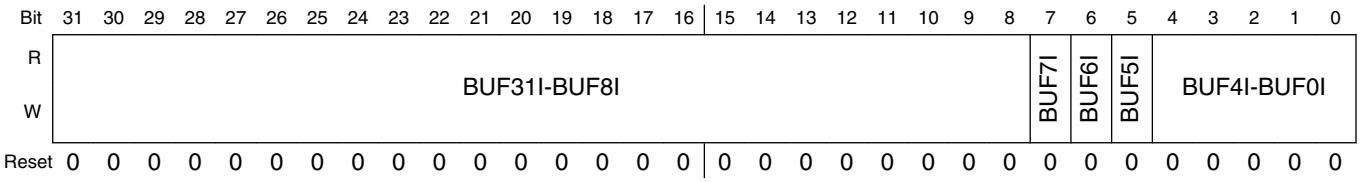
This register defines the flags for 32 Message Buffer interrupts and FIFO interrupts. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the corresponding IFLAG1 bit . If the corresponding IMASK1 bit is set, an interrupt will be generated. The Interrupt flag must be cleared by writing it to 1. Writing 0 has no effect.

When the AEN bit in the MCR is set (Abort enabled), while the IFLAG1 bit is set for a message buffer configured as Tx, the writing access done by ARM into the corresponding message buffer will be blocked.

When the FEN bit in the MCR is set (FIFO enabled), the function of the 8 least significant interrupt flags (BUF7I - BUF0I) is changed to support the FIFO operation. BUF7I, BUF6I and BUF5I indicate operating conditions of the FIFO, while BUF4I to BUF0I are not used.

Addresses: FLEXCAN-1_IFLAG1 is 53FC_8000h base + 30h offset = 53FC_8030h

FLEXCAN-2_IFLAG1 is 53FC_C000h base + 30h offset = 53FC_C030h



FLEXCANx_IFLAG1 field descriptions

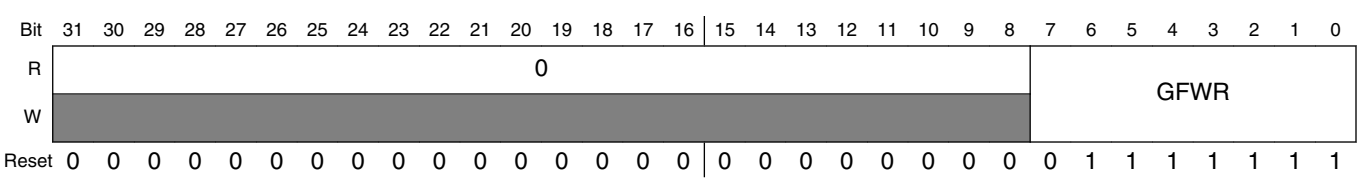
Field	Description
31–8 BUF31I-BUF8I	Each bit flags the respective FLEXCAN Message Buffer (MB8 to MB31) interrupt. 1 The corresponding message buffer has successfully completed transmission or reception 0 No such occurrence
7 BUF7I	If the FIFO is not enabled, this bit flags the interrupt for MB7. If the FIFO is enabled, this flag indicates an overflow condition in the FIFO (frame lost because FIFO is full). 1 MB7 completed transmission/reception or FIFO overflow 0 No such occurrence
6 BUF6I	If the FIFO is not enabled, this bit flags the interrupt for MB6. If the FIFO is enabled, this flag indicates that 4 out of 6 buffers of the FIFO are already occupied (FIFO almost full). 1 MB6 completed transmission/reception or FIFO almost full 0 No such occurrence
5 BUF5I	If the FIFO is not enabled, this bit flags the interrupt for MB5. If the FIFO is enabled, this flag indicates that at least one frame is available to be read from the FIFO. 1 MB5 completed transmission/reception or frames available in the FIFO 0 No such occurrence
4–0 BUF4I-BUF0I	If the FIFO is not enabled, these bits flag the interrupts for MB0 to MB4. If the FIFO is enabled, these flags are not used and must be considered as reserved locations. 1 Corresponding message buffer completed transmission/reception 0 No such occurrence

34.6.13 Glitch Filter Width Register (FLEXCANx_GFWR)

The Glitch Filter just takes effects when FLEXCAN enters the STOP mode.

Addresses: FLEXCAN-1_GFWR is 53FC_8000h base + 34h offset = 53FC_8034h

FLEXCAN-2_GFWR is 53FC_C000h base + 34h offset = 53FC_C034h



FLEXCANx_GFWR field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–0 GFWR	It determines the Glitch Filter Width. The width will be divided from Oscillator clock by GFWR values. By default, it is 5.33us when the oscillator is 24MHz.

34.6.14 Rx Individual Mask Registers (FLEXCANx_RXIMR)

These registers are used as acceptance masks for ID filtering in Rx message buffers and the FIFO. If the FIFO is not enabled, one mask register is provided for each available Message Buffer, providing ID masking capability on a per Message Buffer basis. When the FIFO is enabled (FEN bit in MCR is set), the first 8 Mask Registers apply to the 8 elements of the FIFO filter table (on a one-to-one correspondence) while the rest of the registers apply to the regular message buffers starting from MB8.

The Individual Rx Mask Registers are implemented in RAM so they are not affected by reset and must be explicitly initialized prior to any reception. Furthermore, they can only be accessed by the ARM while the block is in Freeze Mode. Out of Freeze Mode, write accesses are blocked and read accesses will return all zeros. Furthermore, if the BCC bit in the MCR Register is negated, any read or write operation to these registers results in access error.

Addresses: FLEXCAN-1_RX0IMR is 53FC_8000h base + 880h offset = 53FC_8880h
FLEXCAN-1_RX63IMR is 53FC_8000h base + 97Ch offset = 53FC_897Ch
FLEXCAN-2_RX0IMR is 53FC_C000h base + 880h offset = 53FC_C880h
FLEXCAN-2_RX63IMR is 53FC_C000h base + 97Ch offset = 53FC_C97Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MI31-MI0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FLEXCANx_RX*n*IMR field descriptions

Field	Description
31–0 MI31-MI0	For normal Rx message buffers, the mask bits affect the ID filter programmed on the message buffer. For the Rx FIFO, the mask bits affect all bits programmed in the filter table (ID, IDE, RTR). 1 The corresponding bit in the filter is checked against the one received 0 the corresponding bit in the filter is dont care

Chapter 35
Electrical Fuse Array (FUSEBOX)

35.1 Introduction

The FUSEBOX consists of electrically programmable poly fuses (FUSEBOX analog) and a digital control block (FUSEBOX digital) that has a Functional Interface (FI) that can interface with the IC Identification Module (IIM) or other dedicated fusebox control modules. A FUSEBOX Interface (FI) connects directly to the FUSEBOX FI inside FUSEBOX.

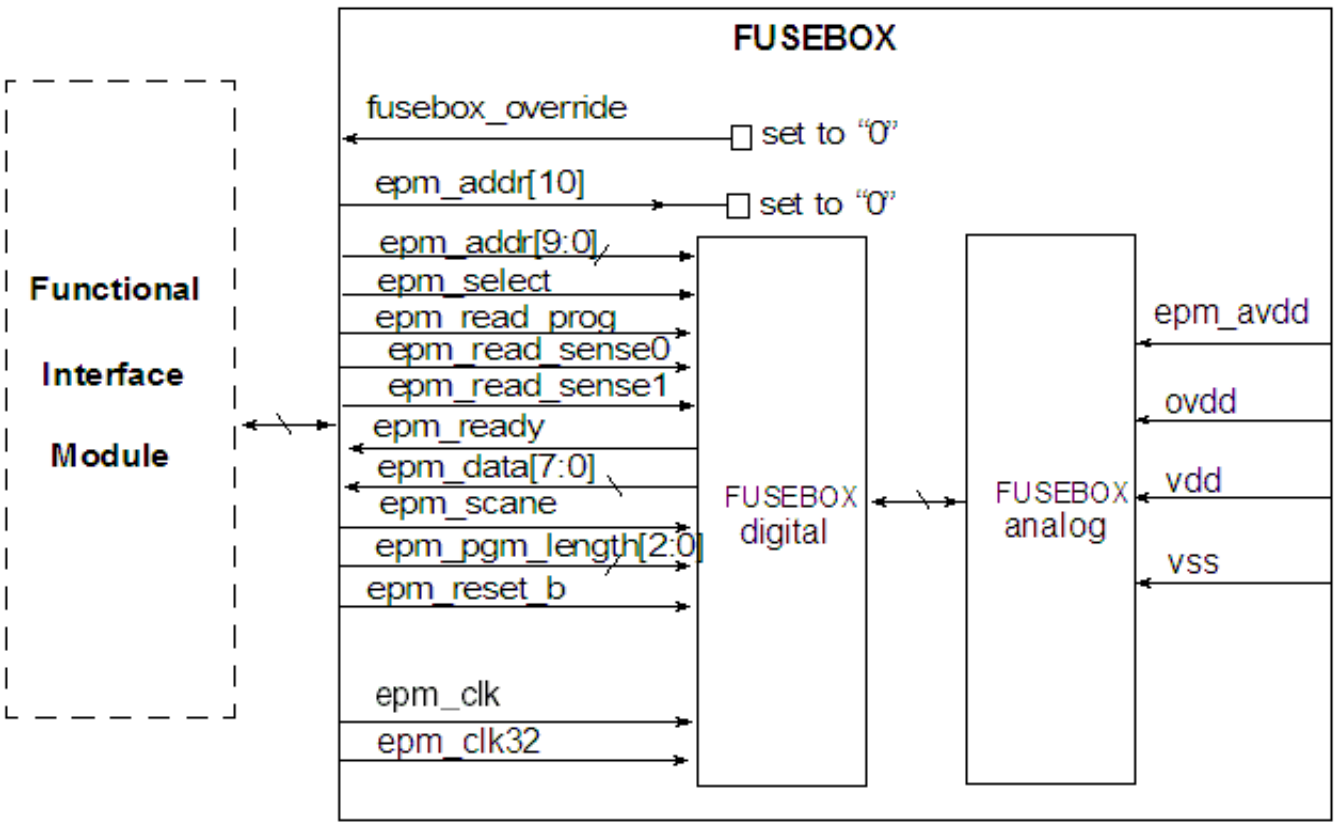


Figure 35-1. FUSEBOX Block Diagram

35.1.1 Overview

There are two major building blocks inside the FUSEBOX, an analog block and a digital block. In addition, fusebox_override signal that is connected to Functional Interface can be used to bypass the FUSEBOX. fusebox_override is set to ground by default, and it can be set to core vdd, if it is necessary, by a metal 6 layer change inside FUSEBOX.

The FUSEBOX digital block decodes the 11 bit address signal and generates control signals to the analog block. The FUSEBOX is able to support up to 1024 bit fuses. One 1024 bit FUSEBOX is used regardless of the size of the fuseboxes to keep a uniform Functional Interface.

There are two sizes of the analog block, 128 and 256 fuses. The interface between the 128 bit analog and the digital block is illustrated in Figure 35-2. The interface between the 256 bit analog and the digital block is illustrated in Figure 35-3. The interface signals

in the two cases are the same except that the 128 bit analog block has 16 additional unconnected decoded address signals. When fusebox_overrid is set to core vdd, all the outputs from the FUSEBOX to the Functional Interface will be grounded.

The 128 bit analog block contains a 16 row x 8 column fuse array (total 2*128 poly fuses due to redundancy), eight sense amplifiers, and other supporting circuitry. Likewise the 256 bit analog block contains a 32 row x 8 column fuse array, eight sense amplifiers, and supporting circuitry.

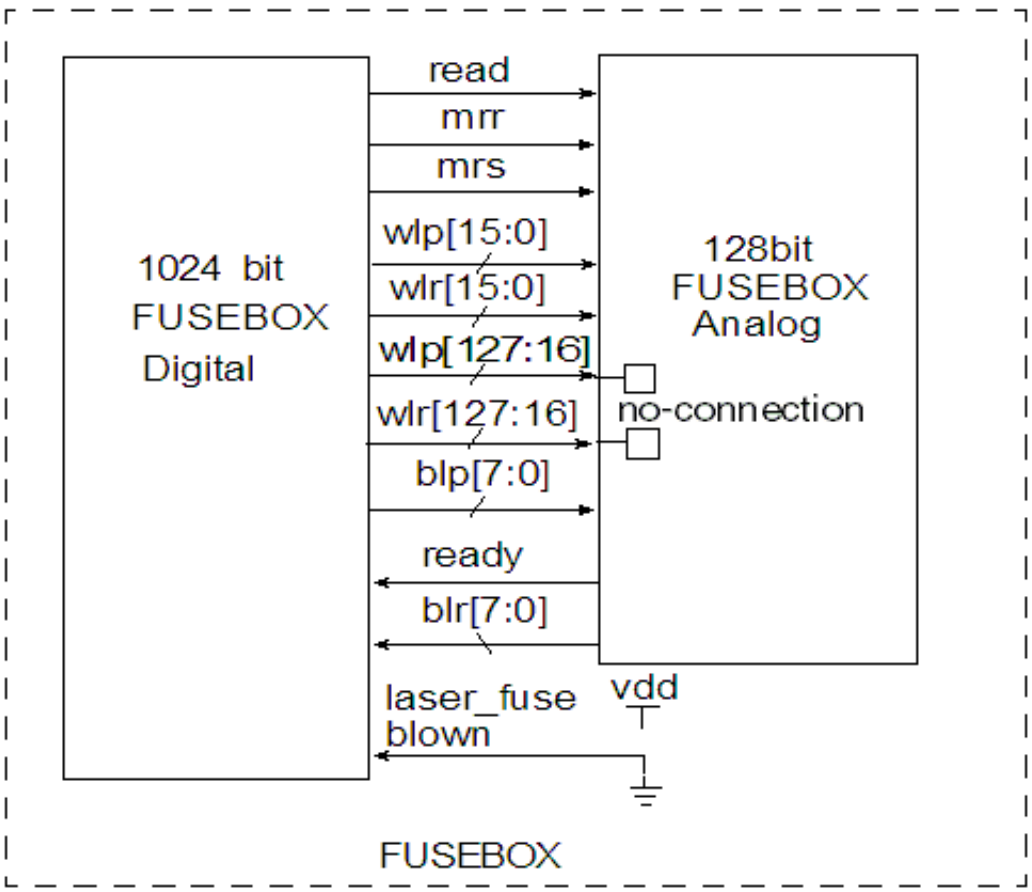


Figure 35-2. Interface Between 128 FUSEBOX Analog and FUSEBOX Digital Block

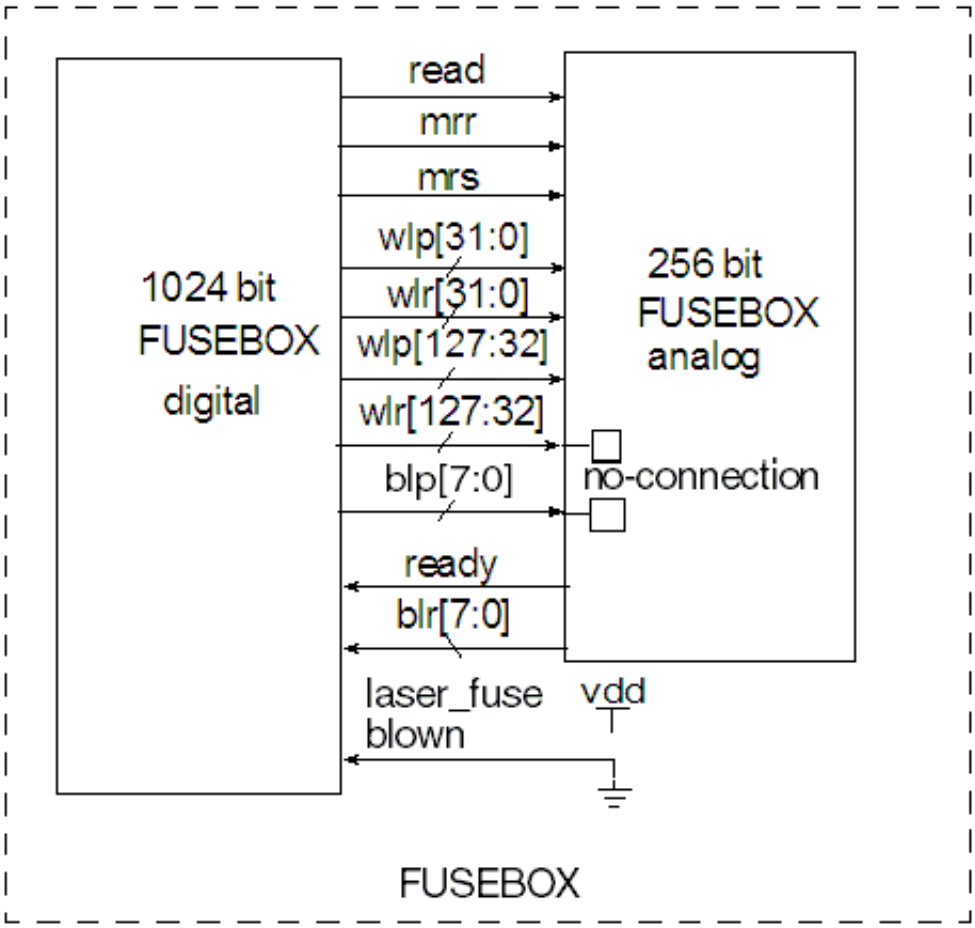


Figure 35-3. Interface Between 256 FUSEBOX Analog and FUSEBOX Digital Block

35.1.2 Modes of Operation

All FUSEBOX operations are self-timed and begin relative to the rising edge on the epm_select line. Most interface signals are shared by all Fuseboxes. Only the FUSEBOX selects (epm_select), read data (epm_data) and ready status (epm_ready) signals are unique to each FUSEBOX. The 32KHz clock is not used within the IIM, but may be used with other FIs. It is passed through to the FUSEBOX to time program operations. The 32KHz clock distributed to the Fuseboxes should be on during FUSEBOX program operations and should be gated off during read operation. The epm_clk is used to sample the epm_select, epm_ready and epm_data signals in order to synchronize these signals with the Functional Interface inside a FUSEBOX. This epm_clk is a gated clock used by the local FIs.

35.1.2.1 Normal Operating Modes

35.1.2.1.1 Word Read

Read operations are done on a word (8-bit) basis. The target word is specified by the address lines epm_addr[9:3]. The bottom three address lines (bit-select lines) are ignored in a read operation. The timing for fuse word read cycles is shown in Figure 1-4. This timing diagram shows one read cycle followed by the start of a second read, demonstrating how quickly one read can follow another. Each read command is registered on the rising edge of epm_select. The address (epm_addr[9:3]) and read command (epm_read_prog) must be stable tsels before the rising edge of epm_select and must remain stable at least tselh hold time beyond the rising edge. epm_ready is negated immediately after the command is registered, indicating that the FUSEBOX is busy. After a tdv delay from the active edge of epm_select, the data is made available on blr by the analog FUSEBOX. The epm_ready acknowledge signal is asserted tready after the initial assertion of epm_select and the data is available on the epm_data. The FUSEBOX synchronizes epm_ready and the data to the epm_clk clock and drives the synchronized data onto the IP Bus. The time between the valid epm_data and epm_ready is signified by tsync in Figure 1-4 and is one-half of one epm_clk cycle. Once the read data has been latched, epm_select is negated by the Functional Interface. epm_select must remain negated a minimum cycle to cycle delay of tccd before the next cycle (read or program) may begin. epm_data remains at a steady state until replaced by data from a subsequent read cycle.

In the read mode operation, the internal signal wlp[127:0] and blp[7:0] remain low ("0") irrespective of epm_addr[2:0].

Table 35-1. FUSEBOX Timing Parameters

Description	Designator	Timing Value
Setup time to epm_select active edge	t_{sels}	2 nS
Hold time after epm_select active edge	t_{selh}	2 nS
Delay from epm_select to epm_ready negation (read). Delay from the rise edge of 32Khz clock to epm_ready (program).	t_{nrdy}	200 pS - 1000 pS
Assert time from epm_select active edge to negation of read from digital	t_{rdd}	1nS - 4 nS
Data valid after epm_select active edge	t_{dv}	30 nS - 100 nS
Epm_ready signal asserted after epm_select active edge (read only)	t_{rdy}	<= (t_{dv} + 1.5 epm_clk)
Epm_ready signal asserted after epm_select active edge (program only)	t_{ready}	(epm_pgm_length) * $period_{32KHz}$

Table continues on the next page...

Table 35-1. FUSEBOX Timing Parameters (continued)

Description	Designator	Timing Value
Data driven out to bus after analog ready signal asserts (read only)	t_{datrdy}	0 - 1 epm_clk
epm_data to epm_ready synchronization delay (read). epm_ready to negation of epm_select (program)	t_{sync}	one-half of 1 epm_clk cycle (rising edge to falling edge) > 3nS if epm_clk duty cycle is not 50%
rising-edge of epm_ready to negative-edge of epm_select (read only)	t_{rdysel}	> 2nS typically one-half of 1 epm_clk cycle (falling edge to rising edge)
Minimum cycle to cycle delay (Consecutive Read)	t_{ccdr}	5 nS
Minimum cycle to cycle delay (Consecutive program)	t_{ccdp}	0.5 uS

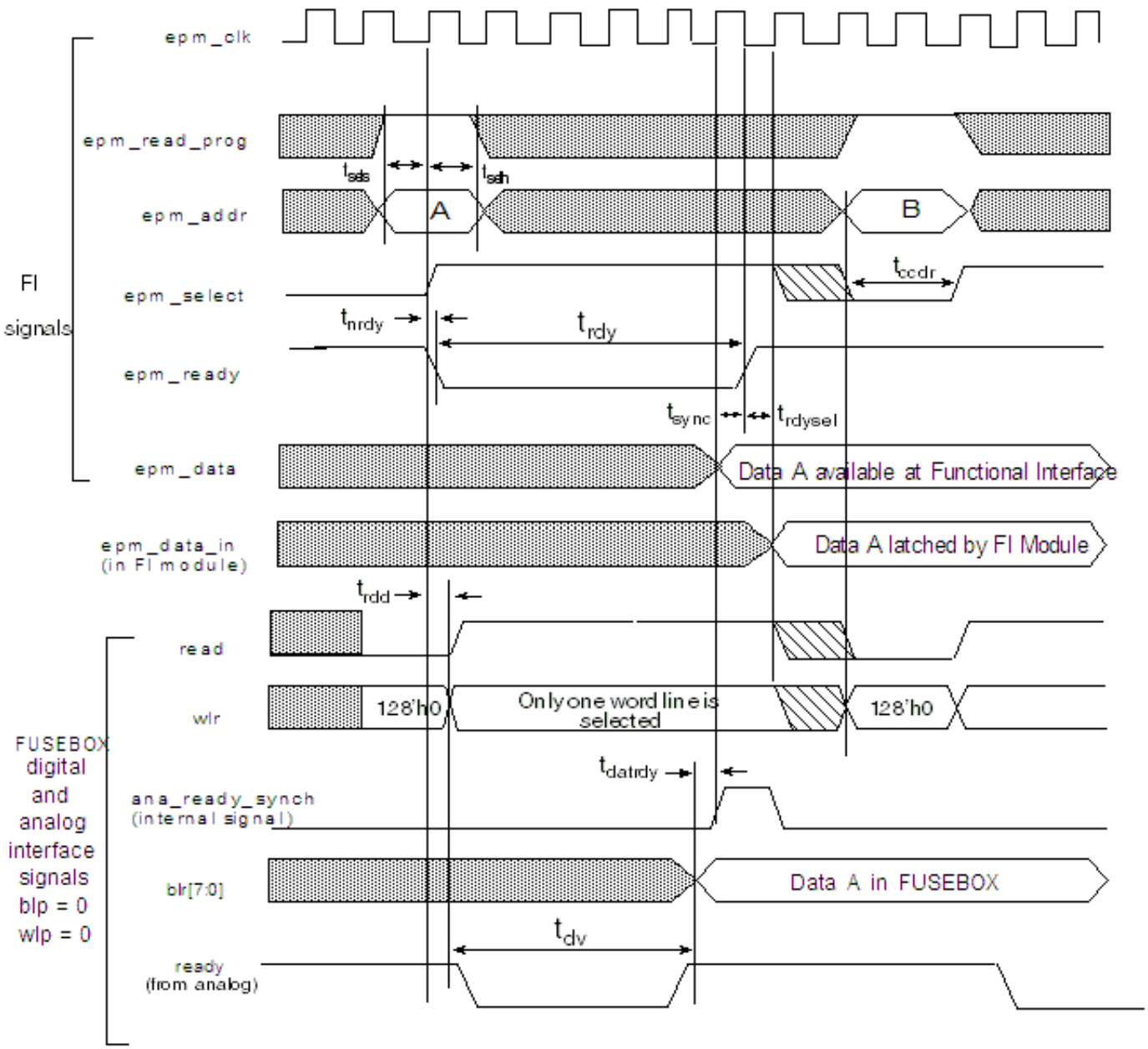


Figure 35-4. FUSEBOX Read Cycle Timing

35.1.2.1.2 Bit Program

Program operations are done on a bit-basis. The target bit is specified by the full address bus (epm_addr[10:0]). Program operations can only blow fuses (change them from logic 0 to logic 1), so there is no program data bus associated with the FUSEBOX. The timing for a program operation is shown in Figure 1-5. As in the read cycle, the address and program command are latched on the rising edge of epm_select. The address must be setup tsels ahead of the rising edge and remain held tselh past the edge of epm_select.

Once the program command is registered, on the next rising edge of the 32KHz clock, the FUSEBOX will initiate the program operation. An internal counter will start to count the number of 32K clock cycles, and the epm_ready signal is de-asserted. After the specified number of 32 KHz clocks (determined by epm_pgm_length[2:0]), the FUSEBOX will complete the program operation and assert the epm_ready signal. epm_pgm_length[2:0] needs to be set tsels ahead of the rising edge of epm_select and to be held until the completion of the internal counter. Consult Table 35-1 for specific timing values. The Functional Interface module synchronizes the epm_ready response to the epm_clk during the period tsync. After recognizing that the program operation has completed, the Functional Interface module negates the epm_select line. To avoid accidental selection of more than one bit line at a time, the 8 bit de-coded bit-line-program (blp) needs to be checked before generate blp[7:0].

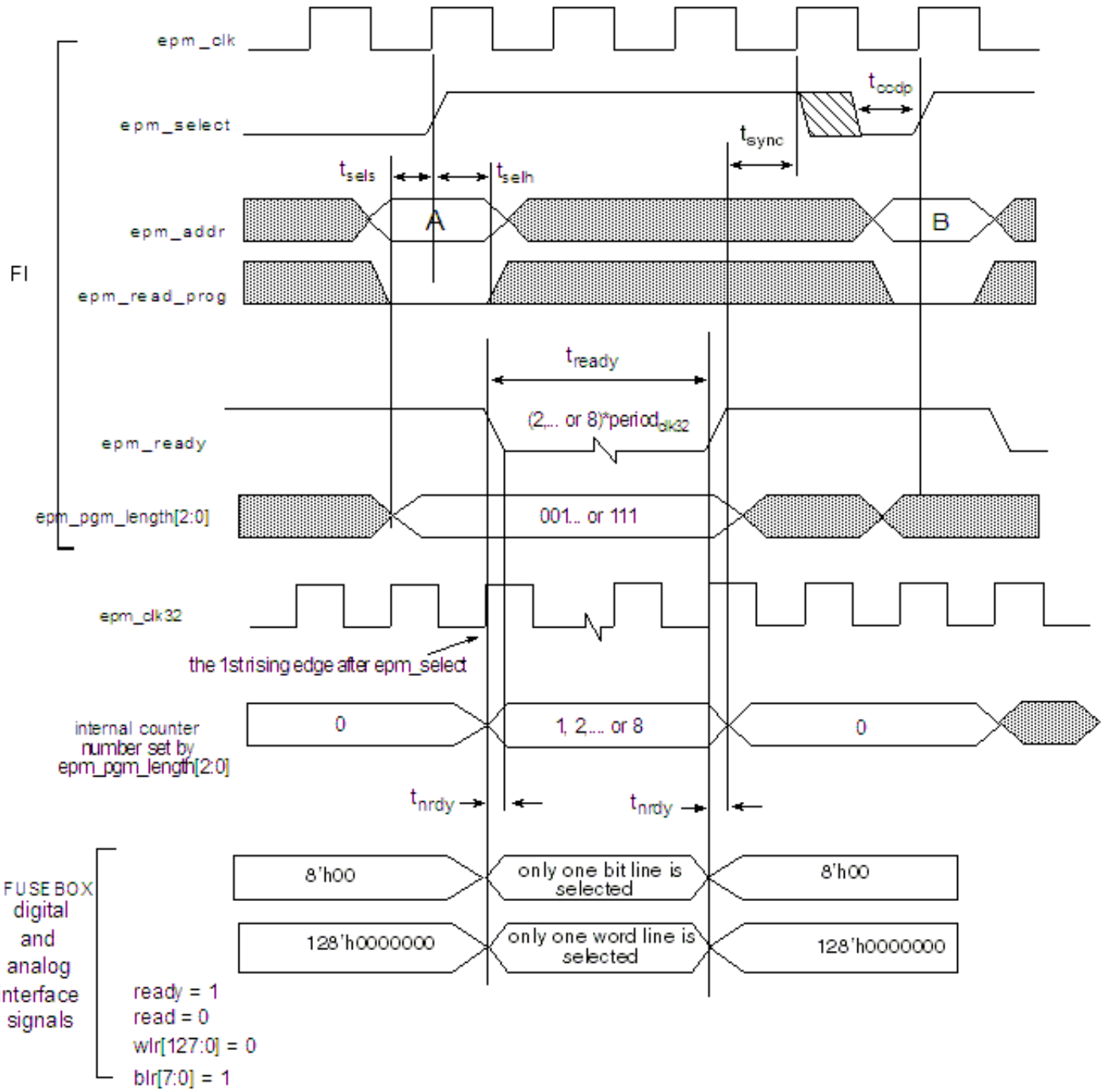


Figure 35-5. Fuse Program Cycle Timing

The fuse stress time is controlled by epm_pgm_length[2:0] and 32Khz clock. The number of 32Khz cycles is counted by an internal counter which is control by epm_pgm_length[2:0]. The counter starts counting at the first rising edge of the 32K clock after both epm_select and epm_pgm_length[2:0] are asserted. Table 35-2 shows the relationship between epm_pgm_length[2:0] and the number of 32K clock cycles.

In program mode operation, the internal signal ready and blr[7:0] remains high ("1"), and wlr[127:0] and read/mrr/mrs remains low ("0") in the program mode operation.

Table 35-2. Fuse Program Length Table

epm_pgm_length[2:0]	32KHz clock cycles	Program Length
000	0	fuses are not stressed
001	2	62.5 uS
010	3	93.75 uS
011	4	125 uS
100	5	156.25 uS
101	6	187.5 uS
110	7	218.75 uS
111	8	250 uS

NOTE

The Functional Interface module controller must determine that the target fuse bank is indeed an e-Fuse bank (not a laser fuse bank) before allowing the program cycle to proceed. This is determined by examining the appropriate fuse_type signal, which is fixed at SoC integration time.

35.1.2.2 Bypass FUSEBOX

A hard wired metal option exists in the analog block of the FUSEBOX to bypass the FUSEBOX outputs. By default, the metal is connected to core vss. When this wire is re-connected to core vdd via metal 6 change, the digital FUSEBOX block asserts the epm_data going to the Functional Interface module to 0 and epm_ready to the Functional Interface module to 1. It also asserts the fusebox_override signal to the Functional Interface module to 1.

35.1.2.3 Low Power Modes

The FUSEBOX is turned off as soon as epm_select is set to "0" state in both program and read operations.

35.2 External Signal Description

External signals are the power supplies shown in Table 35-3. During chip power on, digital core vdd needs to be powered up before epm_avdd. If epm_avdd is connected to ovdd at the chip level, digital core vdd needs to be powered up before the ovdd.

Table 35-3. Signal Properties

Signal	I/O	Function	Routing Constrain
EPM_AVDD	I	Power supply for programming fuses. Can present only in program operation.	The metal resistance needs to be less than 1 Ω from I/O pad to the block input.
OVDD	I	Power supply that needs to be available in both program and read operation. This power supply is usually the I/O pad supply in the segment.	The metal resistance needs to be less than 1.5 Ω from I/O pad to the block input.
VDD	I	Digital core vdd.	N/A.
VSS	I	Power ground. Core vss, ovss, or a dedicated FUSEBOX ground can be used.	The metal resistance needs to be less than 1 Ω from I/O pad to the block input.

35.2.1 Detailed Signal Descriptions

Table 35-4. FUSEBOX Detailed Signal Descriptions

Signal	I/O	Description
EPM_AVDD	I	Power supply for programming fuses. At least a 2.775V voltage source that is able to supply 60mA current in program operation. 3.3V voltage source is preferred.
OVDD	I	Power supply that needs to be available in both program and read operation. Usually, it is the I/O pad supply in the segment. This supply needs to provide 8mA with 1.95V ovdd or 20mA with 2.775V ovdd in read mode operation. Max. 1.95V is preferred to avoid stressing fuses.
VDD	I	Digital core vdd, 1.2V typical. Needs to be powered up before epm_avdd.
VSS	I	Power ground that is able to sustain 60mA current in program operation. Core vss, ovss, or dedicated FUSEBOX ground can be used.

35.3 Functional Description

The FUSEBOX module receives 11 bit address signal and other control signals from a FUSEBOX Interface module, see Figure 35-1. The mode of operation and the location of bit cells of the FUSEBOX are controlled by the FUSEBOX Interface module. See Overview for detail descriptions.

The FUSEBOX generates two outputs, epm_ready and epm_data[7:0]. Both signals are connected to the FUSEBOX Interface module. Epm_ready indicates the completion of the program or read operation. The FUSEBOX Interface module can start the next operation cycle after detecting the rise edge of epm_ready from the FUSEBOX. In read operation, The FUSEBOX Interface module also stores the epm_data[7:0] when the rise edge of epm_ready is detected.

35.3.1 Clocks

EPM_CLK32 is a 32.768Khz clock that needs to be present during FUSEBOX program operation.

EPM_CLK is the same clock that a FUSEBOX Interface module uses to generate FUSEBOX interface signals. The maximum frequency of EPM_CLK is 166 Mhz. The worst duty cycle of EPM_CLK is 40% - 60%.

35.3.2 Reset

EPM_RESET_B needs to be connected to an early system reset signal so that the FUSEBOX can be reset during chip power up.

35.3.3 Interrupts

No interrupt signal is generated by FUSEBOX.

35.4 Initialization Information

The FUSEBOX is initialized and controlled by the IIM. Please refer to the IIM chapter for more details.

Chapter 36
General Power Controller (GPC)

36.1 Introduction

GPC is a General Power Control module.

36.1.1 Overview

GPC block includes the following sub-blocks of Advanced Power Saving techniques:

- Two DVFS sub-blocks (ARM platform and peripherals clock/power domains)
- SRPG sub-block for SRPG-ed module: ARM platform

Each of the sub-blocks has its own IP registers. The GPC also includes an arbitration block of DVFS voltage/frequency updates.

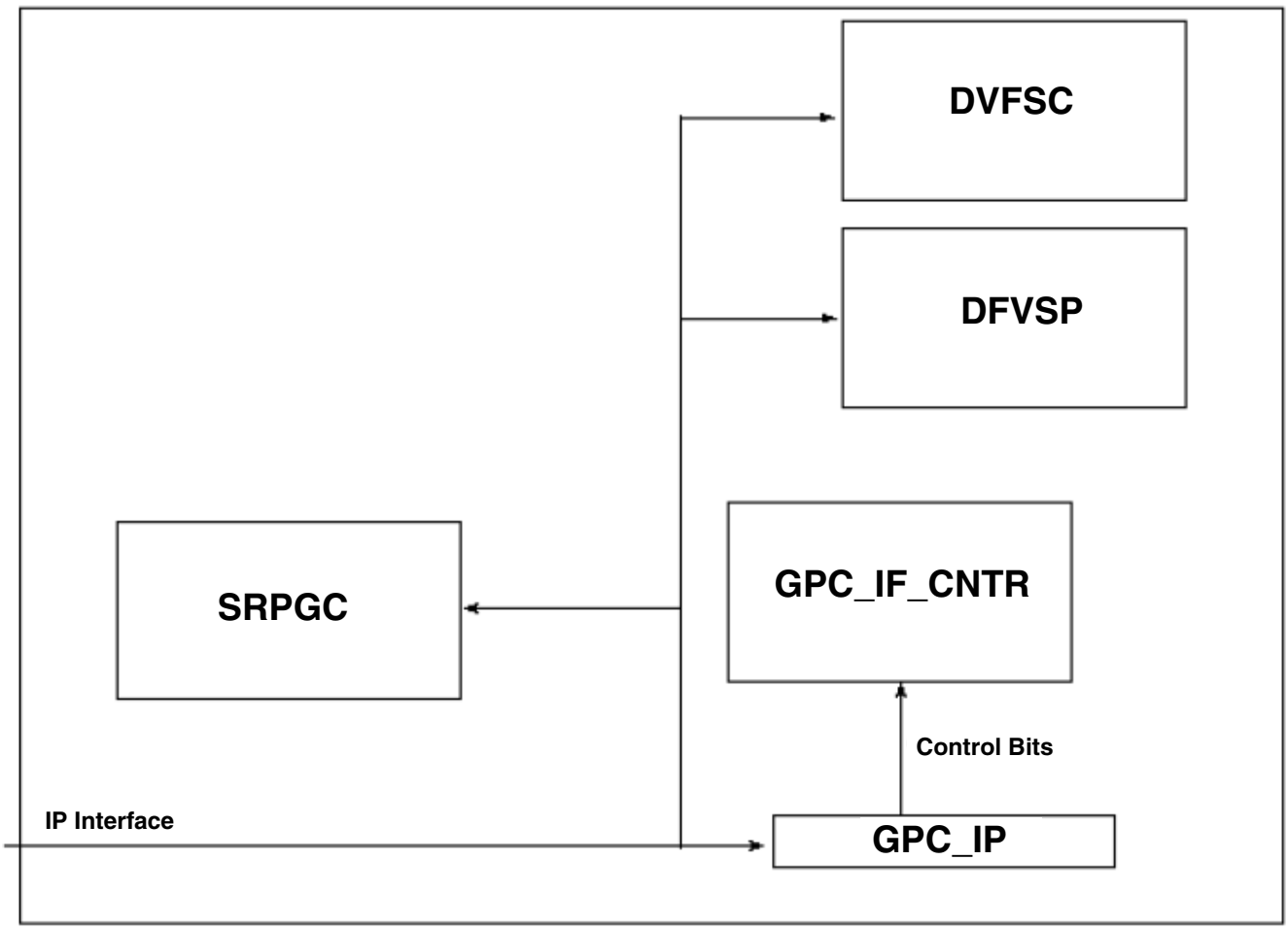


Figure 36-1. GPC Block Diagram

36.1.2 Features

- Simultaneous usage of DVFS of each domain.

- Internal counter for bypass of voltage_ready signal

- Possibility of switching between Configurable SPI (CSPI) usage and I2C Interface (I2C) .

36.2 Functional Description

GPC block includes the following sub-blocks of Advanced Power Saving techniques:

- 2 DVFS sub-blocks (ARM platform and peripherals clock/power domains)
- SRPG sub-block for SRPG-ed module: ARM platform

Each of sub-blocks has its own IP registers. As well, GPC includes an arbitration block of DVFS voltage/frequency updates.

In the following sections the global operation of the sub-blocks is described. For further details of DVFS and SRPG refer appropriate sub-block Block Guide document.

36.2.1 DVFS - Dynamic Voltage & Frequency Scaling

Dynamic Voltage & Frequency Scaling is a well-known technique to reduce power consumption in mobile devices.

In order to improve power saving efficiency, DVFS is applied both on ARM platform domain (DVFSC) and peripherals domain (DVFSP). DVFSC uses mainly ARM's IDLE signal for load monitoring, but also uses s/w writable info as secondary inputs. DVFSP uses hardware accelerators activity signals, bus activity signals and memory activity indicators for load monitoring. Both DVFS load tracking controllers are h/w implemented, providing: high resolution of load tracking (for higher efficiency), no MIPS requirements from ARM platform for continuous operation (more power saving, no additional ARM platform load).

Voltage update requests (to PMIC, via CSPI) may be served in one of the following ways:

- Hardware triggering of CSPI, by GPC
- ARM interrupts serving CSPI

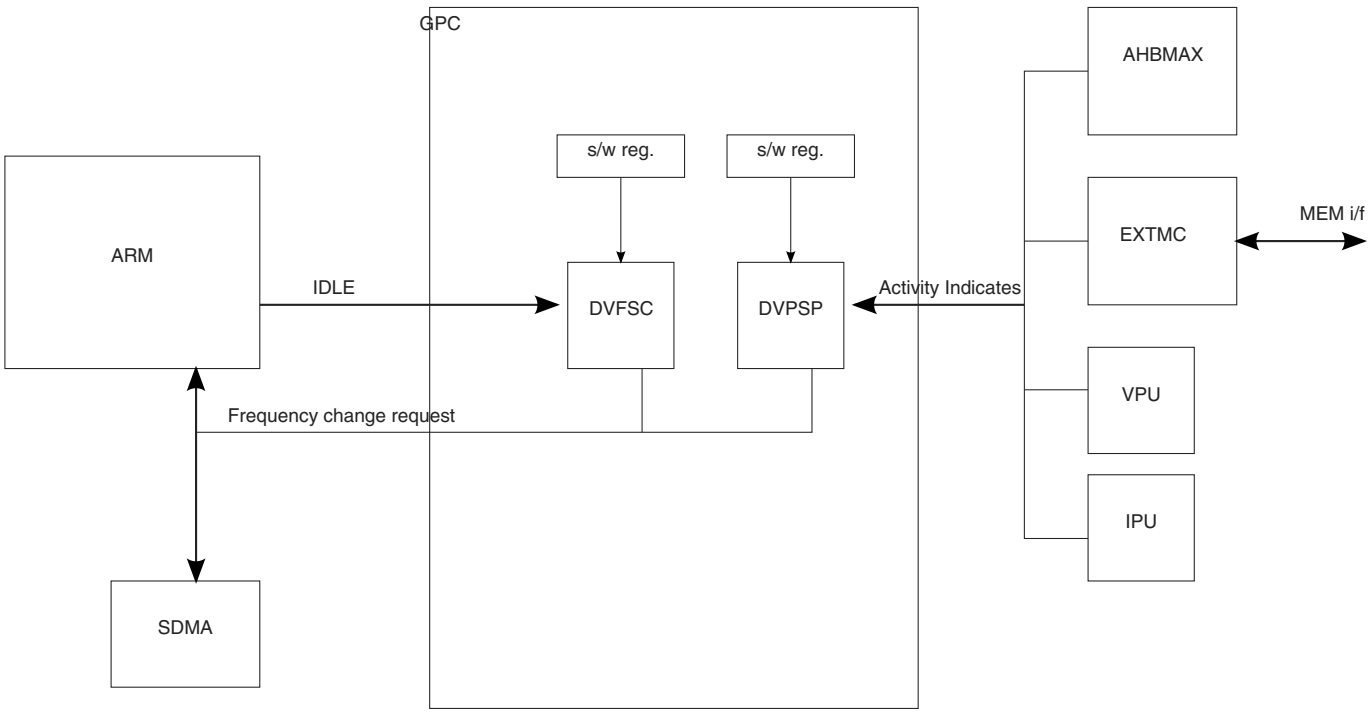


Figure 36-2. DVFS System High Level Diagram

36.2.2 DVFS Change Request Sequence Diagrams

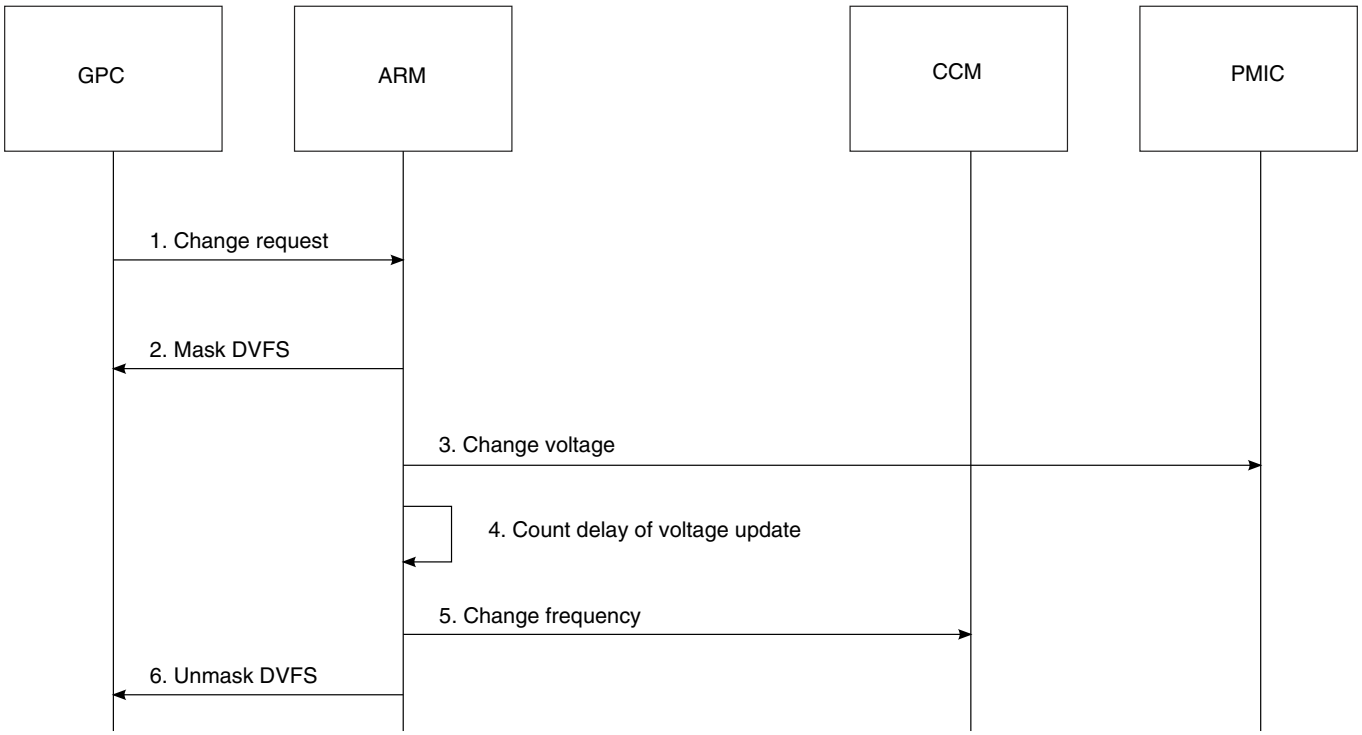


Figure 36-3. DVFS - frequency increase

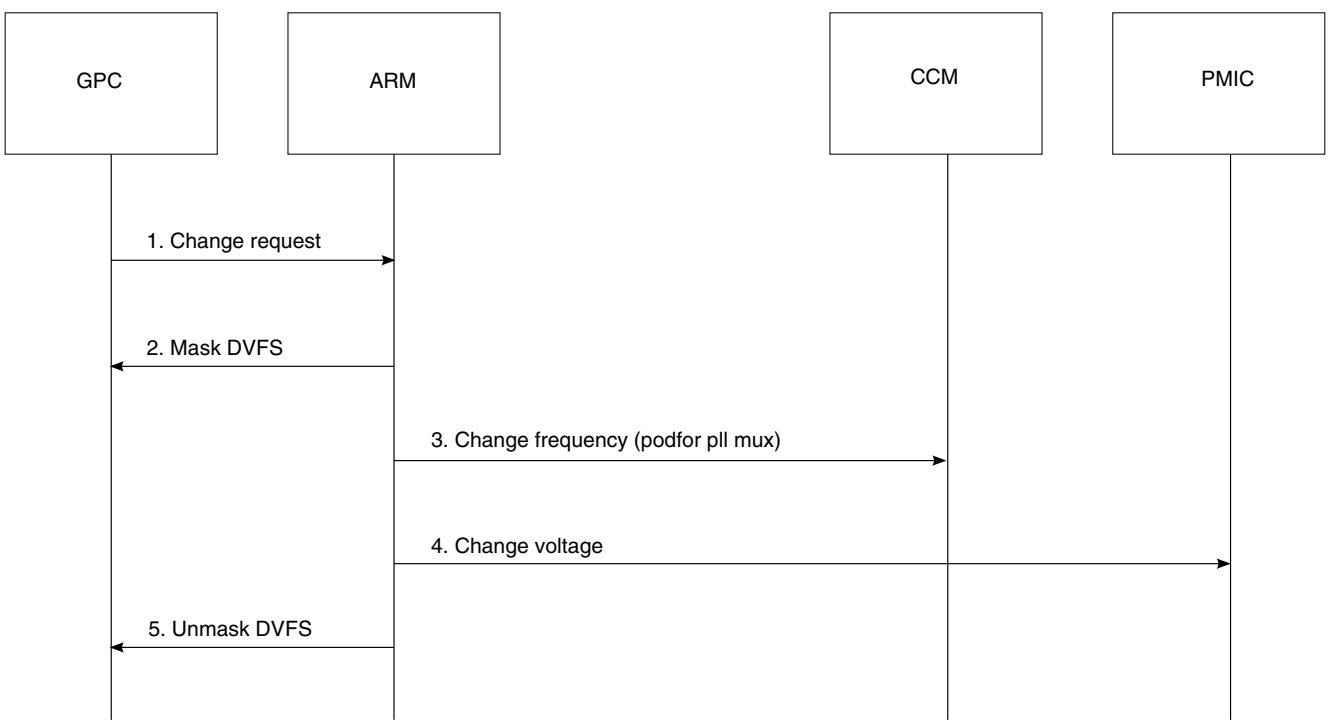


Figure 36-4. DVFS - frequency decrease

DVFSC and DVFSP will have one common interrupt line to ARM and DMA request to SDMA. SDMA (or ARM) will choose only one domain to serve.

DVFSP frequency change will be performed with post-dividers values update only.

DVFSC frequency change can be performed in 2 ways:

- PLL and post-dividers values update
- post-dividers values only update

When PLL update is required, Controller will switch ARM platform clock to another PLL, will wait until the previous PLL is locked on the required frequency, and then will switch the ARM platform clock back to the first PLL.

36.2.3 Frequency / Voltage Change Controller Description

Controller is responsible of correct frequency and voltage update flow.

36.2.3.1 GPC Controller Description

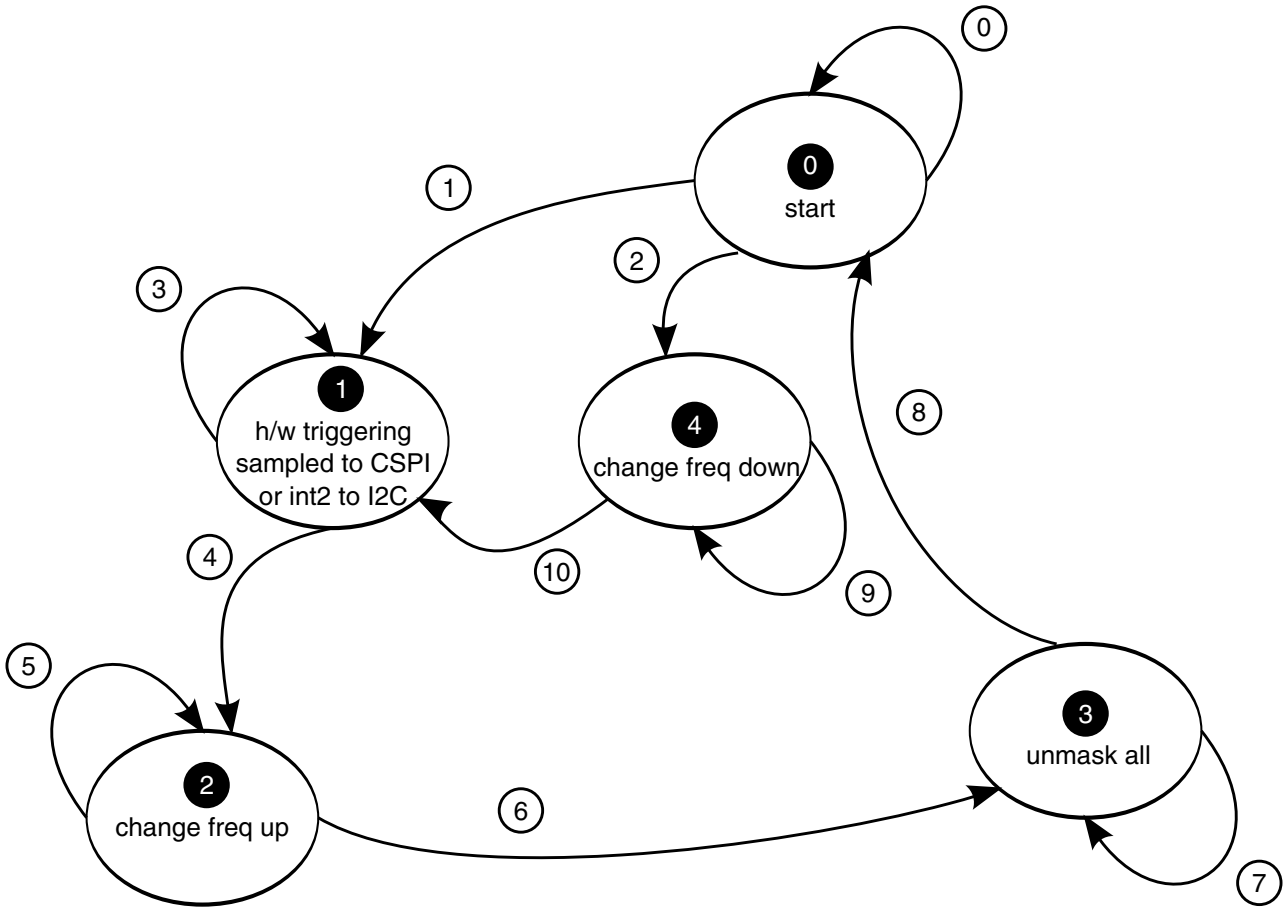


Figure 36-5. GPC DVFS State machine

Table 36-1. GPC Controller Description

Transition	Transition condition	Output
0	not (1)	-
1	STRT bit asserted, and ((freq/voltage increase needed) | (!freq/volt_inc & ! freq_update))	-
2	STRT bit set, and freq/voltage decrease needed and freq_update	-
3	not (4)	Sample h/w triggering selected reg index to CSPI or sending int2 to I2C for PMIC Start counter if VCNTU==1 (pmic ack not needed)
4	(voltage_ready & pmic ack needed) | (counter finished & !pmic ack needed)	-

Table continues on the next page...

Table 36-1. GPC Controller Description (continued)

Transition	Transition condition	Output
5	not (6)	If (dvfs update up), trigger freq. change Reset counter
6	(dvfs update up & clk change ack) | !(dvfs update up)	-
7	1 clk	unmask all, enable all toggle dvfs_cnt_res_arm or dvfs_cnt_res_per
8	not (8)	-
9	not (10)	If (dvfs update dw), trigger freq. change IRQ masked: IRQM=1
10	(dvfs update dw& clk change ack) | !(dvfs update dw) | !freq_update	-

36.2.3.2 Clock Control Module Frequency Update Controller Description

Additional State Machine is required in CCM to take care about frequency update sequence, as defined below:

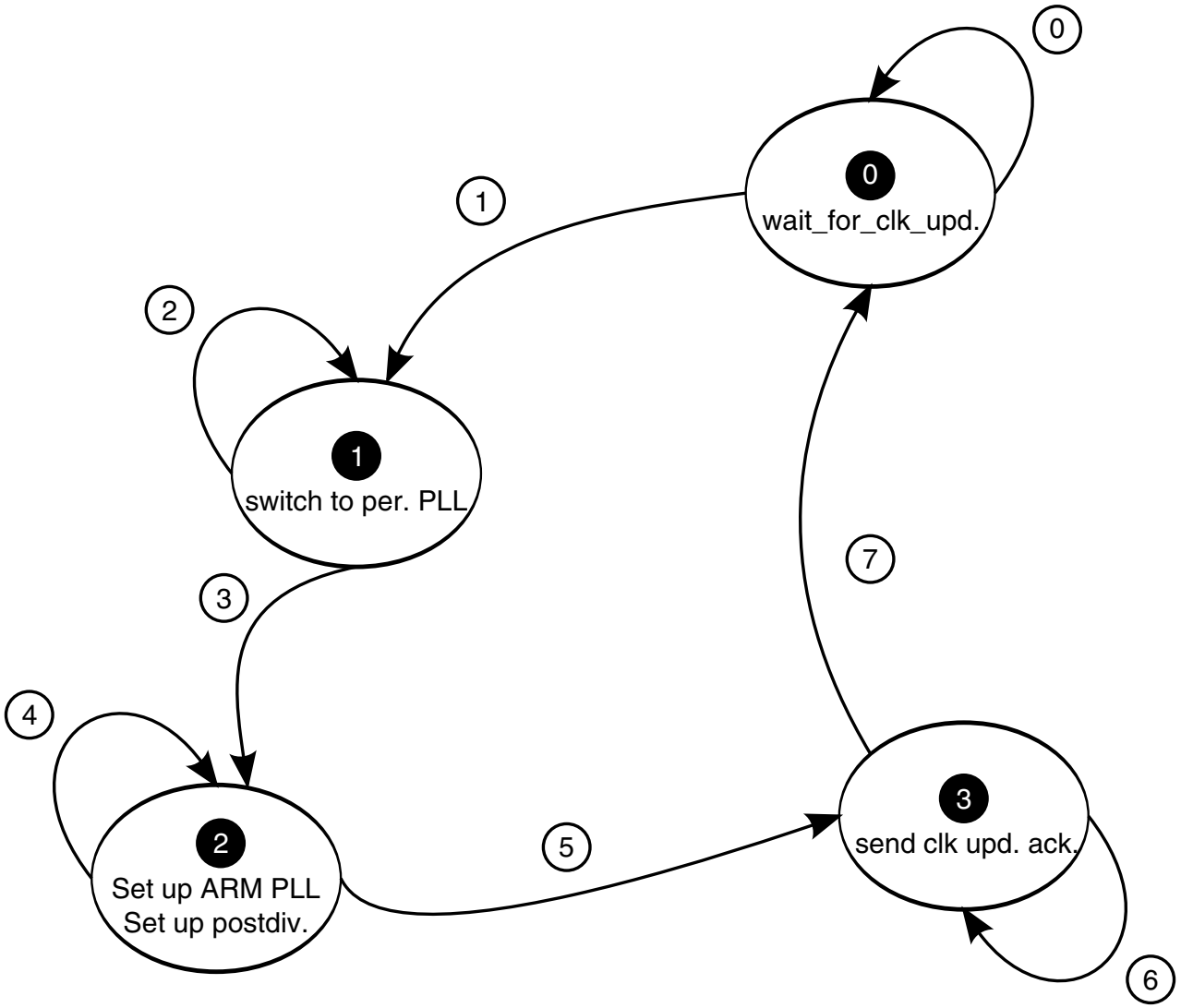


Figure 36-6. CCM Frequency Update Controller

Table 36-2. CCM Frequency Update Controller Description

Transition	Transition condition	Output
0	not (1)	clk upd ack negated
1	clk update request	-
2	not (3)	switch ARM clk to per. PLL
3	clk switched to per. PLL	-
4	not (5)	apply new PLL settings, if needed apply new postdiv settings if needed
5	ARM PLL & postdiv upd ack.	-
6	1 clk	clk update ack to GPC asserted.

Table continues on the next page...

Table 36-2. CCM Frequency Update Controller Description (continued)

Transition	Transition condition	Output
7	not (6)	-

36.2.4 State Retention Power Gating (SRPG)

State Retention Power Gating (SRPG) is an advanced power saving technique - the values of FFs are saved, while the supply for the combinational logic is gated. Such special power gating allows significant power saving during low-power (stand-by) states, when no logical operation is needed. Relatively the simple Power Gating techniques, SRPG allows to restore the state very quickly and continue the processing from the state, sampled before stand-by period.

SRPG Controller manages all the related signals for entering and exiting SRPG state of relevant modules. Not all the modules have SRPG design.

SRPG power down sequence:

CCM sends power down request when the chip enters stop or wait mode. The user should define which modules will be powered down (PGCR registers of corresponding SRPG, bit 0).

SRPG power up sequence:

The modules receive power up request in following order:

NEON and CTA8.

36.2.5 PMIC Interface Requirements for APM Support

PMIC APM interface is done through SPI protocol by enhanced CSPI module with HW triggering. Below is the description of modules requirements for APM support.

- CSPI
  - Will have 16 HW trigger registers.
  - Each register will be 32 bits:
    - 1 bits read/write (always write)
    - 6 bits address
    - 1 bit not used
    - 24 bits data
  - Each register will have an associated signal that starts the transaction to Atlas

- GPC
  - will have a 4-bit select register to select one of 16 HW trigger signals, if CSPI is used.
  - will trigger the signals during DVFS state machine according to select register value.
- PMIC
  - Only one 24-bit register should be updated for voltage change. One CSPI HW trigger will be able to make the change, or I2C will get int2.

Ability to use SDMA for h/w controllers inputs analysis and/or CSPI programming will provide high level of s/w flexibility and will be kept for backup.

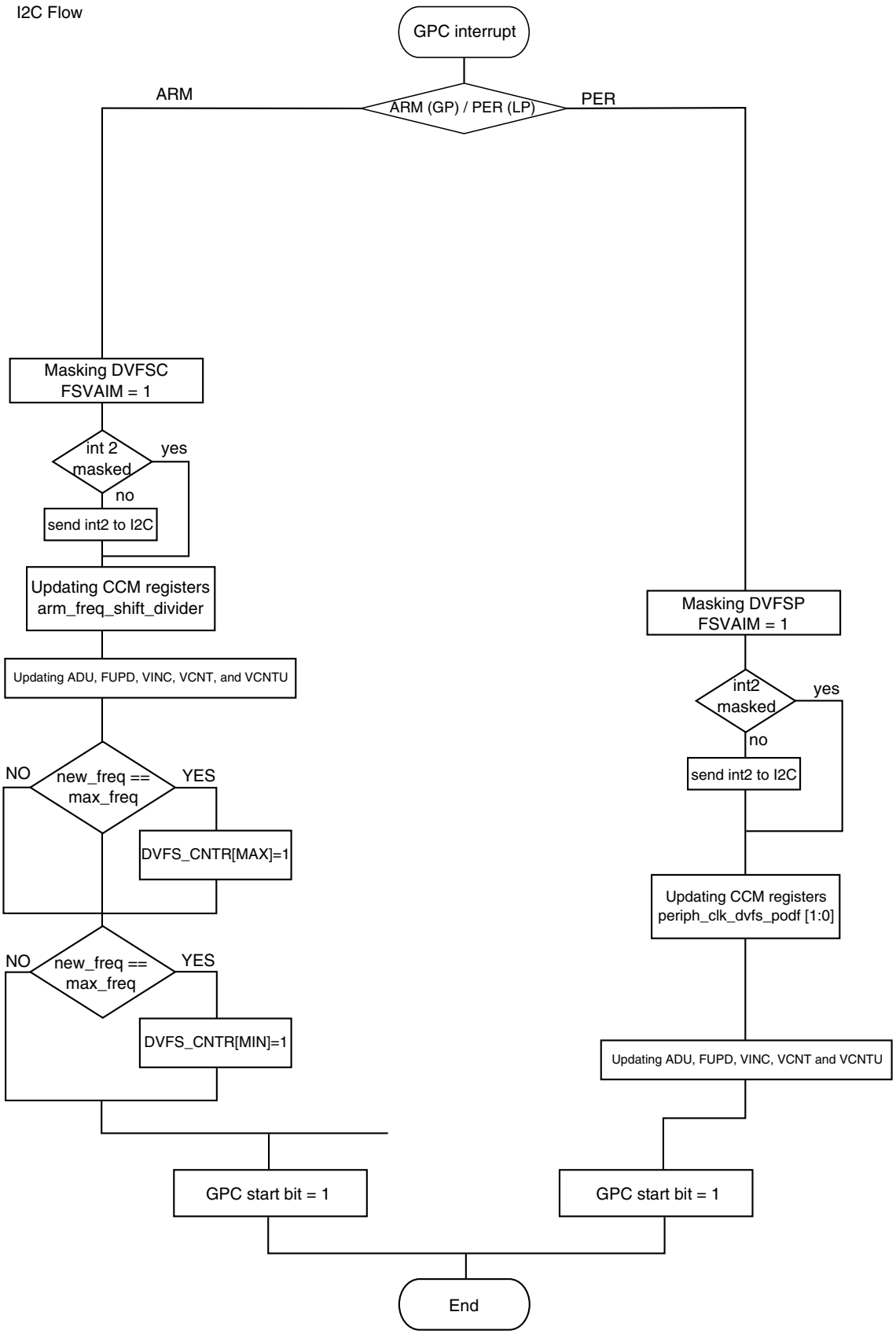


Figure 36-7. PMIC Interface Requirements IPC Flow

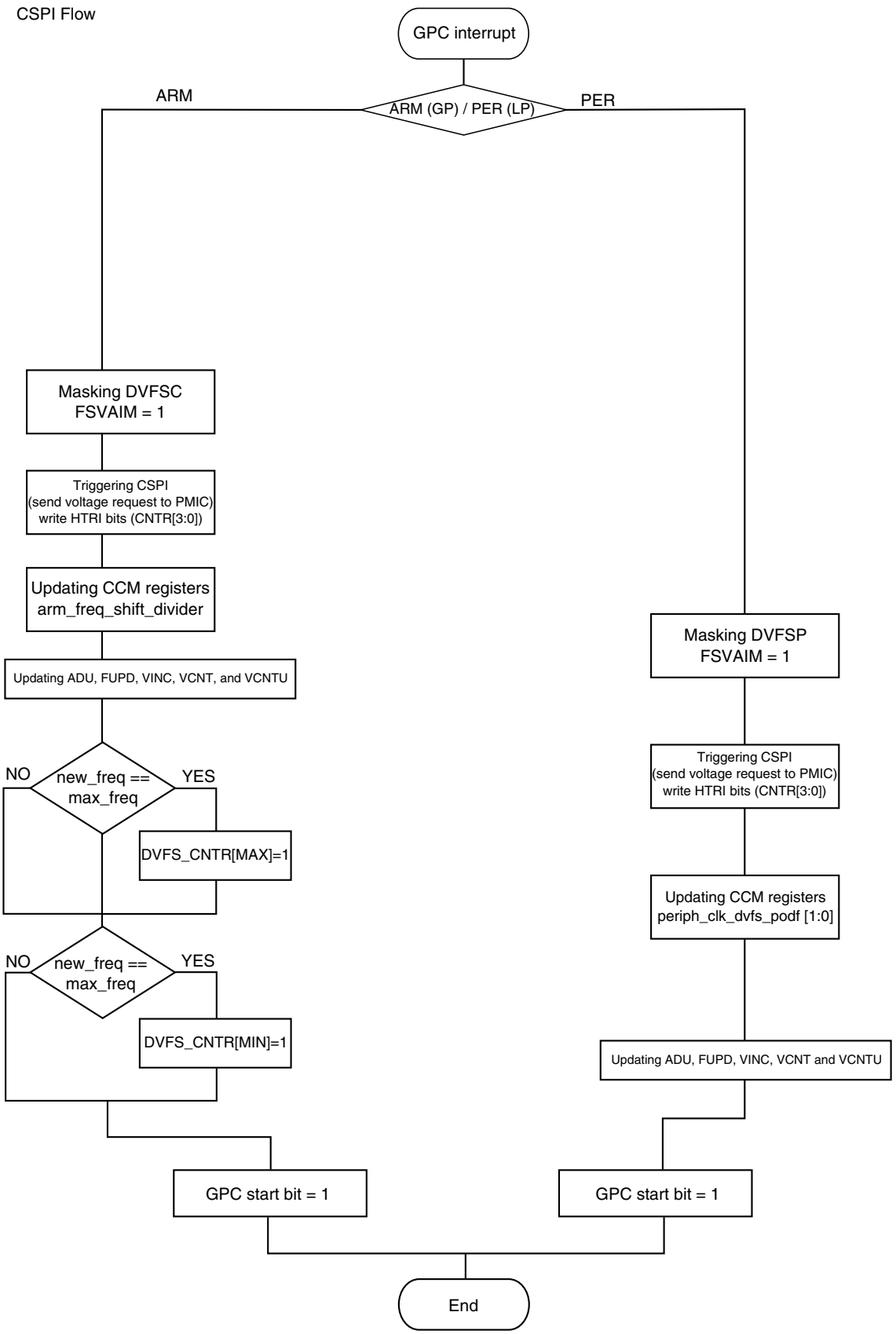


Figure 36-8. PMIC Interface Requirements CSPI Flo

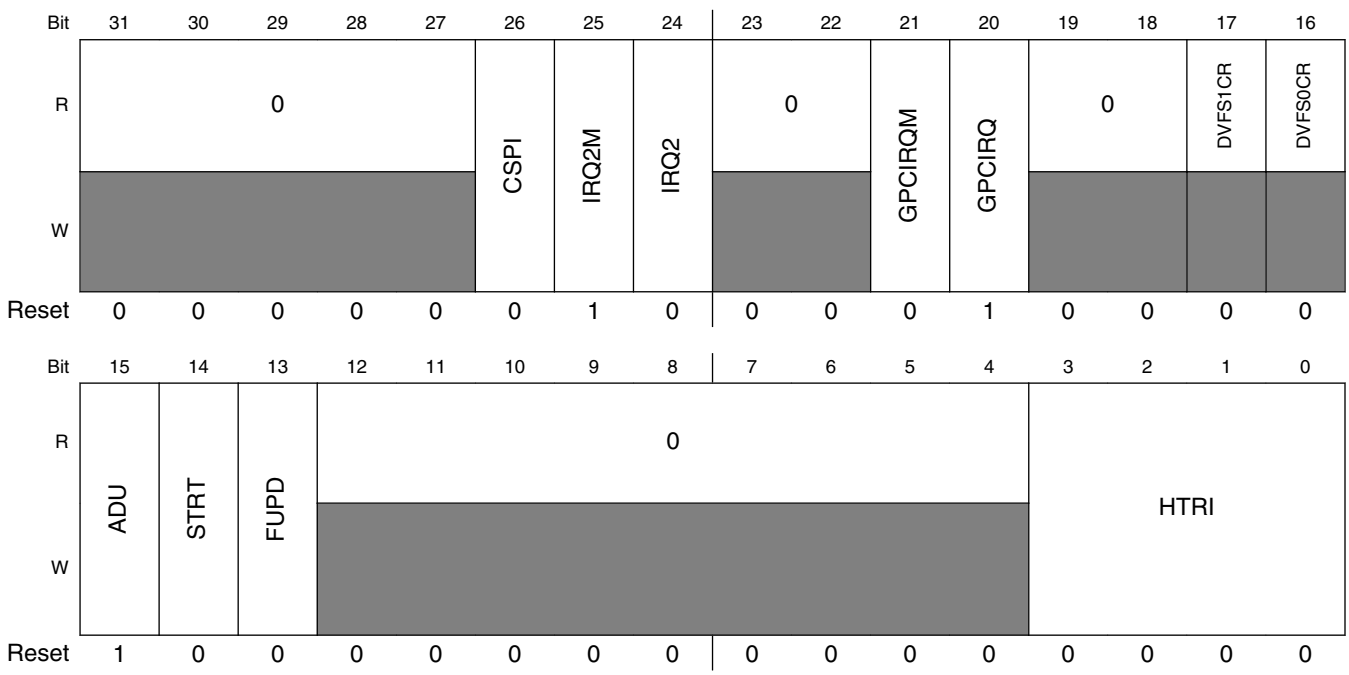
36.3 Programmable Registers

GPC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FD_8000	Interface Control Register (GPC_CNTR)	32	R/W	0210_8000h	36.3.1/ 1716
53FD_8008	Voltage Counter Register (GPC_VCR)	32	R/W	0000_0001h	36.3.2/ 1718
53FD_8010	NEON register (GPC_NEON)	32	R/W	0000_0030h	36.3.3/ 1719

36.3.1 Interface Control Register (GPC_CNTR)

Address: GPC_CNTR is 53FD_8000h base + 0h offset = 53FD_8000h



GPC_CNTR field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

GPC_CNTR field descriptions (continued)

Field	Description
26 CSPI	CSPI or I2C is used (hw trigger will be generated or irq2 will be sent for PMIC) 0 default - I2C is in use 1 CSPI is in use
25 IRQ2M	int2 (for I2C) masking 0 int2 will be sent by GPC FSM if CSPI is '0' (I2C is in use) 1 default - int2 is masked, but GPC FSM will wait for INT2 bit to be negated by writing '1', if CSPI is '0'.
24 IRQ2	Status bit, write 1 to clear.
23–22 Reserved	This read-only field is reserved and always has the value zero. Reserved
21 GPCIRQM	GPC interrupt/event masking 1 interrupt/event is masked 0 not masked
20 GPCIRQ	GPC will generate ARM IRQ or Smart Direct Memory Access (SDMA) event, as a result of DVFS change requests 1 GPC will generate ARM IRQ 0 GPC will generate SDMA event
19–18 Reserved	This read-only field is reserved and always has the value zero. Reserved
17 DVFS1CR	DVFSP Change request (bit is read-only) 1 DVFS1 is requesting for frequency/voltage update 0 DVFS1 has no request
16 DVFS0CR	DVFSC Change request (bit is read-only) 1 DVFS0 is requesting for frequency/voltage update 0 DVFS0 has no request
15 ADU	ARM domain freq/voltage update needed 1 ARM domain frequency and/or voltage update needed 0 PER domain frequency and/or voltage update needed
14 STRT	Controller start Controller operation will be started when bit set to "1". Bit will be set automatically to "0" when frequency / voltage change finished. There is still a possibility to write "0" by s/w. 1 Controller operation in progress. No IRQ is allowed during voltage update procedure (IRQ will be masked) 0 Controller operation finished. New freq/voltage change request is available
13 FUPD	Frequency update needed 1 frequency update needed 0 frequency updated is not needed

Table continues on the next page...

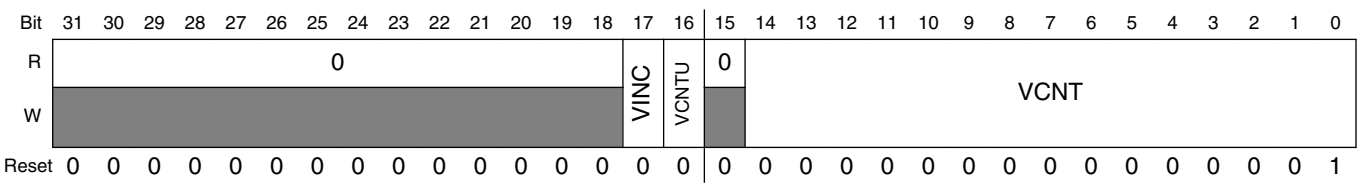
GPC_CNTR field descriptions (continued)

Field	Description
12–4 Reserved	This read-only field is reserved and always has the value zero. Reserved
3–0 HTRI	Hardware Triggering Register Index Indicates which one of the CSPI h/t register will be written to the PMIC. 0000 register #0 will be triggered, 0001 register #1 will be triggered, 0010 register #3 will be triggered...

36.3.2 Voltage Counter Register (GPC_VCR)

VCR Register - Voltage Counter Register

Address: GPC_VCR is 53FD_8000h base + 8h offset = 53FD_8008h



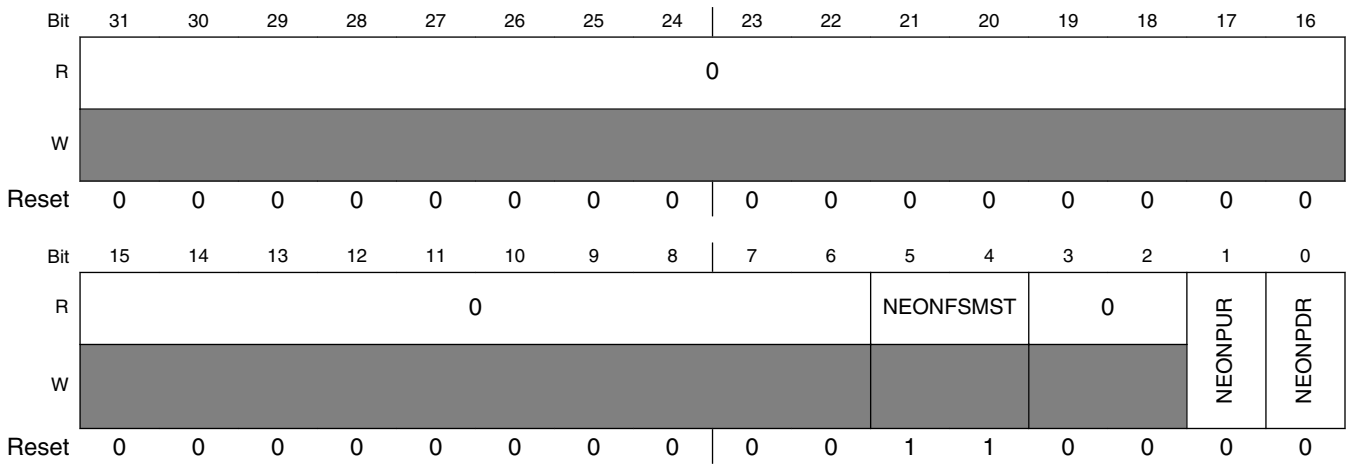
GPC_VCR field descriptions

Field	Description
31–18 Reserved	This read-only field is reserved and always has the value zero. Reserved
17 VINC	Voltage increase 1 Voltage will be increased 0 Voltage will be decreased
16 VCNTU	Voltage Count Used 1 VCNT is used 0 VCNT is not used, "voltage ready" signal of PMIC will be used.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–0 VCNT	Voltage update Count Counting delay of the voltage update, if PMIC's "voltage ready" signal is not used Counter is using SYS_CLK (CKIH) clock

36.3.3 NEON register (GPC_NEON)

NEON Register - Register Control Power for NEON

Address: GPC_NEON is 53FD_8000h base + 10h offset = 53FD_8010h



GPC_NEON field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5–4 NEONFSMST	Neon FSM status bits - state of Neon State Machine: 11 Power up acknowledge / Normal state 01 Power down request 00 Power down acknowledge 10 Power up request
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1 NEONPUR	NEON Power Up Request. Assertion causes Neon PowerDown state machine to change state from PowerDown to PowerUp request. Will clear itself after a single clock.
0 NEONPDR	NEON Power Down Request. Assertion starts Neon PowerDown state machine. Will clear itself after a single clock.

Chapter 37
General Purpose Input/Output (GPIO)

37.1 Introduction

The GPIO generates up to 32 signals for general-purpose.

A block diagram of the GPIO is shown in .

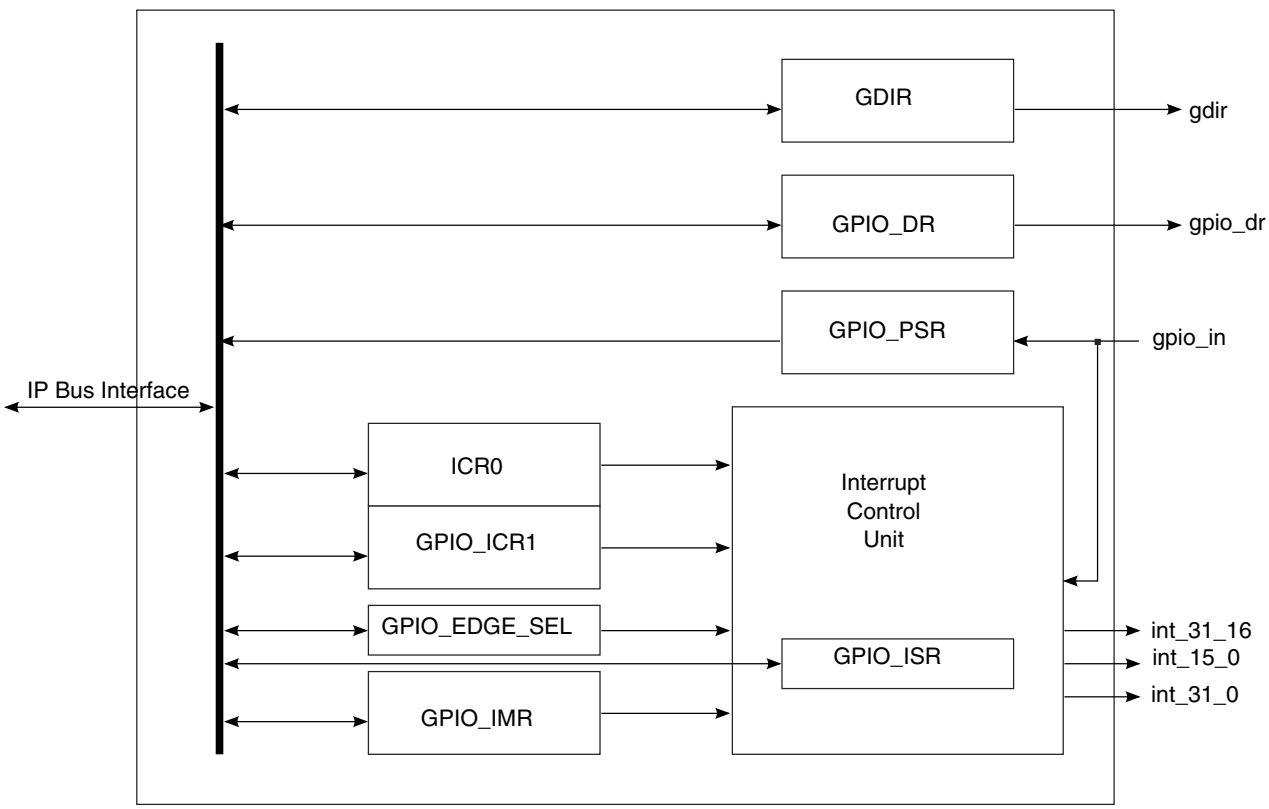


Figure 37-1. GPIO Block Diagram

37.2 General Overview

The GPIO general-purpose input/output peripheral provides dedicated general-purpose pins that can be configured as either inputs or outputs.

When configured as an output, it is possible to write to an internal register to control the state driven on the output pin. When configured as an input, it is possible to detect the state of the input by reading the state of an internal register.

In addition, the GPIO peripheral can produce CORE interrupts.

The GPIO is one of the blocks controlling the IOMUX of the Chip.

shows the Chip multiplexing scheme.

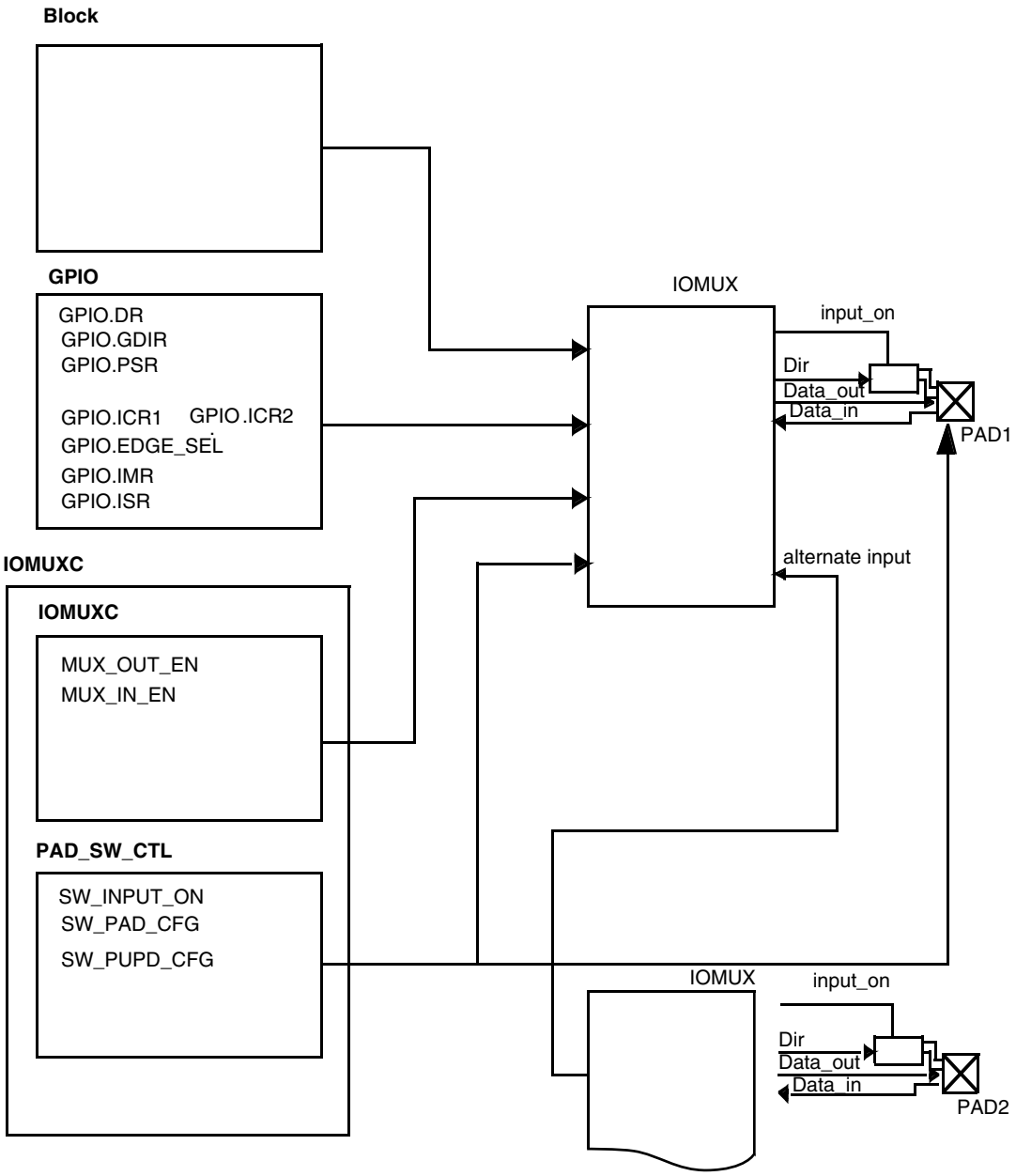


Figure 37-2. Chip IOMUX Scheme

The GPIO functionality is provided through eight registers, an edge-detect circuit, and interrupt generation logic.

The eight registers are:

- Data register (GPIO_DR)
- GPIO direction register (GPIO_GDIR)
- Pad sample register (GPIO_PSR)
- Interrupt control registers (GPIO_ICR1, GPIO_ICR2)

- Edge select register (GPIO_EDGE_SEL)
- Interrupt mask register (GPIO_IMR)
- Interrupt status register (GPIO_ISR)

Each GPIO input has a dedicated edge-detect circuit which can be configured through software to detect rising edges, falling edges, logic low-levels or logic high-levels on the input signals. The outputs of the edge detect circuits are optionally masked by setting the corresponding bit in the interrupt mask register (GPIO_IMR). These qualified outputs are OR'ed together to generate three one-bit interrupt lines:

- interrupt_or_31_16: One-bit interrupt OR of 16 high interrupts.
- interrupt_or_15_0: One-bit interrupt OR of 16 low interrupts.
- ipi_gpio_int: One-bit interrupt OR of 32 interrupts.

In addition, the 32-bit signal ipi_gpio_int32 gives the user access to all 32 individual interrupts.

37.2.1 Features

The GPIO includes the following features:

- General purpose input/output logic capabilities:
  - Drives specific data to output using the data register (GPIO_DR)
  - Controls the direction of the signal using the GPIO direction register (GPIO_GDR)
  - Enables the core to sample the status of the corresponding inputs by reading the pad sample register (GPIO_PSR).
- GPIO interrupt capabilities:
  - Supports up to 32 interrupts
  - Identifies interrupt edges
  - Generates three active-high interrupts to the SoC interrupt controller

37.3 GPIO Functional Description

37.3.1 GPIO Function

A GPIO signal can operate as a general-purpose input/output when the IOMUX is set to GPIO mode. Each GPIO signal may be independently configured as either an input or an output using the GPIO direction register (GPIO_GDIR).

When configured as an output (GPIO_GDIR bit = 1), the value in the data bit in the GPIO data register (GPIO_DR) is driven on the corresponding GPIO line. When a signal is configured as an input (GPIO_GDIR bit = 0), the state of the input can be read from the corresponding PSR bit.

37.3.2 GPIO Programming

37.3.2.1 GPIO Read Mode

The programming sequence for reading input signals should be as follows:

1. Configure IOMUX to select GPIO mode (Via IOMUX Controller (IOMUXC)).
2. Configure GPIO direction register (GPIO_GDR) to input.
3. Read value from data reg*ister*/pad status register.

A pseudocode description to read [input3:input0] values is as follows:

```
write sw_mux_ctl_<input0>_<input1>_<input2>_<input3> , 32'h00000000  // SET INPUTS TO
GPIO MODE.
write GDIR[31:4,input3_bit, input2_bit, input1_bit, input0_bit,] 32'hxxxxxxx0   // SET GDIR
TO
INPUT.
read DR             // READ INPUT VALUE FROM DR.
read PSR            // READ INPUT VALUE FROM PSR.
```

NOTE

While the GPIO direction is set to input (GPIO_GDIR = 0), a read access to GPIO_DR does not return GPIO_DR data. Instead, it returns the PSR data, which is the corresponding input signal value.

37.3.2.2 GPIO Write Mode

The programming sequence for driving output signals should be as follows:

1. Configure IOMUX to select GPIO mode (Via IOMUXC).
2. Configure GPIO direction register (GPIO_GDR) to output.
3. Write value to data register (GPIO_DR).

A pseudocode description to drive 4'b0101 on [output3:output0] is as follows:

```
write sw_mux_ctl_<output0>_<output1>_<output2>_<output3> , 32'h00000000  // SET OUTPUTS
TO GPIO MODE.
write GDIR[31:4,output3_bit,output2_bit, output1_bit, output0_bit,] 32'hxxxxxxxF    // SET
GDIR
TO OUTPUT.
```

```
write DR, 32'hxxxxxxx5          // WRITE OUTPUT VALUE TO DR.
read_cmp PSR, 32'hxxxxxxx5          // READ OUTPUT VALUE FROM PSR ONLY.
```

NOTE

While GPIO direction is set to output, the real internal value can only be verified through the GPIO_PSR, and not through the GPIO_DR.

37.3.3 Interrupt Control Unit

In addition to the general-purpose input/output function, the edge-detect logic in the GPIO peripheral reflects whether a transition has occurred on a given GPIO signal that is configured as an input (GDIR bit = 0). The interrupt control registers (GPIO_ICR1 and GPIO_ICR2) may be used to independently configure the interrupt condition of each input signal (low-to-high transition, high-to-low transition, low, or high).

The interrupt control unit is built of 32 interrupt control subunits, where each subunit handles a single interrupt line.

37.4 Programmable Registers

There are eight 32-bit GPIO registers. All registers are accessible from the IP interface. Only 32-bit access is supported.

The GPIO memory map is shown in the following table.

GPIO memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53F8_4000	GPIO data register (GPIO-1_DR)	32	R/W	0000_0000h	37.4.1/ 1729
53F8_4004	GPIO direction register (GPIO-1_GDIR)	32	R/W	0000_0000h	37.4.2/ 1731
53F8_4008	GPIO pad status register (GPIO-1_PSR)	32	R	0000_0000h	37.4.3/ 1732
53F8_400C	GPIO interrupt configuration register1 (GPIO-1_ICR1)	32	R/W	0000_0000h	37.4.4/ 1733
53F8_4010	GPIO interrupt configuration register2 (GPIO-1_ICR2)	32	R/W	0000_0000h	37.4.5/ 1737

Table continues on the next page...

GPIO memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53F8_4014	GPIO interrupt mask register (GPIO-1_IMR)	32	R/W	0000_0000h	37.4.6/ 1741
53F8_4018	GPIO interrupt status register (GPIO-1_ISR)	32	w1c	0000_0000h	37.4.7/ 1742
53F8_401C	GPIO edge select register (GPIO-1_EDGE_SEL)	32	R/W	0000_0000h	37.4.8/ 1743
53F8_8000	GPIO data register (GPIO-2_DR)	32	R/W	0000_0000h	37.4.1/ 1729
53F8_8004	GPIO direction register (GPIO-2_GDIR)	32	R/W	0000_0000h	37.4.2/ 1731
53F8_8008	GPIO pad status register (GPIO-2_PSR)	32	R	0000_0000h	37.4.3/ 1732
53F8_800C	GPIO interrupt configuration register1 (GPIO-2_ICR1)	32	R/W	0000_0000h	37.4.4/ 1733
53F8_8010	GPIO interrupt configuration register2 (GPIO-2_ICR2)	32	R/W	0000_0000h	37.4.5/ 1737
53F8_8014	GPIO interrupt mask register (GPIO-2_IMR)	32	R/W	0000_0000h	37.4.6/ 1741
53F8_8018	GPIO interrupt status register (GPIO-2_ISR)	32	w1c	0000_0000h	37.4.7/ 1742
53F8_801C	GPIO edge select register (GPIO-2_EDGE_SEL)	32	R/W	0000_0000h	37.4.8/ 1743
53F8_C000	GPIO data register (GPIO-3_DR)	32	R/W	0000_0000h	37.4.1/ 1729
53F8_C004	GPIO direction register (GPIO-3_GDIR)	32	R/W	0000_0000h	37.4.2/ 1731
53F8_C008	GPIO pad status register (GPIO-3_PSR)	32	R	0000_0000h	37.4.3/ 1732
53F8_C00C	GPIO interrupt configuration register1 (GPIO-3_ICR1)	32	R/W	0000_0000h	37.4.4/ 1733
53F8_C010	GPIO interrupt configuration register2 (GPIO-3_ICR2)	32	R/W	0000_0000h	37.4.5/ 1737
53F8_C014	GPIO interrupt mask register (GPIO-3_IMR)	32	R/W	0000_0000h	37.4.6/ 1741
53F8_C018	GPIO interrupt status register (GPIO-3_ISR)	32	w1c	0000_0000h	37.4.7/ 1742
53F8_C01C	GPIO edge select register (GPIO-3_EDGE_SEL)	32	R/W	0000_0000h	37.4.8/ 1743
53F9_0000	GPIO data register (GPIO-4_DR)	32	R/W	0000_0000h	37.4.1/ 1729

Table continues on the next page...

GPIO memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53F9_0004	GPIO direction register (GPIO-4_GDIR)	32	R/W	0000_0000h	37.4.2/ 1731
53F9_0008	GPIO pad status register (GPIO-4_PSR)	32	R	0000_0000h	37.4.3/ 1732
53F9_000C	GPIO interrupt configuration register1 (GPIO-4_ICR1)	32	R/W	0000_0000h	37.4.4/ 1733
53F9_0010	GPIO interrupt configuration register2 (GPIO-4_ICR2)	32	R/W	0000_0000h	37.4.5/ 1737
53F9_0014	GPIO interrupt mask register (GPIO-4_IMR)	32	R/W	0000_0000h	37.4.6/ 1741
53F9_0018	GPIO interrupt status register (GPIO-4_ISR)	32	w1c	0000_0000h	37.4.7/ 1742
53F9_001C	GPIO edge select register (GPIO-4_EDGE_SEL)	32	R/W	0000_0000h	37.4.8/ 1743
53FD_C000	GPIO data register (GPIO-5_DR)	32	R/W	0000_0000h	37.4.1/ 1729
53FD_C004	GPIO direction register (GPIO-5_GDIR)	32	R/W	0000_0000h	37.4.2/ 1731
53FD_C008	GPIO pad status register (GPIO-5_PSR)	32	R	0000_0000h	37.4.3/ 1732
53FD_C00C	GPIO interrupt configuration register1 (GPIO-5_ICR1)	32	R/W	0000_0000h	37.4.4/ 1733
53FD_C010	GPIO interrupt configuration register2 (GPIO-5_ICR2)	32	R/W	0000_0000h	37.4.5/ 1737
53FD_C014	GPIO interrupt mask register (GPIO-5_IMR)	32	R/W	0000_0000h	37.4.6/ 1741
53FD_C018	GPIO interrupt status register (GPIO-5_ISR)	32	w1c	0000_0000h	37.4.7/ 1742
53FD_C01C	GPIO edge select register (GPIO-5_EDGE_SEL)	32	R/W	0000_0000h	37.4.8/ 1743
53FE_0000	GPIO data register (GPIO-6_DR)	32	R/W	0000_0000h	37.4.1/ 1729
53FE_0004	GPIO direction register (GPIO-6_GDIR)	32	R/W	0000_0000h	37.4.2/ 1731
53FE_0008	GPIO pad status register (GPIO-6_PSR)	32	R	0000_0000h	37.4.3/ 1732
53FE_000C	GPIO interrupt configuration register1 (GPIO-6_ICR1)	32	R/W	0000_0000h	37.4.4/ 1733
53FE_0010	GPIO interrupt configuration register2 (GPIO-6_ICR2)	32	R/W	0000_0000h	37.4.5/ 1737

Table continues on the next page...

GPIO memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FE_0014	GPIO interrupt mask register (GPIO-6_IMR)	32	R/W	0000_0000h	37.4.6/ 1741
53FE_0018	GPIO interrupt status register (GPIO-6_ISR)	32	w1c	0000_0000h	37.4.7/ 1742
53FE_001C	GPIO edge select register (GPIO-6_EDGE_SEL)	32	R/W	0000_0000h	37.4.8/ 1743
53FE_4000	GPIO data register (GPIO-7_DR)	32	R/W	0000_0000h	37.4.1/ 1729
53FE_4004	GPIO direction register (GPIO-7_GDIR)	32	R/W	0000_0000h	37.4.2/ 1731
53FE_4008	GPIO pad status register (GPIO-7_PSR)	32	R	0000_0000h	37.4.3/ 1732
53FE_400C	GPIO interrupt configuration register1 (GPIO-7_ICR1)	32	R/W	0000_0000h	37.4.4/ 1733
53FE_4010	GPIO interrupt configuration register2 (GPIO-7_ICR2)	32	R/W	0000_0000h	37.4.5/ 1737
53FE_4014	GPIO interrupt mask register (GPIO-7_IMR)	32	R/W	0000_0000h	37.4.6/ 1741
53FE_4018	GPIO interrupt status register (GPIO-7_ISR)	32	w1c	0000_0000h	37.4.7/ 1742
53FE_401C	GPIO edge select register (GPIO-7_EDGE_SEL)	32	R/W	0000_0000h	37.4.8/ 1743

37.4.1 GPIO data register (GPIOx_DR)

The 32-bit GPIO_DR register stores data that is ready to be driven to the output lines. If the IOMUXC is in GPIO mode and a given GPIO direction bit is set, then the corresponding DR bit is driven to the output. If a given GPIO direction bit is cleared, then a read of GPIO_DR reflects the value of the corresponding signal.Two wait states are required in read access for synchronization.

The results of a read of a DR bit depends on the IOMUXC input mode settings and the corresponding GDIR bit as follows:

- If GDIR[*n*] is set and IOMUXC input mode is GPIO, then reading DR[*n*] returns the contents of DR[*n*].
- If GDIR[*n*] is cleared and IOMUXC input mode is GPIO, then reading DR[*n*] returns the corresponding input signal's value.

- If GDIR[*n*] is set and IOMUXC input mode is not GPIO, then reading DR[*n*] returns the contents of DR[*n*].
- If GDIR[*n*] is cleared and IOMUXC input mode is not GPIO, then reading DR[*n*] always returns zero.

Addresses: GPIO-1_DR is 53F8_4000h base + 0h offset = 53F8_4000h
GPIO-2_DR is 53F8_8000h base + 0h offset = 53F8_8000h
GPIO-3_DR is 53F8_C000h base + 0h offset = 53F8_C000h
GPIO-4_DR is 53F9_0000h base + 0h offset = 53F9_0000h
GPIO-5_DR is 53FD_C000h base + 0h offset = 53FD_C000h
GPIO-6_DR is 53FE_0000h base + 0h offset = 53FE_0000h
GPIO-7_DR is 53FE_4000h base + 0h offset = 53FE_4000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_DR field descriptions

Field	Description
31–0 DR	Data bits. This register defines the value of the GPIO output when the signal is configured as an output (GDIR[n]=1). Writes to this register are stored in a register. Reading GPIO_DR returns the value stored in the register if the signal is configured as an output (GDIR[n]=1), or the input signal's value if configured as an input (GDIR[n]=0). **NOTE:** The I/O multiplexer must be configured to GPIO mode for the GPIO_DR value to connect with the signal. Reading the data register with the input path disabled always returns a zero value.

37.4.2 GPIO direction register (GPIOx_GDIR)

GPIO_GDIR functions as direction control when the IOMUXC is in GPIO mode. Each bit specifies the direction of a one-bit signal. The mapping of each DIR bit to a corresponding SoC signal is determined by the SoC's pin assignment and the IOMUX table-for more details consult the IOMUXC chapter.

Addresses: GPIO-1_GDIR is 53F8_4000h base + 4h offset = 53F8_4004h
GPIO-2_GDIR is 53F8_8000h base + 4h offset = 53F8_8004h
GPIO-3_GDIR is 53F8_C000h base + 4h offset = 53F8_C004h
GPIO-4_GDIR is 53F9_0000h base + 4h offset = 53F9_0004h
GPIO-5_GDIR is 53FD_C000h base + 4h offset = 53FD_C004h
GPIO-6_GDIR is 53FE_0000h base + 4h offset = 53FE_0004h
GPIO-7_GDIR is 53FE_4000h base + 4h offset = 53FE_4004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GDIR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

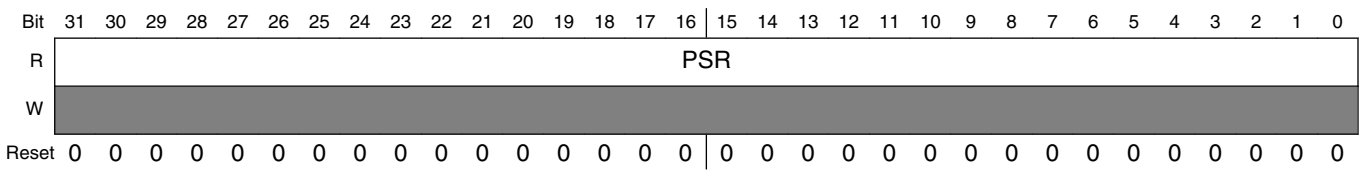
GPIOx_GDIR field descriptions

Field	Description
31–0 GDIR	GPIO direction bits. Bit n of this register defines the direction of the GPIO[n] signal. **NOTE:** GPIO_GDIR affects only the direction of the I/O signal when the corresponding bit in the I/O MUX is configured for GPIO. 0 GPIO is configured as input. 1 GPIO is configured as output.

37.4.3 GPIO pad status register (GPIOx_PSR)

GPIO_PSR is a read-only register. Each bit stores the value of the corresponding input signal (as configured in the IOMUX). This register is clocked with the ipg_clk_s clock, meaning that the input signal is sampled only when accessing this location. Two wait states are required any time this register is accessed for synchronization.

Addresses: GPIO-1_PSR is 53F8_4000h base + 8h offset = 53F8_4008h
GPIO-2_PSR is 53F8_8000h base + 8h offset = 53F8_8008h
GPIO-3_PSR is 53F8_C000h base + 8h offset = 53F8_C008h
GPIO-4_PSR is 53F9_0000h base + 8h offset = 53F9_0008h
GPIO-5_PSR is 53FD_C000h base + 8h offset = 53FD_C008h
GPIO-6_PSR is 53FE_0000h base + 8h offset = 53FE_0008h
GPIO-7_PSR is 53FE_4000h base + 8h offset = 53FE_4008h



GPIOx_PSR field descriptions

Field	Description
31–0 PSR	GPIO pad status bits (status bits). Reading GPIO_PSR returns the state of the corresponding input signal. Settings: **NOTE:** The I/O multiplexer must be configured to GPIO mode for GPIO_PSR to reflect the state of the corresponding signal.

37.4.4 GPIO interrupt configuration register1 (GPIOx_ICR1)

GPIO_ICR1 contains 16 two-bit fields, where each field specifies the interrupt configuration for a different input signal.

Addresses: GPIO-1_ICR1 is 53F8_4000h base + Ch offset = 53F8_400Ch
GPIO-2_ICR1 is 53F8_8000h base + Ch offset = 53F8_800Ch
GPIO-3_ICR1 is 53F8_C000h base + Ch offset = 53F8_C00Ch
GPIO-4_ICR1 is 53F9_0000h base + Ch offset = 53F9_000Ch
GPIO-5_ICR1 is 53FD_C000h base + Ch offset = 53FD_C00Ch
GPIO-6_ICR1 is 53FE_0000h base + Ch offset = 53FE_000Ch
GPIO-7_ICR1 is 53FE_4000h base + Ch offset = 53FE_400Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ICR15		ICR14		ICR13		ICR12		ICR11		ICR10		ICR9		ICR8		ICR7		ICR6		ICR5		ICR4		ICR3		ICR2		ICR1		ICR0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_ICR1 field descriptions

Field	Description
31–30 ICR15	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for lines 15 to 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
29–28 ICR14	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for lines 15 to 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
27–26 ICR13	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for lines 15 to 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive.

Table continues on the next page...

GPIOx_ICR1 field descriptions (continued)

Field	Description
	10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
25–24 ICR12	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for lines 15 to 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
23–22 ICR11	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for lines 15 to 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
21–20 ICR10	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for lines 15 to 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
19–18 ICR9	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for lines 15 to 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
17–16 ICR8	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for lines 15 to 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive.

Table continues on the next page...

GPIOx_ICR1 field descriptions (continued)

Field	Description
	10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
15–14 ICR7	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for lines 15 to 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
13–12 ICR6	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for lines 15 to 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
11–10 ICR5	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for lines 15 to 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
9–8 ICR4	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for lines 15 to 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
7–6 ICR3	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for lines 15 to 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive.

Table continues on the next page...

GPIOx_ICR1 field descriptions (continued)

Field	Description
	10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
5–4 ICR2	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for lines 15 to 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
3–2 ICR1	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for lines 15 to 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
1–0 ICR0	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for lines 15 to 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.

37.4.5 GPIO interrupt configuration register2 (GPIOx_ICR2)

GPIO_ICR2 contains 16 two-bit fields, where each field specifies the interrupt configuration for a different input signal.

Addresses: GPIO-1_ICR2 is 53F8_4000h base + 10h offset = 53F8_4010h
GPIO-2_ICR2 is 53F8_8000h base + 10h offset = 53F8_8010h
GPIO-3_ICR2 is 53F8_C000h base + 10h offset = 53F8_C010h
GPIO-4_ICR2 is 53F9_0000h base + 10h offset = 53F9_0010h
GPIO-5_ICR2 is 53FD_C000h base + 10h offset = 53FD_C010h
GPIO-6_ICR2 is 53FE_0000h base + 10h offset = 53FE_0010h
GPIO-7_ICR2 is 53FE_4000h base + 10h offset = 53FE_4010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ICR31		ICR30		ICR29		ICR28		ICR27		ICR26		ICR25		ICR24		ICR23		ICR22		ICR21		ICR20		ICR19		ICR18		ICR17		ICR16	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_ICR2 field descriptions

Field	Description
31–30 ICR31	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for lines 31 to 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
29–28 ICR30	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for lines 31 to 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
27–26 ICR29	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for lines 31 to 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive.

Table continues on the next page...

GPIOx_ICR2 field descriptions (continued)

Field	Description
	10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
25–24 ICR28	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for lines 31 to 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
23–22 ICR27	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for lines 31 to 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
21–20 ICR26	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for lines 31 to 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
19–18 ICR25	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for lines 31 to 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
17–16 ICR24	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for lines 31 to 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive.

Table continues on the next page...

GPIOx_ICR2 field descriptions (continued)

Field	Description
	10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
15–14 ICR23	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for lines 31 to 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
13–12 ICR22	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for lines 31 to 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
11–10 ICR21	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for lines 31 to 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
9–8 ICR20	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for lines 31 to 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
7–6 ICR19	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for lines 31 to 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive.

Table continues on the next page...

GPIOx_ICR2 field descriptions (continued)

Field	Description
	10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
5–4 ICR18	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for lines 31 to 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
3–2 ICR17	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for lines 31 to 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.
1–0 ICR16	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for lines 31 to 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 Interrupt n is low-level sensitive. 01 Interrupt n is high-level sensitive. 10 Interrupt n is rising-edge sensitive. 11 Interrupt n is falling-edge sensitive.

37.4.6 GPIO interrupt mask register (GPIOx_IMR)

GPIO_IMR contains masking bits for each interrupt line.

Addresses: GPIO-1_IMR is 53F8_4000h base + 14h offset = 53F8_4014h
GPIO-2_IMR is 53F8_8000h base + 14h offset = 53F8_8014h
GPIO-3_IMR is 53F8_C000h base + 14h offset = 53F8_C014h
GPIO-4_IMR is 53F9_0000h base + 14h offset = 53F9_0014h
GPIO-5_IMR is 53FD_C000h base + 14h offset = 53FD_C014h
GPIO-6_IMR is 53FE_0000h base + 14h offset = 53FE_0014h
GPIO-7_IMR is 53FE_4000h base + 14h offset = 53FE_4014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

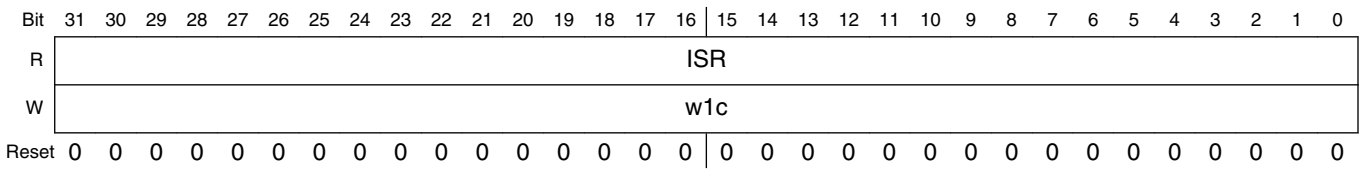
GPIOx_IMR field descriptions

Field	Description
31–0 IMR	Interrupt Mask bits. This register is used to enable or disable the interrupt function on each of the 32 GPIO signals. Settings: Bit IMR[n] (n=0...31) controls interrupt n as follows: 0 Interrupt n is disabled. 1 Interrupt n is enabled.

37.4.7 GPIO interrupt status register (GPIOx_ISR)

The GPIO_ISR functions as an interrupt status indicator. Each bit indicates whether an interrupt condition has been met for the corresponding input signal. When an interrupt condition is met (as determined by the corresponding interrupt condition register field), the corresponding bit in this register is set. Two wait states are required in read access for synchronization. One wait state is required for reset.

Addresses: GPIO-1_ISR is 53F8_4000h base + 18h offset = 53F8_4018h
GPIO-2_ISR is 53F8_8000h base + 18h offset = 53F8_8018h
GPIO-3_ISR is 53F8_C000h base + 18h offset = 53F8_C018h
GPIO-4_ISR is 53F9_0000h base + 18h offset = 53F9_0018h
GPIO-5_ISR is 53FD_C000h base + 18h offset = 53FD_C018h
GPIO-6_ISR is 53FE_0000h base + 18h offset = 53FE_0018h
GPIO-7_ISR is 53FE_4000h base + 18h offset = 53FE_4018h



GPIOx_ISR field descriptions

Field	Description
31–0 ISR	Interrupt status bits - Bit n of this register is asserted (active high) when the active condition (as determined by the corresponding ICR bit) is detected on the GPIO input and is waiting for service. The value of this register is independent of the value in GPIO_IMR. When the active condition has been detected, the corresponding bit remains set until cleared by software. Status flags are cleared by writing a 1 to the corresponding bit position.

37.4.8 GPIO edge select register (GPIOx_EDGE_SEL)

GPIO_EDGE_SEL may be used to override the ICR registers' configuration. If the GPIO_EDGE_SEL bit is set, then a rising edge or falling edge in the corresponding signal generates an interrupt. This register provides backward compatibility. On reset all bits are cleared (ICR is not overridden).

Addresses: GPIO-1_EDGE_SEL is 53F8_4000h base + 1Ch offset = 53F8_401Ch
GPIO-2_EDGE_SEL is 53F8_8000h base + 1Ch offset = 53F8_801Ch
GPIO-3_EDGE_SEL is 53F8_C000h base + 1Ch offset = 53F8_C01Ch
GPIO-4_EDGE_SEL is 53F9_0000h base + 1Ch offset = 53F9_001Ch
GPIO-5_EDGE_SEL is 53FD_C000h base + 1Ch offset = 53FD_C01Ch
GPIO-6_EDGE_SEL is 53FE_0000h base + 1Ch offset = 53FE_001Ch
GPIO-7_EDGE_SEL is 53FE_4000h base + 1Ch offset = 53FE_401Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPIO_EDGE_SEL																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_EDGE_SEL field descriptions

Field	Description
31–0 GPIO_EDGE_SEL	Edge select. When GPIO_EDGE_SEL[*n*] is set, the GPIO disregards the ICR[*n*] setting, and detects any edge on the corresponding input signal.

Chapter 38
General Purpose Timer (GPT)

38.1 Overview

This chapter describes the General Purpose Timer (GPT) module interface. It is also a reference for software driver programming. The GPT has a 32-bit up-counter. The timer counter value can be captured in a register using an event on an external pin. The capture trigger can be programmed to be a rising or/and falling edge. The GPT can also generate an event on the DO_CMPOUT*n* pins and an interrupt when the timer reaches a programmed value. The GPT has a 12-bit prescaler, which provides a programmable clock frequency derived from multiple clock sources.

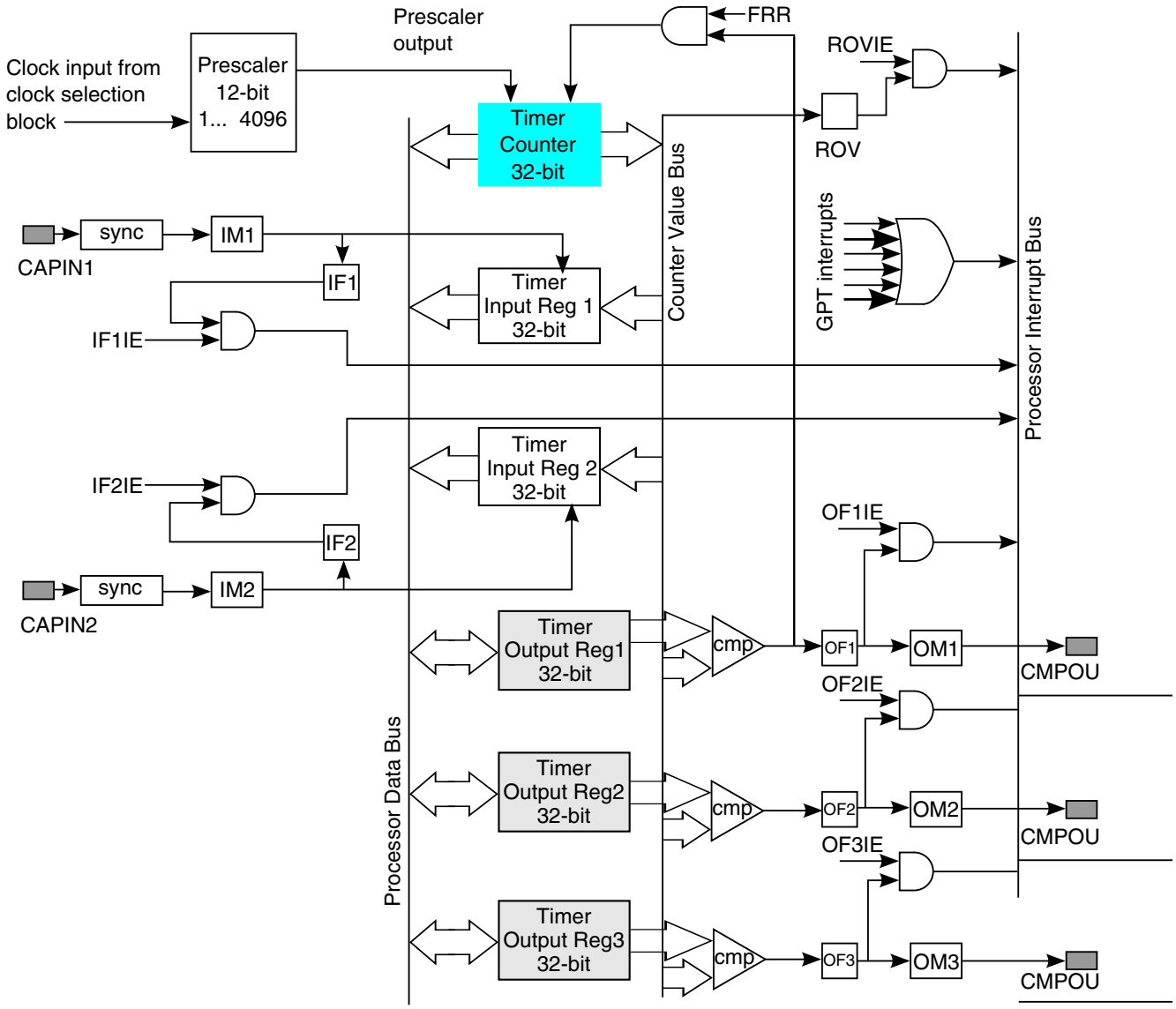


Figure 38-1. GPT Block Diagram

The following figure shows the GPT functional clocking scheme.

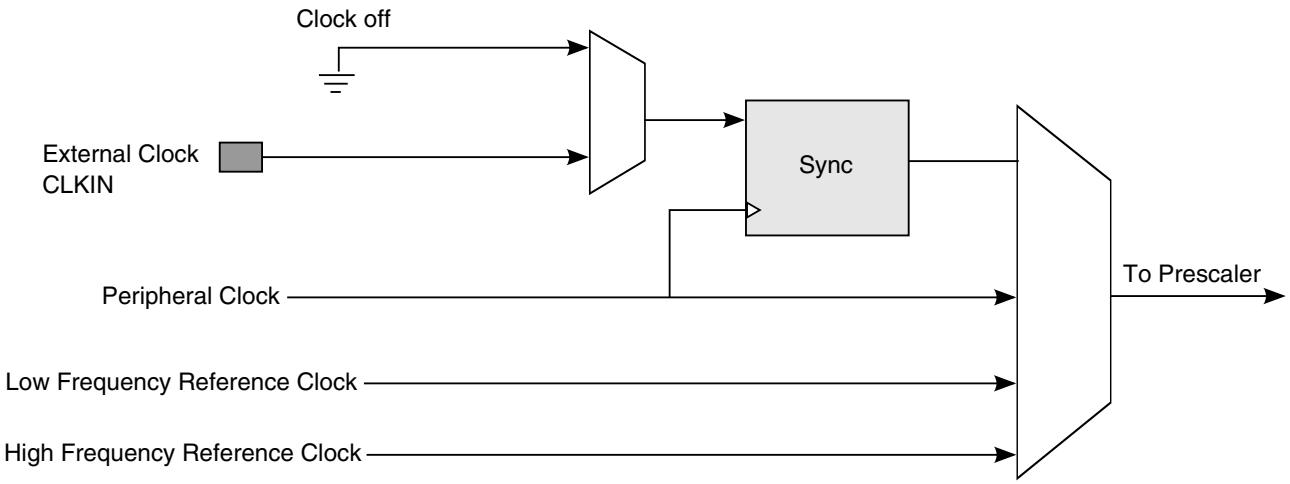


Figure 38-2. GPT Counter Clocks Diagram

38.1.1 Features

- One 32-bit up-counter with clock source selection, including external clock.
- Two input capture channels with a programmable trigger edge.
- Three output compare channels with a programmable output mode. A "forced compare" feature is also available.
- Can be programmed to be *active* in low power and debug modes.
- Interrupt generation at capture, compare, and rollover events.
- Restart or free-run modes for counter operations.

38.1.2 Modes and Operation

The GPT supports the modes described in the indicated sections:

- Operating Modes
  - Restart Mode
  - Free-Run Mode

38.2 External Signals

The GPT follows the IP Bus protocol for interfacing with the processor core. The GPT does not have *any interface signals with any other module inside the chip,* except for the clock and reset inputs (from the clock and reset controller module) and for the interrupt signals *to* the processor interrupt handler. There are functional and clock inputs, and functional output signals going outside the chip boundary.

The following table describes all block signals that connect off-chip.

Table 38-1. Off-Chip Module Signals

Name	Direction	Function	Reset State	Pull-Up
IND_CLKIN	I	Input pin for an external clock that the counter can be operated at.	-	Passive Hysteresis
IND_CAPIN1	I	Input pin for a capture event for Input Capture Channel 1.	-	Passive
IND_CAPIN2	I	Input pin for a capture event for Input Capture Channel 2.	-	Passive
DO_CMPOUT1	O	Output pin that indicates a "compare event" occurrence in Output Compare Channel 1.	0	Passive
DO_CMPOUT2	O	Output pin that indicates a "compare event" occurrence in Output Compare Channel 2.	0	Passive
DO_CMPOUT3	O	Output pin that indicates a "compare event" occurrence in Output Compare Channel 3.	0	Passive

There are six signals (three input, three output) in the GPT module that *can be* connected to the chip pads.

38.2.1 External Clock Input: IND_CLKIN

The GPT counter can be operated using an external clock from outside the device, and this is the input pin used for that purpose. This clock is treated as asynchronous to the peripheral clock. To ensure proper operations of GPT, the external clock input frequency should be less than 1/4 of frequency of the peripheral clock. Hysteresis characteristics on this pad will be required because this is a clock input.

38.2.2 Input Capture Trigger Signals: IND_CAPIN1, IND_CAPIN2

The GPT counter value can be stored in a register, triggered by an event from *outside the device*. A positive or/and negative edge on these signals can trigger this capture event. These signals are treated as asynchronous to the peripheral clock. Only those transitions which occur *at least a single clock cycle* (the clock selected to run the counter) *after the previous recorded transition* are guaranteed to trigger a capture event.

38.2.3 Output Compare Signals: DO_CMPOUT1, DO_CMPOUT2, DO_CMPOUT3

The output compare signals indicate that output compare events have gone through a specified transition.

38.3 Functional Description

This section provides a complete functional description of the GPT.

38.3.1 Operating Modes

The GPT counter can be programmed to work in either of two modes: Restart mode or Free-Run mode.

38.3.1.1 Restart Mode

In Restart mode (selectable through the GPT Control Register GPT_CR), when the counter reaches the compared value, the counter resets and starts again from 0x00000000. The Restart feature is associated only with Compare Channel 1.

Any write access to the Compare register of Channel 1 will reset the GPT counter. This is done to avoid possibly missing a compare event when compare value is changed from a higher value to lower value while counting is proceeding.

For the other two compare channels, when the compare event occurs the counter is *not reset*.

38.3.1.2 Free-Run Mode

In Free-Run mode, when compare events occur for all 3 channels, the counter is *not reset*; instead the counter continues to count until 0xffffffff, and then rolls over (to 0x00000000).

38.3.2 Operation

The General Purpose Timer (GPT) has a single counter (GPT_CNT) that is a 32-bit free-running *up-counter,* which starts counting *after it is enabled by software* (EN=1).

The counter's clock source is the output of the prescaler labelled "Prescaler output" in Figure 38-1.

- If the GPT timer is disabled (EN=0), then the Main Counter *and* Prescaler Counter freeze their current count values. The ENMOD bit determines the value of the GPT counter when the EN bit is set and the Counter is enabled again.
  - If the ENMOD bit is set (=1), then the Main Counter and Prescaler Counter values are reset to 0, when GPT is enabled (EN=1).
  - If ENMOD bit is programmed to 0, then the Main Counter and Prescaler Counter restart counting from their frozen values, when GPT is enabled again (EN=1).
- If GPT is programmed to be disabled in a low power mode (STOP/WAIT), then the Main Counter and Prescaler Counter freeze at their current count values *when* GPT enters low power mode. When GPT exits a low power mode, the Main Counter and Prescaler Counter start counting from their frozen values *regardless* of the ENMOD bit value. Note that the GPT_CNT can be read *at any time* by the processor, and that *both* Input Capture Channels use the *same* counter (GPT_CNT).
- A hardware reset resets all the GPT registers to their respective reset values. All registers except the Output Compare Registers (OCR1, OCR2, OCR3) obtain a value of 0x0. The Compare registers are reset to 0xffffffff.
- The software reset (SWR bit in the GPT_CR control register) resets *all* of the register bits *except* the EN, ENMOD, STOPEN, WAITEN, and DBGEN bits. The state of these bits is not affected by a software reset. Note that a software reset can be given *while the GPT is disabled.*

38.3.2.1 Clocks

The clock that is input to the prescaler can be selected from 4 clock sources:

- High Frequency Clock

Provided by the Clock Controller Module (CCM), the High Frequency Clock is intended to be ON in Normal Power mode when the Peripheral Clock is turned OFF, thereby enabling the GPT to be operated using the High Frequency Clock *in Normal Power mode*. The CCM is expected to provide this clock *after* synchronizing it to the System Bus Clock in Normal functional mode; the CCM is also expected to switch to the *unsynchronized* version of the High Frequency Clock in a Low Power mode.

- Low Reference Clock

  This 32 kHz Low Reference Clock (provided by the CCM) is intended to be ON in Low Power mode when the Peripheral Clock is turned OFF, thereby enabling the GPT to be operated using the Low Reference Clock in Low Power mode. The CCM is expected to provide the Low Reference Clock *after* synchronizing it to the System Bus Clock in Normal functional mode; the CCM is also expected to switch to the *unsynchronized* version of the Low Reference Clock in a Low Power mode.

- External Clock

  The External Clock comes from *outside the device* and can be selected to run the GPT counter. The External Clock is treated as *asynchronous to the Peripheral Clock,* and is synchronized to the Peripheral Clock, *inside* the module. Therefore, the External Clock frequency is limited to < 1/4 frequency of the Peripheral Clock, for proper GPT operations. Note that in Low Power modes, *if* the Peripheral Clock is not available, then the External Clock *cannot be used* to run the counter.

- Peripheral Clock

  If the Peripheral Clock or the External Clock is selected(CLKSRC=001 or 011) as Clock Source, then the Peripheral Clock will be ON in normal GPT operations. In Low Power modes, if the GPT is programmed to be disabled (STOPEN or WAITEN or DOZEN=0), then the Peripheral Clock can be switched OFF.

- Crystal Oscillator Clock

  This 24MHz Crystal Oscillator Clock (provided by the CCM) is intended to be used against frequency change of Peripheral Clock changes to provide a more accruate timer clock for operation system. The CCM is expected to provide the 24M Crystal Oscillator Clock *without* synchronizing it to the System Bus Clock in Normal functional mode. Synchronization is done in GPT module. Before synchronization, the 24M Crystal Oscillator Clock is divided by a 24M clock prescaler, to make sure the clock frequency less than half of System Bus Clock .

The clock input source is configured using the clock source field (CLKSRC, in the GPT_CR control register). The clock input to the prescaler can be disabled by programming the CLKSRC bits (of the GPT_CR control register) to 000. **The CLKSRC field value should be changed** *only after disabling the GPT* (by setting the EN bit in the GPT_CR to 0).

The PRESCALER field selects the divide ratio of the input clock that drives the main counter. The prescaler can divide the input clock by a value (from 1 to 4096) and can be changed *at any time*. A change in the value of the PRESCALER field *immediately affects* the output clock frequency.

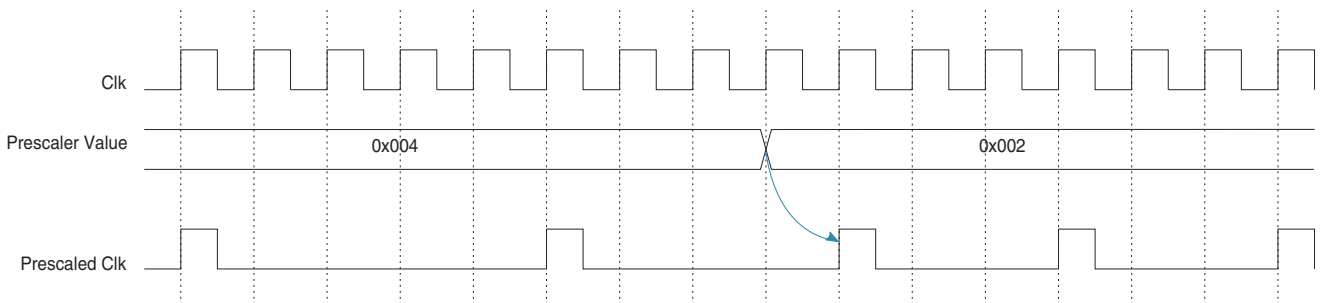


Figure 38-3. Prescaler Value Change Timing Diagram

38.3.2.2 Input Capture

There are two Input Capture Channels, and each Input Capture Channel has a dedicated capture pin, capture register and input edge detection/selection logic. Each input capture function has an associated status flag, and can cause the processor to make an interrupt service request.

When a selected edge transition occurs on an Input Capture pin, the contents of the GPT_CNT is captured on the corresponding capture register and the appropriate interrupt status flag is set. An interrupt request can be generated when the transition is detected *if* its corresponding enable bit is set (in the Interrupt Register). The capture can be programmed to occur on the input pin's rising edge, falling edge, on both rising and falling edges, or the capture can be disabled. The events are synchronized with the clock that was selected to run the counter. Only those transitions that occur at least one clock cycle (clock selected to run the counter) *after* the previous recorded transition will be guaranteed to trigger a capture event. There can be up to one clock cycle of uncertainty in the latching of the input transition. The Input Capture registers can be read *at any time* without affecting their values.

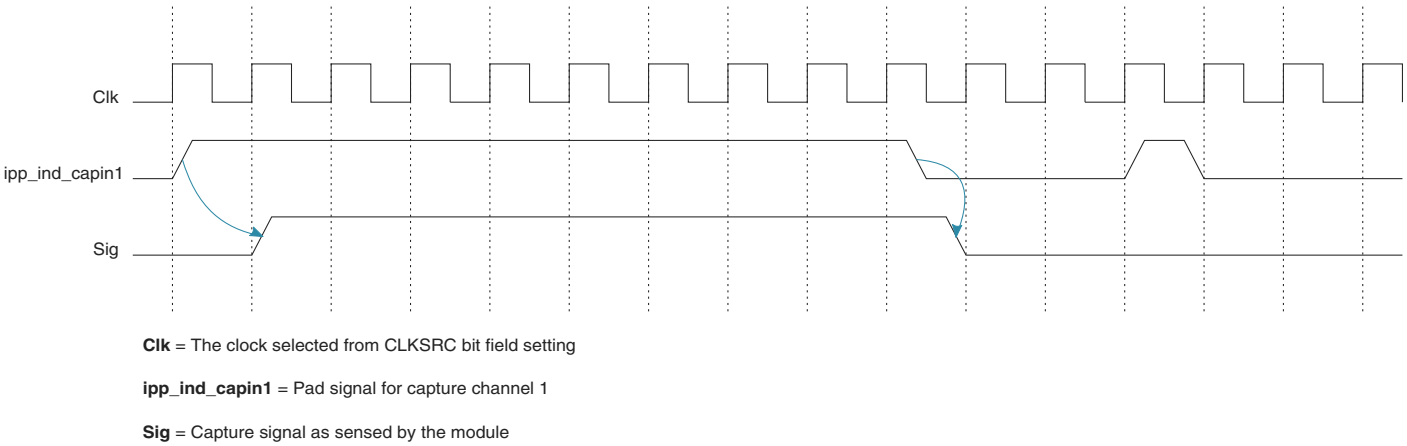


Figure 38-4. Input Capture Event Timing

38.3.2.3 Output Compare

The three Output Compare Channels *use the same counter* (GPT_CNT) as the Input Capture Channels. When the programmed content of an Output Compare register matches the value in GPT_CNT, an output compare status flag is set and an interrupt is generated (if the corresponding bit is set in the interrupt register). Consequently, the Output Compare timer pin will be set, cleared, toggled, not affected at all or provide an active-low pulse for one input clock period (subject to the restriction on the maximum frequency allowed on the pad) according to the mode bits (that were programmed).

There is also a "forced-compare" feature that allows the software to generate a compare event when required, *without the condition of the counter value that is equal to the compare value*. The action taken as a result of a forced compare is the same as when an output compare match occurs, *except that the status flags are not set and no interrupt can be generated*. Forced channels take programmed action immediately after the write to the force-compare bits. These bits are self-negating and always read as zeros.

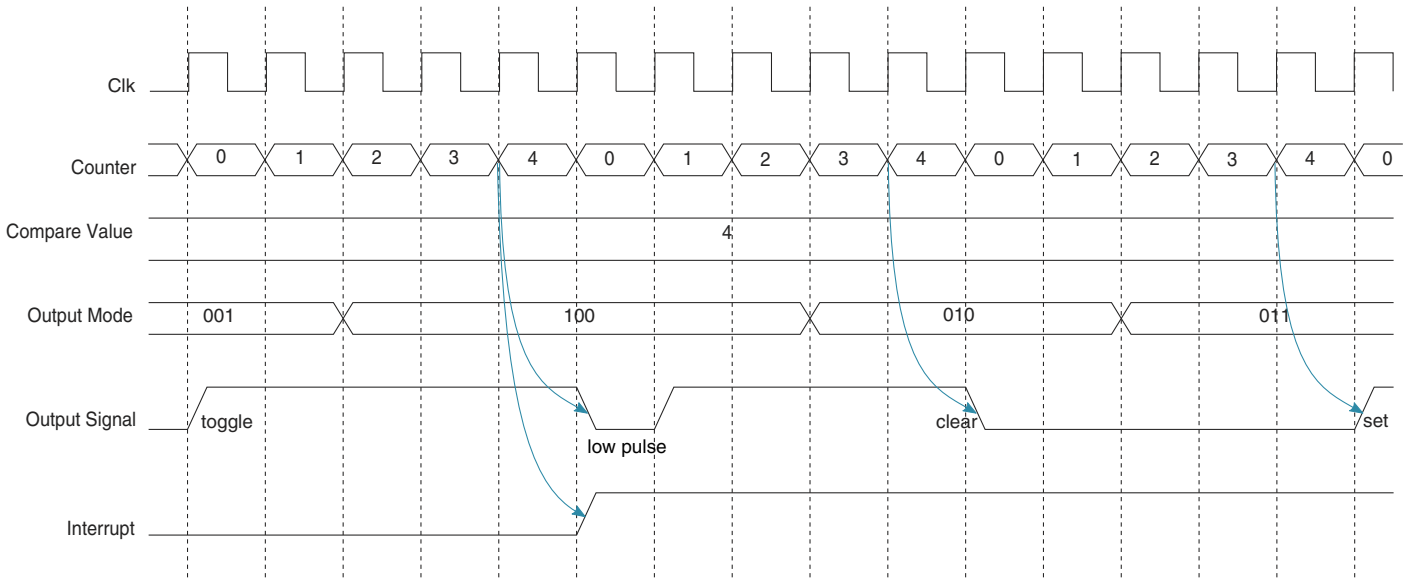


Figure 38-5. Output Compare and Interrupt Timing

38.3.2.4 Interrupts

There are 6 different interrupts that are generated by the GPT. If the selected clock for running the counter is available, then *all interrupts can be generated in Low Power and Debug modes*.

- Rollover Interrupt

  The Rollover Interrupt is generated when the GPT counter reaches 0xffffffff, then resets to 0x00000000 and continues counting. The Rollover Interrupt is enabled by the ROVIE bit in the GPT_IR register; the associated status bit is the ROV bit in the GPT_SR register.

- Input Capture Interrupt 1, 2

  After a capture event occurs, the associated Input Capture Channel generates an interrupt. The "capture event" interrupts are enabled by the IF2IE and IF1IE bits (in the GPT_IR register); the associated status bits are IF2 and IF1 (in the GPT_SR register). The capture of the counter value because of a capture event is *not affected by a pending capture interrupt*. The Capture register is updated with a new counter value when a capture event occurs, regardless of whether that Capture Channels' interrupt has been serviced or not.

- Output Compare Interrupt 1, 2, 3

After a compare event occurs, the associated Output Compare Channel generates an interrupt. The "compare event" interrupts are enabled by the OF3IE, OF2IE, and OF1IE bits (in the GPT_IR register); the associated status bits are OF3, OF2, and OF1 (in the GPT_SR register). A "forced compare" does not generate an interrupt.

A *cumulative* interrupt line is also present, which is asserted whenever any of the above interrupts are posted. The cumulative interrupt line has *no* associated enables or status bits.

38.3.2.5 Low Power Mode Behavior

In Low Power modes, if the clock from the selected clock source is available (except for the External Clock, which can be used *only if* the Peripheral Clock is available), the counter will continue to run depending on whether the control bit for that mode is set. If the clock is not present or if the corresponding low power bit in the GPT_CR control register is 0, the Main Counter and the Prescaler Counter freeze at their current values and resume counting (from their frozen values) when the Low Power mode is exited.

38.3.2.6 Debug Mode Behavior

In Debug mode, the modules in the device have the option of continuing to run or be halted.

- If the DBGEN bit is set, then the GPT timer will continue to run in Debug mode.
- If the DBGEN bit is not set (in the GPT_CR control register), then the GPT timer is halted.

38.4 Programmable Registers

The GPT has 10 user-accessible 32-bit registers, which are used to configure, operate, and monitor the state of the GPT.

An IP bus write access to the GPT Control Register (GPT_CR) and the GPT Output Compare Register1 (GPT_OCR1) results in *one cycle of wait state*, while other valid IP bus accesses incur 0 wait states.

Irrespective of the Response Select signal value, a Write access to the GPT Status Registers (Read-only registers GPT_ICR1, GPT_ICR2, GPT_CNT) will generate a bus exception.

- If the Response Select signal is driven Low, then the Read/Write access to the *unimplemented* address space of GPT (*ips_addr* is greater than or equal to $BASE + $028) will generate a bus exception.
- If the Response Select is driven High, then the Read/Write access to the unimplemented address space of GPT will *not* generate any error response (like a bus exception).

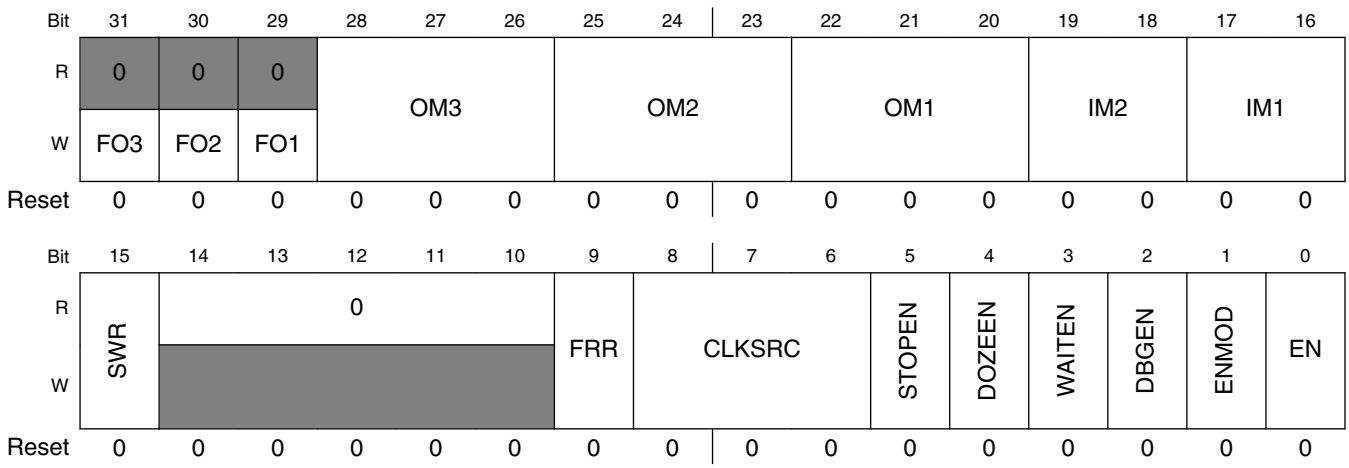
GPT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_0000	GPT Control Register (GPT_CR)	32	R/W	0000_0000h	38.4.1/ 1757
53FA_0004	GPT Prescaler Register (GPT_PR)	32	R/W	0000_0000h	38.4.2/ 1760
53FA_0008	GPT Status Register (GPT_SR)	32	R/W	0000_0000h	38.4.3/ 1761
53FA_000C	GPT Interrupt Register (GPT_IR)	32	R/W	0000_0000h	38.4.4/ 1762
53FA_0010	GPT Output Compare Register 1 (GPT_OCR1)	32	R/W	FFFF_ FFFFh	38.4.5/ 1763
53FA_0014	GPT Output Compare Register 2 (GPT_OCR2)	32	R/W	FFFF_ FFFFh	38.4.6/ 1764
53FA_0018	GPT Output Compare Register 3 (GPT_OCR3)	32	R/W	FFFF_ FFFFh	38.4.7/ 1764
53FA_001C	GPT Input Capture Register 1 (GPT_ICR1)	32	R	0000_0000h	38.4.8/ 1765
53FA_0020	GPT Input Capture Register 2 (GPT_ICR2)	32	R	0000_0000h	38.4.9/ 1765
53FA_0024	GPT Counter Register (GPT_CNT)	32	R	0000_0000h	38.4.10/ 1766

38.4.1 GPT Control Register (GPT_CR)

The GPT Control Register (GPT_CR) is used to program and configure GPT operations. An IP Bus Write to the GPT Control Register occurs after one cycle of wait state, while an IP Bus Read occurs after 0 wait states.

Address: GPT_CR is 53FA_0000h base + 0h offset = 53FA_0000h



GPT_CR field descriptions

Field	Description
31 FO3	FO3 Force Output Compare Channel 3 FO2 Force Output Compare Channel 2 FO1 Force Output Compare Channel 1 The FO*n* bit causes the pin action *programmed* for the timer Output Compare *n* pin (according to the OM*n* bits in this register). • The OF*n* flag (OF3, OF2, OF1) in the status register is **not affected**. • This bit is self-negating and always read as zero. 0 Writing a 0 has no effect. 1 Causes the programmed pin action on the timer Output Compare *n* pin; the OF*n* flag is not set.
30 FO2	See F03
29 FO1	See F03
28–26 OM3	OM3 (bits 28-26) controls the Output Compare Channel 3 operating mode. OM2 (bits 25-23) controls the Output Compare Channel 2 operating mode. OM1 (bits 22-20) controls the Output Compare Channel 1 operating mode. The OM*n* bits specify the response that a compare event will generate on the output pin of Output Compare Channel *n*.

Table continues on the next page...

GPT_CR field descriptions (continued)

Field	Description
	• The toggle, clear, and set options cause a change on the output pin *only* if a compare event occurs. • When OM*n* is programmed as 1xx (active low pulse), the output pin is set to one immediately on the next input clock; a low pulse (that is an input clock in width) occurs when there is a compare event. Note that here, "input clock" refers to the clock selected by the CLKSRC bits of the GPT Control Register. 000 Output disconnected. No response on pin. 001 Toggle output pin 010 Clear output pin 011 Set output pin 1xx Generate an active low pulse (that is one input clock wide) on the output pin.
25–23 OM2	See OM3
22–20 OM1	See OM3
19–18 IM2	IM2 (bits 19-18, Input Capture Channel 2 operating mode) IM1 (bits 17-16, Input Capture Channel 1 operating mode) The IM*n* bit field determines the transition on the input pin (for Input capture channel *n*), which will trigger a capture event. 00 capture disabled 01 capture on rising edge only 10 capture on falling edge only 11 capture on both edges
17–16 IM1	See IM2
15 SWR	Software reset. This is the software reset of the GPT module. It is a self-clearing bit. • The SWR bit is set when the module is in reset state. • The SWR bit is cleared when the reset procedure finishes. • Setting the SWR bit resets **all of the registers** to their default reset values, except for the CLKSRC, EN, ENMOD, STOPEN, WAITEN, and DBGEN bits in the GPT Control Register (this control register). 0 GPT is not in reset state 1 GPT is in reset state
14–10 Reserved	This read-only field is reserved and always has the value zero. Reserved bits. • Writing a value to these reserved bits will not affect GPT operations. • These reserved bits are always read as zero. • It is recommended that all writes to these reserved bits be 0 (for forward compatibility).
9 FRR	Free-Run or Restart mode. The FFR bit determines the behavior of the GPT when a compare event in channel 1 occurs. • In Restart mode, after a compare event, the counter resets to 0x00000000 and resumes counting (after the occurrence of a compare event). • In Free-Run mode, after a compare event, the counter continues counting until 0xFFFFFFFF and then rolls over to 0.

Table continues on the next page...

GPT_CR field descriptions (continued)

Field	Description
	0 Restart mode 1 Free-Run mode
8–6 CLKSRC	Clock Source select. The CLKSRC bits select which clock will go to the prescaler (and subsequently be used to run the GPT counter). • The CLKSRC bit field value should only be changed after disabling the GPT by clearing the EN bit in this register (GPT_CR). • A software reset does not affect the CLKSRC bit. 000 No clock 001 Peripheral Clock 010 High Frequency Reference Clock 011 External Clock (IND_CLKIN) 100 Low Frequency Reference Clock 101 Crystal oscillator as Reference Clock others Reserved
5 STOPEN	GPT Stop Mode enable. The STOPEN read/write control bit enables GPT operation *during Stop mode*. • A hardware reset resets the STOPEN bit. • A software reset *does not affect* the STOPEN bit. 0 GPT is disabled in Stop mode. 1 GPT is enabled in Stop mode.
4 DOZEEN	GPT Doze Mode Enable. • A hardware reset resets the DOZEEN bit. • A software reset *does not affect* the DOZEEN bit. 0 GPT is disabled in doze mode. 1 GPT is enabled in doze mode.
3 WAITEN	GPT Wait Mode enable. The WAITEN read/write control bit enables GPT operation *during Wait mode*. • A hardware reset resets the WAITEN bit. • A software reset *does not affect* the WAITEN bit. 0 GPT is disabled in wait mode. 1 GPT is enabled in wait mode.
2 DBGEN	GPT debug mode enable. The DBGEN read/write control bit enables GPT operation *during Debug mode*. • A hardware reset resets the DBGEN bit. • A software reset *does not affect* the DBGEN bit. 0 GPT is disabled in debug mode. 1 GPT is enabled in debug mode.
1 ENMOD	GPT Enable mode.

Table continues on the next page...

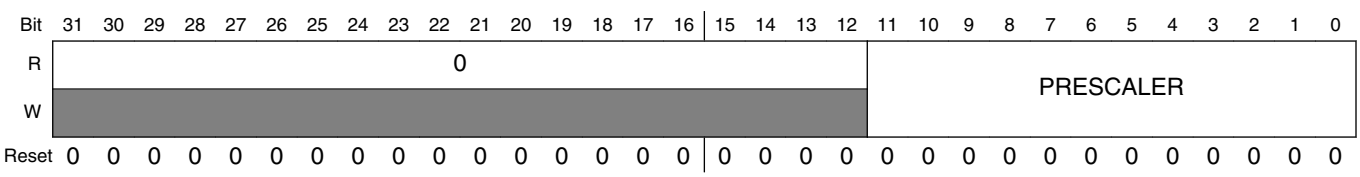
GPT_CR field descriptions (continued)

Field	Description
	When the GPT is disabled (EN=0), then both the Main Counter and Prescaler Counter *freeze their current count values*. The ENMOD bit determines the value of the GPT counter when Counter is enabled again (if the EN bit is set). • If the ENMOD bit is 1, then the Main Counter and Prescaler Counter values are reset to 0 after GPT is enabled (EN=1). • If the ENMOD bit is 0, then the Main Counter and Prescaler Counter restart counting *from their frozen values* after GPT is enabled (EN=1). • If GPT is programmed to be disabled in a low power mode (STOP/WAIT), then the Main Counter and Prescaler Counter *freeze at their current count values* when the GPT enters low power mode. • When GPT exits low power mode, the Main Counter and Prescaler Counter start counting from their frozen values, regardless of the ENMOD bit value. • Setting the SWR bit will clear the Main Counter and Prescaler Counter values, regardless of the value of EN or ENMOD bits. • A hardware reset resets the ENMOD bit. • A software reset *does not affect* the ENMOD bit. 0 GPT counter will retain its value when it is disabled. 1 GPT counter value is reset to 0 when it is disabled.
0 EN	GPT Enable. The EN bit is the GPT module enable bit. **Before setting the EN bit**, we recommend that *all registers be properly programmed*. • A hardware reset resets the EN bit. • A software reset *does not affect* the EN bit. 0 GPT is disabled. 1 GPT is enabled.

38.4.2 GPT Prescaler Register (GPT_PR)

The GPT Prescaler Register (GPT_PR) contains bits that determine the divide value of the clock that runs the counter.

Address: GPT_PR is 53FA_0000h base + 4h offset = 53FA_0004h



GPT_PR field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero. Reserved bits.

Table continues on the next page...

GPT_PR field descriptions (continued)

Field	Description
	• Writing a value to these reserved bits will not affect GPT operations. • These reserved bits are always read as zero.
11–0 PRESCALER	Prescaler bits. The clock selected by the CLKSRC field is divided by [PRESCALER + 1], and then used to run the counter. • A change in the value of the PRESCALER bits cause the Prescaler counter to reset and a new count period to start immediately. • See Figure 38-3 for the timing diagram. 0x000 Divide by 1 0x001 Divide by 2 ... ... 0xFFF Divide by 4096

38.4.3 GPT Status Register (GPT_SR)

The GPT Status Register (GPT_SR) contains bits that indicate that a counter has rolled over, and if any event has occurred on the Input Capture and Output Compare channels. The bits are cleared by writing a 1 to them.

Address: GPT_SR is 53FA_0000h base + 8h offset = 53FA_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										ROV	IF2	IF1	OF3	OF2	OF1
W											w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPT_SR field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero. Reserved bits. • Writing a value to these reserved bits will not affect GPT operations. • These reserved bits are always read as zero.
5 ROV	Rollover Flag. The ROV bit indicates that the counter has reached its *maximum possible value* and *rolled over* to 0 (from which the counter continues counting). The ROV bit is only set if the counter has reached 0xFFFFFFFF in both Restart and Free-Run modes.

Table continues on the next page...

GPT_SR field descriptions (continued)

Field	Description
	0 Rollover has not occurred. 1 Rollover has occurred.
4 IF2	IF2 Input capture 2 Flag IF1 Input capture 1 Flag The IF*n* bit indicates that a capture event has occurred on Input Capture channel *n*. 0 Capture event has not occurred. 1 Capture event has occurred.
3 IF1	See IF2
2 OF3	OF3 Output Compare 3 Flag OF2 Output Compare 2 Flag OF1 Output Compare 1 Flag The OF*n* bit indicates that a compare event has occurred on Output Compare channel *n*. 0 Compare event has not occurred. 1 Compare event has occurred.
1 OF2	See OF3
0 OF1	See OF3

38.4.4 GPT Interrupt Register (GPT_IR)

The GPT Interrupt Register (GPT_IR) contains bits that control whether interrupts are generated after rollover, input capture and output compare events.

Address: GPT_IR is 53FA_0000h base + Ch offset = 53FA_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										ROVIE	IF2IE	IF1IE	OF3IE	OF2IE	OF1IE
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPT_IR field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero. Reserved bits. • Writing a value to these reserved bits will not affect GPT operations. • These reserved bits are always read as zero.
5 ROVIE	Rollover Interrupt Enable.

Table continues on the next page...

GPT_IR field descriptions (continued)

Field	Description
	The ROVIE bit controls the Rollover interrupt. 0 Rollover interrupt is disabled. 1 Rollover interrupt enabled.
4 IF2IE	IF2IE Input capture 2 Interrupt Enable IF1IE Input capture 1 Interrupt Enable The IF*n*IE bit controls the IF*n*IE Input Capture *n* Interrupt Enable. 0 IF2IE Input Capture *n* Interrupt Enable is disabled. 1 IF2IE Input Capture *n* Interrupt Enable is enabled.
3 IF1IE	See IF2IE
2 OF3IE	OF3IE Output Compare 3 Interrupt Enable OF2IE Output Compare 2 Interrupt Enable OF1IE Output Compare 1 Interrupt Enable The OF*n*IE bit controls the Output Compare Channel *n* interrupt. 0 Output Compare Channel *n* interrupt is disabled. 1 Output Compare Channel *n* interrupt is enabled.
1 OF2IE	See OF3IE
0 OF1IE	See OF3IE

38.4.5 GPT Output Compare Register 1 (GPT_OCR1)

The GPT Compare Register 1 (GPT_OCR1) holds the value that determines when a compare event will be generated on Output Compare Channel 1. Any write access to the Compare register of Channel 1 while in Restart mode (FRR=0) will reset the GPT counter.

An IP Bus Write access to the GPT Output Compare Register1 (GPT_OCR1) occurs *after* one cycle of wait state; an IP Bus Read access occurs *immediately* (0 wait states).

Address: GPT_OCR1 is 53FA_0000h base + 10h offset = 53FA_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMP																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

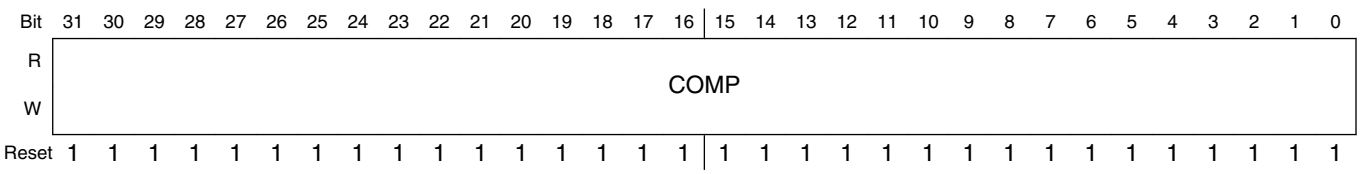
GPT_OCR1 field descriptions

Field	Description
31–0 COMP	Compare Value. When the counter value equals the COMP bit field value, a compare event is generated on Output Compare Channel 1.

38.4.6 GPT Output Compare Register 2 (GPT_OCR2)

The GPT Compare Register 2 (GPT_OCR2) holds the value that determines when a compare event will be generated on Output Compare Channel 2.

Address: GPT_OCR2 is 53FA_0000h base + 14h offset = 53FA_0014h



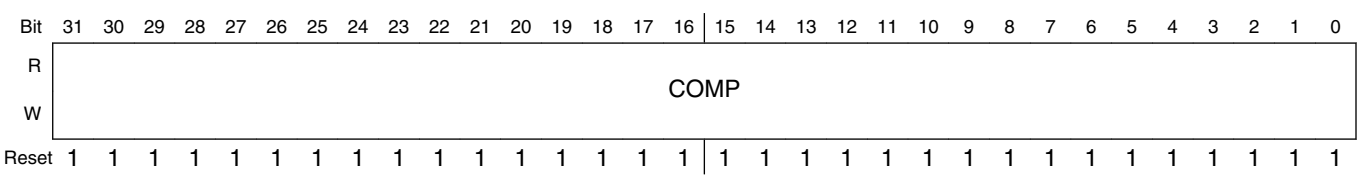
GPT_OCR2 field descriptions

Field	Description
31–0 COMP	Compare Value. When the counter value equals the COMP bit field value, a compare event is generated on Output Compare Channel 2.

38.4.7 GPT Output Compare Register 3 (GPT_OCR3)

The GPT Compare Register 3 (GPT_OCR3) holds the value that determines when a compare event will be generated on Output Compare Channel 3.

Address: GPT_OCR3 is 53FA_0000h base + 18h offset = 53FA_0018h



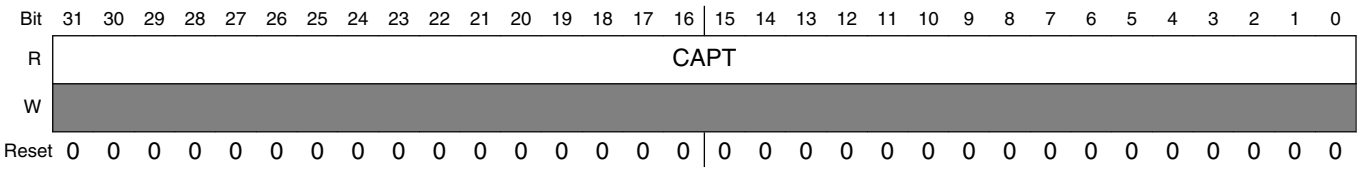
GPT_OCR3 field descriptions

Field	Description
31–0 COMP	Compare Value. When the counter value equals the COMP bit field value, a compare event is generated on Output Compare Channel 3.

38.4.8 GPT Input Capture Register 1 (GPT_ICR1)

The GPT Input Capture Register 1 (GPT_ICR1) is a read-only register that holds the value that was in the counter during the last capture event on Input Capture Channel 1.

Address: GPT_ICR1 is 53FA_0000h base + 1Ch offset = 53FA_001Ch



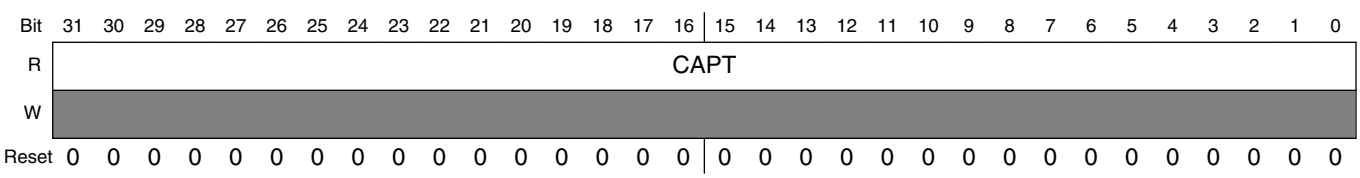
GPT_ICR1 field descriptions

Field	Description
31–0 CAPT	Capture Value. After a capture event on Input Capture Channel 1 occurs, the current value of the counter is loaded into GPT Input Capture Register 1.

38.4.9 GPT Input Capture Register 2 (GPT_ICR2)

The GPT Input capture Register 2 (GPT_ICR2) is a read-only register which holds the value that was in the counter during the last capture event on input capture channel 2.

Address: GPT_ICR2 is 53FA_0000h base + 20h offset = 53FA_0020h



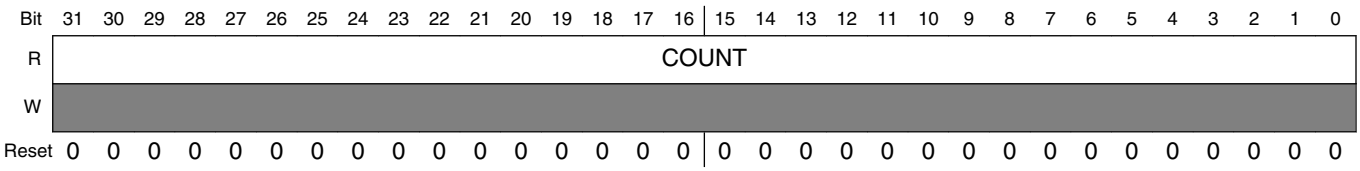
GPT_ICR2 field descriptions

Field	Description
31–0 CAPT	Capture Value. After a capture event on Input Capture Channel 2 occurs, the current value of the counter is loaded into GPT Input Capture Register 2.

38.4.10 GPT Counter Register (GPT_CNT)

The GPT Counter Register (GPT_CNT) is the main counter's register. GPT_CNT is a read-only register and can be read without affecting the counting process of the GPT.

Address: GPT_CNT is 53FA_0000h base + 24h offset = 53FA_0024h



GPT_CNT field descriptions

Field	Description
31–0 COUNT	Counter Value. The COUNT bits show the current count value of the GPT counter.

Chapter 39
2D Graphics Processing Unit (GPU2D)

39.1 Overview

This block guide describes key architectural features of the G12 OpenVG core as well as details of the G12 OpenVG IP customizing and integration in the i.MX53.

The ATI™ Z160(G12) IP is an embedded, 2D and vector graphics accelerator targeting the OpenVG 1.0.1 graphics API and feature set.

It accelerates 2D bitmap graphics operations, such as BitBlt, fill and raster operations using a separate 2D graphics acceleration unit.

Vector graphics rendering is accelerated by a separate anti-aliasing polygon rasterizer, which is connected to the 2D graphics acceleration unit.

The core has a rich, but well-chosen set of features, with emphasis being on very high image quality and low memory bandwidth consumption.

The GPU top level block diagram is presented at GPU2D Block Diagram.

39.2 GPU2D Feature List

The following chapters describe the functional features of the graphics processor.

39.2.1 Frame Buffer

- Frame buffer sizes supported up-to 2048x2048
- ARGB4444, RGB565, ARGB1555, ARGB5551, ARGB8888 frame buffer modes
- Configurable ARGB order in frame buffer: ARGB, BGRA, ABGR, RGBA
- Linear and block-based (4x4 pixels) frame buffer modes
- Fast buffer clears
- Support for OpenVG render to Image

39.2.2 2D Bitmap Graphics (Separate 2D Unit)

- Parallel operation with the 3D pipeline, independent command input
- BitBlt (surface-to-surface copy)
  - Format conversion from monochrome/ARGB/YUV to ARGB during BitBlt
- Block fill
- Internal 32-bit color precision
- Source bitmap format:
  - 1/4/8-bit monochrome
  - ARGB4444, RGB565, ARGB1555, ARGB5551, ARGB8888
  - Configurable ARGB order: ARGB, BGRA, ABGR, RGBA
  - Packed YUV 4:2:2 formats (FOURCC codes YUY2, UYVY, YVYU), two pixels per 32 bits of data.
  - 1-bit bitmap maps to foreground and background colors.
  - 4-bit bitmap is optionally gamma corrected to 8-bit alpha values and can be combined with foreground color to draw anti-aliased fonts.
- Destination bitmap format:
  - ARGB4444, RGB565, ARGB1555, ARGB5551, ARGB8888, B8, A8, AB88
  - Configurable ARGB order: ARGB, BGRA, ABGR, RGBA
- Supports three source bitmaps for separate mask/pattern/alpha bitmap support plus reading destination for ROP, blend and color key operations
- Supports masking source coordinates for wrapping patterns
- Supports ROP4 (ROP3 with separate ROPs for masked and unmasked pixels) logical operations
- Supports inverting mask and alpha values from source
- Supports destination rotation by 0/90/180/270 degrees Supports programmable blending with optional alpha un-premultiply
- Supports per pixel and constant alpha with optional modulation by source color alpha for OpenVG alpha masking Supports color keying by source and destination colors, with optional ignoring of alpha channel
- Supports one scissor rectangle for destination coordinates
- Dithering (ordered)
- Color component masking
- RGB reads and writes
- Non-power of two source and destination bitmap sizes supported (stride must be a multiple of 32-bits)
- BitBlt with scaling implemented with the 3D rendering pipeline, bilinear filtering with texture lookups, programmable filter kernels possible with the programmable Pixel processor

39.2.3 Vector Graphics

- Parallel operation with the 3D pipeline, independent command input
- Rasterization of convex and concave polygons with anti-aliasing
- Efficient native polygon rendering (no tessellation to triangles)
- Non-zero and odd-even fill rules
- Primitives supported:
  - Polygons
  - OpenVG path primitives (except Elliptical Arcs): Horizontal/vertical lines, generic lines, curves, smooth curves, moveto, path closing
  - Curve types supported: cubic and quadratic BÈzier
  - Strokes with thickness, joints and end caps, unlimited stroke thickness
  - Special case handling of singularities for thick strokes
  - Supports paths with a maximum of 256 crossings along a horizontal or a vertical line
- Input coordinates:
  - Absolute and relative coordinate input in floating point
  - Fixed-point (byte, short, int) and floating-point coordinate input - 0.8, 0.16, 16.16 formats
  - Little- and Big-endian support separately selectable for command stream and data.
- Geometry
  - User to surface transform for vertices and stroke shape
  - Hardware curve tessellation
  - Adjustable accuracy for curve and round cap splitting
  - OpenVG/SVG join types: Miter (with miter limit), round, bevel
  - OpenVG/SVG cap types: Butt, round, square
- Pixel processing:
  - Programmable gradient and texturing processor
  - Linear and radial gradients (with focal point)
  - Perspective texture mapping with filtering
  - Two textures supported
  - sRGB and pre-multiply support for textures
  - 16-sample anti-aliasing
  - 4x RGSS AA (Rotated Grid Super Sample)
  - Per-pixel alpha-masking
  - Maximum texture size: 1024x1024 pixels
- Vector graphics rendering system ARM platform load:
  - Display list generation during path creation ñ commands and vertices are stored to an internal format/buffer, no format conversion is performed

- Filling or stroking a path only requires a few register writes to start the operation in hardware
- Display lists are transferred to the vector graphics rasterizer using DMA without ARM platform interaction

39.3 GPU2D Block Diagram

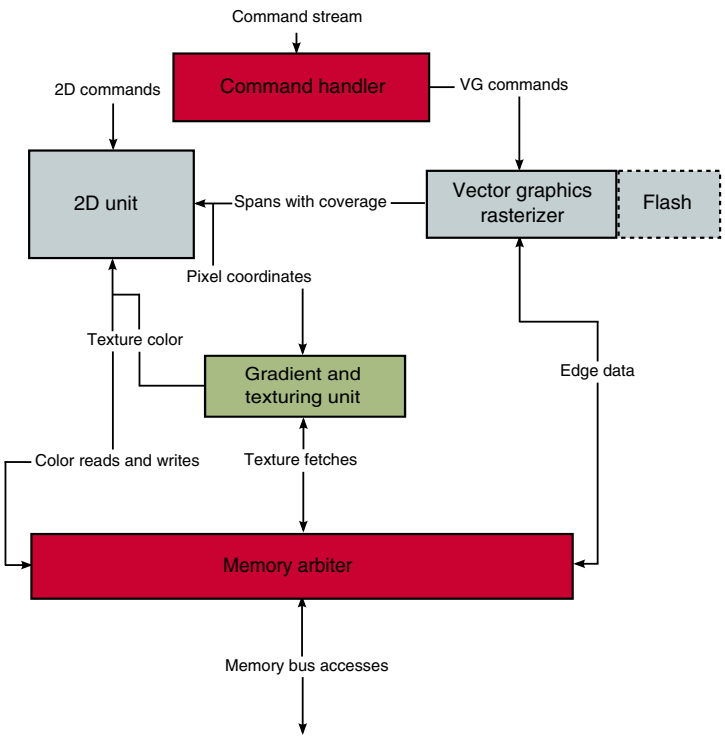


Figure 39-1. GPU2D Block Diagram

39.4 GPU SoC Interface

39.4.1 GPU2D Top Level Diagram

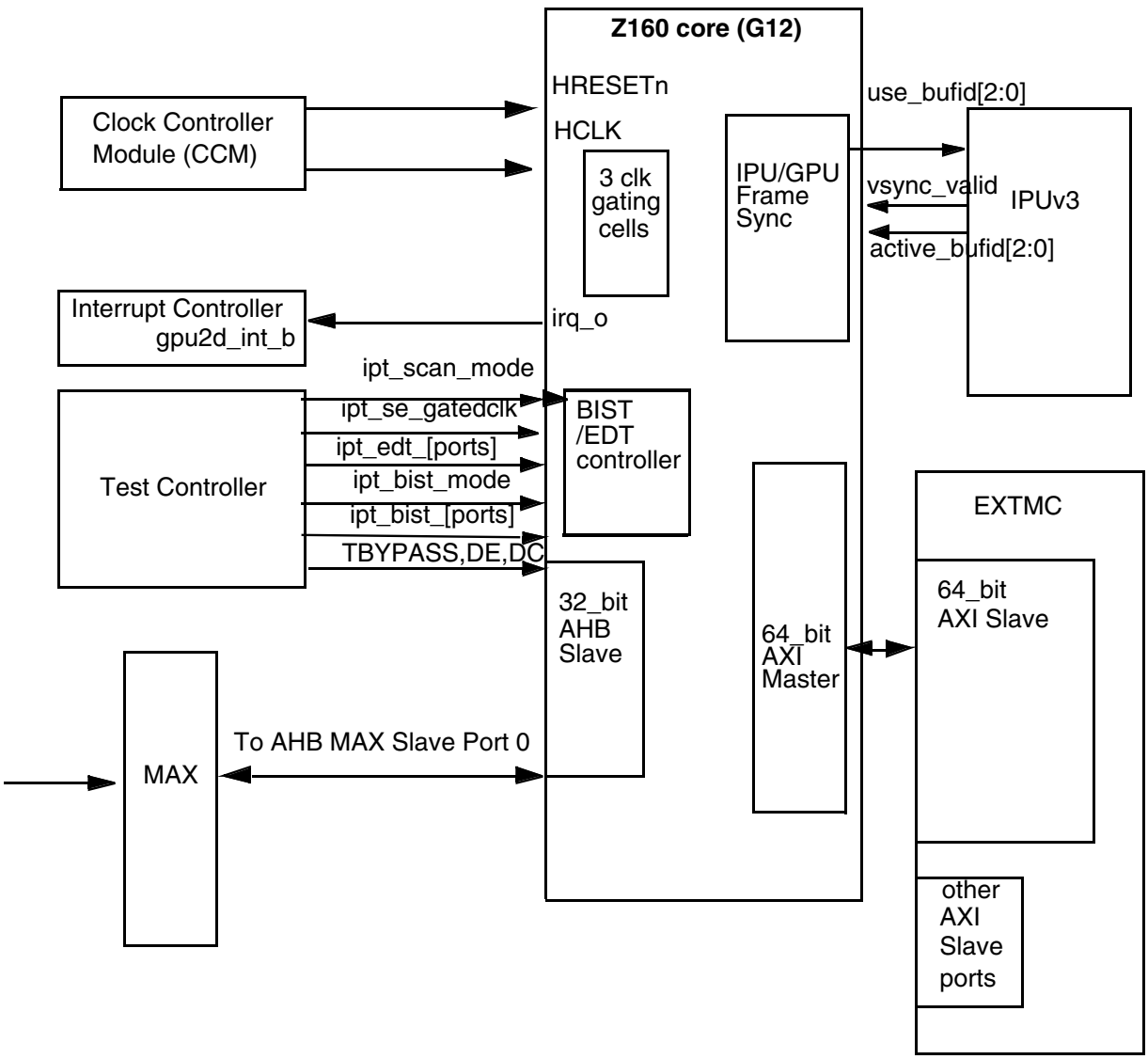


Figure 39-2. GPU2D System Connectivity

39.4.2 SoC Bus Connection

The GPU2D has the following two bus interfaces in SoC:

- 32_bit AHB slave bus.

This port is connected to ARM through MAX slave port 0, it allows access to the control registers and is the command stream input to Z160 core. The AHB interface is a fully compliant ARM AMBA AHB bus interface. It has been designed to comply with Rev 2.0 of the AMBA specification and it implements the Retry capable Slave part of the specification

- 64_bit AXI master bus

The external memory master bus is a standard 64_bit AMBA AXI master bus.

39.5 Other Signals Connection

Table 39-1. Z160(G12) miscellaneous signals

Name	Type	Source/ destination	Description
gpu2d_int_b	Output	Interrupt controller	Interrupt output from the Z160 (G12) OpenVG Graphics Core. Active LOW.
gpu2d_use_bufid[2:0]	Output	Image Processing Unit (IPU) frame sync	Tell IPU which buffer can be displayed
ipu_vsync_valid	Input	IPU frame sync	Tell GPU2D one frame is finished
ipu_active_bufid[2:0]	Input	IPU frame sync	Tell GPU2D which frame is free to be overwritten

39.6 Clocking Architecture

There is one clock input to the GPU2D, namely HCLK. In fact Z160(G12) core needs 5 clock inputs, named bus_clk, clk_0,clk_1,clk_2,clk_3, which are divided from HCLK but balanced and therefore, count as only one clock domain.

39.7 Power Management

Table 39-2. GPU2D miscellaneous signals

Clock name	Enable signal	Clocked sub-blocks	Related area
bus_clk(hclk)		Busif	4%
clk_0(hclk)	Input	Arbiter, Input	1%
clk_1	clock1_ena	Bcache	20%

Table continues on the next page...

Table 39-2. GPU2D miscellaneous signals (continued)

Clock name	Enable signal	Clocked sub-blocks	Related area
clk_2	clock2_ena	2D, V3	45%
clk_3	clock3_ena	V1, V2	30%

The different use cases and the associated clock settings are presented at the following table. For the use-cases where the core is not used at all, also bus_clk and clk_0 should be gated off to maximize the power savings. This cannot be controlled by the core itself.

Use case	bus clk	clk 0	clk 1	clk 2	clk 3
Idle	on	on	off	off	off
Bitmap case - standalone	on	on	on	on	off
Vector graphics	on	on	on	on	on

Figure 39-3. Graphics Core Operating Mode

If the whole chip go into low power mode, all clk source will be shut down by CCM.

39.8 Modes of Operation

The GPU2D supports

- 2D bitmap acceleration mode
- Vector Graphic acceleration mode
- low power mode

39.9 Reset

The GPU2D has only 1 reset port named HRESETn. The internal Z160 core (G12) contains asynchronous reset signals for each corresponding clock domain - namely rst_n for the clk clocks and bus_rst_n for the busclk clock, which are directly connected to HRESETn.

Resets are active low. The reset signals asynchronously reset all of the DFFs in the design.

39.10 Interrupts

The Z160 Core (G12) generates individual interrupts internally. Each interrupt can be enabled or disabled by changing its own enable bit. Setting the enable bit HIGH enables the corresponding interrupt.

The interrupts from all sources in the Z160 Core (G12) are combined (ORed) and output as the active LOW **gpu2d_int_b** interrupt.

39.11 DMA

GPU2D is a DMA master in fact, it read command and write data to system memory through EXTMC.

GPU2D also can be feed by DMA through MAX slave port 0.

39.12 Memory Map

The Z160 Graphics Core (G12) memory map is described in the following sections:

- AHB slave interface
- AXI master memory interface (EXTMC port).

39.12.1 AHB Slave Interface

This is based on the GPU2D occupying slave port 0 of the crossbar., whose base address is 0x20000000. GPU2D has 2 kinds of registers, one is "Interface" registers which can be read/write accessible, the others are "Internal" registers which only can be written through Z160's slave ports.

GPU2D Memory Map

Description	Memory Address Base	Memory Address End
Registers	0x2000 0000	0x200007FC

Description of all user-accessible registers in the design can be found from regs.html

Interface registers can only be accessed by status read / write channel directly through the slave port, which is mapped to 0x400 - 0x7fc address range. These registers are used to read and clear interrupts by the ARM platform, for example.

Internal 2D / VG registers are write only. They can only be accessed by two writing commands through the slave port (GPU2D's base address). The first command is a register address write, followed by a data write command.

In practice, this means that the command stream, which is prepared by the software driver is written through the 0x000-0x3ff address range either directly, or through the DMA operation by the core. These registers are not accessible directly by slave ports channel, like interface registers are not accessible through this channel.

39.12.2 AXI Master Memory Interface (EXTMC Port)

The Z160 Graphics Core (G12 or GPU2D) can access memory with or without a *Memory Management Unit* (MMU):

Translation is performed using a table with 8KB entries, one for each 4KB page in a 32MB linear address space.

Chapter 40
3D Graphics Processing Unit (GPU3D)

40.1 Overview

The 3D Graphics Processing Unit (GPU3D) is an embedded engine capable of DirectX9 Shader Model 3.0+ program execution. The GPU3D is focused on accelerating user level graphics APIs such as OpenGL ES 2.0 & 1.1, OpenVG 1.0, and Direct3D Mobile 1.2.

40.2 GPU3D Features Overview

The GPU3D IP core has the following high-level features:

- Built to accelerate OpenGL ES 2.0 and Direct3D Mobile 1.2
- Unified Shader Architecture
  - Uses dynamically shared shader ALU/memory resources between vertex and pixel shaders.
- General purpose exports to system memory from Vertex & Pixel Shaders
  - Can be used in vertex shader for tessellation of HOS, sub-division surfaces etc.
  - Can be used in pixel shader for generating vertices etc.
- Supports 2- and 4- sample MSAA
- Command Processor:
  - As a DMA master, support advanced packet based command stream manager allowing for efficient and flexible transfer of graphics commands and host data from the host system to the graphics processor core
- Graphics Memory Controller and Graphics Memory (GMEM)
  - Customer configurable on-chip memory used to accelerate 3D rendering using a binning architecture significantly reducing external system bandwidth requirements. i.MX53 adopts a 256KB internal GMEM.
- Integrated Power Management

Block-level clock gating managed automatically in the IP.

40.3 Capabilities and Performance

- 32 bit FP Internal Shader Precision
- Compressed Texture Support: ETC, DXTC, and ATI_TC
- Supports general purpose exports to system memory from Vertex and Pixel shaders.
- Uses indexed vertex data fetches from the Vertex Shader as a high-bandwidth vertex data path.
- Latency hiding through FIFOs and multi-threading
- Sophisticated Shader support
  - 512 4-Component constants
  - 1024 Shader Instructions
  - 16 Textures
  - 16 4-Component Interpolants
  - 64 General Purpose Registers
- Rich texture format and types
  - Compressed
  - 16 bit Floating point
  - Cube map
  - Volume textures
  - Non power-of-2
  - Anisotropic
- 1 Primitive (Triangle/Line/Point) every 6 clocks
- 2 Vector (4 component) and 1 Scalar (single component) ALU instruction per clock
- 1 Control Flow Instructions (Jumps, loops) per clock
- 1 Export per clock
- 14 component Pixel Shader input interpolant per clock
- Early-Z testing at up to 4 pixels per clock
- 1 pixel (at 4 sample MSAA) per clock with alpha blending and depth test

40.4 GPU3D Block Diagram

The GPU3D contains three primary blocks:

- Graphics Core (GC)
  - Programmable graphics engine that performs all of the data processing and memory transactions
- Graphics Arbiter (GARB)
  - Controls the GMEM and arbitrates between requests from the GC and the system
- Graphics Memory (GMEM)

- Local SRAM buffers (4x64KB buffers) that are used for the tile / bin of pixels when the GC is working on

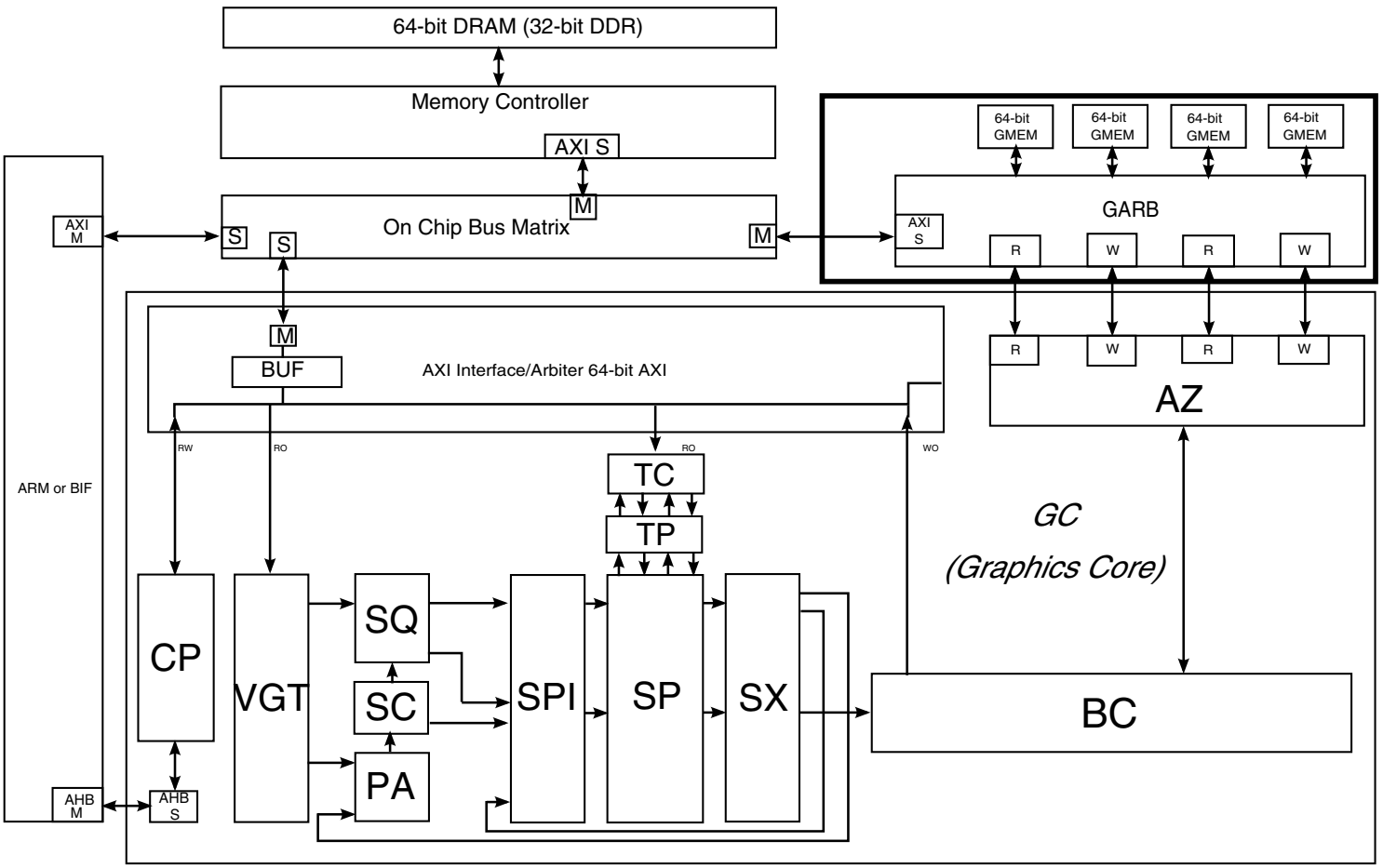


Figure 40-1. GPU3D Top Level Block Diagram

40.5 GPU3D Interface

40.5.1 GPU3D Top Level Diagram

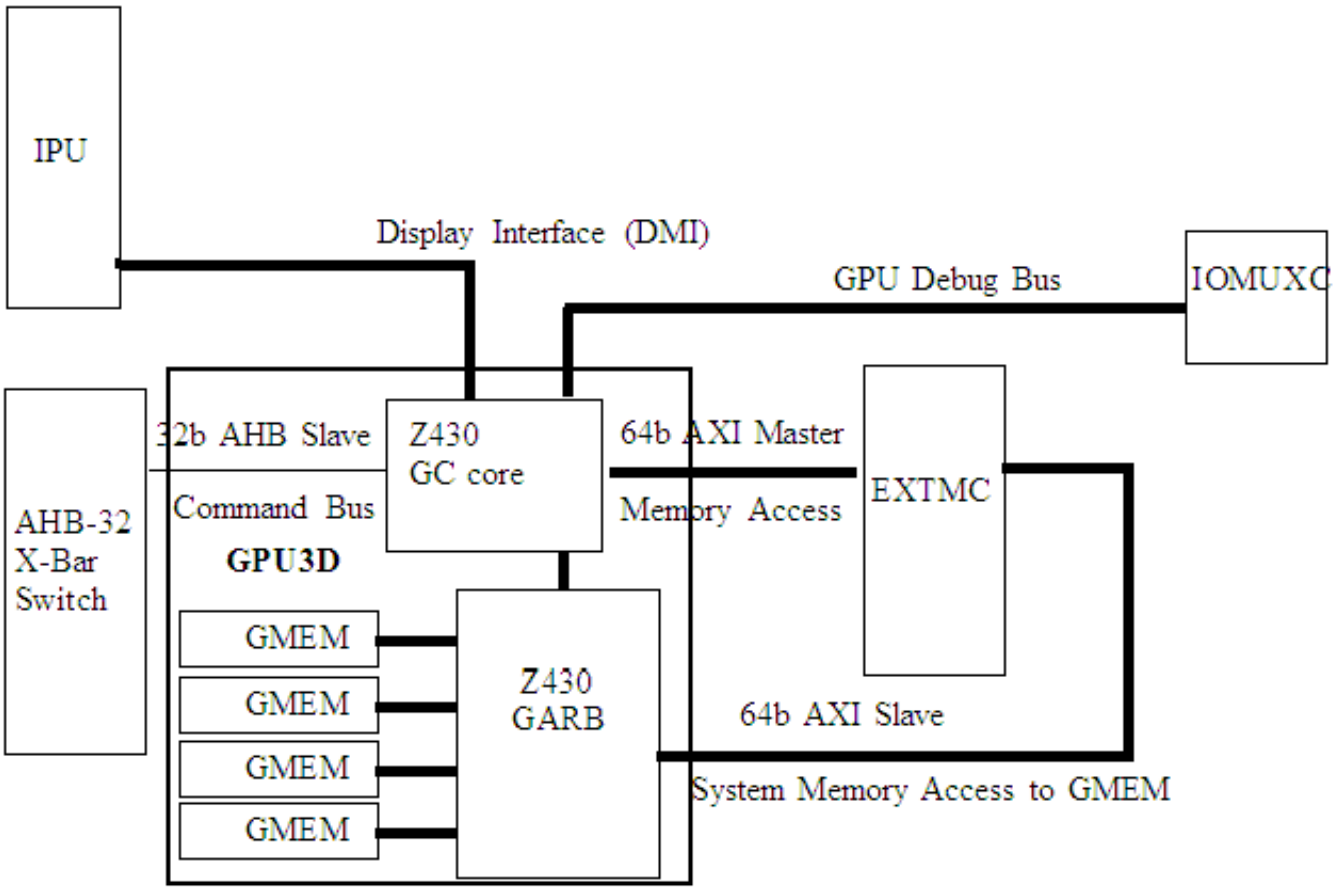


Figure 40-2. GPU3D System Connectivity

40.5.2 SoC Interface Summary

- Slave Interface to ARM platform or controller interface
  - Peripheral AMBA 2.0 AHB-Lite Host Interface
  - Provides host access to graphics core from the system processor.
  - 32b wide, synchronous with core.
  - Support for clock gating for phase removal on AHB clock.
- AXI Master Interface into memory system (AXI V1.0)
  - 64b wide, synchronous
  - Provides access to system memory from the graphics core.
  - Programmable number of read and write requests to the memory system
  - Number of outstanding request limited by system resources.
- AXI slave interface for system access to graphics memory (though GARB to GMEM)

    - 64b wide, synchronous interface
    - 1 outstanding read and write request
- Display Module Interface (DMI)
    - For synchronization with the display processor. (IPU)
    - Provides indication to display processor from graphics core that a newly rendered frame is ready and that the display must perform a buffer flip at the next vertical blank.
    - Provides indication to graphics core from the display processor that the buffer identified by buffer ID has been displayed at least once..
- Control Interface Block (CIB on AHB slave bus)
    - Holding block for clock gating, reset etc.
    - Interrupt Interface (INT)
    - Error and status interrupts to the controller.
- Design For Test Interface (DFT)
    - Memory test collar access.
    - Scan/BIST access.
- Debug bus
    - GPIO register as control functions
    - Set of signals OR chained between internal masters for debug purposes
    - 32b bus muxed to 16b output to pad

40.5.3 Memory Interface Detail

40.5.3.1 Access Type

The GPU3D core generates 128 and 256 bit bursts to external memory from its internal masters, which are arbitrated by a Memory Hub (MH). The MH supports up to 64 outstanding transactions, 8 AXI IDs, and contains a MMU.

Typical graphics use cases (usually games) will utilize approximately 400 to 500 MB/s and require a memory latency of less that 90ns (70ns is preferred).

The Memory Hub selects between the following clients in the graphics core:

- CP - (Command Processor) Read/Write
- VGT - (Vertex Grouper and Tesselator) Read Only
- TC - (Texture Cache) Read only
- RB - (Render Block) Write Only

Sub-Client	Sub-Client Designation	AXI ID	Operation	Transaction Size (Bytes)	Sub-Client Outstanding Transaction Limit
Ring Buffer	CPr0	0	Read	32	Programmable (Note 1)
Indirect Buffer #1	CPr1	1	Read	32	Programmable (Note 1)
Indirect Buffer #2	CPr2	2	Read	32	Programmable (Note 1)
State Sub-Block, Constant, & Shader Instruction Data	CPr3	3	Read	32	Programmable (Note 1)
Micro-Engine & Write Pointer Polling	CPr4	4	Read	32	1 Micro-engine read & 1 write pointer read
VGT Indices	VGTr0	5	Read	32	5 (Note 2)
VGT Bin ID	VGTr1	6	Read	32	8
Texture/Vertex Read	TCr	7	Read	16/32	9
Synchronization Semaphores, Micro-Engine Semaphores, Constant State Data	CPw	Programmable	Write	16/32	8 Unconfirmed writes, unlimited writes past point of confirmation
RB Copy	RBw	Programmable	Write	32	No client limit
PA	PAw	Programmable	Write	32	1
MMU TLB Miss	MMUr	Programmable	Read	32	1

All transactions are incrementing address bursts. The starting address is always aligned to the burst size, that is, all 16 Byte bursts start at a 16-Byte aligned address; all 32 Byte bursts start at a 32-Byte aligned address.

Command stream and vertex data is read in bursts of 256 bits.

The Burst cache combines color accesses from several smaller objects to 4x4 pixel (16-bit), or 4x2 pixel (32-bit), which are transferred to memory in bursts of 256 bits.

Color writes use byte enables when incomplete pixel blocks are written to the memory

40.5.3.2 Memory Management

The GPU3D GC core include a Memory Management Unit (MMU) capable of remapping a programmably sized region of the 32 bit GC memory space. This unit will use a single level mapping table, with each 32 bit table entry mapping a 4K byte region. Hence to map a 1 M byte region will require 256 table entries or a table 1K in size.

To make the remapping more efficient the GC will include a Translation Lookaside Buffer (TLB) which acts as a 'cache' of translations. This cache will hold a total of 128 translations organized as 16 'lines' of 8 translations each. Each of the 16 'lines' is fully associative, allowing translations for sections of up to 16 different surfaces to be simultaneously stored in the TLB.

Each client within the GC can be selected whether or not to use the MMU remapping function.

Any accesses that do not use the MMU may be limited to a programmable range of physical address space, with accesses outside of thatregion resulting in a page fault fatal error.

When the driver requests memory from the OS, the virtual to physical translation is stored in the GPU3D's Page Table. The page table is a contiguous physical chunk of memory whose starting address is defined as the page table base (PTB) and whose extent depends on the amount of virtual memory required by the driver. For example, if supporting 4KB pages and each page table entry is 32b, a virtual address range of 64MB available to the driver would require 16K entries or 64KB of page table storage.

Note that the page table used by the graphics is not the same as the page table used by the ARM processor. Specifically, the GPU page table is a single level page table, and not all access control mechanisms are implemented.

In its simplest form, using a 4KB page, bits [11:0] of the virtual address can be the same as the physical address[11:0] and can be used as an index into the page. Bits [31:12] of the virtual address, in conjunction with the Virtual Address Base register, can be used as an index into the page table to find the address of the relevant page table entry (PTE).

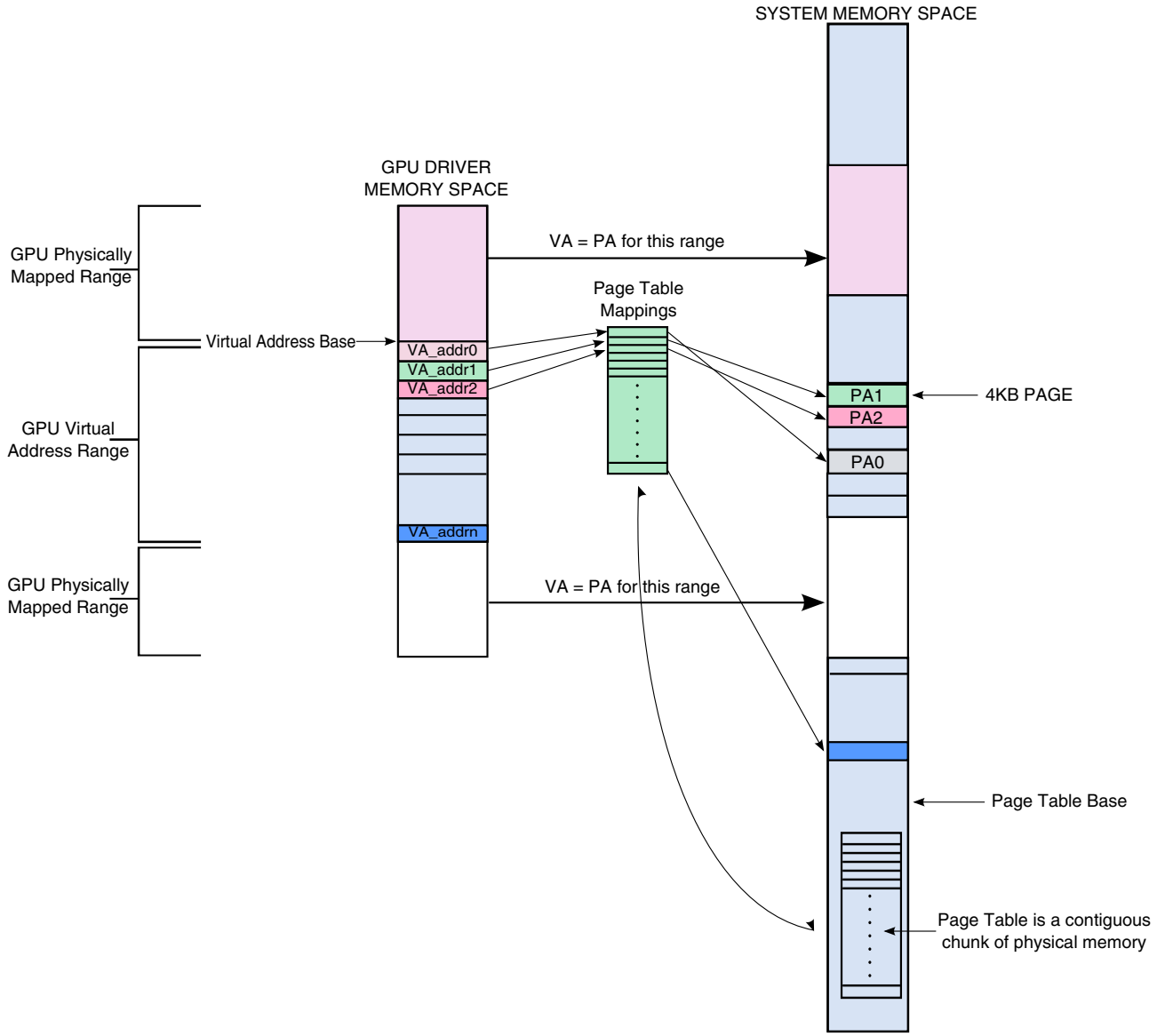


Figure 40-3. GPU3D Memory Mapping Concept

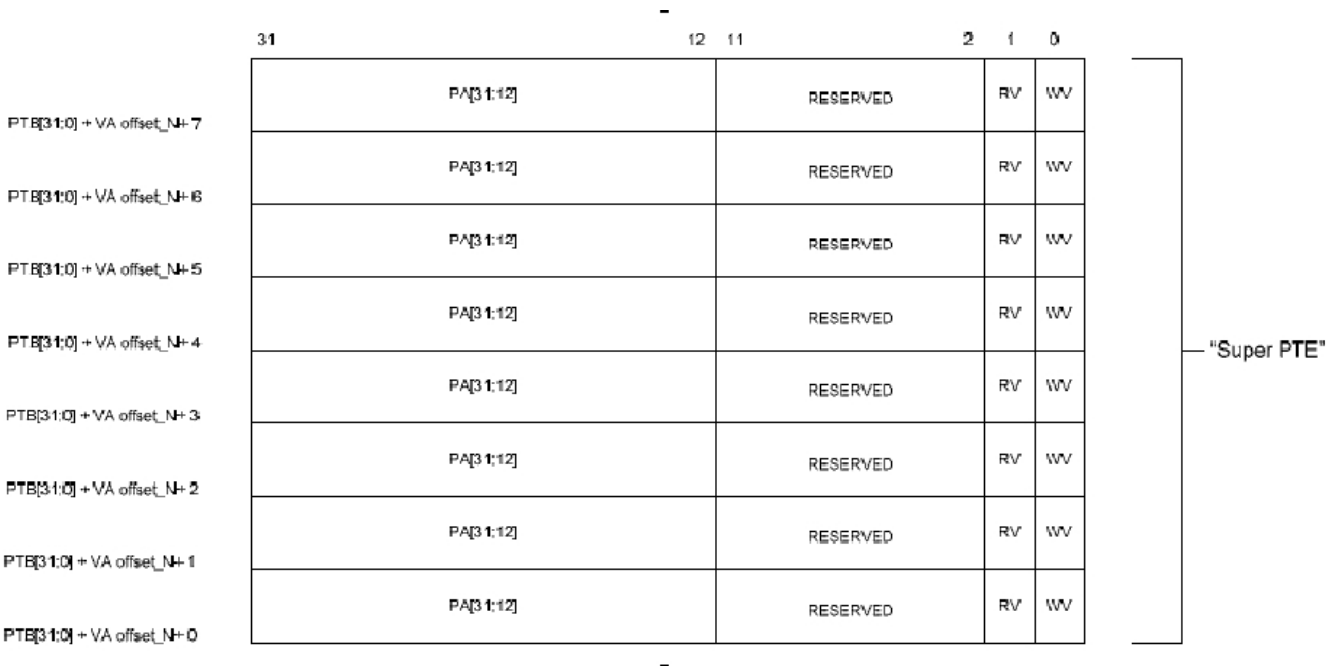


Figure 40-4. Page Table Entry Format in External Memory

40.5.4 DMI Interface Detail

GPU3D DMI ports

Name	Type	Source	Description
gpu_use_bufid[2:0]	Output	GPU3D	Tell IPU which buffer can be displayed, one-hot coded
ipu_vsync_valid	Input	IPU	Tell GPU3D one frame is finished, high in vertical blank period
ipu_active_bufid[2:0]	Input	IPU	Tell GPU3D which frame is free to be overwritten, one-hot coded

The vertical sync (VSYNC) from the display controller occurs after each frame is displayed, during the vertical blanking period.

One-hot encoded USE_BUFID signals are updated by GPU3D whenever there is a new rendered buffer available (and the previous value of USE_BUFID has already been sampled by the IPU / display controller), so it can change anywhere in the frame. It is re-synchronized and sampled a few cycles before the start of the vertical blanking period (rising edge of VSYNC) in display controller, while the currently finished buffer (ACTIVE_BUFID signals) is updated by display controller at the rising edge of VSYNC, ACTIVE_BUFID are then re-synchronized and sampled by GPU3D after synchronizing to VSYNC rising edge.

40.5.5 Debug Bus and GPIO

40.5.5.1 Debug Bus

The GPU3D has a 32b debug bus output that is controlled by a 18b 'GPIO' bus. The 32b debug bus should be mapped to system GPIO. In i.MX53, this 32b bus is MUXed to 16b to reduce the number of external pins required. The MUXing of these pins is controlled by 2 bits in the GPIO register.

40.5.5.2 GPIO Register

The GPU3D supports 16 general purpose input pins that can be configured as soft reset, debug mux select bits etc. The bus is implemented as a system register (whose offset is 0x2FFFC from GPU3D base address) along with 2_bit debug bus control information. The registered inputs are semi-static and have no timing constraints associated with them. The following table shows GPIO register implementation.

GPIO register-R/W 18 bit- (0x2FFFC)

field name	bits	default	Description
debug bus control	17:16	0x0	00 - output lower 16 bits 01 - output upper 16 bits 10 - toggle between outputing lower and upper 16 bits per 83MHz clock 11 - unused
GPIO bus	15:0	0x0000	debug bus control

Table 40-1. GPIO bus functions

GPIO BIT	Debug Function	Reset/Misc Function
15	1'b0	Ignore SQ RTR for AHB
14	1'b0	Ignore VGT RTR for AHB
13	1'b0	Ignore CP RTR for AHB
12	1'b0	Ignore RTR for AHB
11	Sub-Block Select[3]	Ignore RTR
10	Sub-Block Select[2]	VGT Soft Reset
9	Sub-Block Select[1]	SC Soft Reset
8	Sub-Block Select[0]	CIB Soft Reset
7	1'b0	1'b0
6	1'b0	1'b1
5	Sub-Block Addr[5]	SX Soft Reset

Table continues on the next page...

Table 40-1. GPIO bus functions (continued)

GPIO BIT	Debug Function	Reset/Misc Function
4	Sub-Block Addr[4]	SQ Soft Reset
3	Sub-Block Addr[3]	RB Soft Reset
2	Sub-Block Addr[2]	MH Soft Reset
1	Sub-Block Addr[1]	PA Soft Reset
0	Sub-Block Addr[0]	CP Soft Reset

The functionality in the Reset/Misc Function is the same as the functionality in the RBBM_SOFT_RESET register and the RBBM_DEBUG register. The bits are logically OR'd together.

There is latency built into the debug bus due to the registered daisy chain nature of the debug bus. From the time GPIO gets registered (count this as clock 1), it can take up to 10 clock cycles before valid Debug Bus data is resident in the Debug Bus register. This latency applies any time the state of GPIO is changed.

40.6 Clocking Architecture

40.6.1 Clock Input

The GPU3D design has 2 clocks at top level: aclk_gpu for the Graphics Core (GC) and aclk_garb for the Graphice Memory Arbiter (GARB).These 2 clocks are synchronous, they're actually in a single domain. There is no multicycle paths, latches or negative edge flops are used in the design. (The CKGATER uses a negative latch, but that is a special cell).

Both clocks (aclk_gpu and aclk_garb) are synchronous and in phase with core clock. Actually they come from one source in CCM but has separate gating cell so that clock (aclk_gpu) to the Graphics core (GC) can be removed independently of the GARB (aclk_garb), which make GMEM still accessible when Graphics Core (GC) is shut down.

Both clocks (aclk_gpu and aclk_garb) should be implemented with programmable dividers to run at full speed, synchronous bus speed, and half bus speed to meet system power use cases.

40.6.2 Clock Gating

GPU3D graphics core (GC) supports three levels of clock gating, which is done by internal block CIB (Control Interface Block):

- Explicit removal of clocks to the GC core by the system under software control
- Designer controlled removal of clocks to logic blocks (controlled by internal CIB block)
- Instantiated clock gating by synthesis tool.

Explicit clock gating is invoked under system control and as such is transparent to the GPU3D. All clocks (aclk_gpu and aclk_garb) are removed. Prior to clock removal, the system should ensure that the GPU3D is idle (no outstanding bus traffic, no outstanding command activity in command buffer). Removing the clocks to the GPU3D does not impact internal state (unless power is also removed).

Designers can chose to remove clocks to blocks or sections of blocks based on those functions becoming idle. Power Management Override bits are supported which selectively disable designer instantiated clock gating. The GC powers up with the clock gating disabled with the exception of the AHB busclock and the GMEM clocks which power up with clock gating enabled. It is the responsibility of the driver to enable the clock gating functionality by disabling (clearing) the power management override bits. The GARB instantiate default enabled clock gaters for GMEM clock for power saving, which can be overridden (disabled) by pm_override[3:0] from the GC to the GARB.

In particular, the output 'idle' signal (gpu_idle, active high) signal from the CIB should be used to clock gate the core clock signals (aclk_garb) to reduce power consumption.

40.7 Reset

GPU3D has only an asynchronous reset port named hresetn at top level, which is active low and will reset GC, GARB & GMEM.

Please note that **the internal Graphics Core (GC) uses synchronous reset which is propagated through the design as a standard synchronous signal. So clocks (ack_gc and aclk_garb) must be turned on when hard reset (hresetn) asserted.** The supplied reset (hresetn) is passed through a reset conditioning circuit in the CIB. This conditioning circuit extends reset assertion long enough to guarantee that all internal circuits are reset correctly. This circuit counts 20 bus cycles after the de-assertion of reset so GPU3D need 20 bus cycles to complete a reset.

In addition to the top level hard reset (hresetn) there are soft resets for each internal blocks that can be controlled by Software.

40.8 Interrupts

The graphics core provides a single interrupt pin to the system (gpu_int_b). Once asserted the interrupt pin remains asserted (active low) until the ARM platform clears the IRQ bit in the RBBM_IRQ register.

In general interrupts are used to signal error conditions or to support frame buffer swap.

40.9 Memory Map

The GPU3D has 2 memory map spaces in i.MX53:

- AHB slave interface (GC space)
- AXI slave memory interface (GMEM space).

Memory Map

Description	Memory Address Base	Memory Address End
GC space (register, instruction)	0x3000 0000	0x3002 FFFF
GMEM space (256KB)	0xF802 0000	0xF805 FFFF

Chapter 41
I2C Controller (I2C)

41.1 Overview

This chapter describes block-level operation and programming of I2C. The chapter is intended for a block driver software developer. To understand how the block is integrated at the SoC level, a system software developer can refer to discussions of the block in the appropriate SoC-level chapter(s).

References: This document assumes an understanding of the following reference:

1. The I2C Bus Specification, Version 2.1

The Inter IC (I2C) provides functionality of a standard I2C slave and master. The I2C is designed to be compatible with the standard Philips I2C bus protocol.

I2C is a two-wire, bidirectional serial bus that provides a simple, efficient method of data exchange, minimizing the interconnection between devices. This bus is suitable for applications requiring occasional communications over a short distance between many devices. The flexible I2C standard allows additional devices to be connected to the bus for expansion and system development. See the connection diagram in the figure below.

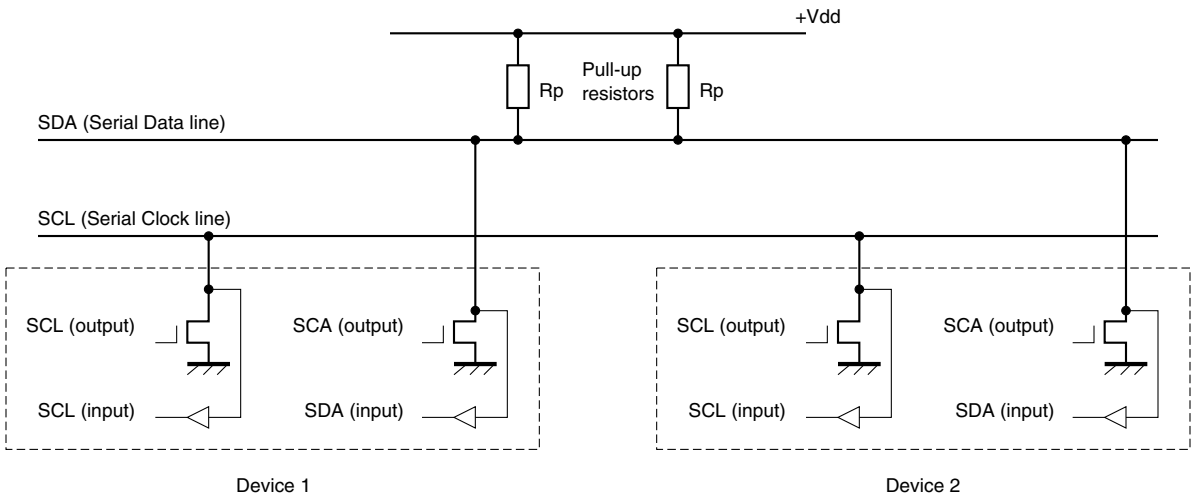


Figure 41-1. Connection of Devices to I2C Bus

The I2C interface operates up to 400 kbps, but it depends on the pin loading and timing characteristics. For pin requirement details, refer to Philips I2C Bus Specification, Version 2.1. The I2C system is a true multiple-master bus including arbitration and collision detection that prevents data corruption if multiple devices attempt to control the bus simultaneously. This feature supports complex applications with multiprocessor control and can be used for rapid testing and alignment of end products through external connections to an assembly-line computer. The figure below shows the block diagram of I2C.

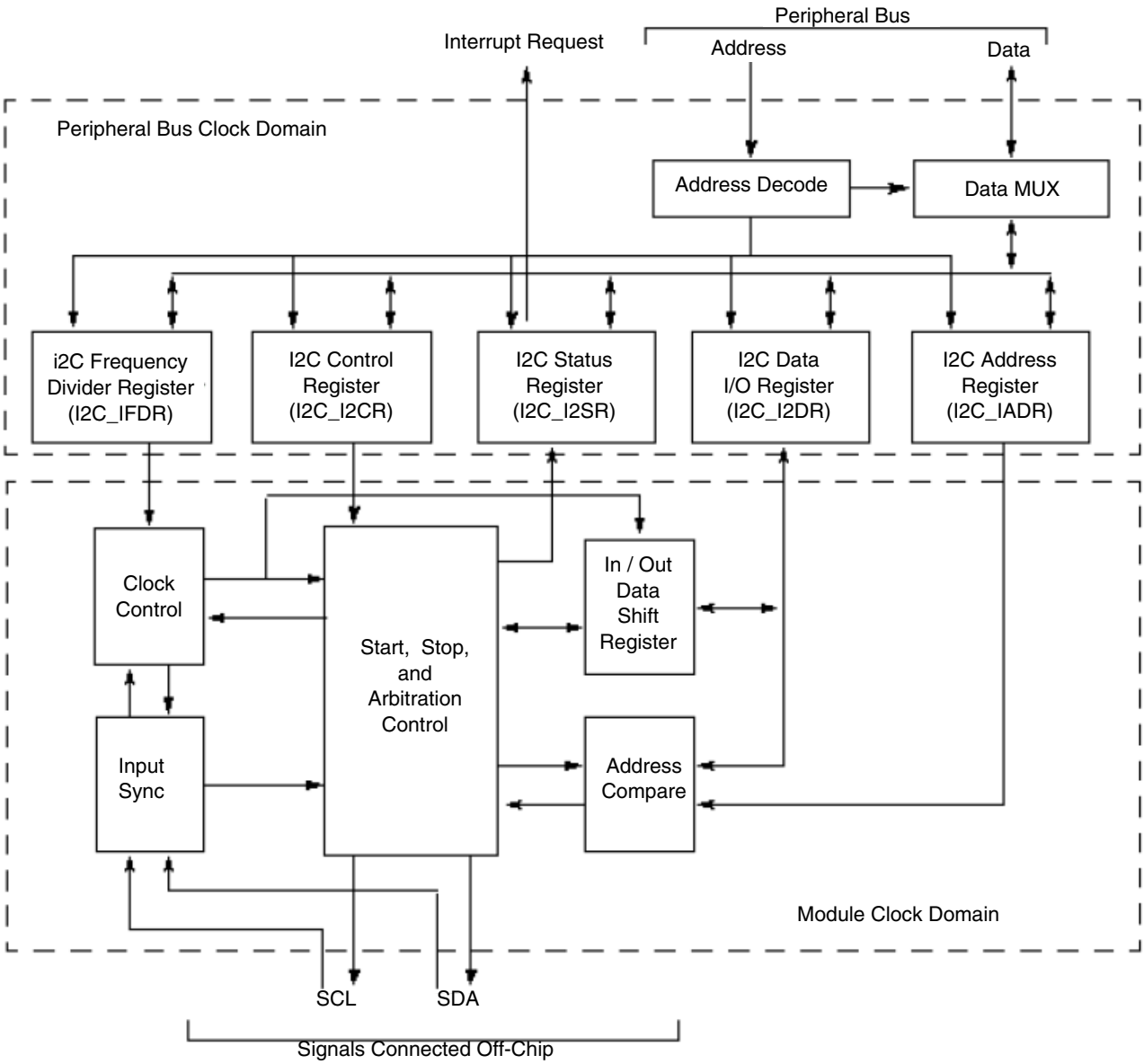


Figure 41-2. I2C Block Diagram

41.1.1 Features

The I2C has the following key features:

- Compatibility with I2C bus standard
- Multiple-master operation
- Software-programmable for one of 64 different serial clock frequencies
- Software-selectable acknowledge bit
- Interrupt-driven, byte-by-byte data transfer
- Arbitration-lost interrupt with automatic mode switching from master to slave

- Calling address identification interrupt
- Start and stop signal generation/detection
- Repeated START signal generation
- Acknowledge bit generation/detection
- Bus-busy detection

41.1.2 Modes and Operations

The I2C primarily operates in two different functional modes.

41.1.2.1 Standard Mode

In Standard mode, I2C supports the data transfer rates up to 100 Kbits/s.

41.1.2.2 Fast Mode

In Fast Mode, data transfer rates up to 400 Kbits/s can be achieved.

As per block operation, there is no special configuration required for Fast and Standard mode. It is the data transfer rate which distinguishes Standard and Fast mode.

41.2 External Signals

The table below describes all I2C signals that connect off-chip.

For I2C compliance, all devices connected to the SCL and SDA signals must have open-drain or open-collector outputs. The logic AND function is exercised on both lines with external pull-up resistors.

Input of SCL and SDA also need to be manually opened by set SION bit in IOMUX after corresponding PADs were selected as I2C function.

Table 41-1. Off-Chip Block Signals

Signal	I/O	Description	Reset State[1]	Pull-Up/Down[1]
SCL	I/O	Serial Clock	1	Active
SDA	I/O	Serial Data	1	Active

1. The reset state values and pull-up/down requirements provided in this table are from the block-level perspective. To understand how the block is integrated at the SoC level, the system software developer must see discussions of the block in the appropriate SoC-level chapter(s). For example, a block signal that requires a pull-up coul

up option built into the SoC. In this case, the system software developer must ensure the proper programming at the SoC level.

41.3 Functional Description

41.3.1 I2C System Configuration

Out of a reset, the I2C defaults to slave receive operations. Thus, when not operating as a master or responding to a slave transmit address, the I2C will default to the slave receiver state.

For exceptions, see Initialization Sequence.

NOTE

The I2C is designed to be compatible with the Philips™ I2C bus protocol. For information on system configuration, protocol, and restrictions, refer to the I2C Bus Specification, Version 2.1. The I2C supports Standard and Fast modes only.

41.3.2 I2C Protocol

The I2C communication protocol consists of six components, as follows:

- START
- Data Source/Recipient
- Data Direction
- Slave Acknowledge
- Data Acknowledge
- STOP

See the figure below for the I2C standard communication protocol, as defined in the following sections.

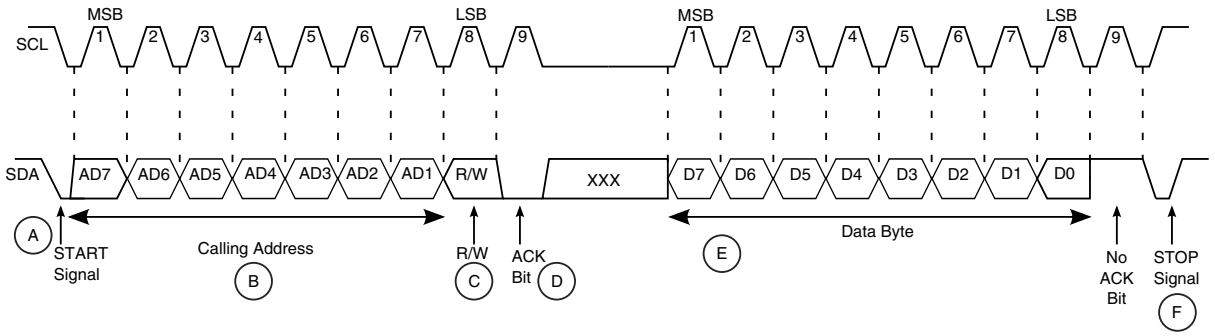


Figure 41-3. I2C Standard Communication Protocol

41.3.2.1 START Signal

When no other device is a bus master (both SCL and SDA lines are at logic high), a device can initiate communication by sending a START signal (see A in Figure 41-3).

A START signal is defined as a high-to-low transition of SDA while SCL is high. This signal denotes the beginning of a data transfer (each data transfer can be several bytes long) and awakens all slaves.

41.3.2.2 Slave Address Transmission

The master sends the slave address in the first byte after the START signal (B). After the seven-bit calling address, it sends the R/W bit (C), which tells the slave data transfer direction.

Each slave must have a unique address. An I2C master must not transmit an address that is the same as its slave address; it cannot be master and slave at the same time.

The slave whose address matches that sent by the master pulls SDA low at the ninth clock (D) to return an acknowledge bit.

41.3.2.3 Data Transfer

When successful slave addressing is achieved, the data transfer can proceed (E) on a byte-by-byte basis in the direction specified by the R/W bit sent by the calling master.

Data can be changed only while SCL is low and must be held stable while SCL is high, as shown in Figure 41-3. SCL is pulsed once for each data bit, most-significant bit first. The receiving device must acknowledge each byte by pulling SDA low at the ninth clock; therefore, a data byte transfer takes nine clock pulses.

If it does not acknowledge the master, the slave receiver must leave SDA high. The master can then generate a STOP signal to abort the data transfer or generate a START signal (a repeated start, as shown in Figure 41-4) to start a new calling sequence.

If the master receiver does not acknowledge the slave transmitter after a byte transmission, it means end-of-data to the slave. The slave releases SDA for the master to generate a STOP or START signal.

NOTE

Writing to the data register triggers transmit operation.

Transmit data should always be written after MTX bit is programmed. Transmit data is not latched inside until the transfer is initiated on the interface bus.

After the transmit data write in I2C, software can either wait for a transfer-done interrupt or it can poll for ICF bit for zero, if new data has to be written during the previous data transfer. The IIF bit may not be polled if the IIEN bit is set because the I2C will generate an interrupt when IIF is set.

41.3.2.4 STOP Signal

The master can terminate communication by generating a STOP signal to free the bus. A STOP signal is defined as a low-to-high transition of SDA while SCL is at logical high (F).

NOTE

A master can generate a STOP even if the slave has made an acknowledgment, at which point the slave must release the bus.

41.3.2.5 Repeat Start

Instead of signalling a STOP, the master can repeat the START signal, followed by a calling command

(see A in the figure below). A repeated START occurs when a START signal is generated without first generating a STOP signal to end the communication. The master uses a repeated START to communicate with another slave or with the same slave in a different mode (transmit/receive mode) without releasing the bus.

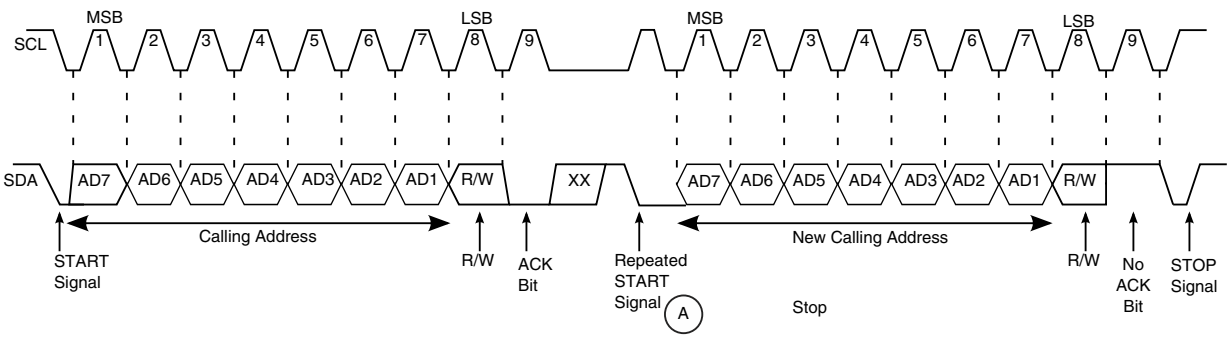


Figure 41-4. Repeated START

41.3.3 Arbitration Procedure

If multiple devices simultaneously request the bus, the bus clock is determined by a synchronization procedure in which the low period equals the longest clock-low period among the devices, and the high period equals the shortest. A data arbitration procedure determines the relative priority of competing devices.

A device loses arbitration if it sends logic high while another sends logic low; it immediately switches to slave-receive mode and stops driving SDA. In this case, the transition from master to slave mode does not generate a STOP condition. Meanwhile, hardware sets the arbitration lost bit in the I2C Status register (I2C_I2SR[IAL] to indicate loss of arbitration).

41.3.4 Clock Synchronization

Because wire-AND logic is used, a high-to-low transition on SCL affects devices connected to the bus. Devices start counting their low period when the master drives SCL low. When a device clock goes low, it holds SCL low until the clock high state is reached. However, the low-to-high change in this device clock may not change the state of SCL if another device clock is still in its low period. Therefore, the device with the longest low period holds the synchronized clock SCL low.

Devices with shorter low periods enter a high wait state during this time (see the figure below). When all devices involved have counted off their low period, the synchronized clock SCL is released and pulled high. There is then no difference between device clocks and the state of SCL, so all of the devices start counting their high periods. The first device to complete its high period pulls SCL low again.

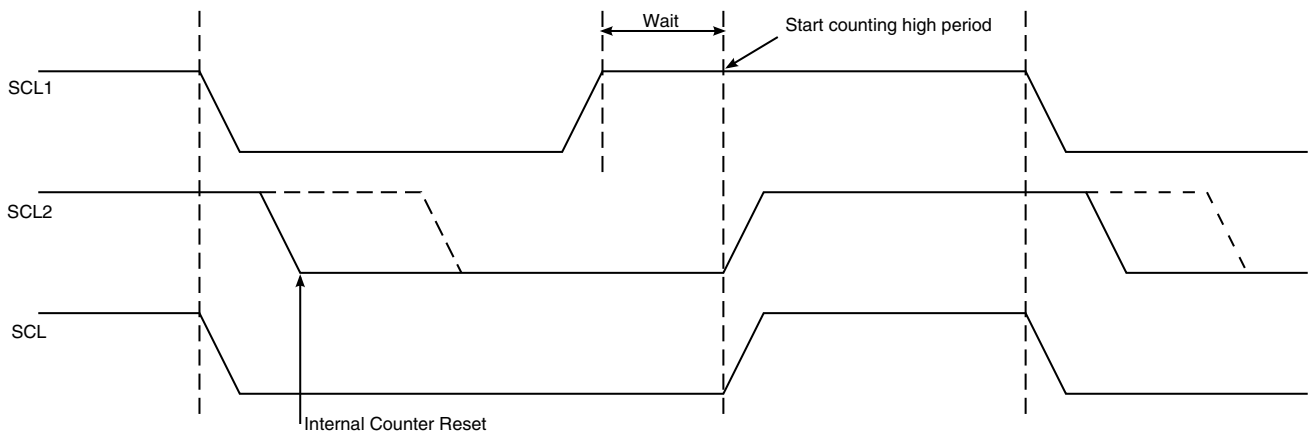


Figure 41-5. Synchronized Clock SCL

41.3.5 Handshaking

The clock synchronization mechanism can be used as a handshake in data transfers. Slave devices can hold SCL low after completing one byte transfer (9 bits). In such a case, the clock mechanism halts the bus clock and forces the master clock into a wait state until the slave releases SCL.

41.3.6 Clock Stretching

Slaves can use the clock synchronization mechanism to slow down the transfer bit rate. After the master has driven SCL low, the slave can drive SCL low for the required period and then release it. If the slave SCL low period is longer than the master SCL low period, the resulting SCL bus signal low period is stretched.

41.3.7 Peripheral Bus Accesses

I2C is a 16-bit block. Only half-word accesses should be performed to the block.

41.3.8 Generation of Transfer Error on IP Bus

If an address is received on the Peripheral slave bus interface that is not implemented, an access error is generated.

41.3.9 Clocks

There are two input clocks for I2C.

1. Peripheral Clock:The clock is used for Peripheral bus register read/writes.
2. Module Clock:This is the functional clock of the I2C. The serial bit clock frequency is derived from the module clock. The module clock and peripheral clocks are synchronous to each other. The minimum frequency of module clock should be 12.8 MHz for Fast Mode to achieve 400 Kbps operation.

41.3.10 Reset

The I2C can be reset in two ways.

1. Global reset: A hard asynchronous reset of the whole I2C.
2. Software reset: An internal reset for whole I2C, except I2C_IADR and I2C_IFDR registers, is initiated by deasserting the I2C_I2CR[IEN] bit.

41.3.11 Interrupts

There is only one interrupt from the block. The interrupt is enabled by setting the I2C_I2CR[IIEN] bit.

The interrupt is generated in any one of the following conditions.

- One byte transfer is completed (the interrupt is set at the falling edge of the ninth clock).
- An address is received that matches its own specific address in slave-receive mode.
- Arbitration is lost.

41.3.12 Byte Order

The block only supports the little endian mode.

41.4 Initialization

41.4.1 Initialization Sequence

Before the interface can transfer serial data, registers must be initialized, as follows:

1. Set the data sampling rate (I2C_IFDR[IC] to obtain SCL frequency from the system bus clock.
2. Update the address in the (I2C_IADR) to define its slave address (address can range from 0 to 0x7f).
3. Set the I2C enable bit (I2C_I2CR[IEN]) to enable the I2C bus interface system.
4. Modify the bits in the I2C_I2CR to select master/slave mode, transmit/receive mode, and interrupt-enable or not.

41.4.2 Generation of START

After completion of the initialization procedure, serial data can be transmitted by selecting the master transmitter mode. On a multiple-master bus system, the busy bus (I2C_I2SR[IBB]) must be tested to determine whether the serial bus is free. If the bus is free (IBB = 0), the START signal and the first byte (the slave address) can be sent. The data written to the data register comprises the address of the desired slave and the LSB indicates the transfer direction.

The free time between a STOP and the next START condition is built into the hardware that generates the START cycle. Depending on the relative frequencies of the system clock and the SCL period, it may be necessary to wait until the I2C is busy after writing the calling address to the data register (I2C_I2DR) before proceeding to load data into the data register (I2C_I2DR).

41.4.3 Post-Transfer Software Response

Sending or receiving a byte sets the data transferring bit (I2C_I2SR[ICF]), which indicates one byte communication is finished. Upon completion, the interrupt status (I2C_I2SR[IIF]) is also set. An external interrupt is generated if the interrupt enable (I2C_I2CR[IIEN]) is set. The software must first clear the interrupt status (I2C_I2SR[IIF]) in the interrupt routine.

(See the flow chart in Figure 41-7.) The data transferring bit (I2C_I2SR[ICF]) is cleared either by reading from I2C_I2DR in receive mode or by writing to this register in transmit mode.

The software can service the I2C I/O in the main program by monitoring the interrupt status (I2C_I2SR[IIF]) if the interrupt enable is deasserted. In this case, the interrupt status should be polled of the data transferring bit (I2C_I2SR[ICF]) because the operation is different when arbitration is lost.

When an interrupt occurs at the end of the address cycle, the master is always in transmit mode; that is, the address is sent. If master receive mode is required, then I2C_I2CR[MTX] should be toggled and dummy read of I2C_I2DR register has to be done for triggering receive data.

During slave-mode address cycles (I2C_I2SR[IAAS] = 1), the slave read/write bit I2C_I2SR[SRW] is read to determine the direction of the next transfer. The transmit/receive bit (I2C_I2CR[MTX]) should also be programmed accordingly. For slave-mode data cycles (IAAS = 0), SRW is invalid. MTX should be read to determine the current transfer direction.

41.4.4 Generation of STOP

A data transfer ends when the master signals a STOP, which can occur after all data is sent.

For a master receiver to terminate a data transfer, it must inform the slave transmitter by not acknowledging the last data byte. This is done by setting the transmit acknowledge bit (I2C_I2CR[TXAK]) before reading the next-to-last byte. Before the last byte is read, a STOP signal must be generated.

41.4.5 Generation of Repeated START

After the data transfer, if the master still wants the bus, it can signal another START followed by another slave address without signalling a STOP.

41.4.6 Slave Mode

In the slave interrupt service routine (see Figure 41-7), the block addressed as slave bit (IAAS) should be tested to check if a calling of its own address has just been received. If IAAS is set, software should set the transmit/receive mode select bit (I2C_I2CR[MTX]) according to the I2C_I2SR[SRW]. Writing to the I2C_I2CR clears the IAAS automatically. The only time IAAS is read as set is from the interrupt at the end of the address cycle where an address match occurred; interrupts resulting from subsequent data transfers will have IAAS cleared. A data transfer can now be initiated by writing information to I2C_I2DR for slave transmits, or read from I2C_I2DR in slave-receive mode. A dummy read of I2C_I2DR in slave/receive mode releases SCL, allowing the master to send data.

In the slave transmitter routine, the receive acknowledge bit (I2C_I2SR[RXAK]) must be tested before sending the next byte of data. Setting RXAK means an end-of-data signal from the master receiver, after which the software must switch it from transmitter to receiver mode. Reading the data register (I2C_I2DR) then releases SCL so that the master can generate a STOP signal.

41.4.7 Arbitration Lost

If several devices try to engage the bus at the same time, one becomes master. Hardware immediately switches devices that lose arbitration to slave receive mode. Data output to SDA stops, but SCL is still generated until the end of the byte during which arbitration is lost. An interrupt occurs at the falling edge of the ninth clock of this transfer if the arbitration is lost (I2C_I2SR[IAL] = 1), and the slave mode is selected (I2C_I2CR[MSTA] = 0).

See the flow chart in Figure 41-7.

If a device that is not a master tries to transmit or do a START, hardware inhibits the transmission, clears MSTA without signalling a STOP, generates an interrupt to the ARM platform, and sets I2C_I2SR[IAL] to indicate a failed attempt to engage the bus. When considering these cases, the slave service routine should first test I2C_I2SR[IAL], and the software should clear it if it is set.

For Multi-master mode, when an I2C is enabled when the bus is busy and asserts START, the I2C_I2SR[IAL] bit gets set only for SDA=0, SCL=0/1, SDA=1, and SCL=0, but not for SDA=1 and SCA=1, which is the same as bus idle state.

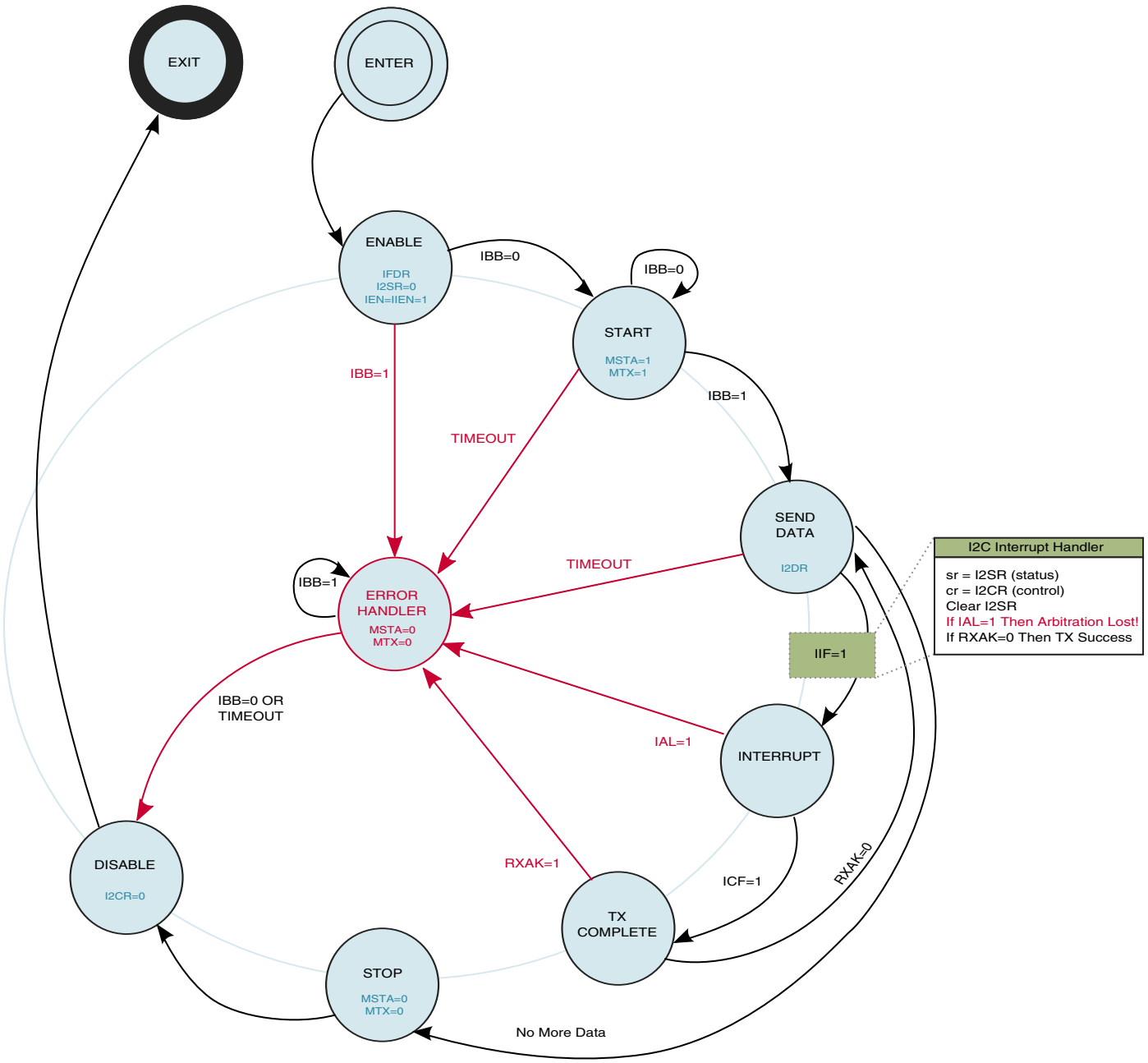


Figure 41-6. I2C Programming State Diagram

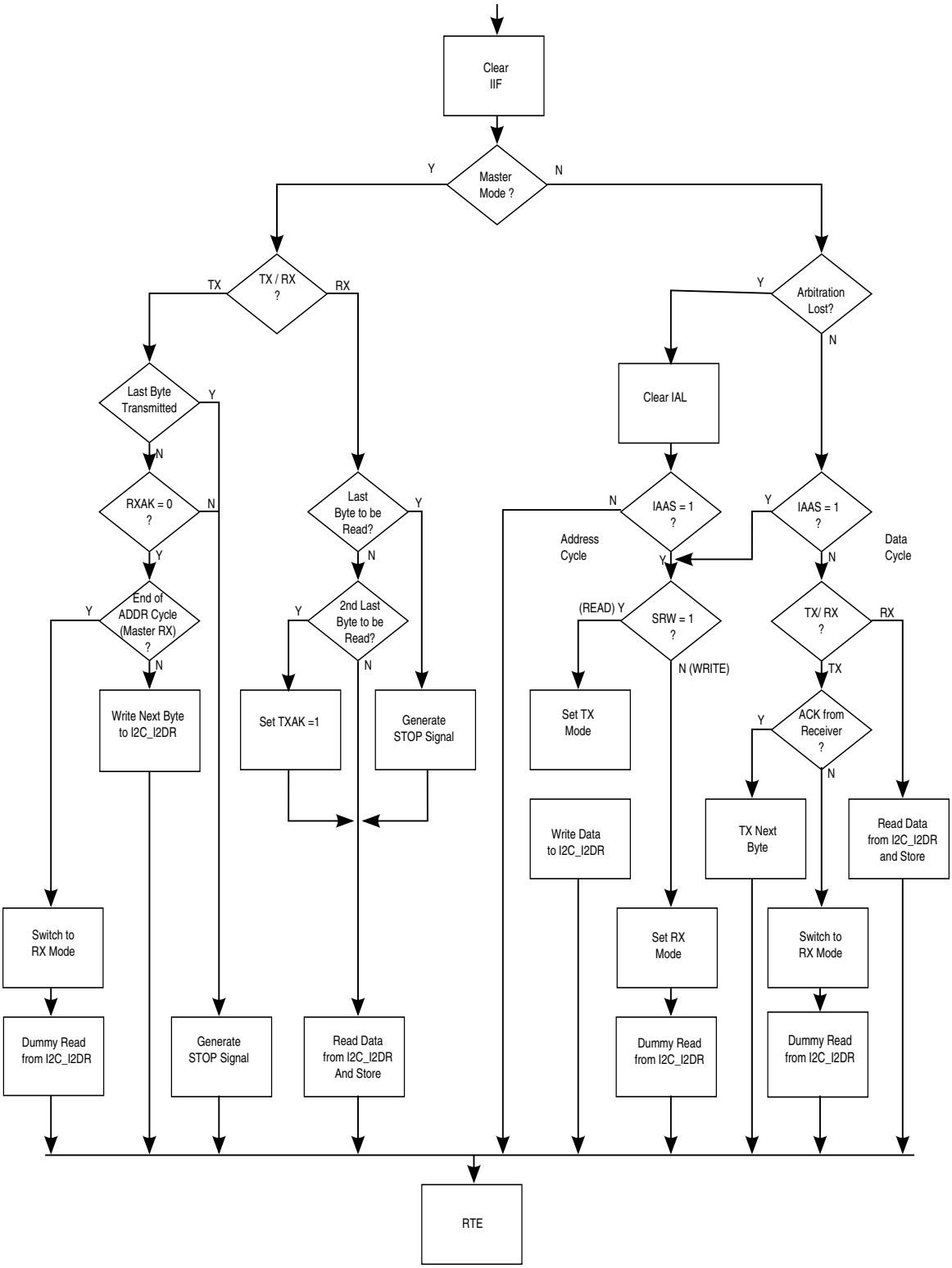


Figure 41-7. Flow-Chart of Typical I2C Interrupt Routine

NOTE

For a repeated start-only, the stop generation stage will not occur in master mode. A loop will repeat itself without stopping for the next start.

For Master Rx mode, I2C is programmed as master transmit during address mode and after slave address transfer, MTX bit should be cleared and Dummy read on I2C_I2DR register should be performed so that I2C can read the next receive data.

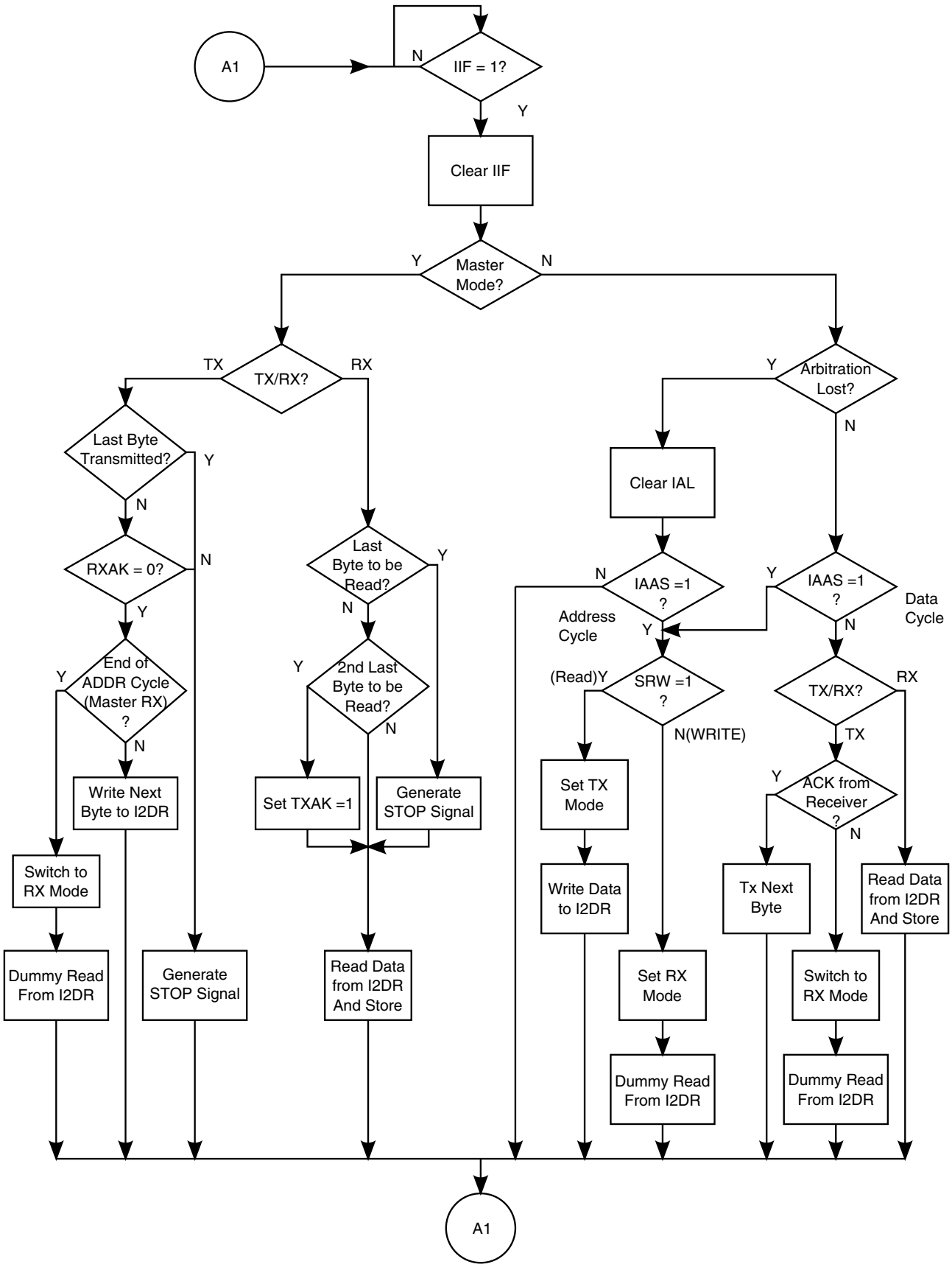


Figure 41-8. Flow Chart for Typical I2C Polling Routi

NOTE

The time-out value will depend on the bus frequency at which I2C is operating. The Min. Time-out for polling IIF bit at I2C Max bus frequency of 400KHz is, T_{min} = 25 μs (= 2.5*10 μs). This value can be interpolated for any bus frequency.

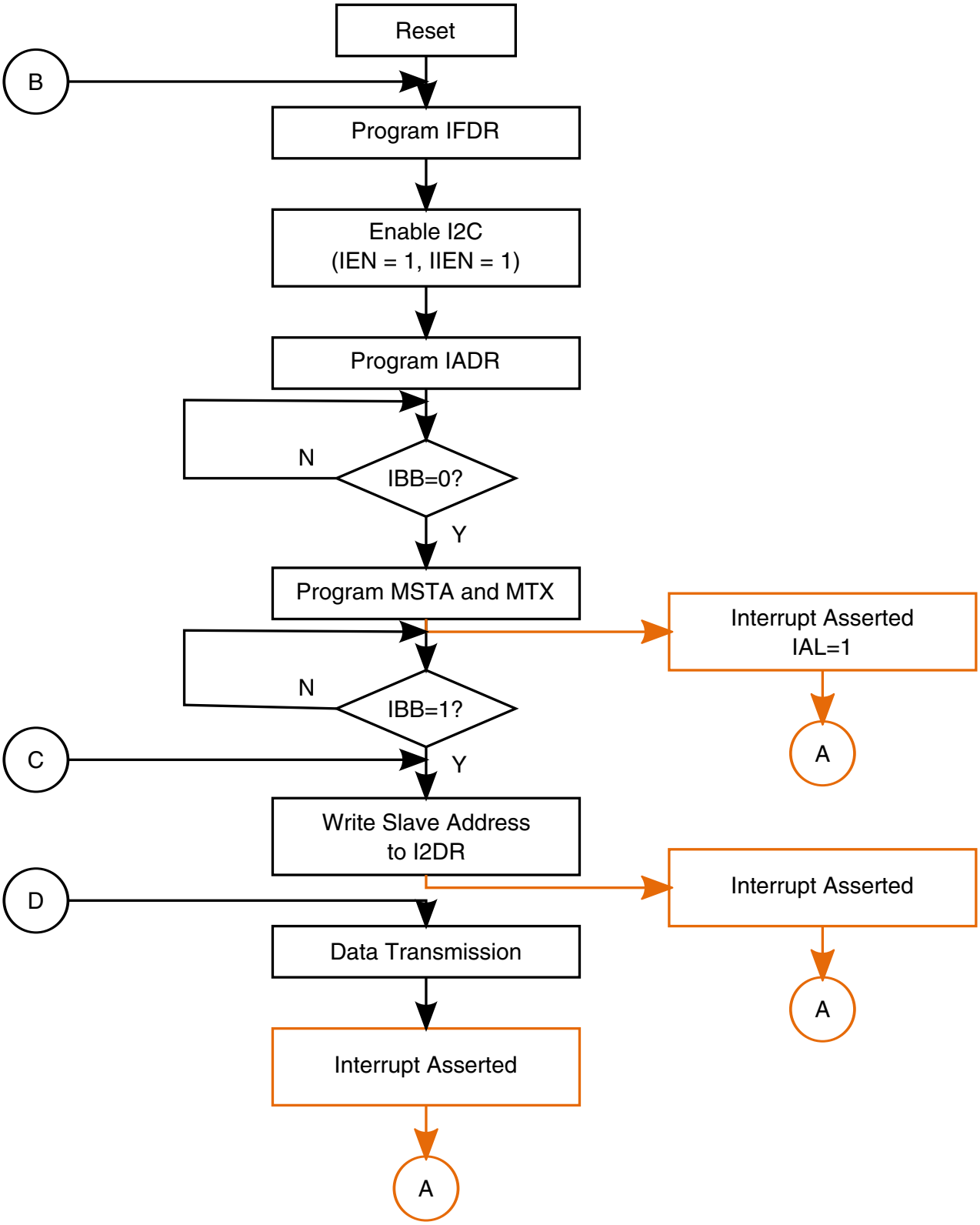


Figure 41-9. Detailed Flow Chart of a Typical I2C Master Tx Mode, Part 1

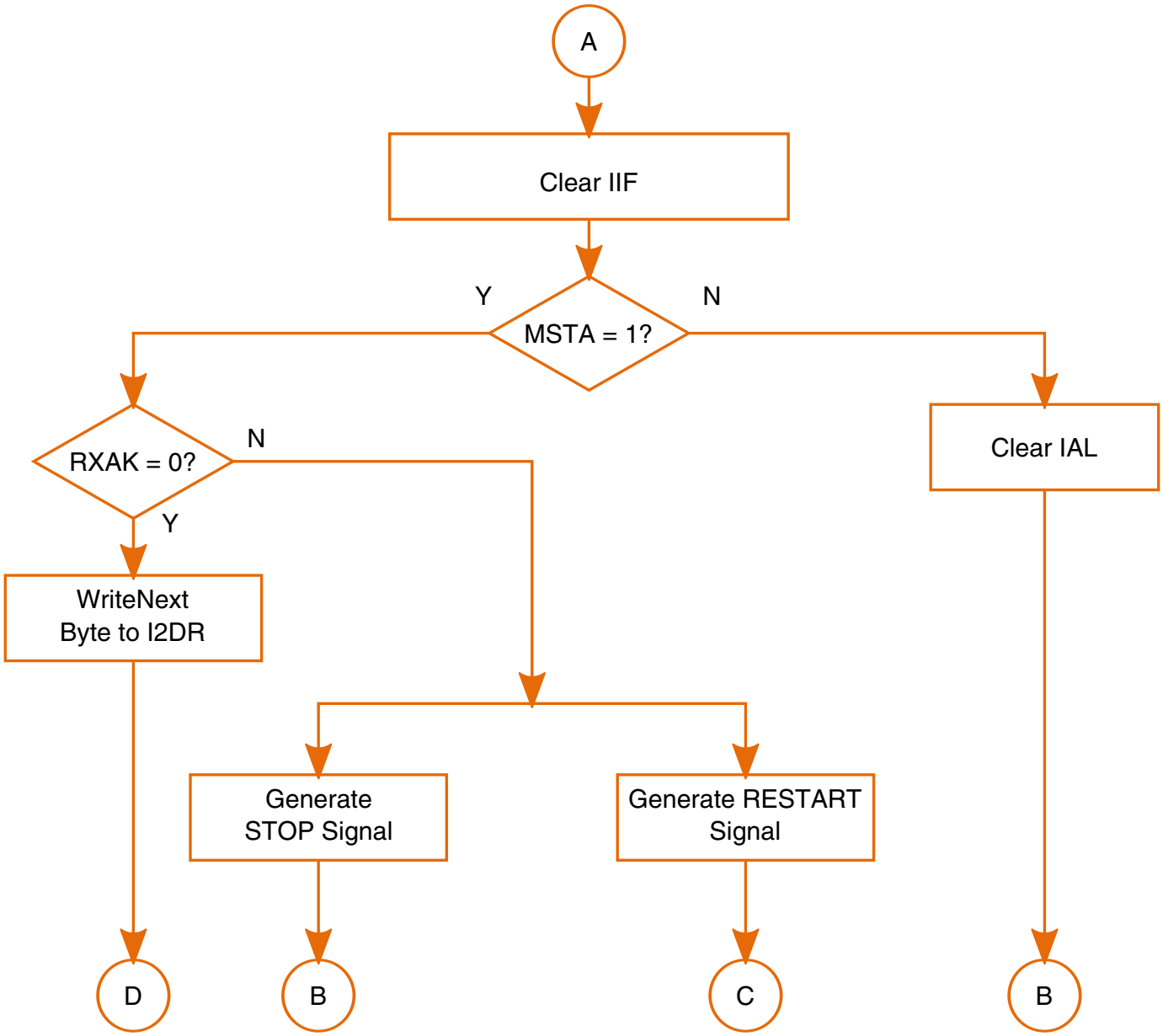


Figure 41-10. Detailed Flow Chart of a Typical I2C Master Tx Mode, Part 2

Figure 41-9 and Figure 41-10 show the Master Transmit mode operation with interrupt sub-routine. In case interrupt is generated and the MSTA bit is 0, then bus arbitration is lost and IAL is set. Software can clear IAL bit and re-program I2C. If MSTA bit is 1, then it will be a transfer done interrupt and software can check RXAK bit for data receive acknowledgement by slave and accordingly decide to either generate STOP, REPEAT START by writing in I2C_I2CR register or next data transfer by writing into I2C_I2DR register.

NOTE

The IBB bit is asserted by START condition on the bus, and it is deasserted by STOP condition on bus. Therefore, if arbitration is lost due to an unexpected STOP condition during transfer, then IBB is cleared. If arbitration is lost due to data mismatch, then it will not be cleared. Software should always clear the IEN bit and then set it if arbitration is lost.

41.5 Software Restriction

Software should take care that there is a delay of at least two Module Clock cycles after it sets the I2C_I2CR[RSTA] bit before writing to the I2C_I2DR register. Maximum possible clock period of Module Clock is 78 ns.

41.6 Programmable Registers

41.6.1 I2C Memory Map/Register Definition

The I2C contains five 16-bit registers.

NOTE

Registers at offsets 0x0002, 0x0006, 0x000A, and 0x000E are reserved for future additions.

I2C memory map

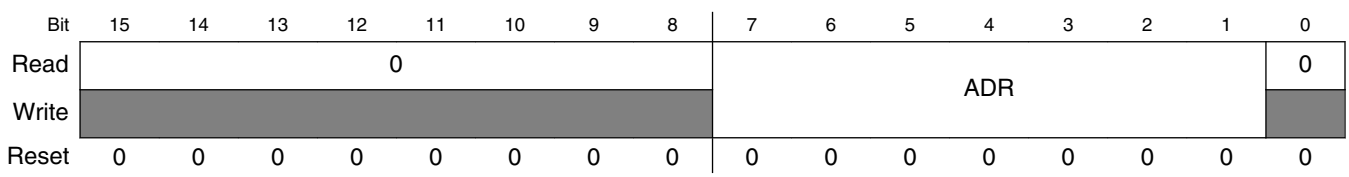
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FE_C000	I2C Address Register (I2C-3_IADR)	16	R/W	0000h	41.61.1/ 1813
53FE_C004	I2C Frequency Divider Register (I2C-3_IFDR)	16	R/W	0000h	41.61.2/ 1814
53FE_C008	I2C Control Register (I2C-3_I2CR)	16	R/W	0000h	41.61.3/ 1815
53FE_C00C	I2C Status Register (I2C-3_I2SR)	16	R/W	0081h	41.61.4/ 1817
53FE_C010	I2C Data I/O Register (I2C-3_I2DR)	16	R/W	0000h	41.61.5/ 1818

I2C memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FC_4000	I2C Address Register (I2C-2_IADR)	16	R/W	0000h	41.61.1/ 1813
63FC_4004	I2C Frequency Divider Register (I2C-2_IFDR)	16	R/W	0000h	41.61.2/ 1814
63FC_4008	I2C Control Register (I2C-2_I2CR)	16	R/W	0000h	41.61.3/ 1815
63FC_400C	I2C Status Register (I2C-2_I2SR)	16	R/W	0081h	41.61.4/ 1817
63FC_4010	I2C Data I/O Register (I2C-2_I2DR)	16	R/W	0000h	41.61.5/ 1818
63FC_8000	I2C Address Register (I2C-1_IADR)	16	R/W	0000h	41.61.1/ 1813
63FC_8004	I2C Frequency Divider Register (I2C-1_IFDR)	16	R/W	0000h	41.61.2/ 1814
63FC_8008	I2C Control Register (I2C-1_I2CR)	16	R/W	0000h	41.61.3/ 1815
63FC_800C	I2C Status Register (I2C-1_I2SR)	16	R/W	0081h	41.61.4/ 1817
63FC_8010	I2C Data I/O Register (I2C-1_I2DR)	16	R/W	0000h	41.61.5/ 1818

41.61.1 I2C Address Register (I2Cx_IADR)

Addresses: I2C-3_IADR is 53FE_C000h base + 0h offset = 53FE_C000h
I2C-2_IADR is 63FC_4000h base + 0h offset = 63FC_4000h
I2C-1_IADR is 63FC_8000h base + 0h offset = 63FC_8000h



I2Cx_IADR field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–1 ADR	Slave address. Contains the specific slave address to be used by the I2C. Slave mode is the default I2C mode for an address match on the bus.

Table continues on the next page...

I2Cx_IADR field descriptions (continued)

Field	Description
	NOTE: The I2C_IADR holds the address the I2C responds to when addressed as a slave. The slave address is not the address sent on the bus during the address transfer. The register is not reset by a software reset.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

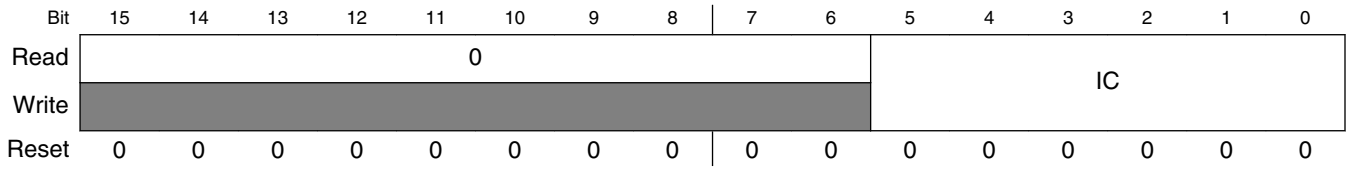
41.61.2 I2C Frequency Divider Register (I2Cx_IFDR)

The I2C_IFDR provides a programmable prescaler to configure the clock for bit-rate selection. The register does not get reset by software reset. The following table describes the Divider values for register field "IC". Table below describes the register values for field "IC".

Table 41-12. I2C_IFDR Register Field Values

IC	Divider		IC	Divider		IC	Divider		IC	Divider
0x00	30		0x10	288		0x20	22		0x30	160
0x01	32		0x11	320		0x21	24		0x31	192
0x02	36		0x12	384		0x22	26		0x32	224
0x03	42		0x13	480		0x23	28		0x33	256
0x04	48		0x14	576		0x24	32		0x34	320
0x05	52		0x15	640		0x25	36		0x35	384
0x06	60		0x16	768		0x26	40		0x36	448
0x07	72		0x17	960		0x27	44		0x37	512
0x08	80		0x18	1152		0x28	48		0x38	640
0x09	88		0x19	1280		0x29	56		0x39	768
0x0A	104		0x1A	1536		0x2A	64		0x3A	896
0x0B	128		0x1B	1920		0x2B	72		0x3B	1024
0x0C	144		0x1C	2304		0x2C	80		0x3C	1280
0x0D	160		0x1D	2560		0x2D	96		0x3D	1536
0x0E	192		0x1E	3072		0x2E	112		0x3E	1792
0x0F	240		0x1F	3840		0x2F	128		0x3F	2048

Addresses: I2C-3_IFDR is 53FE_C000h base + 4h offset = 53FE_C004h
I2C-2_IFDR is 63FC_4000h base + 4h offset = 63FC_4004h
I2C-1_IFDR is 63FC_8000h base + 4h offset = 63FC_8004h



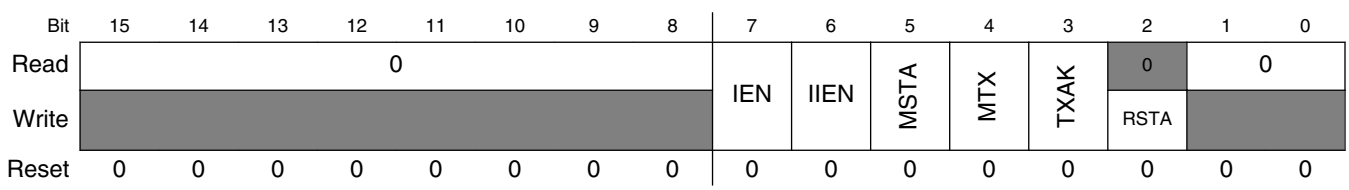
I2Cx_IFDR field descriptions

Field	Description
15–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5–0 IC	I2C clock rate. Pre scales the clock for bit-rate selection. Due to potentially slow SCL and SDA rise and fall times, bus signals are sampled at the prescaler frequency. The serial bit clock frequency is equal to IPG_CLK_ROOT divided by the divider shown in I2C Data I/O Register I2C Frequency Divider Register. **NOTE:** The IC value should not be changed during the data transfer, however, it can be changed before REPEAT START or START programming sequence in I2C. The I2C protocol supports bit rates up to 400 kbps. The IC bits need to be programmed in accordance with this constraint.

41.61.3 I2C Control Register (I2Cx_I2CR)

The I2C_I2CR is used to enable the I2C and the I2C interrupt. It also contains bits that govern operation as a slave or a master.

Addresses: I2C-3_I2CR is 53FE_C000h base + 8h offset = 53FE_C008h
I2C-2_I2CR is 63FC_4000h base + 8h offset = 63FC_4008h
I2C-1_I2CR is 63FC_8000h base + 8h offset = 63FC_8008h



I2Cx_I2CR field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7 IEN	I2C enable. Also controls the software reset of the entire I2C. Resetting the bit generates an internal reset to the block. If the block is enabled in the middle of a byte transfer, slave mode ignores the current bus transfer and starts operating when the next start condition is detected. Master mode is not aware that the bus is busy so initiating a start cycle may corrupt the current bus cycle, ultimately causing either the current master or the I2C to lose arbitration. After which, bus operation returns to normal.

Table continues on the next page...

I2Cx_I2CR field descriptions (continued)

Field	Description
	0 The block is disabled, but registers can still be accessed. 1 The I2C is enabled. This bit must be set before any other I2C_I2CR bits have any effect.
6 IIEN	I2C interrupt enable. **NOTE:** If data is written during the START condition, that is, just after setting the I2C_I2CR[MSTA] and I2C_I2CR[MTX] bits, then the ICF bit is cleared at the falling edge of SCLK after START. If data is written after the START condition and falling edge of SCLK, then ICF bit is cleared as soon as data is written. 0 I2C interrupts are disabled, but the status flag I2C_I2SR[IIF] continues to be set when an interrupt condition occurs. 1 I2C interrupts are enabled. An I2C interrupt occurs if I2C_I2SR[IIF] is also set.
5 MSTA	Master/slave mode select bit. If the master loses arbitration, MSTA is cleared without generating a STOP signal. **NOTE:** Module clock should be on for writing to the MSTA bit. **NOTE:** The MSTA bit is cleared by software to generate a STOP condition; it can also be cleared by hardware when the I2C loses the bus arbitration. 0 Slave mode. Changing MSTA from 1 to 0 generates a STOP and selects slave mode. 1 Master mode. Changing MSTA from 0 to 1 signals a START on the bus and selects master mode.
4 MTX	Transmit/receive mode select bit. Selects the direction of master and slave transfers. 0 Receive. When a slave is addressed, the software should set MTX according to the slave read/write bit in the I2C status register (I2C_I2SR[SRW]). 1 Transmit. In master mode, MTX should be set according to the type of transfer required. Therefore, for address cycles, MTX is always 1.
3 TXAK	Transmit acknowledge enable. Specifies the value driven onto SDA during acknowledge cycles for both master and slave receivers. **NOTE:** Writing TXAK applies only when the I2C bus is a receiver. 0 An acknowledge signal is sent to the bus at the ninth clock bit after receiving one byte of data. 1 No acknowledge signal response is sent (that is, the acknowledge bit = 1).
2 RSTA	Repeat start. Always reads as 0. Attempting a repeat start without bus mastership causes loss of arbitration. 0 No repeat start 1 Generates a repeated START condition
1–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

41.61.4 I2C Status Register (I2Cx_I2SR)

The I2C_I2SR contains bits that indicate transaction direction and status.

Addresses: I2C-3_I2SR is 53FE_C000h base + Ch offset = 53FE_C00Ch
I2C-2_I2SR is 63FC_4000h base + Ch offset = 63FC_400Ch
I2C-1_I2SR is 63FC_8000h base + Ch offset = 63FC_800Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								ICF	IAAS	IBB	IAL	0	SRW	IIF	RXAK
Write																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1

I2Cx_I2SR field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7 ICF	Data transferring bit. While one byte of data is transferred, ICF is cleared. 0 Transfer is in progress. 1 Transfer is complete. This bit is set by the falling edge of the ninth clock of the last byte transfer.
6 IAAS	I2C addressed as a slave bit. The ARM platform is interrupted if the interrupt enable (I2C_I2CR[IIEN]) is set. The ARM platform must check the slave read/write bit (SRW) and set its TX/RX mode accordingly. Writing to I2C_I2CR clears this bit. 0 Not addressed 1 Addressed as a slave. Set when its own address (I2C_IADR) matches the calling address.
5 IBB	I2C bus busy bit. Indicates the status of the bus. **NOTE:** When I2C is enabled (I2C_I2CR[IEN] = 1), it continuously polls the bus data (SDAK) and clock (SCLK) signals to determine a START or STOP condition. 0 Bus is idle. If a STOP signal is detected, IBB is cleared. 1 Bus is busy. When START is detected, IBB is set.
4 IAL	Arbitration lost. Set by hardware in the following circumstances (IAL must be cleared by software by writing a "0" to it at the start of the interrupt service routine): • SDA input sampled low when the master drives high during an address or data-transmit cycle. • SDA input sampled low when the master drives high during the acknowledge bit of a data-receive cycle. For the above two cases, the bit is set at the falling edge of 9th SCL clock during the ACK cycle. • A start cycle is attempted when the bus is busy. • A repeated start cycle is requested in slave mode. • A stop condition is detected when the master did not request it. **NOTE:** Software cannot set the bit. 0 No arbitration lost. 1 Arbitration is lost.

Table continues on the next page...

I2Cx_I2SR field descriptions (continued)

Field	Description
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2 SRW	Slave read/write. When the I2C is addressed as a slave, IAAS is set, and the slave read/write bit (SRW) indicates the value of the R/W command bit of the calling address sent from the master. SRW is valid only when a complete transfer has occurred, no other transfers have been initiated, and the I2C is a slave and has an address match. 0 Slave receive, master writing to slave 1 Slave transmit, master reading from slave
1 IIF	I2C interrupt. Must be cleared by the software by writing a "0" to it in the interrupt routine. **NOTE:** The software cannot set the bit. 0 No I2C interrupt pending. 1 An interrupt is pending. This causes a processor interrupt request (if the interrupt enable is asserted [IIEN = 1]). The interrupt is set when one of the following occurs: • One byte transfer is completed (the interrupt is set at the falling edge of the ninth clock). • An address is received that matches its own specific address in slave-receive mode. • Arbitration is lost.
0 RXAK	Received acknowledge. This is the value received of the SDA input for the acknowledge bit during a bus cycle. 0 An "acknowledge" signal was received after the completion of an 8-bit data transmission on the bus. 1 A "No acknowledge" signal was detected at the ninth clock.

41.61.5 I2C Data I/O Register (I2Cx_I2DR)

In master-receive mode, reading the data register allows a read to occur and initiates the next byte to be received. In slave mode, the same function is available after it is addressed.

Addresses: I2C-3_I2DR is 53FE_C000h base + 10h offset = 53FE_C010h
I2C-2_I2DR is 63FC_4000h base + 10h offset = 63FC_4010h
I2C-1_I2DR is 63FC_8000h base + 10h offset = 63FC_8010h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								DATA							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Cx_I2DR field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–0 DATA	Data Byte. Holds the last data byte received or the next data byte to be transferred. Software writes the next data byte to be transmitted or reads the data byte received. **NOTE:** The core-written value in I2C_I2DR cannot be read back by the core. Only data written by the I2C bus side can be read.

Chapter 42
IC Identification Module (IIM)

42.1 Overview

The IC Identification Module (IIM) provides an interface for reading, and in some cases programming and/or overriding, identification and control information stored in on-chip fuse elements. The block supports electrically-programmable poly fuses (e-Fuses).

The IIM also provides a set of volatile software-accessible signals which can be used for software control of hardware elements, not requiring non-volatility.

The IIM provides the primary user-visible mechanism for interfacing with on-chip fuse elements. Among the uses for the fuses are unique chip identifiers, mask revision numbers, cryptographic keys, and various control signals requiring permanent non-volatility. The IIM also provides up to 28 volatile control signals

42.1.1 Modes of Operation

The IIM is in its functional mode (all specified functionality available) any time it is out of reset and supplied with the proper clocks.

42.2 Functional Description

The IIM is an 8-bit IP Bus peripheral which implements interfaces for laser fuses and/or e-Fuses, as well as the hardware revision codes, cryptographic keys, and miscellaneous SoC-level feature enabling circuitry. Up to eight banks, each with up to 2048 fuses, are supported. Laser fuse banks and e-Fuse banks may be mixed.

42.2.1 Signal Groups

Listed below are the intended uses for the five fuse banks. Each fuse bank may contain up to 2048 fuses, and the unspecified fuses in each bank below mScay be used for implementation-specific purposes. However, access protection is specified on a per-bank basis; this may limit the use of the unspecified fuses.

The IIM implements non-volatile "Hardware-Visible" control signals. These are all capable of driving SoC-level nets to allow them to permanently enable or disable features in the system.

42.2.1.1 System JTAG Control

Among the hardware control signals implemented in Fuse Bank 0 are the Secure JTAG mode control bits. These fuses are used to allow or disallow JTAG access to secured resources.

Three JTAG security levels are envisioned, as shown in the table below.

Table 42-1. JTAG Security Level Control Bits

Security Mode	JTAG_SMODE	JTAG_SCC_DIS	Description
No Debug	2'b11	x	The highest security level.
Secure JTAG	2'b01	x	Limit the JTAG access by using key based authentication mechanism.
JTAG Enable	2'b00	1'b1	Low Security, all JTAG features are enabled.
SCC JTAG	2'b00	1'b0	No security.

JTAG debug access may be semi-permanently enabled by blowing the jtag_bp bit (JTAG Bypass Security). This, however, can be overridden by blowing the jtag_re bit (JTAG Security Re-enable). Blowing the jtag_bp bit effectively changes the security level from 'Secure JTAG' to 'JTAG Enable', without actually modifying the JTAG_SMODE fuses.

NOTE

The IIM must only allow the jtag_bp fuse to be blown if the JTAG debug security level is "Secure JTAG" (JTAG_SMODE = 2'b01) and ipt_response_in = 1, ipt_secur_block = 0. This indicates that the SJC has processed a valid response in its challenge/response protocol, and the JTAG security bypass functionality should be allowed. JTAG security is re-instated to the "Secure JTAG" level by blowing the jtag_re bit.

In addition to the three JTAG security levels, the JTAG debug notification normally passed to the Security Controller (SCC), which causes it to transition out of the secure state when debugging has become active, can be gated off using the jtag_scc fuse in JAC. If jtag_scc is left unblown, the debug active signals are masked, and do not propagate to the SCC. If jtag_scc is blown, the debug active signals propagate to the SCC as normal.

NOTE

Leaving the jtag_scc fuse unblown represents a very significant security risk, and should only be done in as few parts as possible to debug secure operation of the SCC. Once SCC debugging is complete, the jtag_scc fuse should immediately be blown. The reason for including this functionality in a fuse is to avoid the need to modify parts using FIB for SCC debugging, but the convenience afforded by the fuse must not be allowed to foster an unguarded attitude toward the dangers such insecure parts would pose.

42.2.1.2 Fuse Bank 0

Fuse Bank 0 contains manufacturing information such as Unique ID (UID), consists of fab ID, lot and wafer number, and die coordinates. These bits are usable by OSs such Microsoft PocketPC, which require a unique device ID. It also contains JTAG and debug features controls, IMEI information, and 24 bits for user purpose.

42.2.1.3 Fuse Bank 1

Fuse Bank 1 contains response reference value for the system JTAG controller, memories information and 40 bits for user purposes.

42.2.1.4 Fuse Bank 2

Fuse Bank 2 contains the 256-bit SCC key.

42.2.1.4.1 SCC Key Format

The-256 bit keys derived from Fuse Bank 2 are used as the secret keys for the Secure RAM controller in the SCC(s). Each key actually consists of three 56-bit keys. Each of the three 56-bit keys corresponds to a 64-bit DES key with the parity bits removed, and processed through the key permutation.

The value in Fuse Bank 1includes 159 random bits and 9 Hamming Code bits. The Hamming Code will detect all 1, 2, and 3 bit errors in the codeword. It will also detect most (though not all) multiple bit errors. Numbering the bits in the key from 0 to 167, the code bits are bits 0, 1, 2, 4, 8, 16, 32, 64, and 128. All of the remaining bits are data bits. Each of the code bits is the modulo 2 sum (i.e XOR or parity) of a subset of the bits in the entire word, as described below.

To determine which bits are used to form each code bit, first look at the binary representation of the bit position of each code bit (ignoring bit 0 for the time being) as shown in the table below.

Table 42-2. Binary Representation of Bit Position

Code Bit	Binary Representation
0	0b00000000
1	0b00000001
2	0b00000010
4	0b00000100
8	0b00001000
16	0b00010000
32	0b00100000
64	0b01000000
128	0b10000000

For code bit number 1, there is a single '1' bit in its binary representation. This code bit is the modulo 2 sum (XOR) of all bit positions which also have a '1' in this same point in their binary representation (that is, all odd bit positions). Alternatively, if we bitwise AND code bit 1's bit number with a data bit's bit number, and the result is not zero, then that bit gets XOR'ed into code bit 1. The same is true for the other code bits, excepting code bit 0. For example, code bit 2 is the XOR of bits 3, 6, 7, 10, 11, 14, 15...254,255. Code bit 128 is the XOR of all bits from 129 through 255 inclusive. After all of the other code bits have been calculated, code bit 0 is simply the XOR of all of the other bits, including the other code bits.

Another way to look at this is to ask which code bits a data bit affects. If we look at the binary representation of the bit position of a data bit, the 1's represent the code bits that this data bit affects. For example, data bit 99 (0b01100011) is XOR'ed into code bits 1, 2, 32, and 64.

42.2.1.4.2 SCC Key Checking

The SCC checks the key using the Hamming Code. If there are any errors in the key, the SCC refuses to use it. In addition, the SCC checks the key against several known weak keys, which it will also refuse to use. The following figure shows a list of hexadecimal key patterns that are checked (x means don't care); no key matching any of these patterns should be programmed into Fuse Bank 1.

```
00000000000000_xxxxxxxxxxxxxxx_xxxxxxxxxxxxxxx
xxxxxxxxxxxxxxx_00000000000000_xxxxxxxxxxxxxxx
xxxxxxxxxxxxxxx_xxxxxxxxxxxxxxx_00000000000000
FFFFFFFFFFFFFF_xxxxxxxxxxxxxxx_xxxxxxxxxxxxxxx
xxxxxxxxxxxxxxx_FFFFFFFFFFFFFF_xxxxxxxxxxxxxxx
xxxxxxxxxxxxxxx_xxxxxxxxxxxxxxx_FFFFFFFFFFFFFF
0000000FFFFFFF_xxxxxxxxxxxxxxx_xxxxxxxxxxxxxxx
xxxxxxxxxxxxxxx_0000000FFFFFFF_xxxxxxxxxxxxxxx
xxxxxxxxxxxxxxx_xxxxxxxxxxxxxxx_0000000FFFFFFF
FFFFFFF0000000_xxxxxxxxxxxxxxx_xxxxxxxxxxxxxxx
xxxxxxxxxxxxxxx_FFFFFFF0000000_xxxxxxxxxxxxxxx
xxxxxxxxxxxxxxx_xxxxxxxxxxxxxxx_FFFFFFF0000000
55555555555555_AAAAAAAAAAAAAA_33333333333333
AAAAAAAAAAAAAA_55555555555555_CCCCCCCCCCCCCC
```

Figure 42-1. Known Weak SCC Key Patterns

42.2.1.5 Fuse Bank 3

Fuse Bank 3 is intended to hold data relevant to the Super-Root Key (SRK) of DSP domain. It may hold an entire SRK (that is, an up to 1024-bit RSA public key), a hash of an SRK (that is, a 160-bit SHA-1 hash), or an SRK one-of-many selector. The boot code for a specific product must know where to find and validate the SRK.

42.2.1.6 Fuse Bank 4

Fuse Bank 4 is a 16-word (128 fuses) bank added to i.MX53 for general purpose use.

42.2.1.7 Software-Controllable Volatile Signals

The IIM implements up to 28 software-controllable, HW-Visible, volatile control signals. These are implemented in four 8-bit registers, each with a lock bit to inhibit modification of the associated register until the IIM is reset. These 28 signals may all drive SoC-level

nets. They are always software-readable, but can only be modified by software if the Lock bit (bit 7) is not set in the register. The sequence for modifying these bits is shown in the figure below.

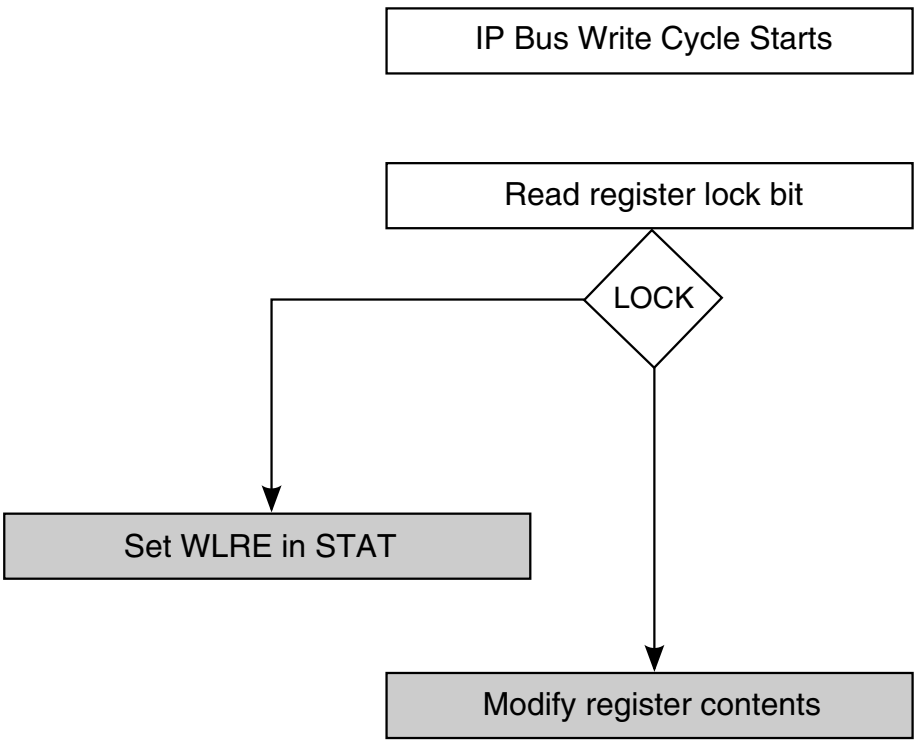


Figure 42-2. IIM_SCS Register Write Sequence

42.2.2 FUSEBOX Signals

The figure below shows the FUSEBOX interface.

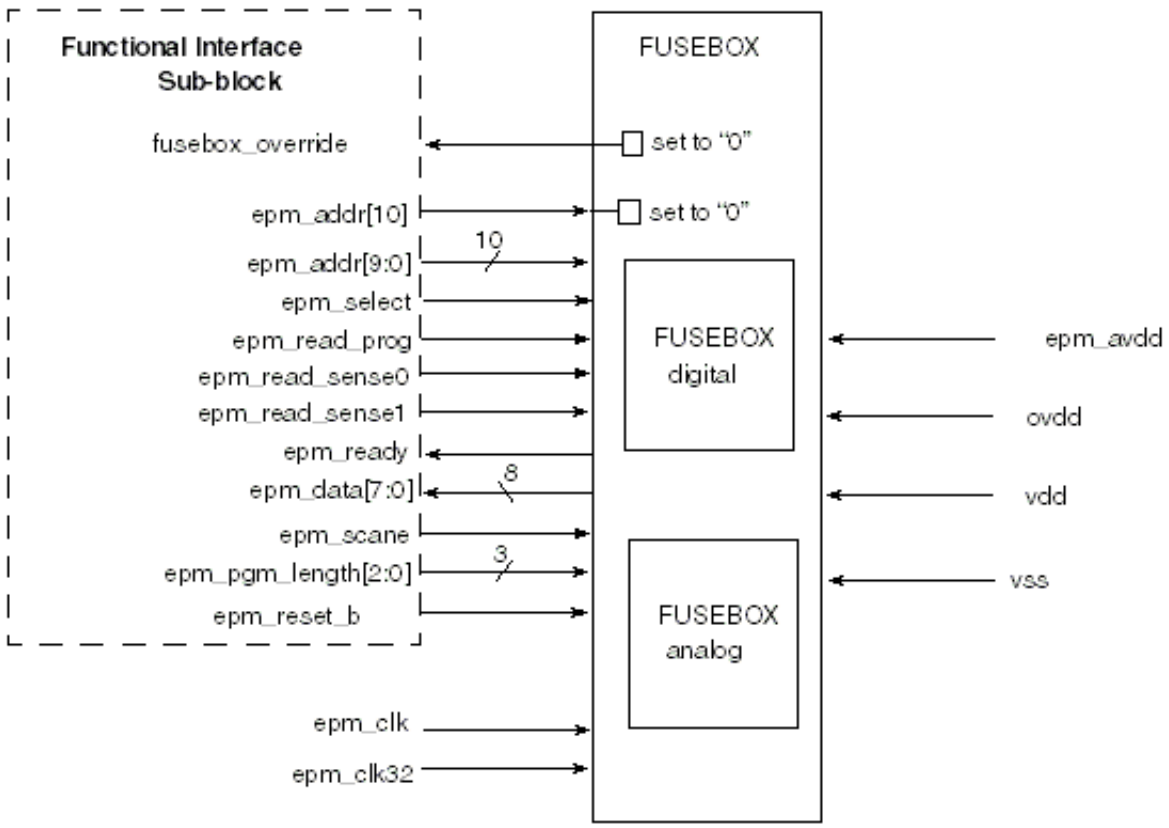


Figure 42-3. FUSEBOX Interface

The signals are summarized from the fuse box's perspective in the following table.

Table 42-3. FUSEBOX Signal Overview

Signal Name	Signal Type	Signal Description
EPM_ADDR[10:0]	I	This bus provides the address for a read or program operation. Read operations are done on a word (8-bit) basis, so the word address is carried on address[10:3]. Program operations are done on a bit-basis, so the bit address is carried on address[10:0]. Address bits l2:0] carry the bit location in a byte, epm_addr[10] is set to "0" for 1024 bit digital block.
		State Meaning
		Timing Needs to be held to desired value at least 2 nS before and after asserting epm_select in both program and read operation.
$\overline{\text{EPM\_CLK}}$	I	Local clock used for synchronization of signals between FI and FUSEBOX. It is the clock used by a FUSEBOX Interface sub-block for creating FUSEBOX interface signals. The max. read clock is 166Mhz with worst 40%-60% duty cycle.
$\overline{\text{EPM\_CLK32}}$	I	The 32.768 KHz clock used in the FUSEBOX to time the program operation in read operation. This clock does not need to be utilized in the FI sub-block.

Table continues on the next page...

Table 42-3. FUSEBOX Signal Overview (continued)

Signal Name	Signal Type	Signal Description
EPM_DATA[7:0]	O	This bus carries the read data from the FUSEBOX during a read cycle.
		State Meaning
		Timing
EPM_PGM_LENGTH [2:0]	I	These signals define the length of the program pulse.
		State Meaning
		Timing
EPM_READY	I	This signal indicates the progress of the current read/program cycle. New commands should only be issued by the FI sub-block while epm_ready is asserted.
		State Meaning Asserted -- The fuse is blown. Negated -- The fuse is not blown.
		Timing Latched at the rise edge of epm_clk half clock cycle ahead of epm_ready
EPM_READ_PROG	I	This signal selects whether the current operation is a read or program cycle. This signal is latched into the FUSEBOX on the rising edge of epm_select.
		State Meaning Assertion- FUSEBOX is in read operation. Negation- FUSEBOX is in program operation.
		Timing Needs to be held to desired value at least 2nS before and after asserting epm_select in both program and read operation.
EPM_READ_SENSE0	I	This signal allows sensing of the 0 (unblown) state to be stressed. This can be useful in detecting marginally blown fuses.
		State Meaning Assertion-When asserted (high), the sense circuitry is configured to stress the sensing of the 0 (unblown) state. Negation- No stress is added.
		Timing This signal is latched by the FUSEBOX on the rising edge of epm_select. It needs to be set to desired value at least 2nS before epm_select is asserted an to be held until epm_select is negated.
EPM_READ_SENSE1	I	This signal allows sensing of the 1 (blown) state to be stressed. This can be useful in detecting marginally blown fuses.
		State Meaning Assertion-When asserted (high), the sense circuitry is configured to stress the sensing of the 1 (blown) state. Negation- No stress is added.
		Timing This signal is latched by the FUSEBOX on the rising edge of epm_select. It needs to be set to desired value at least 2nS before epm_select is asserted and to be held until epm_select is negated.

Table continues on the next page...

Table 42-3. FUSEBOX Signal Overview (continued)

Signal Name	Signal Type	Signal Description
EPM_RESET_B	I	Reset FUSEBOX digital signals during power up. This signal needs to be connected to a early system reset. The signal is active low.
EPM_SCANE	I	This signal indicates whether the FUSEBOX is scannable via JTAG. It shall be always de-asserted because the JTAG function is disabled inside FUSEBOX in cmos065 design.
EPM_SELECT	I	This signal starts a read/program operation and epm_select rising edge needs to be synchronized with the rising edge of epm_clk. There is no restriction for epm_select falling edge to be synchronized with epm_clk. The FUSEBOX senses the rising edge of this signal to start the operation. On the rising edge of epm_select, the address, operation to perform and program data are latched. There is one select signal for each fuse bank.
		State Meaning Assertion- In Read/program operation. Negation- Not in read/program operation.
		Timing This signal must remain asserted throughout the read/program operation.
FUSEBOX_OVERRIDE	O	FUSEBOX override signal to the Functional Interface sub-block. It is connected to VSS by default. It can be re-connected to core VDD via Metal 6 layer change.
		State Meaning Assertion- The FUSEBOX is bypassed. All outputs are set to 0. Negation- The FUSEBOX outputs are sent to FI sub-blocks.
		Timing

42.2.2.1 FUSEBOX Operations

All FUSEBOX operations are self-timed and begin relative to the rising edge on the epm_select line. Most interface signals are shared by all Fuseboxes. Only the FUSEBOX selects (epm_select), read data (epm_data) and ready status (epm_ready) signals are unique to each FUSEBOX. The 32 KHz clock is not used within the IIM, but may be used with other FI sub-blocks. It is passed through to the FUSEBOX to time program operations. The 32 KHz clock distributed to the Fuseboxes should be on during FUSEBOX program operations and should be gated off during read operation. The epm_clk is used to sample the epm_select, epm_ready and epm_data signals in order to synchronize these signals with the Functional Interface inside a FUSEBOX. This epm_clk is a gated clock used by the local FI sub-blocks.

42.2.2.1.1 Word Read

Read operations are done on a word (8-bit) basis. The target word is specified by the address lines epm_addr[9:3]. The bottom three address lines (bit-select lines) are ignored in a read operation.

The timing for fuse word read cycles is shown in the figure below. This timing diagram shows one read cycle followed by the start of a second read, demonstrating how quickly one read can follow another. Each read command is registered on the rising edge of epm_select. The address (epm_addr[9:3]) and read command (epm_read_prog) must be stable tsels before the rising edge of epm_select and must remain stable at least tselh hold time beyond the rising edge. epm_ready is negated immediately after the command is registered, indicating that the FUSEBOX is busy. After a tdv delay from the active edge of epm_select, the data is made available on blr by the analog FUSEBOX. The epm_ready acknowledge signal is asserted tready after the initial assertion of epm_select and the data is available on the epm_data. The FUSEBOX synchronizes epm_ready and the data to the epm_clk clock and drives the synchronized data onto the IP Bus. The time between the valid epm_data and epm_ready is signified by tsync in the figure below and is one-half of one epm_clk cycle. Once the read data has been latched, epm_select is negated by the Functional Interface sub-block. epm_select must remain negated a minimum cycle to cycle delay of tccd before the next cycle (read or program) may begin. epm_data remains at a steady state until replaced by data from a subsequent read cycle.

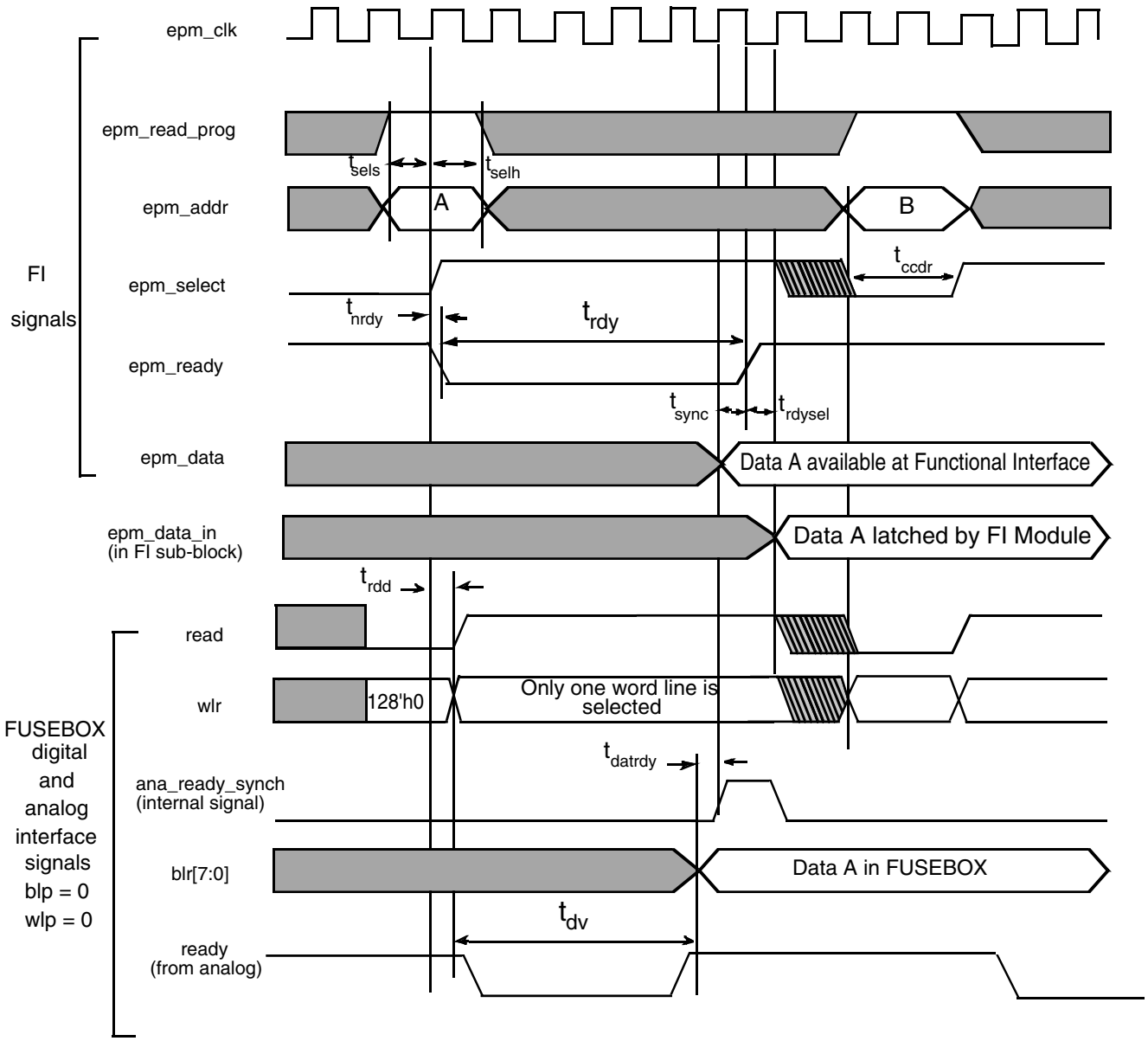


Figure 42-4. FUSEBOX Read Cycle Timing

42.2.2.1.2 Bit Program

Program operations are done on a bit-basis. The target bit is specified by the full address bus (epm_addr[10:0]). Program operations can only blow fuses (change them from logic 0 to logic 1), so there is no program data bus associated with the FUSEBOX. The timing for a program operation is shown in Figure 42-5.

As in the read cycle, the address and program command are latched on the rising edge of epm_select. The address must be setup tsels ahead of the rising edge and remain held tselh past the edge of epm_select. Once the program command is registered, on the next rising edge of the 32 KHz clock, the FUSEBOX initiates the program operation. An internal counter starts to count the number of 32K clock cycles, and the epm_ready signal is de-asserted.

After the specified number of 32 KHz clocks (determined by epm_pgm_length[2:0]), the FUSEBOX completes the program operation and asserts the epm_ready signal. epm_pgm_length[2:0] needs to be set tsels ahead of the rising edge of epm_select and needs to be held until the completion of the internal counter. Consult Table 42-4 for specific timing values. The Functional Interface sub-block synchronizes the epm_ready response to the epm_clk during the period tsync. After recognizing that the program operation has completed, the Functional Interface sub-block negates the epm_select line. To avoid accidental selection of more than one bit line at a time, the 8 bit de-coded bit-line-program (BLP) needs to be checked before generate blp[7:0].

Table 42-4. Timing Values

Designator	Timing Value	Description
t_{sels}	2 nS	Setup time to epm_select active edge
t_{selh}	2 nS	Hold time after epm_select active edge
t_{nrdy}	200 pS -1000 pS	Delay from epm_select to epm_ready negation (read). Delay from the rise edge of 32 Khz clock to epm_ready (program).
t_{rdd}	1 nS-4 nS	Assert time from epm_select active edge to negation of read from digital
t_{dv}	30 nS-100 nS	Data valid after epm_select active edge
t_{rdy}	< (t_{dv} + 1.5 epm_clk)	Epm_ready signal asserted after epm_select active edge (read only)
t_{ready}	(epm_pgm_length) * $period_{32KHz}$	Epm_ready signal asserted after epm_select active edge (program only)
t_{datrdy}	0-1 epm_clk	Data driven out to bus after analog ready signal asserts (read only)
t_{sync}	one-half of 1 epm_clk cycle (rising edge to falling edge) > 3nS if epm_clk duty cycle is not 50%	epm_data to epm_ready synchronization delay (read). epm_ready to negation of epm_select (program)
t_{rdysel}	> 2nS typically one-half of 1 epm_clk cycle (falling edge to rising edge)	rising-edge of epm_ready to negative-edge of epm_select (read only)
t_{ccdr}	5 nS	Minimum cycle to cycle delay (Consecutive Read)
t_{ccdp}	0.5 uS	Minimum cycle to cycle delay (Consecutive program)

Table continues on the next page...

Table 42-4. Timing Values (continued)

Designator	Timing Value	Description
t_{sels}	2 nS	Setup time to epm_select active edge
t_{selh}	2 nS	Hold time after epm_select active edge
t_{nrdy}	200 pS -1000 pS	Delay from epm_select to epm_ready negation (read). Delay from the rise edge of 32 Khz clock to epm_ready (program).

See the timing diagram of the fuse program in the figure below.

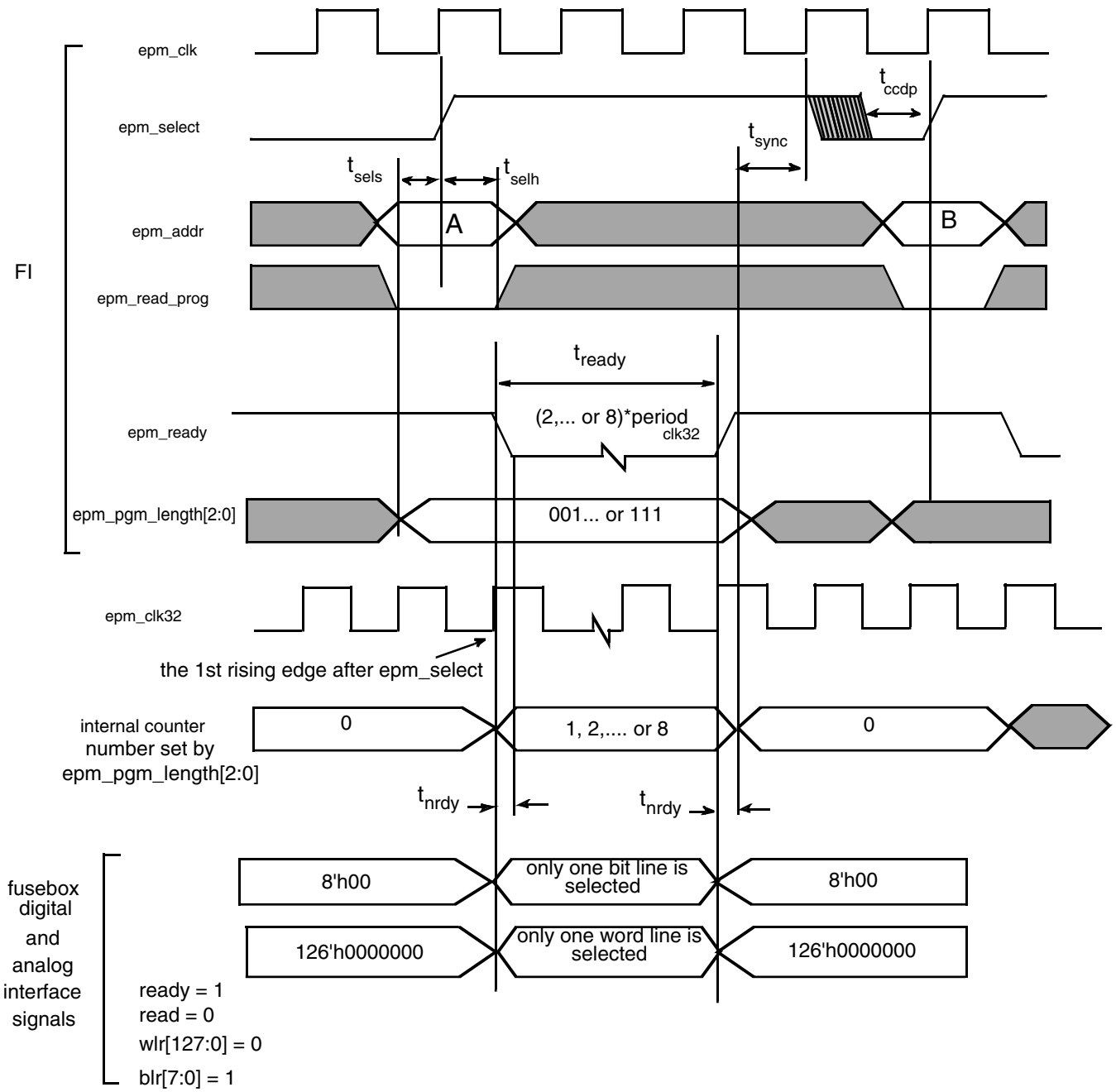


Figure 42-5. Fuse Program Cycle Timing

NOTE

The IIM controller must determine that the target fuse bank is indeed an e-Fuse bank (not a laser fuse bank) before allowing the program cycle to proceed. This is determined by examining the appropriate fuse_type signal, which is fixed at SoC integration time.

42.2.3 Fuse Value Storage

The values of fuses may be read from one of the following three places:

- Software fuse value shadow cache
- Hardware-visible fuse value shadow cache, driving SOC nets
- Fuse elements themselves

The reasons for caching the fuse values are to reduce the risk of accidental programming of e-Fuses due to repeated reads and to reduce power consumption associated with sense cycles.

42.2.3.1 Software Fuse Value Shadow Cache

Each word in the cache RAM includes a valid bit and a parity bit, as shown in the following figure. The valid bit is set if the data bits and parity bit have been set according to the sensed state of the corresponding fuses, Odd parity is used across the data bits only (that is, the valid bit is not included in the parity calculation).

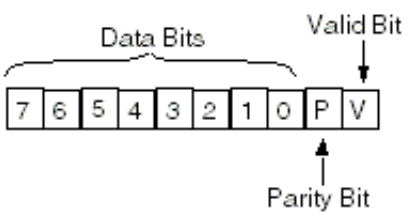


Figure 42-6. Software Cache Word Format

All bits are set to 0 when IIM comes out of reset. When a fuse word is read, the IIM will first attempt to read it from the cache (except for the hardware-visible fuses; see Hardware-Visible Fuse Shadow Cache). A sense cycle will only be run to the fuses if the cached word is invalid or the parity is incorrect. The overall cached fuse word read sequence is shown in Read Sequence. The cache words and fuse words is one to one mapped.

42.2.3.2 Hardware-Visible Fuse Shadow Cache

The hardware-visible fuse include fuses in bank0 and bank1, and FBAC words of each bank.The IIM must sense these fuses and write their values to the appropriate registers when it comes out of reset, and must ensure than any change to the fuses is also immediately reflected in the registers. The overall hardware-visible word read sequence is shown in Read Sequence. Each word has a parity bit as shown below.

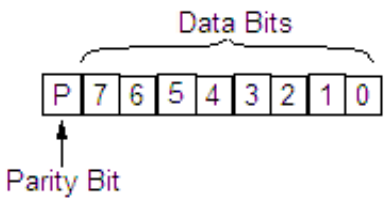


Figure 42-7. Hardware Cache Word Format

If parity error detected on shadow cache, IIM will set error bit PARITYE, and re initialize cache from fuses.

42.2.4 Fuse Protection

The following subsections describe fuses used as protection bits.

42.2.4.1 FUSEBOX Bank Protection Fuse

FBWP of each bank control whether this bank can be programed. FBOP of each bank control whether this bank can be overridden. FBRP of each bank control whether this bank can be read. FBSP control whether this bank can be scanned. FBESP control whether this bank can be explicitly sensed.

42.2.4.2 Scan Out Protection

To prevent the secret keys scanned out by hacker, specific logic is added (shown in the figure below). It resets all flip-flops in IIM before entering scan mode.

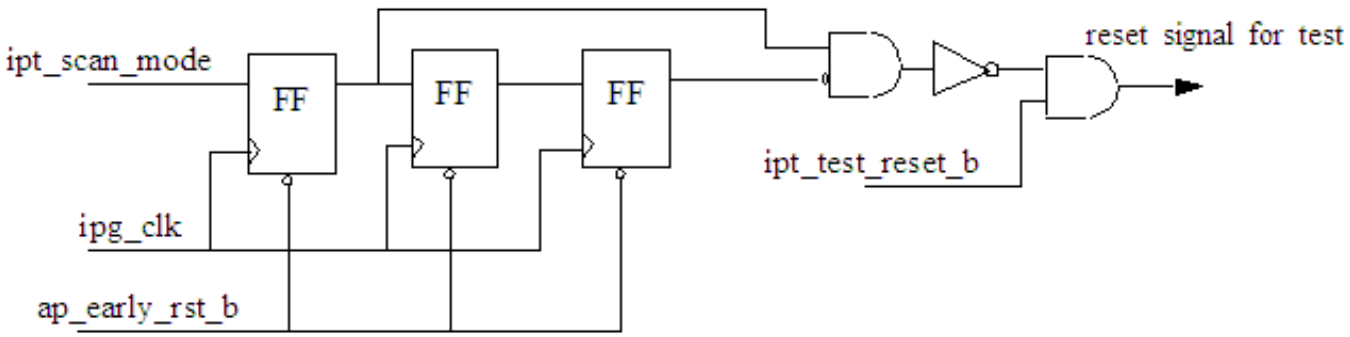


Figure 42-8. Scan Out Protection Circuit

42.2.5 Fuse Bank Operations

This section discusses the following sequences:

- Read Sequence
- Explicit Sense Sequence
- Programming Sequence
- Override Sequence

42.2.5.1 Read Sequence

Fuses may be read using the IP Bus interface from any bank which is not read-inhibited (that is, FBACx[FBRP] is unblown). The read sequence from a hardware-visible signals word is shown below.

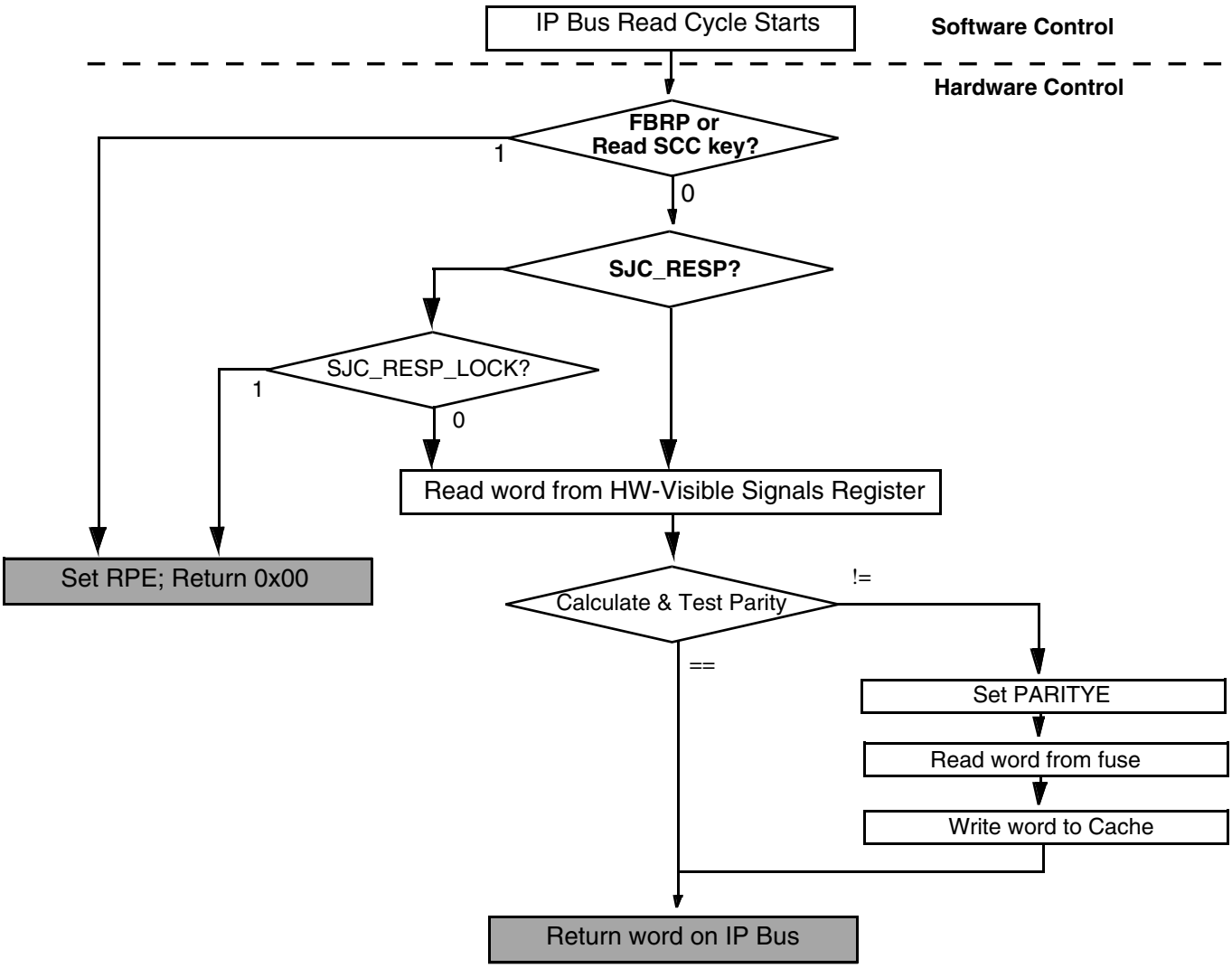


Figure 42-9. Hardware Visible Fuse Read

The read sequence to a cacheable fuse word is shown in the figure below.

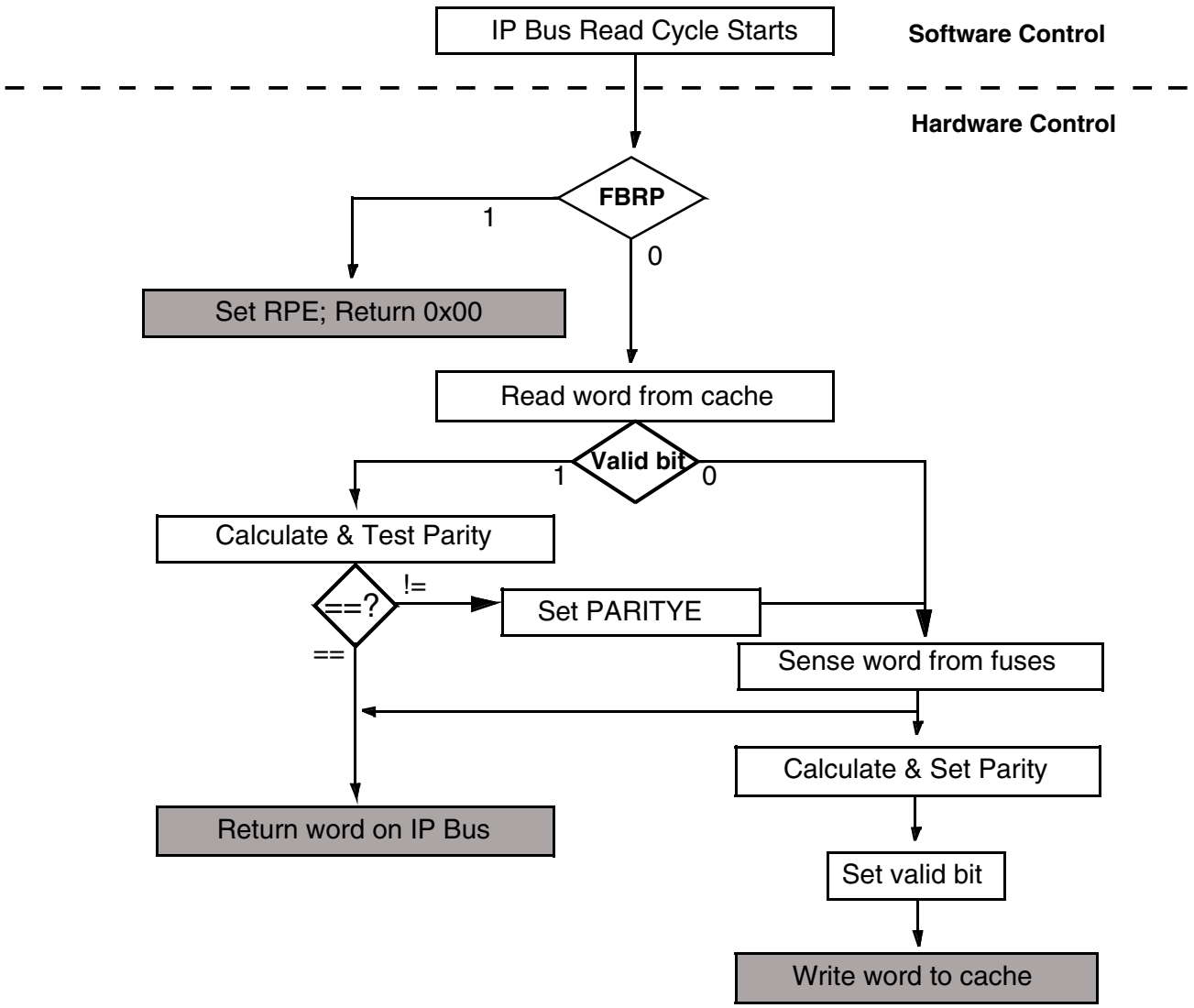


Figure 42-10. Software Fuse Read

42.2.5.2 Explicit Sense Sequence

The explicit sense sequence using IP Bus is depicted in the figure below.

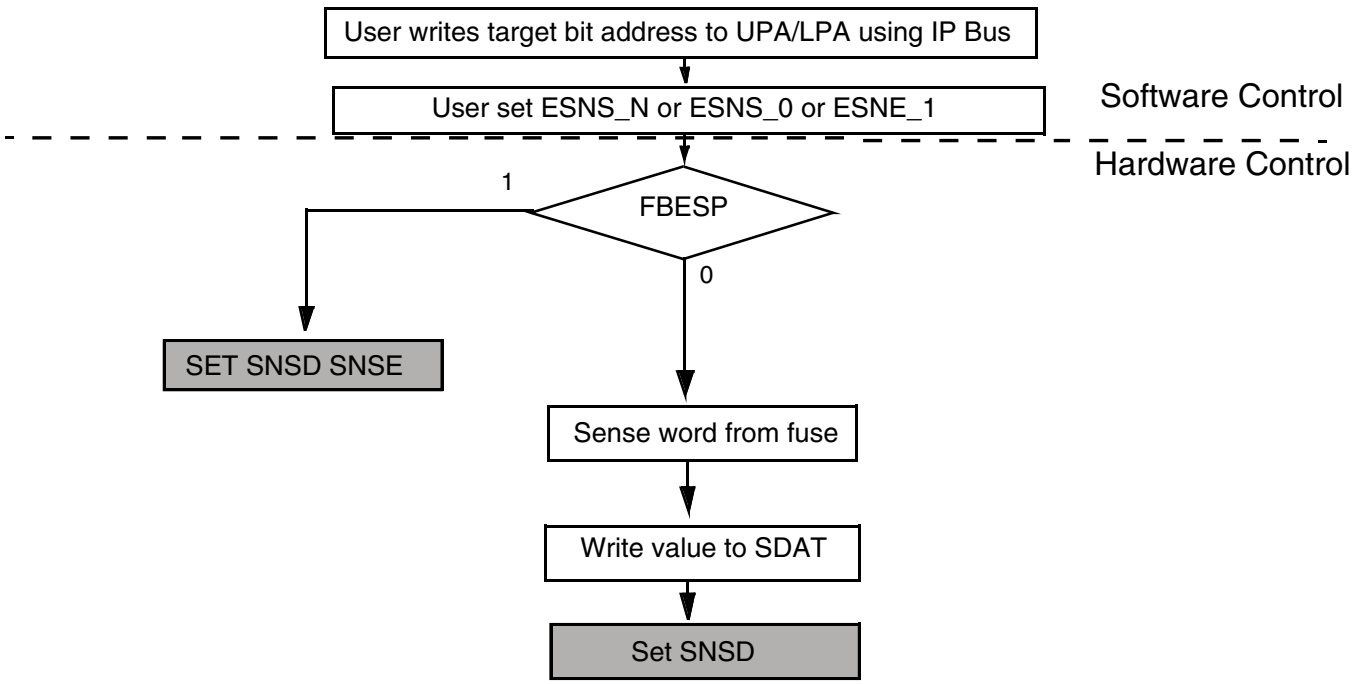


Figure 42-11. Explicit Sense Sequence

42.2.5.3 Programming Sequence

The software-controlled e-Fuse programming sequence using the IP Bus is depicted in the figure below.

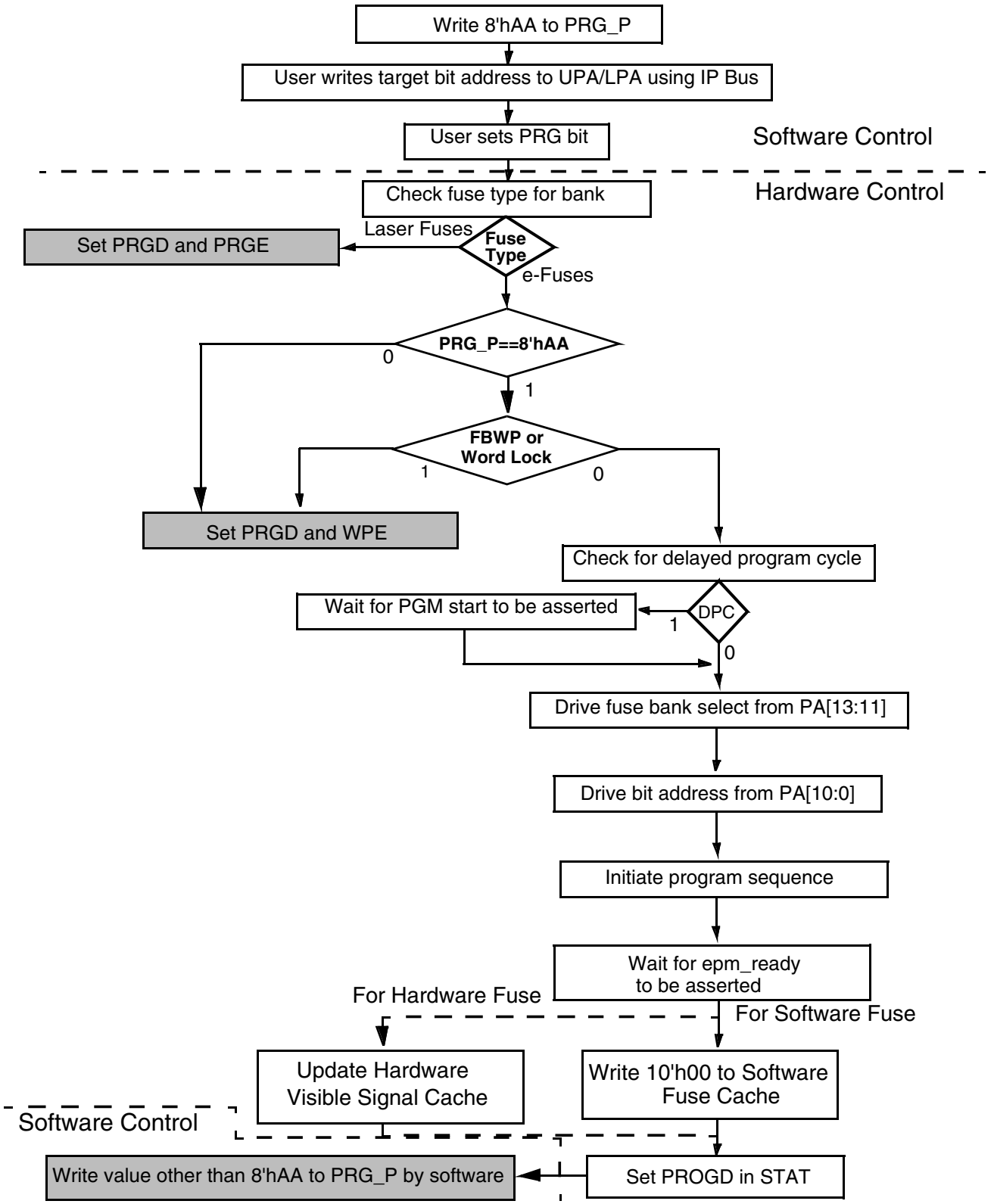


Figure 42-12. e-Fuse Program Sequence

Fuses may also be programmed directly using the JTAG interface, so long as the IIM asserts the appropriate epm_scane signal. The IIM is not involved in this programming sequence aside from allowing or disallowing it using the epm_scane signal.

Note that minimum program voltage is 2.775 V, maximum program voltage is 3.3 V. This must be taken into account when designing the architecture of an IC which includes e-Fuses. For complete parametric specifications, refer to the FUSEBOX specification.

42.2.5.4 Override Sequence

The override sequence is depicted in the following figure. .

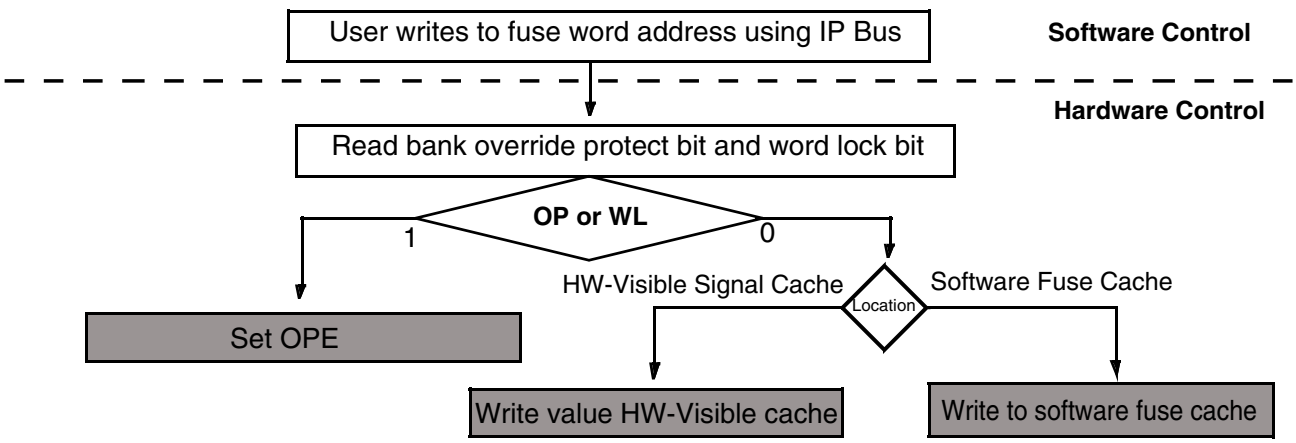


Figure 42-13. Fuse Value Override Sequence

42.3 Initialization/Application Information

42.3.1 Initialization

When the IIM come out of reset, IIM automatically senses the hardware-visible fuses and writes their values into the appropriate registers. IIM also senses the fuse access control word from each of the fuse banks, and set epm_scane according to the values of fbsp bits. Software fuses are not sensed until software has made a read request for them. Software cache will be set to 0 on reset.

After the IIM come out of reset, all the hardware visible fuses must be read out to hardware visible cache within 200us, to ensure the hardware visible fuses are loaded before the second reset at the SoC-level. IIM reads HWV2 first. The initialization time depends on the number of hardware visible fuses.

Initialization time = (number of hardware visible fuse word) x (100 ns + 4 ipg_clk cycles)

After the initialization is complete, fuse_latched will be asserted, as shown in the following figure.

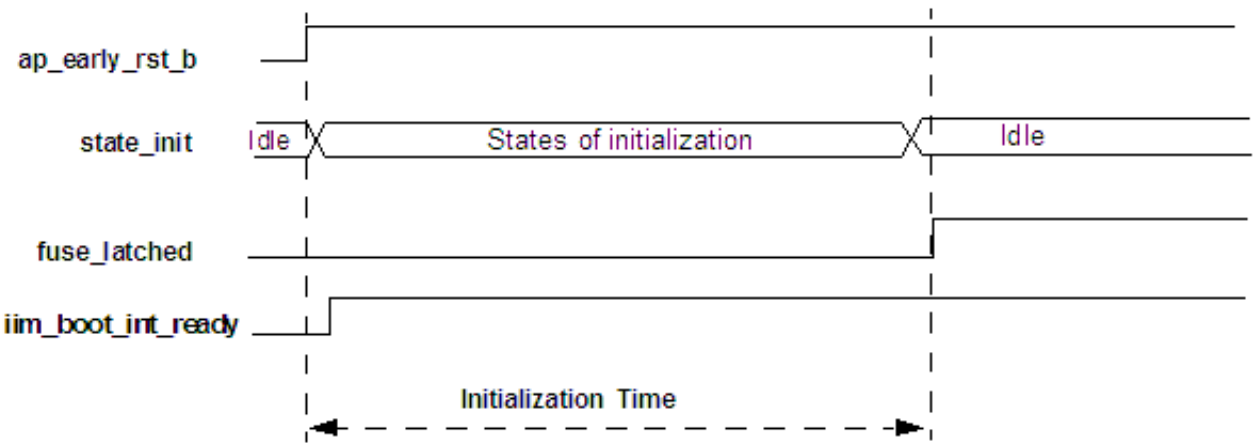


Figure 42-14. Timing Diagram of Initialization

42.3.2 Program

After JATG programming, reset must be applied to synchronize the fuse value to hardware visible signal cache.

A wait period is required between program and read operation.

42.4 Programmable Registers

All IIM registers are 8-bits wide, but addressable on 32-bit boundaries. Only the bottom 8 bits (the usable bits) of each register are shown in the following diagrams. The top 24 bits always read as 0 and writes to them are ignored..

All IIM registers can be accessed in user mode except FCTL, which controls programming functions and can be accessed in supervisor mode only.

IIM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63F9_8000	Status register (IIM_STAT)	8	w1c	00h	42.4.1/ 1844
63F9_8004	Status IRQ Mask register (IIM_STATM)	8	R/W	00h	42.4.2/ 1845
63F9_8008	Module Errors register (IIM_ERR)	8	R/W	See section	42.4.3/ 1846
63F9_800C	Error IRQ Mask register (IIM_EMASK)	8	R/W	See section	42.4.4/ 1847
63F9_8010	Fuse Control register (IIM_FCTL)	8	R/W	30h	42.4.5/ 1848
63F9_8014	Upper Address register (IIM_UA)	8	R/W	00h	42.4.6/ 1849
63F9_8018	Lower Address register (IIM_LA)	8	R/W	00h	42.4.7/ 1850
63F9_801C	Explicit Sense Data register (IIM_SDAT)	8	R	00h	42.4.8/ 1850
63F9_8020	Product Revision register (IIM_PREV)	8	R	Unaffected	42.4.9/ 1851
63F9_8024	Silicon Revision register (IIM_SREV)	8	R	00h	42.4.10/ 1851
63F9_8028	Program Protection register (IIM_PREG_P)	8	R/W	00h	42.4.11/ 1852
63F9_802C	Software-Controllable Signals register 0 (IIM_SCS0)	8	R/W	00h	42.4.12/ 1852
63F9_8034	Software-Controllable Signals register 2 (IIM_SCS2)	8	R/W	00h	42.4.13/ 1853
63F9_8038	Software-Controllable Signals register 3 (IIM_SCS3)	8	R/W	00h	42.4.14/ 1854

42.4.1 Status register (IIM_STAT)

All block status information is read using the STAT register.

Address: IIM_STAT is 63F9_8000h base + 0h offset = 63F9_8000h

Bit	7	6	5	4	3	2	1	0
Read	BUSY	-					PRGD	SNSD
Write							w1c	w1c
Reset	0	0	0	0	0	0	0	0

IIM_STAT field descriptions

Field	Description
7 BUSY	Indicates whether the IIM is busy with a program or sense cycle. Any attempt to access the IIM registers other than STAT while it is busy with a program or sense cycle (BUSY asserted) results in a bus error. 0 The IIM is not busy with a program or sense cycle 1 The IIM is busy with a program or sense cycle
6–2 -	Reserved
1 PRGD	Program Done. Indicates an e-Fuse program operation is done. Assertion causes an interrupt request (irq_b signal asserted) if PRGD_M is set in the Status IRQ Mask Register. This bit is automatically set by hardware upon completion of an e-Fuse program cycle; software must clear the bit by writing 1 to it. 0 Program operation has not finished (read); no meaning (write) 1 Program operation has finished (read); clear bit (write)
0 SNSD	Explicit Sense Cycle Done. Indicates that an explicit fuse sense cycle is done, and the data is available in SDAT. Assertion causes an interrupt request if SNSD_M is set in the Status IRQ Mask Register. This bit is automatically set by hardware and must be cleared by software by writing 1 to it. 0 No explicit sense cycle has finished (read); no meaning (write) 1 An explicit sense cycle has finished (read); clear bit (write)

42.4.2 Status IRQ Mask register (IIM_STATM)

Address: IIM_STATM is 63F9_8000h base + 4h offset = 63F9_8004h

Bit	7	6	5	4	3	2	1	0
Read Write	-						PRGD_M	SNSD_M
Reset	0	0	0	0	0	0	0	0

IIM_STATM field descriptions

Field	Description
7–2 -	Reserved
1 PRGD_M	Program Mask. Masks or unmasks IRQ generation due to PRGD events. 0 PRGD events do not cause an IRQ 1 PRGD events cause an IRQ
0 SNSD_M	Explicitly Sense Cycle Done Mask. Masks or unmasks IRQ generation due to SNSD events. 0 SNSD events do not cause an IRQ 1 SNSD events cause an IRQ

42.4.3 Module Errors register (IIM_ERR)

Address: IIM_ERR is 63F9_8000h base + 8h offset = 63F9_8008h

Bit	7	6	5	4	3	2	1	0
Read Write	PRGE	WPE	OPE	RPE	WLRE	SNSE	PARITYE	-
Reset	0	0	0	0	0	0	0	u*

* Notes:
- u = Unaffected by reset.

IIM_ERR field descriptions

Field	Description
7 PRGE	Program Error. Indicates an e-Fuse program operation ended in failure. Assertion causes an interrupt request if PRGE_M is set in the Errors IRQ Mask Register. A program failure occurs when an attempt is made to program laser fuses. This bit is automatically set by hardware and must be cleared by software by writing 1 to it. 0 Program operation has not finished with error (read); no meaning (write) 1 Program operation has finished with an error (read); clear bit (write)
6 WPE	Write Protect Error. Indicates an e-Fuse program operation was attempted to a write-protected fuse bank, or a locked words, or when the value of PRG_P is not 8'hAA. Assertion causes an interrupt request if WPE_M is set in the Errors IRQ Mask Register. This bit is automatically set by hardware and must be cleared by software by writing 1 to it. 0 There was no write-protect error (read); no meaning (write) 1 There was a write-protect error (read); clear bit (write)
5 OPE	Override Protect Error. Indicates an attempt was made to override the values in an override-protected fuse bank, or a locked words. Assertion causes an interrupt request if OPE_M is set in the Errors IRQ Mask Register. This bit is automatically set by hardware and must be cleared by software by writing 1 to it. 0 There was no override-protect error (read); no meaning (write) 1 There was an override-protect error (read); clear bit (write)
4 RPE	Read Protect Error. Indicates an attempt was made to read values from a read-protected fuse bank or SCC. Assertion causes an interrupt request if RPE_M is set in the Errors IRQ Mask Register. This bit is automatically set by hardware and must be cleared by software by writing 1 to it. 0 There was no read-protect error (read); no meaning (write) 1 There was a read-protect error (read); clear bit (write)
3 WLRE	Write to Locked Register Error. Indicates an attempt was made to write to a locked SCS register. Assertion causes an interrupt request if WLRE_M is set in the Errors IRQ Mask Register. This bit is automatically set by hardware and must be cleared by software by writing 1 to it. 0 There was no write-to-locked-register error (read); no meaning (write) 1 There was a write-to-locked-register error (read); clear bit (write)
2 SNSE	Explicit Sense Cycle Error. Indicates that an explicit fuse sense was refused, because FBESP is set to 1, or more than two bits of SNS_N, SNS_1, SNS_0, PRG are asserted at the same moment. Assertion

Table continues on the next page...

IIM_ERR field descriptions (continued)

Field	Description
	causes an interrupt request if SNSE_M is set in the Errors IRQ Mask Register. This bit is automatically set by hardware and must be cleared by software by writing 1 to it. 0 There was no explicit sense error (read); no meaning (write) 1 There was an explicit sense error (read); clear bit (write)
1 PARITYE	Parity Error of Cache. Indicate that an parity error was detected in hardware fuse cache or software fuse cache. Assertion causes an interrupt request if PARITYE_M is set in the Errors IRQ Mask Register. This bit is automatically set by hardware and must be cleared by software by writing 1 to it. 0 There was no explicit sense error (read); no meaning (write) 1 There was an explicit sense error (read); clear bit (write)
0 -	Reserved

42.4.4 Error IRQ Mask register (IIM_EMASK)

Address: IIM_EMASK is 63F9_8000h base + Ch offset = 63F9_800Ch

Bit	7	6	5	4	3	2	1	0
Read Write	PRGE_M	WPE_M	OPE_M	RPE_M	WLRE_M	SNSE_M	PARITYE_M	-
Reset	0	0	0	0	0	0	0	u*

* Notes:
- u = Unaffected by reset.

IIM_EMASK field descriptions

Field	Description
7 PRGE_M	Program Error Mask. Masks or unmasks IRQ generation due to PRGE events. 0 PRGE events do not cause an IRQ 1 PRGE events cause an IRQ
6 WPE_M	Write Protect Error Mask. Masks or unmasks IRQ generation due to WPE events. 0 WPE events do not cause an IRQ 1 WPE events cause an IRQ
5 OPE_M	Override Protect Error Mask. Masks or unmasks IRQ generation due to OPE events. 0 OPE events do not cause an IRQ 1 OPE events cause an IRQ
4 RPE_M	Read Protect Error Mask. Masks or unmasks IRQ generation due to RPE events. 0 RPE events do not cause an IRQ 1 RPE events cause an IRQ

Table continues on the next page...

IIM_EMASK field descriptions (continued)

Field	Description
3 WLRE_M	Write to Locked Register Error Mask. Masks or unmasks IRQ generation due to WLRE events. 0 WLRE events do not cause an IRQ 1 WLRE events cause an IRQ
2 SNSE_M	Explicit Sense Cycle Error Mask. Masks or unmasks IRQ generation due to SNSE events. 0 SNSE events do not cause an IRQ 1 SNSE events cause an IRQ
1 PARITYE_M	Parity Error of Cache Mask. Masks or unmasks IRQ generation due to PARITYE events. 0 PARITYE events do not cause an IRQ 1 PARITYE events cause an IRQ
0 -	Reserved

42.4.5 Fuse Control register (IIM_FCTL)

Address: IIM_FCTL is 63F9_8000h base + 10h offset = 63F9_8010h

Bit	7	6	5	4	3	2	1	0
Read Write	DPC	PRG_LENGTH[2:0]			ESNS_N	ESNS_0	ESNS_1	PRG
Reset	0	0	1	1	0	0	0	0

IIM_FCTL field descriptions

Field	Description
7 DPC	Delayed Program Cycle. This bit is a control bit, selecting immediate or delayed program. When this bit is cleared, program cycles start immediately upon setting of PRG bit. When this bit is asserted, program cycles is delayed until input signal delayed_pgm_start is asserted. 0 Program cycles begin immediately upon setting of the PRG bit 1 Program cycles are delayed; they do not begin until delayed_pgm_start signal is asserted
6–4 PRG_ LENGTH[2:0]	Program Length. This 3 bits define the length of program pulse PRG_LENGTH* (period of 32k clock) Setting is unknown.
3 ESNS_N	Explicit Sense - Normal. Writing 1 to this bit initiate an unstressed (normal) explicit sense cycle. Read of this bit always returns zero. FSM generate a "done" signal when the operation complete. This bit is cleared automatically by hardware when sense operation completed. Only one of ESNS_N, ESNS_0, ESNS_1, PRG can be asserted, otherwise ESNS_E will be asserted to indicate this error. 0 Return 0 for all read (read); No meaning (write) 1 Initiate an unstressed explicit sense cycle (write)

Table continues on the next page...

IIM_FCTL field descriptions (continued)

Field	Description
2 ESNS_0	Explicit Sense - 0-stressed. Writing 1 to this bit initiate a 0-stressed explicit sense cycle. Read of this bit always returns zero.FSM generate a "done" signal when the operation complete. This bit is cleared automatically by hardware when sense operation completed. During 0-stressed explicit sense cycles, the **epm_readsense0** signal is asserted to the fuse banks. Only one of ESNS_N, ESNS_0, ESNS_1, PRG can be asserted, otherwise ESNS_E will be asserted to indicate this error. 0 Return 0 for all read (read); No meaning (write) 1 Initiate a 0-stressed explicit sense cycle (write)
1 ESNS_1	Explicit Sense - 1-stressed. Writing 1 to this bit initiate a 1-stressed explicit sense cycle. Read of this bit always returns zero. FSM generate a "done" signal when the operation complete. This bit is cleared automatically by hardware when sense operation completed. During 1-stressed explicit sense cycles, the **epm_readsense1** signal is asserted to the fuse banks. Only one of ESNS_N, ESNS_0, ESNS_1, PRG can be asserted, otherwise ESNS_E will be asserted to indicate this error. Only one of ESNS_N, ESNS_0, ESNS_1, PRG can be asserted, otherwise ESNS_E will be asserted to indicate this error. 0 Return 0 for all read (read); No meaning (write) 1 Initiate a 1-stressed explicit sense cycle (write)
0 PRG	Program. Writing 1 to this bit initiate a fuse program cycle. Read of this bit always returns zero. FSM generate a "done" signal when the operation complete. This bit is cleared automatically by hardware when program operation completed. Only one of ESNS_N, ESNS_0, ESNS_1, PRG can be asserted, otherwise ESNS_E will be asserted to indicate this error. 0 Return 0 for all read (read); No meaning (write) 1 Initiate a program cycle (write)

42.4.6 Upper Address register (IIM_UA)

The top part of the address of the e-Fuse bit to be programmed or the word to be sensed in an explicit sense cycle. Note that programming is done by bit, so the program address is a full-bit address. Sensing is done on 8-bit words, so the bottom three bits of the address are ignored.

Address: IIM_UA is 63F9_8000h base + 14h offset = 63F9_8014h

Bit	7	6	5	4	3	2	1	0
Read Write	-				A[13-8]			
Reset	0	0	0	0	0	0	0	0

IIM_UA field descriptions

Field	Description
7–6 -	Reserved

Table continues on the next page...

IIM_UA field descriptions (continued)

Field	Description
5–0 A[13-8]	The top six bits of the address of the e-Fuse bit to be programmed or the word to be sensed explicitly. The address must be written prior to setting the PRG or ESNS_x bit in FCTL to initiate the program/sense operation. A[13-11] select the Fuse bank. A[10-8] provide the most significant portion of the row address within the bank.

42.4.7 Lower Address register (IIM_LA)

The bottom 8 bits of the address of the e-Fuse bit to be programmed or word to be explicitly sensed.

Address: IIM_LA is 63F9_8000h base + 18h offset = 63F9_8018h

Bit	7	6	5	4	3	2	1	0
Read Write	A							
Reset	0	0	0	0	0	0	0	0

IIM_LA field descriptions

Field	Description
7–0 A	The bottom eight bits of the address of the e-Fuse bit to be programmed or word to be sensed explicitly. The address must be written prior to setting the PRG or ESNS_x bit in FCTL to initiate a program or sense operation. A[7:3] provides the least significant portion of the row address. A[2:0] select the bit position within the selected row.

42.4.8 Explicit Sense Data register (IIM_SDAT)

On an explicit sense cycle, the data sensed from the fuses is placed in this register at the conclusion of the sense cycle. Software can recognize the conclusion of the sense cycle by the assertion of SNSD in STAT.

Address: IIM_SDAT is 63F9_8000h base + 1Ch offset = 63F9_801Ch

Bit	7	6	5	4	3	2	1	0
Read	D							
Write								
Reset	0	0	0	0	0	0	0	0

IIM_SDAT field descriptions

Field	Description
7–0 D	The data sensed from the fuses. Setting is unknown.

42.4.9 Product Revision register (IIM_PREV)

The product revision. This corresponds to the top eight bits of the deprecated HW_REV register.

Address: IIM_PREV is 63F9_8000h base + 20h offset = 63F9_8020h

Bit	7	6	5	4	3	2	1	0
Read	PROD_REV					PROD_VT		
Write								
Reset	u*	u*	u*	u*	u*	u*	u*	u*

* Notes:
- u = Unaffected by reset.

IIM_PREV field descriptions

Field	Description
7–3 PROD_REV	Product Revision. The product revision (specific to the product or product family). Setting according to product revision.
2–0 PROD_VT	Product vendor or technology. The product vendor and/or technology (specific to the product or product family). Setting according to product vendor/technology.

42.4.10 Silicon Revision register (IIM_SREV)

The silicon revision (that is, mask revision). This corresponds to the bottom 8 bits of the deprecated HW_REV register.

Address: IIM_SREV is 63F9_8000h base + 24h offset = 63F9_8024h

Bit	7	6	5	4	3	2	1	0
Read	SILICON_REV							
Write								
Reset	0	0	0	0	0	0	0	0

IIM_SREV field descriptions

Field	Description
7–0 SILICON_REV	Mask Set Revision. The mask set revision used in fabrication of the part. The value changes with each change to the mask set. Setting according to mask set revision.

42.4.11 Program Protection register (IIM_PREG_P)

This register is used to against accidental fuse programming. The fuses can be blown only when the value of this register is 8'hAA. Software should only program this register to 8'hAA while actively blowing fuses. After the program operation is complete, this register should be immediately reprogrammed to a different value.

Address: IIM_PREG_P is 63F9_8000h base + 28h offset = 63F9_8028h

Bit	7	6	5	4	3	2	1	0
Read Write	PROTECTION_REG							
Reset	0	0	0	0	0	0	0	0

IIM_PREG_P field descriptions

Field	Description
7–0 PROTECTION_REG	The fuses can be blown only when the value of this register is 8'hAA. Any attempt to program the fuse while the value is other than 8'hAA will be terminated with error. WPE bit will be asserted. Setting is undefined.

42.4.12 Software-Controllable Signals register 0 (IIM_SCS0)

This register is physically located in the hardware-visible signals sub-block. It implements software-controlled, volatile signals which can drive SoC-level nets for feature enabling.

Address: IIM_SCS0 is 63F9_8000h base + 2Ch offset = 63F9_802Ch

Bit	7	6	5	4	3	2	1	0
Read	LOCK	HAB_JDE	SCS[26:21]					
Write								
Reset	0	0	0	0	0	0	0	0

IIM_SCS0 field descriptions

Field	Description
7 LOCK	Lock this register. This bit is used to lock the contents of this register until the next reset. The intended usage is to have trusted software program the register as desired and lock it before allowing distrusted software to run. This bit is write only, read of this bit return 0. 0 The register is not locked, it may be modified 1 The register is locked, all attempts to modify it are ignored
6 HAB_JDE	HAB JTAG Debug Enable. This bit is used by the HAB to enable JTAG debugging, assuming that a properly signed command to do so is found and validated by the HAB. The HAB musts lock the register before passing control to the OS whether or not JTAG debugging has been enabled. Once JTAG is enabled by HAB_JDE bit, it can not be disabled unless the system is reset by POR. 0 JTAG debugging is not enabled by the HAB (it may still be enabled by other mechanisms) 1 JTAG debugging is enabled by the HAB (though this signal may be gated off)
5–0 SCS[26:21]	These bits may be used in an implementation-defined way.

42.4.13 Software-Controllable Signals register 2 (IIM_SCS2)

SCS2 and SCS3 registers are physically located in the hardware-visible signals sub-block. They implement software-controlled, volatile signals that can drive SoC-level nets for feature enabling.

Each IIM fuse bank include a read protection bit to inhibit software fuses read and sense and to support secure computing environments. When a fuse bank's read protection bit is set (i.e.'1'), the bank's fuses cannot be read or sensed by software. The read protection bit cannot be changed if the sticky lock bit was asserted.

This is the fuse bank read lock register. This corresponds to five i.MX53 fuse banks.

Address: IIM_SCS2 is 63F9_8000h base + 34h offset = 63F9_8034h

Bit	7	6	5	4	3	2	1	0
Read	LOCK	-		FBRL4	FBRL3	FBRL2	FBRL1	FBRL0
Write								
Reset	0	0	0	0	0	0	0	0

IIM_SCS2 field descriptions

Field	Description
7 LOCK	Lock this register. 0 The register is not locked, it may be modified 1 The register is locked, all attempts to modify it are ignored

Table continues on the next page...

IIM_SCS2 field descriptions (continued)

Field	Description
6–5 -	Reserved
4 FBRL4	Read lock for fuse bank #4
3 FBRL3	Read lock for fuse bank #3
2 FBRL2	Read lock for fuse bank #2
1 FBRL1	Read lock for fuse bank #1
0 FBRL0	Read lock for fuse bank #0

42.4.14 Software-Controllable Signals register 3 (IIM_SCS3)

SCS2 and SCS3 registers are physically located in the hardware-visible signals sub-block. They implement software-controlled, volatile signals that can drive SoC-level nets for feature enabling.

Each IIM fuse bank include a read protection bit to inhibit software fuses read and sense and to support secure computing environments. When a fuse bank's read protection bit is set (i.e.'1'), the bank's fuses cannot be read or sensed by software. The read protection bit cannot be changed if the sticky lock bit was asserted.

This is the fuse bank write lock register. This corresponds to the five i.MX53 fuse banks.

Address: IIM_SCS3 is 63F9_8000h base + 38h offset = 63F9_8038h

Bit	7	6	5	4	3	2	1	0
Read Write	LOCK	-		FBWL4	FBWL3	FBWL2	FBWL1	FBWL0
Reset	0	0	0	0	0	0	0	0

IIM_SCS3 field descriptions

Field	Description
7 LOCK	Lock this register. 0 The register is not locked, it may be modified 1 The register is locked, all attempts to modify it are ignored
6–5 -	Reserved

Table continues on the next page...

IIM_SCS3 field descriptions (continued)

Field	Description
4 FBWL4	Write lock for fuse bank #4
3 FBWL3	Write lock for fuse bank #3
2 FBWL2	Write lock for fuse bank #2
1 FBWL1	Write lock for fuse bank #1
0 FBWL0	Write lock for fuse bank #0

Chapter 43
IOMUX Controller (IOMUXC)

43.1 Introduction

43.1.1 Overview

The IOMUX Controller (IOMUXC), together with the IOMUX, enables the SoC to share one pad with several functional blocks. This is done by multiplexing the pad input/output signals. For each pad there are up to 8 muxing options (called ALT modes). Since different blocks require different pad settings (like pull up, keeper, etc.), the IOMUXC also controls the pad setting parameters.

The IOMUXC consists of only combinatorial logic combined from several basic IOMUXC cells. Each basic IOMUXC cell handles the muxing of only one pad, and each pad connects to one package I/O, called a pin.

Figure below illustrates the IOMUX/IOMUXC connectivity in the system.

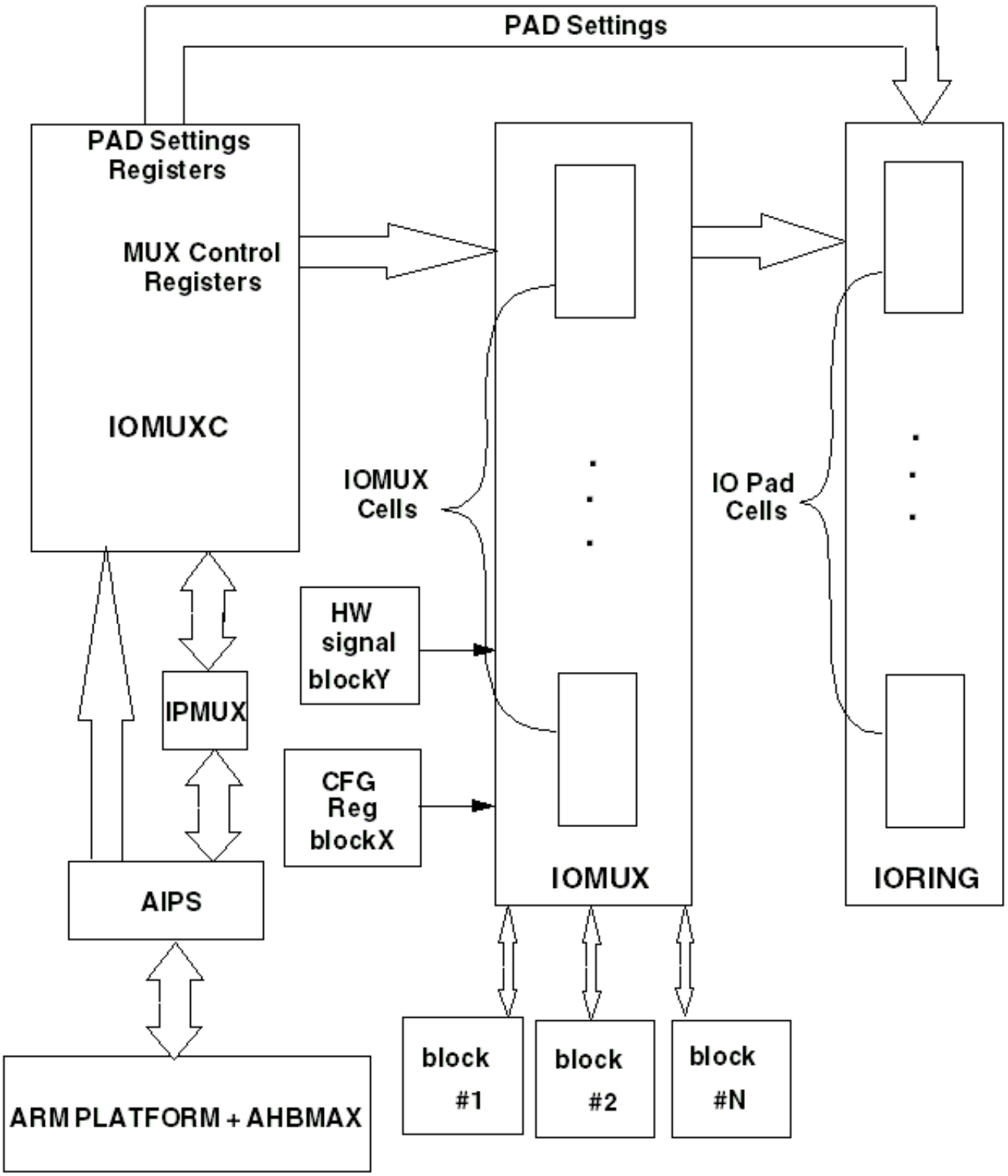


Figure 43-1. IOMUX SoC-Level Block Diagram

43.1.2 Features

The IOMUXC features are:

- 32-bit General Purpose Registers - Two 32 bits registers according to SoC requirements for any usage
- 32-bit SW Mux control registers (IOMUXC_SW_MUX_CTL_PAD_<PAD NAME> or IOMUXC_SW_MUX_CTL_GRP_<GROUP NAME>) to configure a specific mux mode for each pad or a predefined group of pads and to enable forcing the input path of the pad/s (SION bit).
- 32-bit SW Pad control registers (IOMUXC_SW_PAD_CTL_PAD_<PAD_NAME> or IOMUXC_SW_PAD_CTL_GRP_<GROUP NAME>) to configure specific pad settings for each pad or a predefined group of pads.
- 5 32-bit OBSERVE_MUX registers (IOMUXC_OBSERVE_MUX_0-4)
- 32-bit Daisy Chain Control registers - register to control the input path to a block when more than one pad may drive this block input.
- IP Bus Interface - IP peripheral bus to allow registers configuration and readiness (all registers are read/write registers).

Each SW MUX/PAD CTL IOMUXC register handles only one pad or one group of pads.

Only the minimum number of registers required by SW will be implemented by HW. For example, if only ALT0 and ALT1 modes are used on Pad X, then only one bit register will be generated as the MUX_MODE control field in the SW Mux control register of Pad X.

The SW Mux control registers may allow forcing the pads to become input (input path enabled), regardless of the functional direction driven. This may be useful for loopback and GPIO data capture.

43.1.3 Modes of Operation

There is only one mode of operation - normal mode.

43.2 External Signal Description

There are no signals in the IOMUXC that connect off chip (they all control the pads functionality).

43.3 Programmable Registers

There are four main groups of registers;

1. The General Purpose Registers (GPR) - used to select operating modes for general features in the SoC, usually not related to the IOMUXC itself.

2. The Observability Registers - can be used for debugging in order to reflect internal signals values.

3. The Software MUX Control Registers - used to configure the IOMUXC multiplexing, and "connect" the pad to a given port in a block.

4. The pad settings Registers - Used to control the pad settings configuration. For some pads (in order to save chip route) the pad settings are grouped in one register, i.e., changing the group register will affect the settings for all pads in the group.

NOTE

The pad settings for SDCLK_0_B and SDCLK_1_B are identical with the SDCLK_0 and SDCLK_1 respectively.

Table 43-2. DDR Output Driver Average Impedance

Parameter	Symbol	Test Conditions	Drive strength (DSE)								Unit
			000	001	010	011	100	101	110	111	
Output Driver Impedance[1]	Rdrv2[2]	LPDDR1/DDR2 mode NVCC_DRAM = 1.8 V DDR_SEL = 00 Calibration resistance = 300 Ω[3]	Hi-Z[4]	300	150	100	75	60	50	43	Ω
		DDR2 mode NVCC_DRAM = 1.8 V DDR_SEL = 01 Calibration resistance = 180 Ω[3]	Hi-Z[4]	180	90	60	45	36	30	26	
		DDR2 mode NVCC_DRAM = 1.8 V DDR_SEL = 10 Calibration resistance = 200 Ω[3]	Hi-Z[4]	200	100	66	50	40	33	28	
		DDR2 mode NVCC_DRAM= 1.8 V DDR_SEL = 11 Calibration resistance = 140 Ω[3]	Hi-Z[4]	140	70	46	35	28	23	20	
		LPDDR2 mode NVCC_DRAM= 1.2 V DDR_SEL = 01[5] Calibration resistance = 160 Ω[3]	Hi-Z[4]	160	80	53	40	32	27	23	
		LPDDR2 mode NVCC_DRAM = 1.2 V DDR_SEL = 10 Calibration resistance = 240 Ω[3]	Hi-Z[4]	240	120	80	60	48	40	34	
		LPDDR2 mode NVCC_DRAM = 1.2 V DDR_SEL = 11 Calibration resistance = 160 Ω[3]	Hi-Z[4]	160	80	53	40	32	27	23	
		DDR3 mode NVCC_DRAM = 1.5 V DDR_SEL = 00 Calibration resistance = 200 Ω[3]	Hi-Z[4]	240	120	80	60	48	48	34	

1. Output driver impedance is controlled across PVTs (process, voltages, and temperatures) using calibration procedure and pu_*cal, pd_*cal input pins.
2. Output driver impedance deviation (calibration accuracy) is ±5% (max/min impedance) across PVTs.
3. Calibration is done against external reference resistor. Value of the resistor should be varied depending on DDR mode and DDR_SEL setting.
4. Disabled.

5. If DDR_SEL = 01 or DDR_SEL = 11 are selected with NVCC_DRAM = 1.2 V for LPDDR2 operation, the external reference resistor value must be 160 Ω for a correct ZQ calibration. In any case, reference resistors attached to the DDR memory devices should be kept to 240 Ω per the JEDEC standard.

IOMUXC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_8000	General Purpose Register 0 (IOMUXC_GPR0)	32	R/W	0000_0000h	43.3.1/ 1890
53FA_8004	General Purpose Register 1 (IOMUXC_GPR1)	32	R/W	0019_0005h	43.3.2/ 1893
53FA_8008	General Purpose Register 2 (IOMUXC_GPR2)	32	R/W	0000_0000h	43.3.3/ 1894
53FA_800C	OBSERVE_MUX 0 Register (IOMUXC_OMUX0)	32	R/W	0000_0000h	43.3.4/ 1896
53FA_8010	OBSERVE_MUX 1 Register 1 (IOMUXC_OMUX1)	32	R/W	0000_0000h	43.3.5/ 1898
53FA_8014	OBSERVE_MUX 2 Register (IOMUXC_OMUX2)	32	R/W	0000_0000h	43.3.6/ 1900
53FA_8018	OBSERVE_MUX 3 Register (IOMUXC_OMUX3)	32	R/W	0000_0000h	43.3.7/ 1901
53FA_801C	OBSERVE_MUX 4 Register (IOMUXC_OMUX4)	32	R/W	0000_0000h	43.3.8/ 1902
53FA_8020	IOMUXC_SW_MUX_CTL_PAD_GPIO_19 (IOMUXC_GPIO_19)	32	R/W	0000_0001h	43.3.9/ 1904
53FA_8024	IOMUXC_SW_MUX_CTL_PAD_KEY_COL0 (IOMUXC_KEY_COL0)	32	R/W	0000_0001h	43.3.10/ 1905
53FA_8028	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW0 (IOMUXC_KEY_ROW0)	32	R/W	0000_0001h	43.3.11/ 1906
53FA_802C	IOMUXC_SW_MUX_CTL_PAD_KEY_COL1 (IOMUXC_KEY_COL1)	32	R/W	0000_0001h	43.3.12/ 1906
53FA_8030	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW1 (IOMUXC_KEY_ROW1)	32	R/W	0000_0001h	43.3.13/ 1907
53FA_8034	IOMUXC_SW_MUX_CTL_PAD_KEY_COL2 (IOMUXC_KEY_COL2)	32	R/W	0000_0001h	43.3.14/ 1908
53FA_8038	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW2 (IOMUXC_KEY_ROW2)	32	R/W	0000_0001h	43.3.15/ 1909
53FA_803C	IOMUXC_SW_MUX_CTL_PAD_KEY_COL3 (IOMUXC_KEY_COL3)	32	R/W	0000_0001h	43.3.16/ 1910
53FA_8040	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW3 (IOMUXC_KEY_ROW3)	32	R/W	0000_0001h	43.3.17/ 1911
53FA_8044	IOMUXC_SW_MUX_CTL_PAD_KEY_COL4 (IOMUXC_KEY_COL4)	32	R/W	0000_0001h	43.3.18/ 1912
53FA_8048	IOMUXC_SW_MUX_CTL_PAD_KEY_ROW4 (IOMUXC_KEY_ROW4)	32	R/W	0000_0001h	43.3.19/ 1912

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_804C	IOMUXC_SW_MUX_CTL_PAD_DI0_DISP_CLK (IOMUXC_DI0_DISP_CLK)	32	R/W	0000_0001h	43.3.20/ 1913
53FA_8050	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN15 (IOMUXC_DI0_PIN15)	32	R/W	0000_0001h	43.3.21/ 1914
53FA_8054	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN2 (IOMUXC_DI0_PIN2)	32	R/W	0000_0001h	43.3.22/ 1915
53FA_8058	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN3 (IOMUXC_DI0_PIN3)	32	R/W	0000_0001h	43.3.23/ 1915
53FA_805C	IOMUXC_SW_MUX_CTL_PAD_DI0_PIN4 (IOMUXC_DI0_PIN4)	32	R/W	0000_0001h	43.3.24/ 1916
53FA_8060	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT0 (IOMUXC_DISP0_DAT0)	32	R/W	0000_0001h	43.3.25/ 1917
53FA_8064	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT1 (IOMUXC_DISP0_DAT1)	32	R/W	0000_0001h	43.3.26/ 1918
53FA_8068	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT2 (IOMUXC_DISP0_DAT2)	32	R/W	0000_0001h	43.3.27/ 1918
53FA_806C	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT3 (IOMUXC_DISP0_DAT3)	32	R/W	0000_0001h	43.3.28/ 1919
53FA_8070	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT4 (IOMUXC_DISP0_DAT4)	32	R/W	0000_0001h	43.3.29/ 1920
53FA_8074	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT5 (IOMUXC_DISP0_DAT5)	32	R/W	0000_0001h	43.3.30/ 1921
53FA_8078	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT6 (IOMUXC_DISP0_DAT6)	32	R/W	0000_0001h	43.3.31/ 1921
53FA_807C	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT7 (IOMUXC_DISP0_DAT7)	32	R/W	0000_0001h	43.3.32/ 1922
53FA_8080	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT8 (IOMUXC_DISP0_DAT8)	32	R/W	0000_0001h	43.3.33/ 1923
53FA_8084	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT9 (IOMUXC_DISP0_DAT9)	32	R/W	0000_0001h	43.3.34/ 1924
53FA_8088	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT10 (IOMUXC_DISP0_DAT10)	32	R/W	0000_0001h	43.3.35/ 1924
53FA_808C	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT11 (IOMUXC_DISP0_DAT11)	32	R/W	0000_0001h	43.3.36/ 1925
53FA_8090	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT12 (IOMUXC_DISP0_DAT12)	32	R/W	0000_0001h	43.3.37/ 1926
53FA_8094	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT13 (IOMUXC_DISP0_DAT13)	32	R/W	0000_0001h	43.3.38/ 1927
53FA_8098	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT14 (IOMUXC_DISP0_DAT14)	32	R/W	0000_0001h	43.3.39/ 1927

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_809C	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT15 (IOMUXC_DISP0_DAT15)	32	R/W	0000_0001h	43.3.40/ 1928
53FA_80A0	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT16 (IOMUXC_DISP0_DAT16)	32	R/W	0000_0001h	43.3.41/ 1929
53FA_80A4	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT17 (IOMUXC_DISP0_DAT17)	32	R/W	0000_0001h	43.3.42/ 1930
53FA_80A8	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT18 (IOMUXC_DISP0_DAT18)	32	R/W	0000_0001h	43.3.43/ 1931
53FA_80AC	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT19 (IOMUXC_DISP0_DAT19)	32	R/W	0000_0001h	43.3.44/ 1932
53FA_80B0	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT20 (IOMUXC_DISP0_DAT20)	32	R/W	0000_0001h	43.3.45/ 1933
53FA_80B4	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT21 (IOMUXC_DISP0_DAT21)	32	R/W	0000_0001h	43.3.46/ 1933
53FA_80B8	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT22 (IOMUXC_DISP0_DAT22)	32	R/W	0000_0001h	43.3.47/ 1934
53FA_80BC	IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT23 (IOMUXC_DISP0_DAT23)	32	R/W	0000_0001h	43.3.48/ 1935
53FA_80C0	IOMUXC_SW_MUX_CTL_PAD_CSI0_PIXCLK (IOMUXC_CSI0_PIXCLK)	32	R/W	0000_0001h	43.3.49/ 1936
53FA_80C4	IOMUXC_SW_MUX_CTL_PAD_CSI0_MCLK (IOMUXC_CSI0_MCLK)	32	R/W	0000_0001h	43.3.50/ 1937
53FA_80C8	IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA_EN (IOMUXC_CSI0_DATA_EN)	32	R/W	0000_0001h	43.3.51/ 1937
53FA_80CC	IOMUXC_SW_MUX_CTL_PAD_CSI0_VSYNC (IOMUXC_CSI0_VSYNC)	32	R/W	0000_0001h	43.3.52/ 1938
53FA_80D0	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT4 (IOMUXC_CSI0_DAT4)	32	R/W	0000_0001h	43.3.53/ 1939
53FA_80D4	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT5 (IOMUXC_CSI0_DAT5)	32	R/W	0000_0001h	43.3.54/ 1939
53FA_80D8	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT6 (IOMUXC_CSI0_DAT6)	32	R/W	0000_0001h	43.3.55/ 1940
53FA_80DC	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT7 (IOMUXC_CSI0_DAT7)	32	R/W	0000_0001h	43.3.56/ 1941
53FA_80E0	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT8 (IOMUXC_CSI0_DAT8)	32	R/W	0000_0001h	43.3.57/ 1942
53FA_80E4	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT9 (IOMUXC_CSI0_DAT9)	32	R/W	0000_0001h	43.3.58/ 1943
53FA_80E8	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT10 (IOMUXC_CSI0_DAT10)	32	R/W	0000_0001h	43.3.59/ 1944

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_80EC	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT11 (IOMUXC_CSI0_DAT11)	32	R/W	0000_0001h	43.3.60/ 1944
53FA_80F0	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT12 (IOMUXC_CSI0_DAT12)	32	R/W	0000_0001h	43.3.61/ 1945
53FA_80F4	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT13 (IOMUXC_CSI0_DAT13)	32	R/W	0000_0001h	43.3.62/ 1946
53FA_80F8	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT14 (IOMUXC_CSI0_DAT14)	32	R/W	0000_0001h	43.3.63/ 1947
53FA_80FC	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT15 (IOMUXC_CSI0_DAT15)	32	R/W	0000_0001h	43.3.64/ 1948
53FA_8100	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT16 (IOMUXC_CSI0_DAT16)	32	R/W	0000_0001h	43.3.65/ 1948
53FA_8104	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT17 (IOMUXC_CSI0_DAT17)	32	R/W	0000_0001h	43.3.66/ 1949
53FA_8108	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT18 (IOMUXC_CSI0_DAT18)	32	R/W	0000_0001h	43.3.67/ 1950
53FA_810C	IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT19 (IOMUXC_CSI0_DAT19)	32	R/W	0000_0001h	43.3.68/ 1951
53FA_8110	IOMUXC_SW_MUX_CTL_PAD_EIM_A25 (IOMUXC_EIM_A25)	32	R/W	0000_0000h	43.3.69/ 1951
53FA_8114	IOMUXC_SW_MUX_CTL_PAD_EIM_EB2 (IOMUXC_EIM_EB2)	32	R/W	0000_0001h	43.3.70/ 1952
53FA_8118	IOMUXC_SW_MUX_CTL_PAD_EIM_D16 (IOMUXC_EIM_D16)	32	R/W	0000_0001h	43.3.71/ 1953
53FA_811C	IOMUXC_SW_MUX_CTL_PAD_EIM_D17 (IOMUXC_EIM_D17)	32	R/W	0000_0001h	43.3.72/ 1954
53FA_8120	IOMUXC_SW_MUX_CTL_PAD_EIM_D18 (IOMUXC_EIM_D18)	32	R/W	0000_0001h	43.3.73/ 1954
53FA_8124	IOMUXC_SW_MUX_CTL_PAD_EIM_D19 (IOMUXC_EIM_D19)	32	R/W	0000_0001h	43.3.74/ 1955
53FA_8128	IOMUXC_SW_MUX_CTL_PAD_EIM_D20 (IOMUXC_EIM_D20)	32	R/W	0000_0001h	43.3.75/ 1956
53FA_812C	IOMUXC_SW_MUX_CTL_PAD_EIM_D21 (IOMUXC_EIM_D21)	32	R/W	0000_0001h	43.3.76/ 1957
53FA_8130	IOMUXC_SW_MUX_CTL_PAD_EIM_D22 (IOMUXC_EIM_D22)	32	R/W	0000_0001h	43.3.77/ 1958
53FA_8134	IOMUXC_SW_MUX_CTL_PAD_EIM_D23 (IOMUXC_EIM_D23)	32	R/W	0000_0001h	43.3.78/ 1958
53FA_8138	IOMUXC_SW_MUX_CTL_PAD_EIM_EB3 (IOMUXC_EIM_EB3)	32	R/W	0000_0001h	43.3.79/ 1959

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_813C	IOMUXC_SW_MUX_CTL_PAD_EIM_D24 (IOMUXC_EIM_D24)	32	R/W	0000_0001h	43.3.80/ 1960
53FA_8140	IOMUXC_SW_MUX_CTL_PAD_EIM_D25 (IOMUXC_EIM_D25)	32	R/W	0000_0001h	43.3.81/ 1961
53FA_8144	IOMUXC_SW_MUX_CTL_PAD_EIM_D26 (IOMUXC_EIM_D26)	32	R/W	0000_0001h	43.3.82/ 1962
53FA_8148	IOMUXC_SW_MUX_CTL_PAD_EIM_D27 (IOMUXC_EIM_D27)	32	R/W	0000_0001h	43.3.83/ 1962
53FA_814C	IOMUXC_SW_MUX_CTL_PAD_EIM_D28 (IOMUXC_EIM_D28)	32	R/W	0000_0001h	43.3.84/ 1963
53FA_8150	IOMUXC_SW_MUX_CTL_PAD_EIM_D29 (IOMUXC_EIM_D29)	32	R/W	0000_0001h	43.3.85/ 1964
53FA_8154	IOMUXC_SW_MUX_CTL_PAD_EIM_D30 (IOMUXC_EIM_D30)	32	R/W	0000_0001h	43.3.86/ 1965
53FA_8158	IOMUXC_SW_MUX_CTL_PAD_EIM_D31 (IOMUXC_EIM_D31)	32	R/W	0000_0001h	43.3.87/ 1966
53FA_815C	IOMUXC_SW_MUX_CTL_PAD_EIM_A24 (IOMUXC_EIM_A24)	32	R/W	0000_0000h	43.3.88/ 1966
53FA_8160	IOMUXC_SW_MUX_CTL_PAD_EIM_A23 (IOMUXC_EIM_A23)	32	R/W	0000_0000h	43.3.89/ 1967
53FA_8164	IOMUXC_SW_MUX_CTL_PAD_EIM_A22 (IOMUXC_EIM_A22)	32	R/W	0000_0000h	43.3.90/ 1968
53FA_8168	IOMUXC_SW_MUX_CTL_PAD_EIM_A21 (IOMUXC_EIM_A21)	32	R/W	0000_0000h	43.3.91/ 1968
53FA_816C	IOMUXC_SW_MUX_CTL_PAD_EIM_A20 (IOMUXC_EIM_A20)	32	R/W	0000_0000h	43.3.92/ 1969
53FA_8170	IOMUXC_SW_MUX_CTL_PAD_EIM_A19 (IOMUXC_EIM_A19)	32	R/W	0000_0000h	43.3.93/ 1970
53FA_8174	IOMUXC_SW_MUX_CTL_PAD_EIM_A18 (IOMUXC_EIM_A18)	32	R/W	0000_0000h	43.3.94/ 1970
53FA_8178	IOMUXC_SW_MUX_CTL_PAD_EIM_A17 (IOMUXC_EIM_A17)	32	R/W	0000_0000h	43.3.95/ 1971
53FA_817C	IOMUXC_SW_MUX_CTL_PAD_EIM_A16 (IOMUXC_EIM_A16)	32	R/W	0000_0000h	43.3.96/ 1972
53FA_8180	IOMUXC_SW_MUX_CTL_PAD_EIM_CS0 (IOMUXC_EIM_CS0)	32	R/W	0000_0000h	43.3.97/ 1972
53FA_8184	IOMUXC_SW_MUX_CTL_PAD_EIM_CS1 (IOMUXC_EIM_CS1)	32	R/W	0000_0000h	43.3.98/ 1973
53FA_8188	IOMUXC_SW_MUX_CTL_PAD_EIM_OE (IOMUXC_EIM_OE)	32	R/W	0000_0000h	43.3.99/ 1974

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_818C	IOMUXC_SW_MUX_CTL_PAD_EIM_RW (IOMUXC_EIM_RW)	32	R/W	0000_0000h	43.3.100/ 1974
53FA_8190	IOMUXC_SW_MUX_CTL_PAD_EIM_LBA (IOMUXC_EIM_LBA)	32	R/W	0000_0000h	43.3.101/ 1975
53FA_8194	IOMUXC_SW_MUX_CTL_PAD_EIM_EB0 (IOMUXC_EIM_EB0)	32	R/W	0000_0000h	43.3.102/ 1976
53FA_8198	IOMUXC_SW_MUX_CTL_PAD_EIM_EB1 (IOMUXC_EIM_EB1)	32	R/W	0000_0000h	43.3.103/ 1977
53FA_819C	IOMUXC_SW_MUX_CTL_PAD_EIM_DA0 (IOMUXC_EIM_DA0)	32	R/W	0000_0000h	43.3.104/ 1977
53FA_81A0	IOMUXC_SW_MUX_CTL_PAD_EIM_DA1 (IOMUXC_EIM_DA1)	32	R/W	0000_0000h	43.3.105/ 1978
53FA_81A4	IOMUXC_SW_MUX_CTL_PAD_EIM_DA2 (IOMUXC_EIM_DA2)	32	R/W	0000_0000h	43.3.106/ 1979
53FA_81A8	IOMUXC_SW_MUX_CTL_PAD_EIM_DA3 (IOMUXC_EIM_DA3)	32	R/W	0000_0000h	43.3.107/ 1979
53FA_81AC	IOMUXC_SW_MUX_CTL_PAD_EIM_DA4 (IOMUXC_EIM_DA4)	32	R/W	0000_0000h	43.3.108/ 1980
53FA_81B0	IOMUXC_SW_MUX_CTL_PAD_EIM_DA5 (IOMUXC_EIM_DA5)	32	R/W	0000_0000h	43.3.109/ 1981
53FA_81B4	IOMUXC_SW_MUX_CTL_PAD_EIM_DA6 (IOMUXC_EIM_DA6)	32	R/W	0000_0000h	43.3.110/ 1981
53FA_81B8	IOMUXC_SW_MUX_CTL_PAD_EIM_DA7 (IOMUXC_EIM_DA7)	32	R/W	0000_0000h	43.3.111/ 1982
53FA_81BC	IOMUXC_SW_MUX_CTL_PAD_EIM_DA8 (IOMUXC_EIM_DA8)	32	R/W	0000_0000h	43.3.112/ 1983
53FA_81C0	IOMUXC_SW_MUX_CTL_PAD_EIM_DA9 (IOMUXC_EIM_DA9)	32	R/W	0000_0000h	43.3.113/ 1983
53FA_81C4	IOMUXC_SW_MUX_CTL_PAD_EIM_DA10 (IOMUXC_EIM_DA10)	32	R/W	0000_0000h	43.3.114/ 1984
53FA_81C8	IOMUXC_SW_MUX_CTL_PAD_EIM_DA11 (IOMUXC_EIM_DA11)	32	R/W	0000_0000h	43.3.115/ 1985
53FA_81CC	IOMUXC_SW_MUX_CTL_PAD_EIM_DA12 (IOMUXC_EIM_DA12)	32	R/W	0000_0000h	43.3.116/ 1985
53FA_81D0	IOMUXC_SW_MUX_CTL_PAD_EIM_DA13 (IOMUXC_EIM_DA13)	32	R/W	0000_0000h	43.3.117/ 1986
53FA_81D4	IOMUXC_SW_MUX_CTL_PAD_EIM_DA14 (IOMUXC_EIM_DA14)	32	R/W	0000_0000h	43.3.118/ 1987
53FA_81D8	IOMUXC_SW_MUX_CTL_PAD_EIM_DA15 (IOMUXC_EIM_DA15)	32	R/W	0000_0000h	43.3.119/ 1988

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_81DC	IOMUXC_SW_MUX_CTL_PAD_NANDF_WE_B (IOMUXC_NANDF_WE_B)	32	R/W	0000_0001h	43.3.120/ 1988
53FA_81E0	IOMUXC_SW_MUX_CTL_PAD_NANDF_RE_B (IOMUXC_NANDF_RE_B)	32	R/W	0000_0001h	43.3.121/ 1989
53FA_81E4	IOMUXC_SW_MUX_CTL_PAD_EIM_WAIT (IOMUXC_EIM_WAIT)	32	R/W	0000_0001h	43.3.122/ 1990
53FA_81E8	IOMUXC_SW_MUX_CTL_PAD_EIM_BCLK (IOMUXC_EIM_BCLK)	32	R/W	0000_0000h	43.3.123/ 1990
53FA_81EC	IOMUXC_SW_MUX_CTL_PAD_LVDS1_TX3_P (IOMUXC_LVDS1_TX3_P)	32	R/W	0000_0000h	43.3.124/ 1991
53FA_81F0	IOMUXC_SW_MUX_CTL_PAD_LVDS1_TX2_P (IOMUXC_LVDS1_TX2_P)	32	R/W	0000_0000h	43.3.125/ 1992
53FA_81F4	IOMUXC_SW_MUX_CTL_PAD_LVDS1_CLK_P (IOMUXC_LVDS1_CLK_P)	32	R/W	0000_0000h	43.3.126/ 1992
53FA_81F8	IOMUXC_SW_MUX_CTL_PAD_LVDS1_TX1_P (IOMUXC_LVDS1_TX1_P)	32	R/W	0000_0000h	43.3.127/ 1993
53FA_81FC	IOMUXC_SW_MUX_CTL_PAD_LVDS1_TX0_P (IOMUXC_LVDS1_TX0_P)	32	R/W	0000_0000h	43.3.128/ 1994
53FA_8200	IOMUXC_SW_MUX_CTL_PAD_LVDS0_TX3_P (IOMUXC_LVDS0_TX3_P)	32	R/W	0000_0000h	43.3.129/ 1994
53FA_8204	IOMUXC_SW_MUX_CTL_PAD_LVDS0_CLK_P (IOMUXC_LVDS0_CLK_P)	32	R/W	0000_0000h	43.3.130/ 1995
53FA_8208	IOMUXC_SW_MUX_CTL_PAD_LVDS0_TX2_P (IOMUXC_LVDS0_TX2_P)	32	R/W	0000_0000h	43.3.131/ 1996
53FA_820C	IOMUXC_SW_MUX_CTL_PAD_LVDS0_TX1_P (IOMUXC_LVDS0_TX1_P)	32	R/W	0000_0000h	43.3.132/ 1996
53FA_8210	IOMUXC_SW_MUX_CTL_PAD_LVDS0_TX0_P (IOMUXC_LVDS0_TX0_P)	32	R/W	0000_0000h	43.3.133/ 1997
53FA_8214	IOMUXC_SW_MUX_CTL_PAD_GPIO_10 (IOMUXC_GPIO_10)	32	R/W	0000_0000h	43.3.134/ 1998
53FA_8218	IOMUXC_SW_MUX_CTL_PAD_GPIO_11 (IOMUXC_GPIO_11)	32	R/W	0000_0000h	43.3.135/ 1998
53FA_821C	IOMUXC_SW_MUX_CTL_PAD_GPIO_12 (IOMUXC_GPIO_12)	32	R/W	0000_0000h	43.3.136/ 1999
53FA_8220	IOMUXC_SW_MUX_CTL_PAD_GPIO_13 (IOMUXC_GPIO_13)	32	R/W	0000_0000h	43.3.137/ 1999
53FA_8224	IOMUXC_SW_MUX_CTL_PAD_GPIO_14 (IOMUXC_GPIO_14)	32	R/W	0000_0000h	43.3.138/ 2000
53FA_8228	IOMUXC_SW_MUX_CTL_PAD_NANDF_CLE (IOMUXC_NANDF_CLE)	32	R/W	0000_0001h	43.3.139/ 2000

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_822C	IOMUXC_SW_MUX_CTL_PAD_NANDF_ALE (IOMUXC_NANDF_ALE)	32	R/W	0000_0001h	43.3.140/ 2001
53FA_8230	IOMUXC_SW_MUX_CTL_PAD_NANDF_WP_B (IOMUXC_NANDF_WP_B)	32	R/W	0000_0001h	43.3.141/ 2002
53FA_8234	IOMUXC_SW_MUX_CTL_PAD_NANDF_RB0 (IOMUXC_NANDF_RB0)	32	R/W	0000_0001h	43.3.142/ 2002
53FA_8238	IOMUXC_SW_MUX_CTL_PAD_NANDF_CS0 (IOMUXC_NANDF_CS0)	32	R/W	0000_0001h	43.3.143/ 2003
53FA_823C	IOMUXC_SW_MUX_CTL_PAD_NANDF_CS1 (IOMUXC_NANDF_CS1)	32	R/W	0000_0001h	43.3.144/ 2004
53FA_8240	IOMUXC_SW_MUX_CTL_PAD_NANDF_CS2 (IOMUXC_NANDF_CS2)	32	R/W	0000_0001h	43.3.145/ 2004
53FA_8244	IOMUXC_SW_MUX_CTL_PAD_NANDF_CS3 (IOMUXC_NANDF_CS3)	32	R/W	0000_0001h	43.3.146/ 2005
53FA_8248	IOMUXC_SW_MUX_CTL_PAD_FEC_MDIO (IOMUXC_FEC_MDIO)	32	R/W	0000_0001h	43.3.147/ 2006
53FA_824C	IOMUXC_SW_MUX_CTL_PAD_FEC_REF_CLK (IOMUXC_FEC_REF_CLK)	32	R/W	0000_0001h	43.3.148/ 2007
53FA_8250	IOMUXC_SW_MUX_CTL_PAD_FEC_RX_ER (IOMUXC_FEC_RX_ER)	32	R/W	0000_0001h	43.3.149/ 2008
53FA_8254	IOMUXC_SW_MUX_CTL_PAD_FEC_CRS_DV (IOMUXC_FEC_CRS_DV)	32	R/W	0000_0001h	43.3.150/ 2008
53FA_8258	IOMUXC_SW_MUX_CTL_PAD_FEC_RXD1 (IOMUXC_FEC_RXD1)	32	R/W	0000_0001h	43.3.151/ 2009
53FA_825C	IOMUXC_SW_MUX_CTL_PAD_FEC_RXD0 (IOMUXC_FEC_RXD0)	32	R/W	0000_0001h	43.3.152/ 2010
53FA_8260	IOMUXC_SW_MUX_CTL_PAD_FEC_TX_EN (IOMUXC_FEC_TX_EN)	32	R/W	0000_0001h	43.3.153/ 2011
53FA_8264	IOMUXC_SW_MUX_CTL_PAD_FEC_TXD1 (IOMUXC_FEC_TXD1)	32	R/W	0000_0001h	43.3.154/ 2011
53FA_8268	IOMUXC_SW_MUX_CTL_PAD_FEC_TXD0 (IOMUXC_FEC_TXD0)	32	R/W	0000_0001h	43.3.155/ 2012
53FA_826C	IOMUXC_SW_MUX_CTL_PAD_FEC_MDC (IOMUXC_FEC_MDC)	32	R/W	0000_0001h	43.3.156/ 2013
53FA_8270	IOMUXC_SW_MUX_CTL_PAD_PATA_DIOW (IOMUXC_PATA_DIOW)	32	R/W	0000_0001h	43.3.157/ 2014
53FA_8274	IOMUXC_SW_MUX_CTL_PAD_PATA_DMACK (IOMUXC_PATA_DMACK)	32	R/W	0000_0001h	43.3.158/ 2014
53FA_8278	IOMUXC_SW_MUX_CTL_PAD_PATA_DMARQ (IOMUXC_PATA_DMARQ)	32	R/W	0000_0001h	43.3.159/ 2015

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_827C	IOMUXC_SW_MUX_CTL_PAD_PATA_BUFFER_EN (IOMUXC_PATA_BUFFER_EN)	32	R/W	0000_0001h	43.3.160/ 2016
53FA_8280	IOMUXC_SW_MUX_CTL_PAD_PATA_INTRQ (IOMUXC_PATA_INTRQ)	32	R/W	0000_0001h	43.3.161/ 2017
53FA_8284	IOMUXC_SW_MUX_CTL_PAD_PATA_DIOR (IOMUXC_PATA_DIOR)	32	R/W	0000_0001h	43.3.162/ 2017
53FA_8288	IOMUXC_SW_MUX_CTL_PAD_PATA_RESET_B (IOMUXC_PATA_RESET_B)	32	R/W	0000_0001h	43.3.163/ 2018
53FA_828C	IOMUXC_SW_MUX_CTL_PAD_PATA_IORDY (IOMUXC_PATA_IORDY)	32	R/W	0000_0001h	43.3.164/ 2019
53FA_8290	IOMUXC_SW_MUX_CTL_PAD_PATA_DA_0 (IOMUXC_PATA_DA_0)	32	R/W	0000_0001h	43.3.165/ 2020
53FA_8294	IOMUXC_SW_MUX_CTL_PAD_PATA_DA_1 (IOMUXC_PATA_DA_1)	32	R/W	0000_0001h	43.3.166/ 2020
53FA_8298	IOMUXC_SW_MUX_CTL_PAD_PATA_DA_2 (IOMUXC_PATA_DA_2)	32	R/W	0000_0001h	43.3.167/ 2021
53FA_829C	IOMUXC_SW_MUX_CTL_PAD_PATA_CS_0 (IOMUXC_PATA_CS_0)	32	R/W	0000_0001h	43.3.168/ 2022
53FA_829C	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA14 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA14)	32	R/W	0000_0001h	43.3.169/ 2023
53FA_82A0	IOMUXC_SW_MUX_CTL_PAD_PATA_CS_1 (IOMUXC_SW_MUX_CTL_PAD_PATA_CS_1)	32	R/W	0000_0001h	43.3.170/ 2023
53FA_82A4	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA0 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA0)	32	R/W	0000_0001h	43.3.171/ 2024
53FA_82A8	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA1 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA1)	32	R/W	0000_0001h	43.3.172/ 2025
53FA_82AC	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA2 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA2)	32	R/W	0000_0001h	43.3.173/ 2026
53FA_82B0	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA3 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA3)	32	R/W	0000_0001h	43.3.174/ 2026
53FA_82B4	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA4 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA4)	32	R/W	0000_0001h	43.3.175/ 2027
53FA_82B8	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA5 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA5)	32	R/W	0000_0001h	43.3.176/ 2028
53FA_82BC	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA6 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA6)	32	R/W	0000_0001h	43.3.177/ 2029
53FA_82C0	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA7 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA7)	32	R/W	0000_0001h	43.3.178/ 2029
53FA_82C4	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA8 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA8)	32	R/W	0000_0001h	43.3.179/ 2030

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_82C8	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA9 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA9)	32	R/W	0000_0001h	43.3.180/ 2031
53FA_82CC	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA10 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA10)	32	R/W	0000_0001h	43.3.181/ 2032
53FA_82D0	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA11 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA11)	32	R/W	0000_0001h	43.3.182/ 2032
53FA_82D4	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA12 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA12)	32	R/W	0000_0001h	43.3.183/ 2033
53FA_82D8	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA13 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA13)	32	R/W	0000_0001h	43.3.184/ 2034
53FA_82E0	IOMUXC_SW_MUX_CTL_PAD_PATA_DATA15 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA15)	32	R/W	0000_0001h	43.3.185/ 2035
53FA_82E4	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0 (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0)	32	R/W	0000_0001h	43.3.186/ 2035
53FA_82E8	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1 (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1)	32	R/W	0000_0001h	43.3.187/ 2036
53FA_82EC	IOMUXC_SW_MUX_CTL_PAD_SD1_CMD (IOMUXC_SW_MUX_CTL_PAD_SD1_CMD)	32	R/W	0000_0001h	43.3.188/ 2037
53FA_82F0	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2 (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2)	32	R/W	0000_0001h	43.3.189/ 2038
53FA_82F4	IOMUXC_SW_MUX_CTL_PAD_SD1_CLK (IOMUXC_SW_MUX_CTL_PAD_SD1_CLK)	32	R/W	0000_0001h	43.3.190/ 2038
53FA_82F8	IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3 (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3)	32	R/W	0000_0001h	43.3.191/ 2039
53FA_82FC	IOMUXC_SW_MUX_CTL_PAD_SD2_CLK (IOMUXC_SW_MUX_CTL_PAD_SD2_CLK)	32	R/W	0000_0001h	43.3.192/ 2040
53FA_8300	IOMUXC_SW_MUX_CTL_PAD_SD2_CMD (IOMUXC_SW_MUX_CTL_PAD_SD2_CMD)	32	R/W	0000_0001h	43.3.193/ 2041
53FA_8304	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3 (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3)	32	R/W	0000_0001h	43.3.194/ 2041
53FA_8308	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2 (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2)	32	R/W	0000_0001h	43.3.195/ 2042
53FA_830C	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1 (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1)	32	R/W	0000_0001h	43.3.196/ 2043
53FA_8310	IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0 (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0)	32	R/W	0000_0001h	43.3.197/ 2044
53FA_8314	IOMUXC_SW_MUX_CTL_PAD_GPIO_0 (IOMUXC_SW_MUX_CTL_PAD_GPIO_0)	32	R/W	0000_0001h	43.3.198/ 2045
53FA_8318	IOMUXC_SW_MUX_CTL_PAD_GPIO_1 (IOMUXC_SW_MUX_CTL_PAD_GPIO_1)	32	R/W	0000_0001h	43.3.199/ 2045

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_831C	IOMUXC_SW_MUX_CTL_PAD_GPIO_9 (IOMUXC_SW_MUX_CTL_PAD_GPIO_9)	32	R/W	0000_0001h	43.3.200/ 2046
53FA_8320	IOMUXC_SW_MUX_CTL_PAD_GPIO_3 (IOMUXC_SW_MUX_CTL_PAD_GPIO_3)	32	R/W	0000_0001h	43.3.201/ 2047
53FA_8324	IOMUXC_SW_MUX_CTL_PAD_GPIO_6 (IOMUXC_SW_MUX_CTL_PAD_GPIO_6)	32	R/W	0000_0001h	43.3.202/ 2048
53FA_8328	IOMUXC_SW_MUX_CTL_PAD_GPIO_2 (IOMUXC_SW_MUX_CTL_PAD_GPIO_2)	32	R/W	0000_0001h	43.3.203/ 2049
53FA_832C	IOMUXC_SW_MUX_CTL_PAD_GPIO_4 (IOMUXC_SW_MUX_CTL_PAD_GPIO_4)	32	R/W	0000_0001h	43.3.204/ 2050
53FA_8330	IOMUXC_SW_MUX_CTL_PAD_GPIO_5 (IOMUXC_SW_MUX_CTL_PAD_GPIO_5)	32	R/W	0000_0001h	43.3.205/ 2050
53FA_8334	IOMUXC_SW_MUX_CTL_PAD_GPIO_7 (IOMUXC_SW_MUX_CTL_PAD_GPIO_7)	32	R/W	0000_0001h	43.3.206/ 2051
53FA_8338	IOMUXC_SW_MUX_CTL_PAD_GPIO_8 (IOMUXC_SW_MUX_CTL_PAD_GPIO_8)	32	R/W	0000_0001h	43.3.207/ 2052
53FA_833C	IOMUXC_SW_MUX_CTL_PAD_GPIO_16 (IOMUXC_SW_MUX_CTL_PAD_GPIO_16)	32	R/W	0000_0001h	43.3.208/ 2053
53FA_8340	IOMUXC_SW_MUX_CTL_PAD_GPIO_17 (IOMUXC_SW_MUX_CTL_PAD_GPIO_17)	32	R/W	0000_0001h	43.3.209/ 2054
53FA_8344	IOMUXC_SW_MUX_CTL_PAD_GPIO_18 (IOMUXC_SW_MUX_CTL_PAD_GPIO_18)	32	R/W	0000_0001h	43.3.210/ 2055
53FA_8348	IOMUXC_SW_PAD_CTL_PAD_GPIO_19 (IOMUXC_SW_PAD_CTL_PAD_GPIO_19)	32	R/W	0000_01E4h	43.3.211/ 2056
53FA_834C	IOMUXC_SW_PAD_CTL_PAD_KEY_COL0 (IOMUXC_SW_PAD_CTL_PAD_KEY_COL0)	32	R/W	0000_01E4h	43.3.212/ 2058
53FA_8350	IOMUXC_SW_PAD_CTL_PAD_KEY_ROW0 (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW0)	32	R/W	0000_01C4h	43.3.213/ 2060
53FA_8354	IOMUXC_SW_PAD_CTL_PAD_KEY_COL1 (IOMUXC_SW_PAD_CTL_PAD_KEY_COL1)	32	R/W	0000_01E4h	43.3.214/ 2062
53FA_8358	IOMUXC_SW_PAD_CTL_PAD_KEY_ROW1 (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW1)	32	R/W	0000_01E4h	43.3.215/ 2064
53FA_835C	IOMUXC_SW_PAD_CTL_PAD_KEY_COL2 (IOMUXC_SW_PAD_CTL_PAD_KEY_COL2)	32	R/W	0000_01E4h	43.3.216/ 2066
53FA_8360	IOMUXC_SW_PAD_CTL_PAD_KEY_ROW2 (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW2)	32	R/W	0000_01E4h	43.3.217/ 2068
53FA_8364	IOMUXC_SW_PAD_CTL_PAD_KEY_COL3 (IOMUXC_SW_PAD_CTL_PAD_KEY_COL3)	32	R/W	0000_01E4h	43.3.218/ 2070
53FA_8368	IOMUXC_SW_PAD_CTL_PAD_KEY_ROW3 (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW3)	32	R/W	0000_01E4h	43.3.219/ 2072

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_836C	IOMUXC_SW_PAD_CTL_PAD_KEY_COL4 (IOMUXC_SW_PAD_CTL_PAD_KEY_COL4)	32	R/W	0000_01E4h	43.3.220/ 2074
53FA_8370	IOMUXC_SW_PAD_CTL_PAD_KEY_ROW4 (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW4)	32	R/W	0000_01C4h	43.3.221/ 2076
53FA_8374	IOMUXC_SW_PAD_CTL_PAD_NVCC_KEYPAD (IOMUXC_SW_PAD_CTL_PAD_NVCC_KEYPAD)	32	R/W	0002_0000h	43.3.222/ 2078
53FA_8378	IOMUXC_SW_PAD_CTL_PAD_DI0_DISP_CLK (IOMUXC_SW_PAD_CTL_PAD_DI0_DISP_CLK)	32	R/W	0000_05E5h	43.3.223/ 2079
53FA_837C	IOMUXC_SW_PAD_CTL_PAD_DI0_PIN15 (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN15)	32	R/W	0000_05E5h	43.3.224/ 2081
53FA_8380	IOMUXC_SW_PAD_CTL_PAD_DI0_PIN2 (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN2)	32	R/W	0000_05E5h	43.3.225/ 2083
53FA_8384	IOMUXC_SW_PAD_CTL_PAD_DI0_PIN3 (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN3)	32	R/W	0000_05E5h	43.3.226/ 2085
53FA_8388	IOMUXC_SW_PAD_CTL_PAD_DI0_PIN4 (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN4)	32	R/W	0000_05E5h	43.3.227/ 2087
53FA_838C	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT0 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT0)	32	R/W	0000_05C5h	43.3.228/ 2089
53FA_8390	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT1 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT1)	32	R/W	0000_05C5h	43.3.229/ 2091
53FA_8394	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT2 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT2)	32	R/W	0000_05C5h	43.3.230/ 2093
53FA_8398	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT3 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT3)	32	R/W	0000_05C5h	43.3.231/ 2095
53FA_839C	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT4 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT4)	32	R/W	0000_05C5h	43.3.232/ 2097
53FA_83A0	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT5 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT5)	32	R/W	0000_05C5h	43.3.233/ 2099
53FA_83A4	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT6 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT6)	32	R/W	0000_05C5h	43.3.234/ 2101
53FA_83A8	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT7 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT7)	32	R/W	0000_05C5h	43.3.235/ 2103
53FA_83AC	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT8 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT8)	32	R/W	0000_05E5h	43.3.236/ 2105
53FA_83B0	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT9 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT9)	32	R/W	0000_05E5h	43.3.237/ 2107
53FA_83B4	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT10 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT10)	32	R/W	0000_05E5h	43.3.238/ 2109
53FA_83B8	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT11 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT11)	32	R/W	0000_05C5h	43.3.239/ 2111

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_83BC	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT12 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT12)	32	R/W	0000_05E5h	43.3.240/ 2113
53FA_83C0	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT13 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT13)	32	R/W	0000_05E5h	43.3.241/ 2115
53FA_83C4	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT14 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT14)	32	R/W	0000_05E5h	43.3.242/ 2117
53FA_83C8	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT15 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT15)	32	R/W	0000_05E5h	43.3.243/ 2119
53FA_83CC	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT16 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT16)	32	R/W	0000_05E5h	43.3.244/ 2121
53FA_83D0	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT17 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT17)	32	R/W	0000_05E5h	43.3.245/ 2123
53FA_83D4	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT18 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT18)	32	R/W	0000_05E5h	43.3.246/ 2125
53FA_83D8	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT19 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT19)	32	R/W	0000_05E5h	43.3.247/ 2127
53FA_83DC	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT20 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT20)	32	R/W	0000_05E5h	43.3.248/ 2129
53FA_83E0	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT21 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT21)	32	R/W	0000_05E5h	43.3.249/ 2131
53FA_83E4	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT22 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT22)	32	R/W	0000_05E5h	43.3.250/ 2133
53FA_83E8	IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT23 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT23)	32	R/W	0000_05E5h	43.3.251/ 2135
53FA_83EC	IOMUXC_SW_PAD_CTL_PAD_CSI0_PIXCLK (IOMUXC_SW_PAD_CTL_PAD_CSI0_PIXCLK)	32	R/W	0000_01E4h	43.3.252/ 2137
53FA_83F0	IOMUXC_SW_PAD_CTL_PAD_CSI0_MCLK (IOMUXC_SW_PAD_CTL_PAD_CSI0_MCLK)	32	R/W	0000_01E4h	43.3.253/ 2139
53FA_83F4	IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA_EN (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA_EN)	32	R/W	0000_01E4h	43.3.254/ 2141
53FA_83F8	IOMUXC_SW_PAD_CTL_PAD_CSI0_VSYNC (IOMUXC_SW_PAD_CTL_PAD_CSI0_VSYNC)	32	R/W	0000_01E4h	43.3.255/ 2143
53FA_83FC	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT4 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT4)	32	R/W	0000_01E4h	43.3.256/ 2145
53FA_8400	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT5 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT5)	32	R/W	0000_01C4h	43.3.257/ 2147
53FA_8404	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT6 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT6)	32	R/W	0000_01E4h	43.3.258/ 2149
53FA_8408	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT7 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT7)	32	R/W	0000_01E4h	43.3.259/ 2151

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_840C	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT8 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT8)	32	R/W	0000_01E4h	43.3.260/ 2153
53FA_8410	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT9 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT9)	32	R/W	0000_01C4h	43.3.261/ 2155
53FA_8414	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT10 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT10)	32	R/W	0000_01E4h	43.3.262/ 2157
53FA_8418	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT11 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT11)	32	R/W	0000_01E4h	43.3.263/ 2159
53FA_841C	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT12 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT12)	32	R/W	0000_01C4h	43.3.264/ 2161
53FA_8420	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT13 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT13)	32	R/W	0000_01C4h	43.3.265/ 2163
53FA_8424	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT14 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT14)	32	R/W	0000_01C4h	43.3.266/ 2165
53FA_8428	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT15 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT15)	32	R/W	0000_01C4h	43.3.267/ 2167
53FA_842C	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT16 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT16)	32	R/W	0000_01C4h	43.3.268/ 2169
53FA_8430	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT17 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT17)	32	R/W	0000_01C4h	43.3.269/ 2171
53FA_8434	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT18 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT18)	32	R/W	0000_01C4h	43.3.270/ 2173
53FA_8438	IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT19 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT19)	32	R/W	0000_01C4h	43.3.271/ 2175
53FA_843C	IOMUXC_SW_PAD_CTL_PAD_NVCC_CSI__0 (IOMUXC_SW_PAD_CTL_PAD_NVCC_CSI__0)	32	R/W	0002_0000h	43.3.272/ 2177
53FA_8440	IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS (IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS)	32	R/W	0000_05D0h	43.3.273/ 2178
53FA_8444	IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD (IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD)	32	R/W	0000_04E0h	43.3.274/ 2180
53FA_8448	IOMUXC_SW_PAD_CTL_PAD_JTAG_TRSTB (IOMUXC_SW_PAD_CTL_PAD_JTAG_TRSTB)	32	R/W	0000_04D0h	43.3.275/ 2182
53FA_844C	IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI)	32	R/W	0000_05D0h	43.3.276/ 2184
53FA_8450	IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK (IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK)	32	R/W	0000_05C0h	43.3.277/ 2186
53FA_8454	IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO)	32	R/W	0000_0485h	43.3.278/ 2188
53FA_8458	IOMUXC_SW_PAD_CTL_PAD_EIM_A25 (IOMUXC_SW_PAD_CTL_PAD_EIM_A25)	32	R/W	0000_00E4h	43.3.279/ 2190

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_845C	IOMUXC_SW_PAD_CTL_PAD_EIM_EB2 (IOMUXC_SW_PAD_CTL_PAD_EIM_EB2)	32	R/W	0000_01E4h	43.3.280/ 2192
53FA_8460	IOMUXC_SW_PAD_CTL_PAD_EIM_D16 (IOMUXC_SW_PAD_CTL_PAD_EIM_D16)	32	R/W	0000_01E4h	43.3.281/ 2194
53FA_8464	IOMUXC_SW_PAD_CTL_PAD_EIM_D17 (IOMUXC_SW_PAD_CTL_PAD_EIM_D17)	32	R/W	0000_01E4h	43.3.282/ 2196
53FA_8468	IOMUXC_SW_PAD_CTL_PAD_EIM_D18 (IOMUXC_SW_PAD_CTL_PAD_EIM_D18)	32	R/W	0000_01E4h	43.3.283/ 2198
53FA_846C	IOMUXC_SW_PAD_CTL_PAD_EIM_D19 (IOMUXC_SW_PAD_CTL_PAD_EIM_D19)	32	R/W	0000_01E4h	43.3.284/ 2200
53FA_8470	IOMUXC_SW_PAD_CTL_PAD_EIM_D20 (IOMUXC_SW_PAD_CTL_PAD_EIM_D20)	32	R/W	0000_01E4h	43.3.285/ 2202
53FA_8474	IOMUXC_SW_PAD_CTL_PAD_EIM_D21 (IOMUXC_SW_PAD_CTL_PAD_EIM_D21)	32	R/W	0000_01E4h	43.3.286/ 2204
53FA_8478	IOMUXC_SW_PAD_CTL_PAD_EIM_D22 (IOMUXC_SW_PAD_CTL_PAD_EIM_D22)	32	R/W	0000_01C4h	43.3.287/ 2206
53FA_847C	IOMUXC_SW_PAD_CTL_PAD_EIM_D23 (IOMUXC_SW_PAD_CTL_PAD_EIM_D23)	32	R/W	0000_01E4h	43.3.288/ 2208
53FA_8480	IOMUXC_SW_PAD_CTL_PAD_EIM_EB3 (IOMUXC_SW_PAD_CTL_PAD_EIM_EB3)	32	R/W	0000_01E4h	43.3.289/ 2210
53FA_8484	IOMUXC_SW_PAD_CTL_PAD_EIM_D24 (IOMUXC_SW_PAD_CTL_PAD_EIM_D24)	32	R/W	0000_01E4h	43.3.290/ 2212
53FA_8488	IOMUXC_SW_PAD_CTL_PAD_EIM_D25 (IOMUXC_SW_PAD_CTL_PAD_EIM_D25)	32	R/W	0000_01E4h	43.3.291/ 2214
53FA_848C	IOMUXC_SW_PAD_CTL_PAD_EIM_D26 (IOMUXC_SW_PAD_CTL_PAD_EIM_D26)	32	R/W	0000_01E4h	43.3.292/ 2216
53FA_8490	IOMUXC_SW_PAD_CTL_PAD_EIM_D27 (IOMUXC_SW_PAD_CTL_PAD_EIM_D27)	32	R/W	0000_01E4h	43.3.293/ 2218
53FA_8494	IOMUXC_SW_PAD_CTL_PAD_EIM_D28 (IOMUXC_SW_PAD_CTL_PAD_EIM_D28)	32	R/W	0000_01E4h	43.3.294/ 2220
53FA_8498	IOMUXC_SW_PAD_CTL_PAD_EIM_D29 (IOMUXC_SW_PAD_CTL_PAD_EIM_D29)	32	R/W	0000_01E4h	43.3.295/ 2222
53FA_849C	IOMUXC_SW_PAD_CTL_PAD_EIM_D30 (IOMUXC_SW_PAD_CTL_PAD_EIM_D30)	32	R/W	0000_01E4h	43.3.296/ 2224
53FA_84A0	IOMUXC_SW_PAD_CTL_PAD_EIM_D31 (IOMUXC_SW_PAD_CTL_PAD_EIM_D31)	32	R/W	0000_01C4h	43.3.297/ 2226
53FA_84A4	IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__1 (IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__1)	32	R/W	0002_0000h	43.3.298/ 2228
53FA_84A8	IOMUXC_SW_PAD_CTL_PAD_EIM_A24 (IOMUXC_SW_PAD_CTL_PAD_EIM_A24)	32	R/W	0000_01E4h	43.3.299/ 2229

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_84AC	IOMUXC_SW_PAD_CTL_PAD_EIM_A23 (IOMUXC_SW_PAD_CTL_PAD_EIM_A23)	32	R/W	0000_01E4h	43.3.300/ 2231
53FA_84B0	IOMUXC_SW_PAD_CTL_PAD_EIM_A22 (IOMUXC_SW_PAD_CTL_PAD_EIM_A22)	32	R/W	0000_00E4h	43.3.301/ 2233
53FA_84B4	IOMUXC_SW_PAD_CTL_PAD_EIM_A21 (IOMUXC_SW_PAD_CTL_PAD_EIM_A21)	32	R/W	0000_00E4h	43.3.302/ 2235
53FA_84B8	IOMUXC_SW_PAD_CTL_PAD_EIM_A20 (IOMUXC_SW_PAD_CTL_PAD_EIM_A20)	32	R/W	0000_00E4h	43.3.303/ 2237
53FA_84BC	IOMUXC_SW_PAD_CTL_PAD_EIM_A19 (IOMUXC_SW_PAD_CTL_PAD_EIM_A19)	32	R/W	0000_00E4h	43.3.304/ 2239
53FA_84C0	IOMUXC_SW_PAD_CTL_PAD_EIM_A18 (IOMUXC_SW_PAD_CTL_PAD_EIM_A18)	32	R/W	0000_00E4h	43.3.305/ 2241
53FA_84C4	IOMUXC_SW_PAD_CTL_PAD_EIM_A17 (IOMUXC_SW_PAD_CTL_PAD_EIM_A17)	32	R/W	0000_00E4h	43.3.306/ 2243
53FA_84C8	IOMUXC_SW_PAD_CTL_PAD_EIM_A16 (IOMUXC_SW_PAD_CTL_PAD_EIM_A16)	32	R/W	0000_00E4h	43.3.307/ 2245
53FA_84CC	IOMUXC_SW_PAD_CTL_PAD_EIM_CS0 (IOMUXC_SW_PAD_CTL_PAD_EIM_CS0)	32	R/W	0000_00E4h	43.3.308/ 2247
53FA_84D0	IOMUXC_SW_PAD_CTL_PAD_EIM_CS1 (IOMUXC_SW_PAD_CTL_PAD_EIM_CS1)	32	R/W	0000_00E4h	43.3.309/ 2249
53FA_84D4	IOMUXC_SW_PAD_CTL_PAD_EIM_OE (IOMUXC_SW_PAD_CTL_PAD_EIM_OE)	32	R/W	0000_00E4h	43.3.310/ 2251
53FA_84D8	IOMUXC_SW_PAD_CTL_PAD_EIM_RW (IOMUXC_SW_PAD_CTL_PAD_EIM_RW)	32	R/W	0000_00E4h	43.3.311/ 2253
53FA_84DC	IOMUXC_SW_PAD_CTL_PAD_EIM_LBA (IOMUXC_SW_PAD_CTL_PAD_EIM_LBA)	32	R/W	0000_00E4h	43.3.312/ 2255
53FA_84E0	IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__4 (IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__4)	32	R/W	0002_0000h	43.3.313/ 2257
53FA_84E4	IOMUXC_SW_PAD_CTL_PAD_EIM_EB0 (IOMUXC_SW_PAD_CTL_PAD_EIM_EB0)	32	R/W	0000_00E4h	43.3.314/ 2258
53FA_84E8	IOMUXC_SW_PAD_CTL_PAD_EIM_EB1 (IOMUXC_SW_PAD_CTL_PAD_EIM_EB1)	32	R/W	0000_00E4h	43.3.315/ 2260
53FA_84EC	IOMUXC_SW_PAD_CTL_PAD_EIM_DA0 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA0)	32	R/W	0000_00E4h	43.3.316/ 2262
53FA_84F0	IOMUXC_SW_PAD_CTL_PAD_EIM_DA1 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA1)	32	R/W	0000_00E4h	43.3.317/ 2264
53FA_84F4	IOMUXC_SW_PAD_CTL_PAD_EIM_DA2 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA2)	32	R/W	0000_00E4h	43.3.318/ 2266
53FA_84F8	IOMUXC_SW_PAD_CTL_PAD_EIM_DA3 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA3)	32	R/W	0000_00E4h	43.3.319/ 2268

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_84FC	IOMUXC_SW_PAD_CTL_PAD_EIM_DA4 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA4)	32	R/W	0000_00E4h	43.3.320/ 2270
53FA_8500	IOMUXC_SW_PAD_CTL_PAD_EIM_DA5 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA5)	32	R/W	0000_00E4h	43.3.321/ 2272
53FA_8504	IOMUXC_SW_PAD_CTL_PAD_EIM_DA6 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA6)	32	R/W	0000_00E4h	43.3.322/ 2274
53FA_8508	IOMUXC_SW_PAD_CTL_PAD_EIM_DA7 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA7)	32	R/W	0000_00E4h	43.3.323/ 2276
53FA_850C	IOMUXC_SW_PAD_CTL_PAD_EIM_DA8 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA8)	32	R/W	0000_00E4h	43.3.324/ 2278
53FA_8510	IOMUXC_SW_PAD_CTL_PAD_EIM_DA9 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA9)	32	R/W	0000_00E4h	43.3.325/ 2280
53FA_8514	IOMUXC_SW_PAD_CTL_PAD_EIM_DA10 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA10)	32	R/W	0000_00E4h	43.3.326/ 2282
53FA_8518	IOMUXC_SW_PAD_CTL_PAD_EIM_DA11 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA11)	32	R/W	0000_00E4h	43.3.327/ 2284
53FA_851C	IOMUXC_SW_PAD_CTL_PAD_EIM_DA12 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA12)	32	R/W	0000_00E4h	43.3.328/ 2286
53FA_8520	IOMUXC_SW_PAD_CTL_PAD_EIM_DA13 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA13)	32	R/W	0000_00E4h	43.3.329/ 2288
53FA_8524	IOMUXC_SW_PAD_CTL_PAD_EIM_DA14 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA14)	32	R/W	0000_00E4h	43.3.330/ 2290
53FA_8528	IOMUXC_SW_PAD_CTL_PAD_EIM_DA15 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA15)	32	R/W	0000_00E4h	43.3.331/ 2292
53FA_852C	IOMUXC_SW_PAD_CTL_PAD_NANDF_WE_B (IOMUXC_SW_PAD_CTL_PAD_NANDF_WE_B)	32	R/W	0000_01E4h	43.3.332/ 2294
53FA_8530	IOMUXC_SW_PAD_CTL_PAD_NANDF_RE_B (IOMUXC_SW_PAD_CTL_PAD_NANDF_RE_B)	32	R/W	0000_01E4h	43.3.333/ 2296
53FA_8534	IOMUXC_SW_PAD_CTL_PAD_EIM_WAIT (IOMUXC_SW_PAD_CTL_PAD_EIM_WAIT)	32	R/W	0000_01E0h	43.3.334/ 2298
53FA_8538	IOMUXC_SW_PAD_CTL_PAD_EIM_BCLK (IOMUXC_SW_PAD_CTL_PAD_EIM_BCLK)	32	R/W	0000_00E4h	43.3.335/ 2300
53FA_853C	IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__7 (IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__7)	32	R/W	0002_0000h	43.3.336/ 2302
53FA_8540	IOMUXC_SW_PAD_CTL_PAD_GPIO_10 (IOMUXC_SW_PAD_CTL_PAD_GPIO_10)	32	R/W	0000_05EDh	43.3.337/ 2303
53FA_8544	IOMUXC_SW_PAD_CTL_PAD_GPIO_11 (IOMUXC_SW_PAD_CTL_PAD_GPIO_11)	32	R/W	0000_05EDh	43.3.338/ 2305
53FA_8548	IOMUXC_SW_PAD_CTL_PAD_GPIO_12 (IOMUXC_SW_PAD_CTL_PAD_GPIO_12)	32	R/W	0000_05EDh	43.3.339/ 2307

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_854C	IOMUXC_SW_PAD_CTL_PAD_GPIO_13 (IOMUXC_SW_PAD_CTL_PAD_GPIO_13)	32	R/W	0000_05EDh	43.3.340/ 2309
53FA_8550	IOMUXC_SW_PAD_CTL_PAD_GPIO_14 (IOMUXC_SW_PAD_CTL_PAD_GPIO_14)	32	R/W	0000_05EDh	43.3.341/ 2311
53FA_8554	IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM3 (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM3)	32	R/W	0008_0020h	43.3.342/ 2313
53FA_8558	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS3 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS3)	32	R/W	0048_0040h	43.3.343/ 2315
53FA_855C	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE1 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE1)	32	R/W	0000_00C0h	43.3.344/ 2317
53FA_8560	IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM2 (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM2)	32	R/W	0008_0020h	43.3.345/ 2319
53FA_8564	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT1 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT1)	32	R/W	0008_00C0h	43.3.346/ 2321
53FA_8568	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS2 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS2)	32	R/W	0048_0040h	43.3.347/ 2323
53FA_856C	IOMUXC_SW_PAD_CTL_PAD_DRAM_RESET (IOMUXC_SW_PAD_CTL_PAD_DRAM_RESET)	32	R/W	0008_00C0h	43.3.348/ 2325
53FA_8570	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_1 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_1)	32	R/W	0008_0220h	43.3.349/ 2327
53FA_8574	IOMUXC_SW_PAD_CTL_PAD_DRAM_CAS (IOMUXC_SW_PAD_CTL_PAD_DRAM_CAS)	32	R/W	0008_0220h	43.3.350/ 2329
53FA_8578	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_0 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_0)	32	R/W	0008_0220h	43.3.351/ 2331
53FA_857C	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS0 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS0)	32	R/W	0048_0040h	43.3.352/ 2333
53FA_8580	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT0 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT0)	32	R/W	0008_00C0h	43.3.353/ 2335
53FA_8584	IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM0 (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM0)	32	R/W	0008_0020h	43.3.354/ 2337
53FA_8588	IOMUXC_SW_PAD_CTL_PAD_DRAM_RAS (IOMUXC_SW_PAD_CTL_PAD_DRAM_RAS)	32	R/W	0008_0020h	43.3.355/ 2339
53FA_858C	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE0 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE0)	32	R/W	0000_00C0h	43.3.356/ 2341
53FA_8590	IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS1 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS1)	32	R/W	0048_0040h	43.3.357/ 2343
53FA_8594	IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM1 (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM1)	32	R/W	0008_0040h	43.3.358/ 2345
53FA_8598	IOMUXC_SW_PAD_CTL_PAD_PMIC_ON_REQ (IOMUXC_SW_PAD_CTL_PAD_PMIC_ON_REQ)	32	R/W	0000_0845h	43.3.359/ 2347

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_859C	IOMUXC_SW_PAD_CTL_PAD_PMIC_STBY_REQ (IOMUXC_SW_PAD_CTL_PAD_PMIC_STBY_REQ)	32	R/W	0000_0845h	43.3.360/ 2349
53FA_85A0	IOMUXC_SW_PAD_CTL_PAD_NANDF_CLE (IOMUXC_SW_PAD_CTL_PAD_NANDF_CLE)	32	R/W	0000_01E4h	43.3.361/ 2351
53FA_85A4	IOMUXC_SW_PAD_CTL_PAD_NANDF_ALE (IOMUXC_SW_PAD_CTL_PAD_NANDF_ALE)	32	R/W	0000_01E4h	43.3.362/ 2353
53FA_85A8	IOMUXC_SW_PAD_CTL_PAD_NANDF_WP_B (IOMUXC_SW_PAD_CTL_PAD_NANDF_WP_B)	32	R/W	0000_01E4h	43.3.363/ 2355
53FA_85AC	IOMUXC_SW_PAD_CTL_PAD_NANDF_RB0 (IOMUXC_SW_PAD_CTL_PAD_NANDF_RB0)	32	R/W	0000_01E4h	43.3.364/ 2357
53FA_85B0	IOMUXC_SW_PAD_CTL_PAD_NANDF_CS0 (IOMUXC_SW_PAD_CTL_PAD_NANDF_CS0)	32	R/W	0000_01E4h	43.3.365/ 2359
53FA_85B4	IOMUXC_SW_PAD_CTL_PAD_NANDF_CS1 (IOMUXC_SW_PAD_CTL_PAD_NANDF_CS1)	32	R/W	0000_01E4h	43.3.366/ 2361
53FA_85B8	IOMUXC_SW_PAD_CTL_PAD_NANDF_CS2 (IOMUXC_SW_PAD_CTL_PAD_NANDF_CS2)	32	R/W	0000_01E4h	43.3.367/ 2363
53FA_85BC	IOMUXC_SW_PAD_CTL_PAD_NANDF_CS3 (IOMUXC_SW_PAD_CTL_PAD_NANDF_CS3)	32	R/W	0000_01E4h	43.3.368/ 2365
53FA_85C0	IOMUXC_SW_PAD_CTL_PAD_NVCC_NANDF (IOMUXC_SW_PAD_CTL_PAD_NVCC_NANDF)	32	R/W	0002_0000h	43.3.369/ 2367
53FA_85C4	IOMUXC_SW_PAD_CTL_PAD_FEC_MDIO (IOMUXC_SW_PAD_CTL_PAD_FEC_MDIO)	32	R/W	0000_01E4h	43.3.370/ 2368
53FA_85C8	IOMUXC_SW_PAD_CTL_PAD_FEC_REF_CLK (IOMUXC_SW_PAD_CTL_PAD_FEC_REF_CLK)	32	R/W	0000_01E4h	43.3.371/ 2370
53FA_85CC	IOMUXC_SW_PAD_CTL_PAD_FEC_RX_ER (IOMUXC_SW_PAD_CTL_PAD_FEC_RX_ER)	32	R/W	0000_01E4h	43.3.372/ 2372
53FA_85D0	IOMUXC_SW_PAD_CTL_PAD_FEC_CRS_DV (IOMUXC_SW_PAD_CTL_PAD_FEC_CRS_DV)	32	R/W	0000_01E4h	43.3.373/ 2374
53FA_85D4	IOMUXC_SW_PAD_CTL_PAD_FEC_RXD1 (IOMUXC_SW_PAD_CTL_PAD_FEC_RXD1)	32	R/W	0000_01E4h	43.3.374/ 2376
53FA_85D8	IOMUXC_SW_PAD_CTL_PAD_FEC_RXD0 (IOMUXC_SW_PAD_CTL_PAD_FEC_RXD0)	32	R/W	0000_01E4h	43.3.375/ 2378
53FA_85DC	IOMUXC_SW_PAD_CTL_PAD_FEC_TX_EN (IOMUXC_SW_PAD_CTL_PAD_FEC_TX_EN)	32	R/W	0000_01C4h	43.3.376/ 2380
53FA_85E0	IOMUXC_SW_PAD_CTL_PAD_FEC_TXD1 (IOMUXC_SW_PAD_CTL_PAD_FEC_TXD1)	32	R/W	0000_01E4h	43.3.377/ 2382
53FA_85E4	IOMUXC_SW_PAD_CTL_PAD_FEC_TXD0 (IOMUXC_SW_PAD_CTL_PAD_FEC_TXD0)	32	R/W	0000_01E4h	43.3.378/ 2384
53FA_85E8	IOMUXC_SW_PAD_CTL_PAD_FEC_MDC (IOMUXC_SW_PAD_CTL_PAD_FEC_MDC)	32	R/W	0000_01E4h	43.3.379/ 2386

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_85EC	IOMUXC_SW_PAD_CTL_PAD_NVCC_FEC (IOMUXC_SW_PAD_CTL_PAD_NVCC_FEC)	32	R/W	0002_0000h	43.3.380/ 2388
53FA_85F0	IOMUXC_SW_PAD_CTL_PAD_PATA_DIOW (IOMUXC_SW_PAD_CTL_PAD_PATA_DIOW)	32	R/W	0000_01E4h	43.3.381/ 2389
53FA_85F4	IOMUXC_SW_PAD_CTL_PAD_PATA_DMACK (IOMUXC_SW_PAD_CTL_PAD_PATA_DMACK)	32	R/W	0000_01E4h	43.3.382/ 2391
53FA_85F8	IOMUXC_SW_PAD_CTL_PAD_PATA_DMARQ (IOMUXC_SW_PAD_CTL_PAD_PATA_DMARQ)	32	R/W	0000_01E4h	43.3.383/ 2393
53FA_85FC	IOMUXC_SW_PAD_CTL_PAD_PATA_BUFFER_EN (IOMUXC_SW_PAD_CTL_PAD_PATA_BUFFER_EN)	32	R/W	0000_01E4h	43.3.384/ 2395
53FA_8600	IOMUXC_SW_PAD_CTL_PAD_PATA_INTRQ (IOMUXC_SW_PAD_CTL_PAD_PATA_INTRQ)	32	R/W	0000_01E4h	43.3.385/ 2397
53FA_8604	IOMUXC_SW_PAD_CTL_PAD_PATA_DIOR (IOMUXC_SW_PAD_CTL_PAD_PATA_DIOR)	32	R/W	0000_01E4h	43.3.386/ 2399
53FA_8608	IOMUXC_SW_PAD_CTL_PAD_PATA_RESET_B (IOMUXC_SW_PAD_CTL_PAD_PATA_RESET_B)	32	R/W	0000_01E4h	43.3.387/ 2401
53FA_860C	IOMUXC_SW_PAD_CTL_PAD_PATA_IORDY (IOMUXC_SW_PAD_CTL_PAD_PATA_IORDY)	32	R/W	0000_01E4h	43.3.388/ 2403
53FA_8610	IOMUXC_SW_PAD_CTL_PAD_PATA_DA_0 (IOMUXC_SW_PAD_CTL_PAD_PATA_DA_0)	32	R/W	0000_01E4h	43.3.389/ 2405
53FA_8614	IOMUXC_SW_PAD_CTL_PAD_PATA_DA_1 (IOMUXC_SW_PAD_CTL_PAD_PATA_DA_1)	32	R/W	0000_01E4h	43.3.390/ 2407
53FA_8618	IOMUXC_SW_PAD_CTL_PAD_PATA_DA_2 (IOMUXC_SW_PAD_CTL_PAD_PATA_DA_2)	32	R/W	0000_01E4h	43.3.391/ 2409
53FA_861C	IOMUXC_SW_PAD_CTL_PAD_PATA_CS_0 (IOMUXC_SW_PAD_CTL_PAD_PATA_CS_0)	32	R/W	0000_01E4h	43.3.392/ 2411
53FA_8620	IOMUXC_SW_PAD_CTL_PAD_PATA_CS_1 (IOMUXC_SW_PAD_CTL_PAD_PATA_CS_1)	32	R/W	0000_01E4h	43.3.393/ 2413
53FA_8624	IOMUXC_SW_PAD_CTL_PAD_NVCC_PATA__2 (IOMUXC_SW_PAD_CTL_PAD_NVCC_PATA__2)	32	R/W	0002_0000h	43.3.394/ 2415
53FA_8628	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA0 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA0)	32	R/W	0000_01E4h	43.3.395/ 2416
53FA_862C	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA1 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA1)	32	R/W	0000_01E4h	43.3.396/ 2418
53FA_8630	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA2 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA2)	32	R/W	0000_01E4h	43.3.397/ 2420
53FA_8634	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA3 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA3)	32	R/W	0000_01E4h	43.3.398/ 2422
53FA_8638	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA4 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA4)	32	R/W	0000_01E4h	43.3.399/ 2424

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_863C	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA5 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA5)	32	R/W	0000_01E4h	43.3.400/ 2426
53FA_8640	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA6 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA6)	32	R/W	0000_01E4h	43.3.401/ 2428
53FA_8644	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA7 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA7)	32	R/W	0000_01E4h	43.3.402/ 2430
53FA_8648	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA8 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA8)	32	R/W	0000_01E4h	43.3.403/ 2432
53FA_864C	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA9 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA9)	32	R/W	0000_01E4h	43.3.404/ 2434
53FA_8650	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA10 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA10)	32	R/W	0000_01E4h	43.3.405/ 2436
53FA_8654	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA11 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA11)	32	R/W	0000_01E4h	43.3.406/ 2438
53FA_8658	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA12 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA12)	32	R/W	0000_01E4h	43.3.407/ 2440
53FA_865C	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA13 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA13)	32	R/W	0000_01E4h	43.3.408/ 2442
53FA_8660	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA14 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA14)	32	R/W	0000_01E4h	43.3.409/ 2444
53FA_8664	IOMUXC_SW_PAD_CTL_PAD_PATA_DATA15 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA15)	32	R/W	0000_01E4h	43.3.410/ 2446
53FA_8668	IOMUXC_SW_PAD_CTL_PAD_NVCC_PATA__0 (IOMUXC_SW_PAD_CTL_PAD_NVCC_PATA__0)	32	R/W	0002_0000h	43.3.411/ 2448
53FA_866C	IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0 (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0)	32	R/W	0000_01E4h	43.3.412/ 2449
53FA_8670	IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1 (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1)	32	R/W	0000_01E4h	43.3.413/ 2451
53FA_8674	IOMUXC_SW_PAD_CTL_PAD_SD1_CMD (IOMUXC_SW_PAD_CTL_PAD_SD1_CMD)	32	R/W	0000_01E4h	43.3.414/ 2453
53FA_8678	IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2 (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2)	32	R/W	0000_01E4h	43.3.415/ 2455
53FA_867C	IOMUXC_SW_PAD_CTL_PAD_SD1_CLK (IOMUXC_SW_PAD_CTL_PAD_SD1_CLK)	32	R/W	0000_01E4h	43.3.416/ 2457
53FA_8680	IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3 (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3)	32	R/W	0000_01E4h	43.3.417/ 2459
53FA_8684	IOMUXC_SW_PAD_CTL_PAD_NVCC_SD1 (IOMUXC_SW_PAD_CTL_PAD_NVCC_SD1)	32	R/W	0002_0000h	43.3.418/ 2461
53FA_8688	IOMUXC_SW_PAD_CTL_PAD_SD2_CLK (IOMUXC_SW_PAD_CTL_PAD_SD2_CLK)	32	R/W	0000_01E4h	43.3.419/ 2462

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_868C	IOMUXC_SW_PAD_CTL_PAD_SD2_CMD (IOMUXC_SW_PAD_CTL_PAD_SD2_CMD)	32	R/W	0000_01E4h	43.3.420/ 2464
53FA_8690	IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3 (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3)	32	R/W	0000_01E4h	43.3.421/ 2466
53FA_8694	IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2 (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2)	32	R/W	0000_01E4h	43.3.422/ 2468
53FA_8698	IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1 (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1)	32	R/W	0000_01E4h	43.3.423/ 2470
53FA_869C	IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0 (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0)	32	R/W	0000_01E4h	43.3.424/ 2472
53FA_86A0	IOMUXC_SW_PAD_CTL_PAD_NVCC_SD2 (IOMUXC_SW_PAD_CTL_PAD_NVCC_SD2)	32	R/W	0002_0000h	43.3.425/ 2474
53FA_86A4	IOMUXC_SW_PAD_CTL_PAD_GPIO_0 (IOMUXC_SW_PAD_CTL_PAD_GPIO_0)	32	R/W	0000_01C4h	43.3.426/ 2475
53FA_86A8	IOMUXC_SW_PAD_CTL_PAD_GPIO_1 (IOMUXC_SW_PAD_CTL_PAD_GPIO_1)	32	R/W	0000_01C4h	43.3.427/ 2477
53FA_86AC	IOMUXC_SW_PAD_CTL_PAD_GPIO_9 (IOMUXC_SW_PAD_CTL_PAD_GPIO_9)	32	R/W	0000_01E4h	43.3.428/ 2479
53FA_86B0	IOMUXC_SW_PAD_CTL_PAD_GPIO_3 (IOMUXC_SW_PAD_CTL_PAD_GPIO_3)	32	R/W	0000_01C4h	43.3.429/ 2481
53FA_86B4	IOMUXC_SW_PAD_CTL_PAD_GPIO_6 (IOMUXC_SW_PAD_CTL_PAD_GPIO_6)	32	R/W	0000_01C4h	43.3.430/ 2483
53FA_86B8	IOMUXC_SW_PAD_CTL_PAD_GPIO_2 (IOMUXC_SW_PAD_CTL_PAD_GPIO_2)	32	R/W	0000_01C4h	43.3.431/ 2485
53FA_86BC	IOMUXC_SW_PAD_CTL_PAD_GPIO_4 (IOMUXC_SW_PAD_CTL_PAD_GPIO_4)	32	R/W	0000_01E4h	43.3.432/ 2487
53FA_86C0	IOMUXC_SW_PAD_CTL_PAD_GPIO_5 (IOMUXC_SW_PAD_CTL_PAD_GPIO_5)	32	R/W	0000_01C4h	43.3.433/ 2489
53FA_86C4	IOMUXC_SW_PAD_CTL_PAD_GPIO_7 (IOMUXC_SW_PAD_CTL_PAD_GPIO_7)	32	R/W	0000_01C4h	43.3.434/ 2491
53FA_86C8	IOMUXC_SW_PAD_CTL_PAD_GPIO_8 (IOMUXC_SW_PAD_CTL_PAD_GPIO_8)	32	R/W	0000_01C4h	43.3.435/ 2493
53FA_86CC	IOMUXC_SW_PAD_CTL_PAD_GPIO_16 (IOMUXC_SW_PAD_CTL_PAD_GPIO_16)	32	R/W	0000_01C4h	43.3.436/ 2495
53FA_86D0	IOMUXC_SW_PAD_CTL_PAD_GPIO_17 (IOMUXC_SW_PAD_CTL_PAD_GPIO_17)	32	R/W	0000_01C4h	43.3.437/ 2497
53FA_86D4	IOMUXC_SW_PAD_CTL_PAD_GPIO_18 (IOMUXC_SW_PAD_CTL_PAD_GPIO_18)	32	R/W	0000_01C4h	43.3.438/ 2499
53FA_86D8	IOMUXC_SW_PAD_CTL_PAD_NVCC_GPIO (IOMUXC_SW_PAD_CTL_PAD_NVCC_GPIO)	32	R/W	0002_0000h	43.3.439/ 2501

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_86DC	IOMUXC_SW_PAD_CTL_PAD_POR_B (IOMUXC_SW_PAD_CTL_PAD_POR_B)	32	R/W	0000_05E0h	43.3.440/ 2502
53FA_86E0	IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE1 (IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE1)	32	R/W	0000_05C0h	43.3.441/ 2504
53FA_86E4	IOMUXC_SW_PAD_CTL_PAD_RESET_IN_B (IOMUXC_SW_PAD_CTL_PAD_RESET_IN_B)	32	R/W	0000_05E0h	43.3.442/ 2506
53FA_86E8	IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE0 (IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE0)	32	R/W	0000_05C0h	43.3.443/ 2508
53FA_86EC	IOMUXC_SW_PAD_CTL_PAD_TEST_MODE (IOMUXC_SW_PAD_CTL_PAD_TEST_MODE)	32	R/W	0000_04C0h	43.3.444/ 2510
53FA_86F0	IOMUXC_SW_PAD_CTL_GRP_ADDDS (IOMUXC_SW_PAD_CTL_GRP_ADDDS)	32	R/W	0008_0000h	43.3.445/ 2511
53FA_86F4	IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL (IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL)	32	R/W	0000_0200h	43.3.446/ 2512
53FA_86FC	IOMUXC_SW_PAD_CTL_GRP_DDRPKE (IOMUXC_SW_PAD_CTL_GRP_DDRPKE)	32	R/W	0000_0080h	43.3.447/ 2513
53FA_8708	IOMUXC_SW_PAD_CTL_GRP_DDRPK (IOMUXC_SW_PAD_CTL_GRP_DDRPK)	32	R/W	0000_0040h	43.3.448/ 2513
53FA_8710	IOMUXC_SW_PAD_CTL_GRP_DDRHYS (IOMUXC_SW_PAD_CTL_GRP_DDRHYS)	32	R/W	0000_0000h	43.3.449/ 2514
53FA_8714	IOMUXC_SW_PAD_CTL_GRP_DDRMODE (IOMUXC_SW_PAD_CTL_GRP_DDRMODE)	32	R/W	0000_0000h	43.3.450/ 2515
53FA_8718	IOMUXC_SW_PAD_CTL_GRP_B0DS (IOMUXC_SW_PAD_CTL_GRP_B0DS)	32	R/W	0008_0000h	43.3.451/ 2515
53FA_871C	IOMUXC_SW_PAD_CTL_GRP_B1DS (IOMUXC_SW_PAD_CTL_GRP_B1DS)	32	R/W	0008_0000h	43.3.452/ 2516
53FA_8720	IOMUXC_SW_PAD_CTL_GRP_CTLDS (IOMUXC_SW_PAD_CTL_GRP_CTLDS)	32	R/W	0008_0000h	43.3.453/ 2516
53FA_8724	IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE (IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE)	32	R/W	0000_0000h	43.3.454/ 2517
53FA_8728	IOMUXC_SW_PAD_CTL_GRP_B2DS (IOMUXC_SW_PAD_CTL_GRP_B2DS)	32	R/W	0008_0000h	43.3.455/ 2518
53FA_872C	IOMUXC_SW_PAD_CTL_GRP_B3DS (IOMUXC_SW_PAD_CTL_GRP_B3DS)	32	R/W	0008_0000h	43.3.456/ 2518
53FA_8730	IOMUXC_AUDMUX_P4_INPUT_DA_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P4_INPUT_DA_AMX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.457/ 2519
53FA_8734	IOMUXC_AUDMUX_P4_INPUT_DB_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P4_INPUT_DB_AMX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.458/ 2519

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_8738	IOMUXC_AUDMUX_P4_INPUT_RXCLK_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P4_INPUT_RXCLK_AMX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.459/ 2520
53FA_873C	IOMUXC_AUDMUX_P4_INPUT_RXFS_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P4_INPUT_RXFS_AMX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.460/ 2520
53FA_8740	IOMUXC_AUDMUX_P4_INPUT_TXCLK_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P4_INPUT_TXCLK_AMX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.461/ 2521
53FA_8744	IOMUXC_AUDMUX_P4_INPUT_TXFS_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P4_INPUT_TXFS_AMX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.462/ 2521
53FA_8748	IOMUXC_AUDMUX_P5_INPUT_DA_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P5_INPUT_DA_AMX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.463/ 2522
53FA_874C	IOMUXC_AUDMUX_P5_INPUT_DB_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P5_INPUT_DB_AMX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.464/ 2522
53FA_8750	IOMUXC_AUDMUX_P5_INPUT_RXCLK_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P5_INPUT_RXCLK_AMX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.465/ 2523
53FA_8754	IOMUXC_AUDMUX_P5_INPUT_RXFS_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P5_INPUT_RXFS_AMX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.466/ 2523
53FA_8758	IOMUXC_AUDMUX_P5_INPUT_TXCLK_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P5_INPUT_TXCLK_AMX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.467/ 2524
53FA_875C	IOMUXC_AUDMUX_P5_INPUT_TXFS_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P5_INPUT_TXFS_AMX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.468/ 2524
53FA_8760	IOMUXC_CAN1_IPP_IND_CANRX_SELECT_INPUT (IOMUXC_CAN1_IPP_IND_CANRX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.469/ 2525
53FA_8764	IOMUXC_CAN2_IPP_IND_CANRX_SELECT_INPUT (IOMUXC_CAN2_IPP_IND_CANRX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.470/ 2525
53FA_8768	IOMUXC_CCM_IPP_ASRC_EXT_SELECT_INPUT (IOMUXC_CCM_IPP_ASRC_EXT_SELECT_INPUT)	32	R/W	0000_0000h	43.3.471/ 2526

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_876C	IOMUXC_CCM_IPP_DI1_CLK_SELECT_INPUT (IOMUXC_CCM_IPP_DI1_CLK_SELECT_INPUT)	32	R/W	0000_0000h	43.3.472/ 2526
53FA_8770	IOMUXC_CCM_PLL1_BYPASS_CLK_SELECT_INPUT (IOMUXC_CCM_PLL1_BYPASS_CLK_SELECT_INPUT)	32	R/W	0000_0000h	43.3.473/ 2527
53FA_8774	IOMUXC_CCM_PLL2_BYPASS_CLK_SELECT_INPUT (IOMUXC_CCM_PLL2_BYPASS_CLK_SELECT_INPUT)	32	R/W	0000_0000h	43.3.474/ 2527
53FA_8778	IOMUXC_CCM_PLL3_BYPASS_CLK_SELECT_INPUT (IOMUXC_CCM_PLL3_BYPASS_CLK_SELECT_INPUT)	32	R/W	0000_0000h	43.3.475/ 2528
53FA_877C	IOMUXC_CCM_PLL4_BYPASS_CLK_SELECT_INPUT (IOMUXC_CCM_PLL4_BYPASS_CLK_SELECT_INPUT)	32	R/W	0000_0000h	43.3.476/ 2528
53FA_8780	IOMUXC_CSPI_IPP_CSPI_CLK_IN_SELECT_INPUT (IOMUXC_CSPI_IPP_CSPI_CLK_IN_SELECT_INPUT)	32	R/W	0000_0000h	43.3.477/ 2529
53FA_8784	IOMUXC_CSPI_IPP_IND_MISO_SELECT_INPUT (IOMUXC_CSPI_IPP_IND_MISO_SELECT_INPUT)	32	R/W	0000_0000h	43.3.478/ 2529
53FA_8788	IOMUXC_CSPI_IPP_IND_MOSI_SELECT_INPUT (IOMUXC_CSPI_IPP_IND_MOSI_SELECT_INPUT)	32	R/W	0000_0000h	43.3.479/ 2530
53FA_878C	IOMUXC_CSPI_IPP_IND_SS0_B_SELECT_INPUT (IOMUXC_CSPI_IPP_IND_SS0_B_SELECT_INPUT)	32	R/W	0000_0000h	43.3.480/ 2531
53FA_8790	IOMUXC_CSPI_IPP_IND_SS1_B_SELECT_INPUT (IOMUXC_CSPI_IPP_IND_SS1_B_SELECT_INPUT)	32	R/W	0000_0000h	43.3.481/ 2531
53FA_8794	IOMUXC_CSPI_IPP_IND_SS2_B_SELECT_INPUT (IOMUXC_CSPI_IPP_IND_SS2_B_SELECT_INPUT)	32	R/W	0000_0000h	43.3.482/ 2532
53FA_8798	IOMUXC_CSPI_IPP_IND_SS3_B_SELECT_INPUT (IOMUXC_CSPI_IPP_IND_SS3_B_SELECT_INPUT)	32	R/W	0000_0000h	43.3.483/ 2532
53FA_879C	IOMUXC_ECSPI1_IPP_CSPI_CLK_IN_SELECT_INPUT (IOMUXC_ECSPI1_IPP_CSPI_CLK_IN_SELECT_INPUT)	32	R/W	0000_0000h	43.3.484/ 2533
53FA_87A0	IOMUXC_ECSPI1_IPP_IND_MISO_SELECT_INPUT (IOMUXC_ECSPI1_IPP_IND_MISO_SELECT_INPUT)	32	R/W	0000_0000h	43.3.485/ 2533
53FA_87A4	IOMUXC_ECSPI1_IPP_IND_MOSI_SELECT_INPUT (IOMUXC_ECSPI1_IPP_IND_MOSI_SELECT_INPUT)	32	R/W	0000_0000h	43.3.486/ 2534
53FA_87A8	IOMUXC_ECSPI1_IPP_IND_SS_B_0_SELECT_INPUT (IOMUXC_ECSPI1_IPP_IND_SS_B_0_SELECT_INPUT)	32	R/W	0000_0000h	43.3.487/ 2535
53FA_87AC	IOMUXC_ECSPI1_IPP_IND_SS_B_1_SELECT_INPUT (IOMUXC_ECSPI1_IPP_IND_SS_B_1_SELECT_INPUT)	32	R/W	0000_0000h	43.3.488/ 2535
53FA_87B0	IOMUXC_ECSPI1_IPP_IND_SS_B_2_SELECT_INPUT (IOMUXC_ECSPI1_IPP_IND_SS_B_2_SELECT_INPUT)	32	R/W	0000_0000h	43.3.489/ 2536
53FA_87B4	IOMUXC_ECSPI1_IPP_IND_SS_B_3_SELECT_INPUT (IOMUXC_ECSPI1_IPP_IND_SS_B_3_SELECT_INPUT)	32	R/W	0000_0000h	43.3.490/ 2536
53FA_87B8	IOMUXC_ECSPI2_IPP_CSPI_CLK_IN_SELECT_INPUT (IOMUXC_ECSPI2_IPP_CSPI_CLK_IN_SELECT_INPUT)	32	R/W	0000_0000h	43.3.491/ 2537

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_87BC	IOMUXC_ECSPI2_IPP_IND_MISO_SELECT_INPUT (IOMUXC_ECSPI2_IPP_IND_MISO_SELECT_INPUT)	32	R/W	0000_0000h	43.3.492/ 2537
53FA_87C0	IOMUXC_ECSPI2_IPP_IND_MOSI_SELECT_INPUT (IOMUXC_ECSPI2_IPP_IND_MOSI_SELECT_INPUT)	32	R/W	0000_0000h	43.3.493/ 2538
53FA_87C4	IOMUXC_ECSPI2_IPP_IND_SS_B_0_SELECT_INPUT (IOMUXC_ECSPI2_IPP_IND_SS_B_0_SELECT_INPUT)	32	R/W	0000_0000h	43.3.494/ 2538
53FA_87C8	IOMUXC_ECSPI2_IPP_IND_SS_B_1_SELECT_INPUT (IOMUXC_ECSPI2_IPP_IND_SS_B_1_SELECT_INPUT)	32	R/W	0000_0000h	43.3.495/ 2539
53FA_87CC	IOMUXC_ESAI1_IPP_IND_FSR_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_FSR_SELECT_INPUT)	32	R/W	0000_0000h	43.3.496/ 2539
53FA_87D0	IOMUXC_ESAI1_IPP_IND_FST_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_FST_SELECT_INPUT)	32	R/W	0000_0000h	43.3.497/ 2540
53FA_87D4	IOMUXC_ESAI1_IPP_IND_HCKR_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_HCKR_SELECT_INPUT)	32	R/W	0000_0000h	43.3.498/ 2540
53FA_87D8	IOMUXC_ESAI1_IPP_IND_HCKT_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_HCKT_SELECT_INPUT)	32	R/W	0000_0000h	43.3.499/ 2541
53FA_87DC	IOMUXC_ESAI1_IPP_IND_SCKR_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_SCKR_SELECT_INPUT)	32	R/W	0000_0000h	43.3.500/ 2541
53FA_87E0	IOMUXC_ESAI1_IPP_IND_SCKT_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_SCKT_SELECT_INPUT)	32	R/W	0000_0000h	43.3.501/ 2542
53FA_87E4	IOMUXC_ESAI1_IPP_IND_SDO0_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_SDO0_SELECT_INPUT)	32	R/W	0000_0000h	43.3.502/ 2542
53FA_87E8	IOMUXC_ESAI1_IPP_IND_SDO1_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_SDO1_SELECT_INPUT)	32	R/W	0000_0000h	43.3.503/ 2543
53FA_87EC	IOMUXC_ESAI1_IPP_IND_SDO2_SDI3_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_SDO2_SDI3_SELECT_INPUT)	32	R/W	0000_0000h	43.3.504/ 2543
53FA_87F0	IOMUXC_ESAI1_IPP_IND_SDO3_SDI2_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_SDO3_SDI2_SELECT_INPUT)	32	R/W	0000_0000h	43.3.505/ 2544
53FA_87F4	IOMUXC_ESAI1_IPP_IND_SDO4_SDI1_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_SDO4_SDI1_SELECT_INPUT)	32	R/W	0000_0000h	43.3.506/ 2544
53FA_87F8	IOMUXC_ESAI1_IPP_IND_SDO5_SDI0_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_SDO5_SDI0_SELECT_INPUT)	32	R/W	0000_0000h	43.3.507/ 2545
53FA_87FC	IOMUXC_ESDHC1_IPP_WP_ON_SELECT_INPUT (IOMUXC_ESDHC1_IPP_WP_ON_SELECT_INPUT)	32	R/W	0000_0000h	43.3.508/ 2545
53FA_8800	IOMUXC_FEC_FEC_COL_SELECT_INPUT (IOMUXC_FEC_FEC_COL_SELECT_INPUT)	32	R/W	0000_0000h	43.3.509/ 2546
53FA_8804	IOMUXC_FEC_FEC_MDI_SELECT_INPUT (IOMUXC_FEC_FEC_MDI_SELECT_INPUT)	32	R/W	0000_0000h	43.3.510/ 2546
53FA_8808	IOMUXC_FEC_FEC_RX_CLK_SELECT_INPUT (IOMUXC_FEC_FEC_RX_CLK_SELECT_INPUT)	32	R/W	0000_0000h	43.3.511/ 2547

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_880C	IOMUXC_FIRI_IPP_IND_RXD_SELECT_INPUT (IOMUXC_FIRI_IPP_IND_RXD_SELECT_INPUT)	32	R/W	0000_0000h	43.3.512/ 2547
53FA_8810	IOMUXC_GPC_PMIC_RDY_SELECT_INPUT (IOMUXC_GPC_PMIC_RDY_SELECT_INPUT)	32	R/W	0000_0000h	43.3.513/ 2548
53FA_8814	IOMUXC_I2C1_IPP_SCL_IN_SELECT_INPUT (IOMUXC_I2C1_IPP_SCL_IN_SELECT_INPUT)	32	R/W	0000_0000h	43.3.514/ 2548
53FA_8818	IOMUXC_I2C1_IPP_SDA_IN_SELECT_INPUT (IOMUXC_I2C1_IPP_SDA_IN_SELECT_INPUT)	32	R/W	0000_0000h	43.3.515/ 2549
53FA_881C	IOMUXC_I2C2_IPP_SCL_IN_SELECT_INPUT (IOMUXC_I2C2_IPP_SCL_IN_SELECT_INPUT)	32	R/W	0000_0000h	43.3.516/ 2549
53FA_8820	IOMUXC_I2C2_IPP_SDA_IN_SELECT_INPUT (IOMUXC_I2C2_IPP_SDA_IN_SELECT_INPUT)	32	R/W	0000_0000h	43.3.517/ 2550
53FA_8824	IOMUXC_I2C3_IPP_SCL_IN_SELECT_INPUT (IOMUXC_I2C3_IPP_SCL_IN_SELECT_INPUT)	32	R/W	0000_0000h	43.3.518/ 2550
53FA_8828	IOMUXC_I2C3_IPP_SDA_IN_SELECT_INPUT (IOMUXC_I2C3_IPP_SDA_IN_SELECT_INPUT)	32	R/W	0000_0000h	43.3.519/ 2551
53FA_882C	IOMUXC_IPU_IPP_DI_0_IND_DISPB_SD_D_SELECT_INPUT (IOMUXC_IPU_IPP_DI_0_IND_DISPB_SD_D_SELECT_INPUT)	32	R/W	0000_0000h	43.3.520/ 2551
53FA_8830	IOMUXC_IPU_IPP_DI_1_IND_DISPB_SD_D_SELECT_INPUT (IOMUXC_IPU_IPP_DI_1_IND_DISPB_SD_D_SELECT_INPUT)	32	R/W	0000_0000h	43.3.521/ 2552
53FA_8834	IOMUXC_IPU_IPP_IND_SENS1_DATA_EN_SELECT_INPUT (IOMUXC_IPU_IPP_IND_SENS1_DATA_EN_SELECT_INPUT)	32	R/W	0000_0000h	43.3.522/ 2552
53FA_8838	IOMUXC_IPU_IPP_IND_SENS1_HSYNC_SELECT_INPUT (IOMUXC_IPU_IPP_IND_SENS1_HSYNC_SELECT_INPUT)	32	R/W	0000_0000h	43.3.523/ 2553
53FA_883C	IOMUXC_IPU_IPP_IND_SENS1_VSYNC_SELECT_INPUT (IOMUXC_IPU_IPP_IND_SENS1_VSYNC_SELECT_INPUT)	32	R/W	0000_0000h	43.3.524/ 2553
53FA_8840	IOMUXC_KPP_IPP_IND_COL_5_SELECT_INPUT (IOMUXC_KPP_IPP_IND_COL_5_SELECT_INPUT)	32	R/W	0000_0000h	43.3.525/ 2554
53FA_8844	IOMUXC_KPP_IPP_IND_COL_6_SELECT_INPUT (IOMUXC_KPP_IPP_IND_COL_6_SELECT_INPUT)	32	R/W	0000_0000h	43.3.526/ 2554
53FA_8848	IOMUXC_KPP_IPP_IND_COL_7_SELECT_INPUT (IOMUXC_KPP_IPP_IND_COL_7_SELECT_INPUT)	32	R/W	0000_0000h	43.3.527/ 2555
53FA_884C	IOMUXC_KPP_IPP_IND_ROW_5_SELECT_INPUT (IOMUXC_KPP_IPP_IND_ROW_5_SELECT_INPUT)	32	R/W	0000_0000h	43.3.528/ 2555

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_8850	IOMUXC_KPP_IPP_IND_ROW_6_SELECT_INPUT (IOMUXC_KPP_IPP_IND_ROW_6_SELECT_INPUT)	32	R/W	0000_0000h	43.3.529/ 2556
53FA_8854	IOMUXC_KPP_IPP_IND_ROW_7_SELECT_INPUT (IOMUXC_KPP_IPP_IND_ROW_7_SELECT_INPUT)	32	R/W	0000_0000h	43.3.530/ 2556
53FA_8858	IOMUXC_MLB_MLBCLK_IN_SELECT_INPUT (IOMUXC_MLB_MLBCLK_IN_SELECT_INPUT)	32	R/W	0000_0000h	43.3.531/ 2557
53FA_885C	IOMUXC_MLB_MLBDAT_IN_SELECT_INPUT (IOMUXC_MLB_MLBDAT_IN_SELECT_INPUT)	32	R/W	0000_0000h	43.3.532/ 2557
53FA_8860	IOMUXC_MLB_MLBSIG_IN_SELECT_INPUT (IOMUXC_MLB_MLBSIG_IN_SELECT_INPUT)	32	R/W	0000_0000h	43.3.533/ 2558
53FA_8864	IOMUXC_OWIRE_BATTERY_LINE_IN_SELECT_INPUT (IOMUXC_OWIRE_BATTERY_LINE_IN_SELECT_INPUT)	32	R/W	0000_0000h	43.3.534/ 2558
53FA_8868	IOMUXC_SDMA_EVENTS_14_SELECT_INPUT (IOMUXC_SDMA_EVENTS_14_SELECT_INPUT)	32	R/W	0000_0000h	43.3.535/ 2559
53FA_886C	IOMUXC_SDMA_EVENTS_15_SELECT_INPUT (IOMUXC_SDMA_EVENTS_15_SELECT_INPUT)	32	R/W	0000_0000h	43.3.536/ 2559
53FA_8870	IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT (IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT)	32	R/W	0000_0000h	43.3.537/ 2560
53FA_8874	IOMUXC_UART1_IPP_UART_RTS_B_SELECT_INPUT (IOMUXC_UART1_IPP_UART_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	43.3.538/ 2560
53FA_8878	IOMUXC_UART1_IPP_UART_RXD_MUX_SELECT_INPUT (IOMUXC_UART1_IPP_UART_RXD_MUX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.539/ 2561
53FA_887C	IOMUXC_UART2_IPP_UART_RTS_B_SELECT_INPUT (IOMUXC_UART2_IPP_UART_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	43.3.540/ 2561
53FA_8880	IOMUXC_UART2_IPP_UART_RXD_MUX_SELECT_INPUT (IOMUXC_UART2_IPP_UART_RXD_MUX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.541/ 2562
53FA_8884	IOMUXC_UART3_IPP_UART_RTS_B_SELECT_INPUT (IOMUXC_UART3_IPP_UART_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	43.3.542/ 2562
53FA_8888	IOMUXC_UART3_IPP_UART_RXD_MUX_SELECT_INPUT (IOMUXC_UART3_IPP_UART_RXD_MUX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.543/ 2563
53FA_888C	IOMUXC_UART4_IPP_UART_RTS_B_SELECT_INPUT (IOMUXC_UART4_IPP_UART_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	43.3.544/ 2564
53FA_8890	IOMUXC_UART4_IPP_UART_RXD_MUX_SELECT_INPUT (IOMUXC_UART4_IPP_UART_RXD_MUX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.545/ 2564

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_8894	IOMUXC_UART5_IPP_UART_RTS_B_SELECT_INPUT (IOMUXC_UART5_IPP_UART_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	43.3.546/ 2565
53FA_8898	IOMUXC_UART5_IPP_UART_RXD_MUX_SELECT_INPUT (IOMUXC_UART5_IPP_UART_RXD_MUX_SELECT_INPUT)	32	R/W	0000_0000h	43.3.547/ 2565
53FA_889C	IOMUXC_USBOH3_IPP_IND_OTG_OC_SELECT_INPUT (IOMUXC_USBOH3_IPP_IND_OTG_OC_SELECT_INPUT)	32	R/W	0000_0000h	43.3.548/ 2566
53FA_88A0	IOMUXC_USBOH3_IPP_IND_UH1_OC_SELECT_INPUT (IOMUXC_USBOH3_IPP_IND_UH1_OC_SELECT_INPUT)	32	R/W	0000_0000h	43.3.549/ 2566
53FA_88A4	IOMUXC_USBOH3_IPP_IND_UH2_OC_SELECT_INPUT (IOMUXC_USBOH3_IPP_IND_UH2_OC_SELECT_INPUT)	32	R/W	0000_0000h	43.3.550/ 2567

43.3.1 General Purpose Register 0 (IOMUXC_GPR0)

Address: IOMUXC_GPR0 is 53FA_8000h base + 0h offset = 53FA_8000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CLOCK_8_MUX_SEL		CLOCK_0_MUX_SEL		CLOCK_B_MUX_SEL		CLOCK_3_MUX_SEL		CLOCK_A_MUX_SEL		CLOCK_2_MUX_SEL		CLOCK_9_MUX_SEL		CLOCK_1_MUX_SEL	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TX_CLK3_MUX_SEL		0		MLBCLK_IN_INV	DMAREQ_MUX_SEL10	DMAREQ_MUX_SEL9	DMAREQ_MUX_SEL8	DMAREQ_MUX_SEL7	DMAREQ_MUX_SEL6	DMAREQ_MUX_SEL5	DMAREQ_MUX_SEL4	DMAREQ_MUX_SEL3	DMAREQ_MUX_SEL2	DMAREQ_MUX_SEL1	DMAREQ_MUX_SEL0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_GPR0 field descriptions

Field	Description
31–30 CLOCK_8_MUX_SEL	Selects the source of asrck_clock_8 in ASRC according to clock muxing scheme: 00 audmux.amx_output_rxclk_p7 muxed with ssi3.ssi_srck 01 audmux.amx_output_rxclk_p7 10 ssi3.ssi_srck 11 ssi3.rx_bit_clk
29–28 CLOCK_0_MUX_SEL	Selects the source of asrck_clock_0 in ASRC according to clock muxing scheme: 00 esai1.ipp_ind_sckr muxed with esai1.ipp_do_sckr 01 esai1.ipp_ind_sckr

Table continues on the next page...

IOMUXC_GPR0 field descriptions (continued)

Field	Description
	10 esai1.ipp_do_sckr 11 Reserved
27–26 CLOCK_B_ MUX_SEL	Selects the source of asrck_clock_b in ASRC according to clock muxing scheme: 00 audmux.amx_output_txclk_p7 muxed with ssi3.ssi_stck 01 audmux.amx_output_txclk_p7 10 ssi3.ssi_stck 11 ssi3.tx_bit_clk
25–24 CLOCK_3_ MUX_SEL	Selects the source of asrck_clock_3 in ASRC according to clock muxing scheme: 00 audmux.amx_output_rxclk_p7 muxed with ssi3.ssi_srck 01 audmux.amx_output_rxclk_p7 10 ssi3.ssi_srck 11 ssi3.rx_bit_clk
23–22 CLOCK_A_ MUX_SEL	Selects the source of asrck_clock_a in ASRC according to clock muxing scheme: 00 audmux.amx_output_txclk_p2 muxed with ssi2.ssi_stck 01 audmux.amx_output_txclk_p2 10 ssi2.ssi_stck 11 ssi2.tx_bit_clk
21–20 CLOCK_2_ MUX_SEL	Selects the source of asrck_clock_2 in ASRC according to clock muxing scheme: 00 audmux.amx_output_rxclk_p2 muxed with ssi2.ssi_srck 01 audmux.amx_output_rxclk_p2 10 ssi2.ssi_srck 11 ssi2.rx_bit_clk
19–18 CLOCK_9_ MUX_SEL	Selects the source of asrck_clock_9 in ASRC according to clock muxing scheme: 00 audmux.amx_output_txclk_p1 muxed with ssi1.ssi_stck 01 audmux.amx_output_txclk_p1 10 ssi1.ssi_stck 11 ssi1.tx_bit_clk
17–16 CLOCK_1_ MUX_SEL	Selects the source of asrck_clock_1 in ASRC according to clock muxing scheme: 00 audmux.amx_output_rxclk_p1 muxed with ssi1.ssi_srck 01 audmux.amx_output_rxclk_p1 10 ssi1.ssi_srck 11 ssi1.rx_bit_clk
15–14 TX_CLK3_MUX_ SEL	Selects the source of tx_clk3 in SPDIF according to ASRC clock muxing scheme: 00 same source as for asrc.asrck_clock_1 01 same source as for asrc.asrck_clock_2 10 same source as for asrc.asrck_clock_3 11 Reserved
13–12 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_GPR0 field descriptions (continued)

Field	Description
11 MLBCLK_IN_ INV	MLBCLK_IN to MLB block is inverted (comparing to MLB device on board) for the following MLB outputs, DATAOUT and SIGOUT. (in addition also the pads output enable are driven by the inverted clock DATA_OE and SIG_OE). This bit should be used for high speed (1024fs) in case of timing issues. 0 MLBCLK_IN is not inverted.[default] 1 MLBCLK_IN is inverted.
10 DMAREQ_ MUX_SEL10	Selects between two possible sources for SDMA_EVENT[13]: 0 uart2.ipd_uart_tx_dmareq_b 1 firi.dma_req_b[1]
9 DMAREQ_ MUX_SEL9	Selects between two possible sources for SDMA_EVENT[12]: 0 uart2.ipd_uart_rx_dmareq_b 1 firi.dma_req_b[0]
8 DMAREQ_ MUX_SEL8	Selects between two possible sources for SDMA_EVENT[3]: 0 uart4.ipd_uart_tx_dmareq_b 1 pata.ata_tx_fifo_alarm
7 DMAREQ_ MUX_SEL7	Selects between two possible sources for SDMA_EVENT[2]: 0 uart4.ipd_uart_rx_dmareq_b 1 pata.ata_rcv_fifo_alarm
6 DMAREQ_ MUX_SEL6	Selects between two possible sources for SDMA_EVENT[39]: 0 cspi.ipd_req_cspi_tdma_b 1 iomux.sdma_events[15] - External DMA Request from pad DISP0_DAT16 or GPIO_17
5 DMAREQ_ MUX_SEL5	Selects between two possible sources for SDMA_EVENT[38]: 0 cspi.ipd_req_cspi_rdma_b 1 epit2.ipi_int_epit_oc
4 DMAREQ_ MUX_SEL4	Selects between two possible sources for SDMA_EVENT[14]: 0 spdif.drq0_spdif_b 1 iomux.sdma_events[14] - External DMA Request from pad DISP0_DAT17 or GPIO_18
3 DMAREQ_ MUX_SEL3	Selects between two possible sources for SDMA_EVENT[11]: 0 esdhc4.ipd_esdhcv2_dreq_b 1 cti2.CTITRIGOUT[0]
2 DMAREQ_ MUX_SEL2	Selects between two possible sources for SDMA_EVENT[10]: 0 i2c3.ipi_int_b 1 esdhc3.ipd_esdhcv3_dreq_b
1 DMAREQ_ MUX_SEL1	Selects between two possible sources for SDMA_EVENT[21]: 0 i2c2.ipi_int_b 1 esdhc2.ipd_esdhcv2_dreq_b

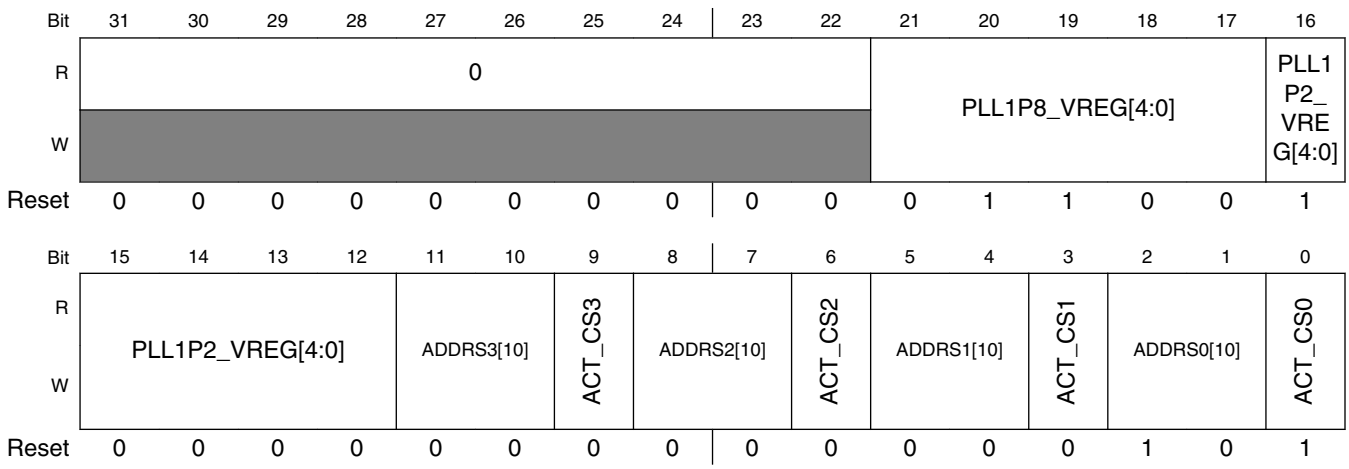
Table continues on the next page...

IOMUXC_GPR0 field descriptions (continued)

Field	Description
0 DMAREQ_ MUX_SEL0	Selects between two possible sources for SDMA_EVENT[20]: 0 i2c1.ipi_int_b 1 esdhc1.ipd_esdhcv2_dreq_b

43.3.2 General Purpose Register 1 (IOMUXC_GPR1)

Address: IOMUXC_GPR1 is 53FA_8000h base + 4h offset = 53FA_8004h



IOMUXC_GPR1 field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value zero. Reserved
21–17 PLL1P8_ VREG[4:0]	Defines the output for voltage regulators. For the 1.8V regulator: 0x00 = 1.5V and 0x1F = 2.275V in 25mV steps. This puts the reset value for 1.8V at 0x0C
16–12 PLL1P2_ VREG[4:0]	Defines the output for voltage regulators. For the 1.2V regulator: 0x00 = 0.8V and 0x1F = 1.575 in 25mV steps. This puts the reset value for 1.2V at 0x10.
11–10 ADDRS3[10]	Active Chip Select and Address Space. Each of the ACT_CSx represents one of the four chip selects of the WEIM. When ACT_CSx=1'b1, the corresponding chip select is active and has a valid address space according to its address space configuration determined by ADDRSx[10] bits ADDRSx[10] is setting the space for each chip select which is active. The address space of the first active chip select must be the biggest one, the following active chip select address spaces may be equal or lower. Total address space size is 128 MByte. Address Space Configuration options (ADDRSx[10]):

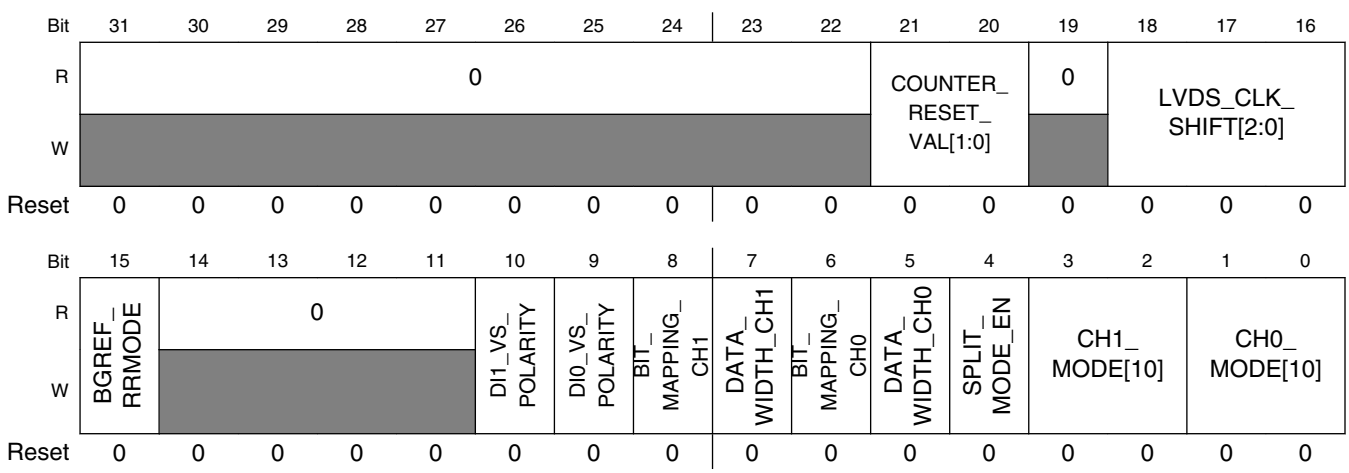
Table continues on the next page...

IOMUXC_GPR1 field descriptions (continued)

Field	Description
	2'b00 32 MByte 2'b01 64 MByte 2'b10 128 MByte 2'b11 Reserved The supported configurations are: CS0(128M), CS1 (0M), CS2 (0M), CS3(0M) [default configuration] CS0(64M), CS1(64M), CS2(0M), CS3(0M) CS0(64M), CS1(32M), CS2(32M), CS3(0M) CS0(32M), CS1(32M), CS2(32M), CS3(32M)
9 ACT_CS3	See description above.
8–7 ADDRS2[10]	See description above.
6 ACT_CS2	See description above.
5–4 ADDRS1[10]	See description above.
3 ACT_CS1	See description above.
2–1 ADDRS0[10]	See description above.
0 ACT_CS0	See description above.

43.3.3 General Purpose Register 2 (IOMUXC_GPR2)

Address: IOMUXC_GPR2 is 53FA_8000h base + 8h offset = 53FA_8008h



IOMUXC_GPR2 field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value zero. Reserved
21–20 COUNTER_ RESET_ VAL[1:0]	Reset value for the LDB counter which determines when the shift registers are loaded with data. Note: Used for debug purposes only. In normal functional operation must be '00' 00 Reset value is 5 01 Reset value is 3 10 Reset value is 4 11 Reset value is 6
19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18–16 LVDS_CLK_ SHIFT[2:0]	Shifts the LVDS output clock in relation to the data. Note: Used for debug purposes only. In normal functional operation must be '000' 000 Output clock is '1100011' (normal operation) 001 Output clock is '1110001' 010 Output clock is '1111000' 011 Output clock is '1000111' 100 Output clock is '0001111' 101 Output clock is '0011111' 110 Output clock is '0111100' 111 Output clock is '1100011'
15 BGREF_ RRMODE	Select reference resistor for bandgap 0 External resistor of 28 kOhms is selected 1 Internal resistor is selected
14–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10 DI1_VS_ POLARITY	Vsync polarity for IPU's DI1 interface. 0 ipu_di1_vsync is active high. 1 ipu_di1_vsync is active low.
9 DI0_VS_ POLARITY	Vsync polarity for IPU's DI0 interface. 0 ipu_di0_vsync is active high. 1 ipu_di0_vsync is active low.
8 BIT_MAPPING_ CH1	Data mapping for LVDS channel 1. 0 Use SPWG standard. 1 Use JEIDA standard.
7 DATA_WIDTH_ CH1	Data width for LVDS channel 1. Note: This bit must be set when using JEIDA standard (bit_mapping_ch1 is set) 0 Data width is 18 bits wide (lvds1_tx3 is not used) 1 Data width is 24 bits wide.

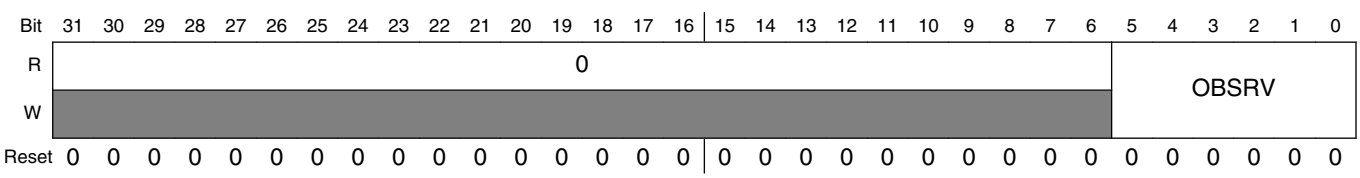
Table continues on the next page...

IOMUXC_GPR2 field descriptions (continued)

Field	Description
6 BIT_MAPPING_CH0	Data mapping for LVDS channel 0. 0 Use SPWG standard. 1 Use JEIDA standard.
5 DATA_WIDTH_CH0	Data width for LVDS channel 0. Note: This bit must be set when using JEIDA standard (bit_mapping_ch0 is set) 0 Data width is 18 bits wide (lvds0_tx3 is not used) 1 Data width is 24 bits wide.
4 SPLIT_MODE_EN	Enable split mode. 0 Split mode is disabled. 1 Split mode is enabled. In this mode both channels should be enabled and working with the same DI (ch0_mode and ch1_mode should both be either '01' or '11')
3–2 CH1_MODE[10]	LVDS channel 1 operation mode 00 Channel disabled. 01 Channel enabled, routed to DI0 10 Channel disabled. 11 Channel enabled, routed to DI1.
1–0 CH0_MODE[10]	LVDS channel 0 operation mode 00 Channel disabled. 01 Channel enabled, routed to DI0 10 Channel disabled. 11 Channel enabled, routed to DI1.

43.3.4 OBSERVE_MUX 0 Register (IOMUXC_OMUX0)

Address: IOMUXC_OMUX0 is 53FA_8000h base + Ch offset = 53FA_800Ch



IOMUXC_OMUX0 field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5–0 OBSRV	Select Instance Pin for Observability OBSERVE_MUX_0 000000 Select Instance ahbmax, Pin max_halted 000001 Select Instance ccm, Pin ccm_clk_switch_ack

Table continues on the next page...

IOMUXC_OMUX0 field descriptions

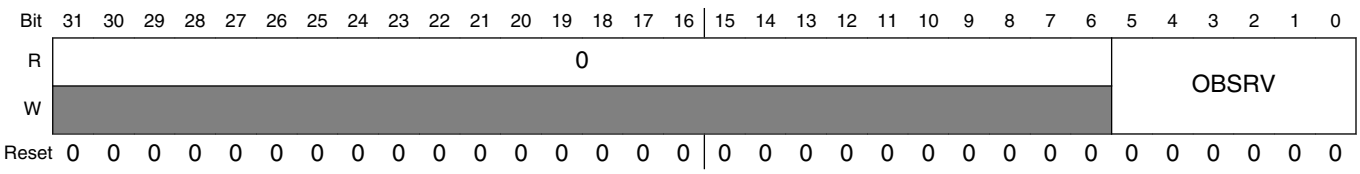
Field	Description
	000010 Select Instance ccm, Pin ccm_ipg_stop
	000011 Select Instance ccm, Pin ccm_ipg_wait
	000100 Select Instance ccm, Pin ccm_lpsr_ipu
	000101 Select Instance ccm, Pin ccm_pdn_4all_req
	000110 Select Instance ccm, Pin hndsk_current_state[0]
	000111 Select Instance ccm, Pin ipi_int_1
	001000 Select Instance ccm, Pin ipi_int_2
	001001 Select Instance ccm, Pin lpm_current_state[0]
	001010 Select Instance ccm, Pin shd_current_state[0]
	001011 Select Instance cspi, Pin ~ipi_int_cspi_b
	001100 Select Instance csu, Pin ~ipi_int_csu_b
	001101 Select Instance dpllip1, Pin dpllip_cpen
	001110 Select Instance ecspi1, Pin ipd_req_cspi_rdma_b
	001111 Select Instance ecspi1, Pin ipd_req_cspi_tdma_b
	010000 Select Instance ecspi1, Pin ~ipi_int_cspi_b
	010001 Select Instance ecspi2, Pin ipd_req_cspi_rdma_b
	010010 Select Instance ecspi2, Pin ~ipi_int_cspi_b
	010011 Select Instance emi, Pin dvfs_ack_fast
	010100 Select Instance emi, Pin dvfs_ack_int1
	010101 Select Instance emi, Pin ~ipi_emi_ap_int_b
	010110 Select Instance emi, Pin ~ipi_int_nfc_b
	010111 Select Instance emi, Pin ipp_obe_data_dir[3]
	011000 Select Instance emi, Pin ipp_obe_maddr_dir[1]
	011001 Select Instance emi, Pin lpack
	011010 Select Instance emi, Pin lpmd_int1_s1_mem
	011011 Select Instance epit1, Pin ipi_int_epit_oc
	011100 Select Instance epit2, Pin ipi_int_epit_oc
	011101 Select Instance esdhc1, Pin ~ipi_esdhcv2_irq_b
	011110 Select Instance esdhc2, Pin ~ipi_esdhcv2_irq_b
	011111 Select Instance esdhc3, Pin ~ipi_esdhcv3_irq_b
	100000 Select Instance esdhc4, Pin ~ipi_esdhcv2_irq_b
	100001 Select Instance fec, Pin fec_ipi_int
	100010 Select Instance firi, Pin ~ipi_int_b
	100011 Select Instance gpc, Pin gpc_cta8_pg[0]
	100100 Select Instance gpc, Pin gpc_cta8_pg[1]
	100101 Select Instance gpc, Pin gpc_cta8_pg[2]
	100110 Select Instance gpc, Pin gpc_cta8_pg[3]
	100111 Select Instance gpc, Pin gpc_cta8_pg[4]
	101000 Select Instance gpc, Pin gpc_emi_pg[0]
	101001 Select Instance gpc, Pin gpc_emi_short_b
	101010 Select Instance gpc, Pin gpc_event
	101011 Select Instance gpc, Pin gpc_int
	101100 Select Instance gpc, Pin gpc_int2
	101101 Select Instance gpc, Pin gpc_neon_pg[0]
	101110 Select Instance pata, Pin ata_rcv_fifo_alarm
	101111 Select Instance pata, Pin ata_tx_fifo_alarm
	110000 Select Instance pata, Pin ata_txfer_end_alarm
	110001 Select Instance src, Pin arm_por_rst

IOMUXC_OMUX0 field descriptions (continued)

Field	Description
	110010 Select Instance src, Pin warm_reset 110011 Select Instance p_platform_ne_32k_256k, Pin dsm_request 110100 Select Instance p_platform_ne_32k_256k, Pin ~nm_irq_b 110101 Select Instance tzic, Pin tzic_fiq_b 110110 Select Instance tzic, Pin tzic_irq_b 110111 Select Instance vpu, Pin ipi_int_vpu

43.3.5 OBSERVE_MUX 1 Register 1 (IOMUXC_OMUX1)

Address: IOMUXC_OMUX1 is 53FA_8000h base + 10h offset = 53FA_8010h



IOMUXC_OMUX1 field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5–0 OBSRV	Select Instance Pin for Observability OBSERVE_MUX_1 000000 Select Instance ccm, Pin ahbmax_halt_req 000001 Select Instance ccm, Pin ccm_pdn_4arm_req 000010 Select Instance ccm, Pin ccm_pup_req 000011 Select Instance ccm, Pin ccm_system_in_stop_mode 000100 Select Instance ccm, Pin ccm_system_in_wait_mode 000101 Select Instance ccm, Pin dpll_en_dpllip 000110 Select Instance ccm, Pin emi_dvfs_req_fast 000111 Select Instance ccm, Pin emi_dvfs_req_int1 001000 Select Instance ccm, Pin emi_dvfs_req_slow 001001 Select Instance ccm, Pin emi_lpmd 001010 Select Instance ccm, Pin emi_lpmd_fast 001011 Select Instance ccm, Pin hndsk_current_state[1] 001100 Select Instance ccm, Pin lpm_current_state[1] 001101 Select Instance ccm, Pin shd_current_state[1] 001110 Select Instance dpllip2, Pin dpllip_cpen 001111 Select Instance ecspi2, Pin ipd_req_cspi_tdma_b 010000 Select Instance emi, Pin dvfs_ack 010001 Select Instance emi, Pin ipp_obe_data_dir[2] 010010 Select Instance emi, Pin ipp_obe_maddr_dir[0] 010011 Select Instance emi, Pin lpack_fast 010100 Select Instance emi, Pin lpmd_int1_s0_mem

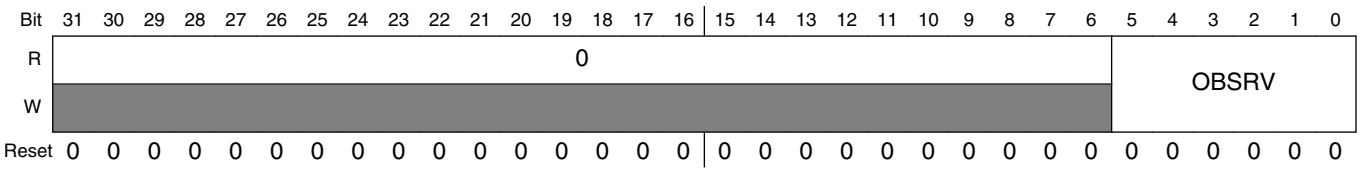
Table continues on the next page...

IOMUXC_OMUX1 field descriptions (continued)

Field	Description
	010101 Select Instance firi, Pin dma_req_b[0] 010110 Select Instance firi, Pin dma_req_b[1] 010111 Select Instance gpc, Pin gpc_cta8_clk_upd_req 011000 Select Instance gpc, Pin gpc_cta8_pg[5] 011001 Select Instance gpc, Pin gpc_cta8_pg[6] 011010 Select Instance gpc, Pin gpc_cta8_pg[7] 011011 Select Instance gpc, Pin gpc_cta8_pg[8] 011100 Select Instance gpc, Pin gpc_cta8_pg[9] 011101 Select Instance gpc, Pin gpc_emi_pg[1] 011110 Select Instance gpc, Pin gpc_ipu_switch_b 011111 Select Instance gpc, Pin gpc_l1bits_pwrdwn 100000 Select Instance gpc, Pin gpc_l2bits_pwrdwn 100001 Select Instance gpc, Pin gpc_neon_pg[1] 100010 Select Instance gpio1, Pin ipi_gpio_int15_0 100011 Select Instance gpio1, Pin ipi_gpio_int31_16 100100 Select Instance gpio1, Pin ipi_gpio_int32[0] 100101 Select Instance gpio1, Pin ipi_gpio_int32[1] 100110 Select Instance gpio1, Pin ipi_gpio_int32[2] 100111 Select Instance gpio1, Pin ipi_gpio_int32[3] 101000 Select Instance gpio1, Pin ipi_gpio_int32[4] 101001 Select Instance gpio1, Pin ipi_gpio_int32[5] 101010 Select Instance gpio1, Pin ipi_gpio_int32[6] 101011 Select Instance gpio1, Pin ipi_gpio_int32[7] 101100 Select Instance gpio2, Pin ipi_gpio_int15_0 101101 Select Instance gpio2, Pin ipi_gpio_int31_16 101110 Select Instance gpio3, Pin ipi_gpio_int15_0 101111 Select Instance gpio3, Pin ipi_gpio_int31_16 110000 Select Instance gpio4, Pin ipi_gpio_int15_0 110001 Select Instance gpio4, Pin ipi_gpio_int31_16 110010 Select Instance gpt, Pin ipi_int_gpt 110011 Select Instance gpu3d, Pin ~gpu_int_b 110100 Select Instance src, Pin arm_soc_rst_b 110101 Select Instance p_platform_ne_32k_256k, Pin dsm_request 110110 Select Instance uart1, Pin ipd_uart_rx_dmareq_b 110111 Select Instance uart1, Pin ipd_uart_tx_dmareq_b 111000 Select Instance uart2, Pin ipd_uart_rx_dmareq_b 111001 Select Instance uart2, Pin ipd_uart_tx_dmareq_b 111010 Select Instance wdog1, Pin wdog_rst_b

43.3.6 OBSERVE_MUX 2 Register (IOMUXC_OMUX2)

Address: IOMUXC_OMUX2 is 53FA_8000h base + 14h offset = 53FA_8014h



IOMUXC_OMUX2 field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5–0 OBSRV	Select Instance Pin for Observability OBSERVE_MUX_2 000000 Select Instance ccm, Pin emi_freq_change_req 000001 Select Instance ccm, Pin emi_lpmd_int1 000010 Select Instance ccm, Pin hndsk_current_state[2] 000011 Select Instance ccm, Pin ipu_clk_changed 000100 Select Instance ccm, Pin lpm_current_state[2] 000101 Select Instance dpllip3, Pin dpllip_cpen 000110 Select Instance emi, Pin ipp_obe_weim_nfc_dir[0] 000111 Select Instance emi, Pin lpack 001000 Select Instance emi, Pin lpack_fast 001001 Select Instance emi, Pin lpack_int1 001010 Select Instance emi, Pin lpack_slow 001011 Select Instance emi, Pin lpmd_int1_s1_mem 001100 Select Instance gpc, Pin gpc_core_pwrdwn 001101 Select Instance gpc, Pin gpc_cta8_clk_dvfs_operation 001110 Select Instance gpc, Pin gpc_cta8_pg[10] 001111 Select Instance gpc, Pin gpc_cta8_pg[11] 010000 Select Instance gpc, Pin gpc_cta8_pg[12] 010001 Select Instance gpc, Pin gpc_cta8_pg[13] 010010 Select Instance gpc, Pin gpc_cta8_pg[14] 010011 Select Instance gpc, Pin gpc_emi_pg[2] 010100 Select Instance gpc, Pin gpc_neon_pg[2] 010101 Select Instance gpc, Pin gpc_neon_pwrdwn 010110 Select Instance gpc, Pin gpc_pdn_ack 010111 Select Instance gpc, Pin gpc_pup_ack 011000 Select Instance gpu3d, Pin ~gpu_idle 011001 Select Instance i2c1, Pin ~ipi_int_b 011010 Select Instance i2c2, Pin ~ipi_int_b 011011 Select Instance iim, Pin ~iim_irq_b 011100 Select Instance ipu, Pin ipi_int_ipu_err 011101 Select Instance ipu, Pin ipi_int_ipu_func 011110 Select Instance kpp, Pin ~ipi_int_kpp_b 011111 Select Instance owire, Pin ipi_owire_int

Table continues on the next page...

IOMUXC_OMUX2 field descriptions (continued)

Field	Description
	100000 Select Instance pata, Pin ipbus_int 100001 Select Instance pwm1, Pin ipi_int_pwm 100010 Select Instance pwm2, Pin ipi_int_pwm 100011 Select Instance rtic, Pin ipi_rtic_done_int 100100 Select Instance sahara, Pin ~ipi_int_sahara_host0_b 100101 Select Instance src, Pin sjc_por_rst_b 100110 Select Instance src, Pin system_rst_b 100111 Select Instance ssi1, Pin ipd_ssi_rx0_dmareq_b 101000 Select Instance ssi1, Pin ipd_ssi_rx1_dmareq_b 101001 Select Instance ssi1, Pin ipd_ssi_tx1_dmareq_b 101010 Select Instance ssi2, Pin ipd_ssi_rx1_dmareq_b

43.3.7 OBSERVE_MUX 3 Register (IOMUXC_OMUX3)

Address: IOMUXC_OMUX3 is 53FA_8000h base + 18h offset = 53FA_8018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																OBSRV															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_OMUX3 field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5–0 OBSRV	Select Instance Pin for Observability OBSERVE_MUX_3 000000 Select Instance ccm, Pin emi_lpmd_slow 000001 Select Instance ccm, Pin rtic_ipg_stop_req 000010 Select Instance cspi, Pin ipd_req_cspi_rdma_b 000011 Select Instance cspi, Pin ipd_req_cspi_tdma_b 000100 Select Instance dpllip1, Pin dpllip_lrf_sticky 000101 Select Instance dpllip2, Pin dpllip_lrf_sticky 000110 Select Instance dpllip3, Pin dpllip_lrf_sticky 000111 Select Instance emi, Pin dvfs_ack_slow 001000 Select Instance emi, Pin ipp_obe_weim_nfc_dir[1] 001001 Select Instance emi, Pin lpack_slow 001010 Select Instance emi, Pin nfc_dma_rd_req 001011 Select Instance emi, Pin nfc_dma_wr_req 001100 Select Instance emi, Pin weim_idle 001101 Select Instance epit2, Pin ipi_int_epit_oc 001110 Select Instance gpc, Pin gpc_cta8_pg[15] 001111 Select Instance gpc, Pin gpc_cta8_pg[16]

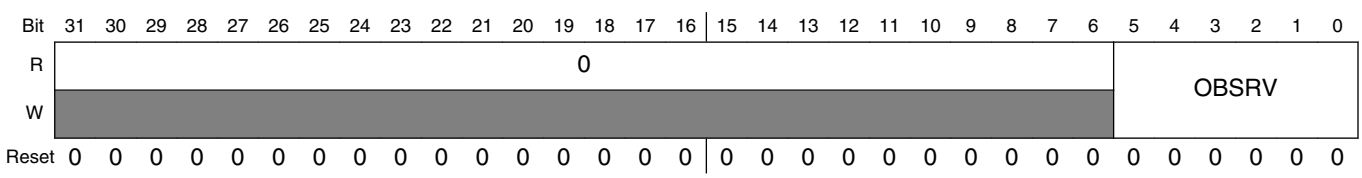
Table continues on the next page...

IOMUXC_OMUX3 field descriptions (continued)

Field	Description
	010000 Select Instance gpc, Pin gpc_cta8_pg[17] 010001 Select Instance gpc, Pin gpc_cta8_pg[18] 010010 Select Instance gpc, Pin gpc_cta8_pg[19] 010011 Select Instance gpc, Pin gpc_emi_pg[3] 010100 Select Instance gpc, Pin gpc_freq_change_mode 010101 Select Instance gpc, Pin gpc_neon_pg[3] 010110 Select Instance gpc, Pin gpc_neon_short_b 010111 Select Instance gpu3d, Pin gpu_idle 011000 Select Instance ipu, Pin ipg_clk_change_ack 011001 Select Instance ipu, Pin ipu_sdma_event 011010 Select Instance ipu, Pin ipu_stby_ack 011011 Select Instance sahara, Pin ipg_stop_ack 011100 Select Instance scc, Pin ipg_stop_ack 011101 Select Instance scc, Pin scc_ipi_sctl_irq 011110 Select Instance scc, Pin scc_ipi_sctl_ns_irq 011111 Select Instance scc, Pin scc_ipi_smon_irq 100000 Select Instance sdma, Pin ipg_stop_ack 100001 Select Instance sdma, Pin ~ipi_host_intr_b 100010 Select Instance spdif, Pin ~mirq_tx_b 100011 Select Instance src, Pin any_pu_rst_b 100100 Select Instance src, Pin emi_rst_b 100101 Select Instance src, Pin ipi_int_1 100110 Select Instance src, Pin system_early_rst_b 100111 Select Instance srtc, Pin ~ipi_srtc_int_b 101000 Select Instance srtc, Pin ~ipi_srtc_sec_int_b 101001 Select Instance ssi1, Pin ipd_ssi_tx0_dmareq_b 101010 Select Instance ssi1, Pin ~ipi_int_b 101011 Select Instance ssi2, Pin ~ipi_int_b 101100 Select Instance ssi3, Pin ~ipi_int_b 101101 Select Instance p_platform_ne_32k_256k, Pin nm_irq_b

43.3.8 OBSERVE_MUX 4 Register (IOMUXC_OMUX4)

Address: IOMUXC_OMUX4 is 53FA_8000h base + 1Ch offset = 53FA_801Ch



IOMUXC_OMUX4 field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5–0 OBSRV	Select Instance Pin for Observability OBSERVE_MUX_4 000000 Select Instance ccm, Pin emi_lpmd_garb 000001 Select Instance ccm, Pin ipu_freq_change_req 000010 Select Instance ccm, Pin ipu_stop_clk_at_stop_req 000011 Select Instance ccm, Pin pll_lvs 000100 Select Instance ccm, Pin sahara_ipg_stop_req 000101 Select Instance ccm, Pin scc_ipg_stop_req 000110 Select Instance ccm, Pin sdma_ipg_stop_req 000111 Select Instance ccm, Pin src_clock_ready 001000 Select Instance emi, Pin esdctl_idle 001001 Select Instance emi, Pin nfc_idle 001010 Select Instance esdhc3, Pin ipd_esdhcv3_dreq_b 001011 Select Instance esdhc4, Pin ipd_esdhcv2_dreq_b 001100 Select Instance gpc, Pin gpc_cta8_iso 001101 Select Instance gpc, Pin gpc_cta8_pg[20] 001110 Select Instance gpc, Pin gpc_emi_pg[4] 001111 Select Instance gpc, Pin gpc_ipu_pg_event 010000 Select Instance gpc, Pin gpc_ipu_stat_pg 010001 Select Instance gpc, Pin volt_chng 010010 Select Instance iim, Pin fuse_latched 010011 Select Instance ipu, Pin ipu_stby_ack 010100 Select Instance rtic, Pin ipg_stop_ack 010101 Select Instance sfp, Pin sfp_ready 010110 Select Instance spdif, Pin drq1_spdif_b 010111 Select Instance src, Pin emi_dvfs_req 011000 Select Instance src, Pin en_system_clk 011001 Select Instance src, Pin memory_repair_mode 011010 Select Instance src, Pin power_gating_reset_done 011011 Select Instance p_platform_ne_32k_256k, Pin cti1_trigout0 011100 Select Instance p_platform_ne_32k_256k, Pin cti1_trigout1 011101 Select Instance p_platform_ne_32k_256k, Pin cti1_trigout2 011110 Select Instance p_platform_ne_32k_256k, Pin cti1_trigout3 011111 Select Instance p_platform_ne_32k_256k, Pin ~cti_irq_b 100000 Select Instance p_platform_ne_32k_256k, Pin dbgack 100001 Select Instance p_platform_ne_32k_256k, Pin ~pmu_irq_b 100010 Select Instance tve, Pin ipi_tve_int 100011 Select Instance tzic, Pin tzic_wakeup_request 100100 Select Instance uart1, Pin ~ipi_uart_anded_b 100101 Select Instance uart2, Pin ~ipi_uart_anded_b 100110 Select Instance uart3, Pin ipd_uart_rx_dmareq_b 100111 Select Instance uart3, Pin ipd_uart_tx_dmareq_b 101000 Select Instance uart3, Pin ~ipi_uart_anded_b 101001 Select Instance usboh3, Pin ipi_int_uh1 101010 Select Instance usboh3, Pin ipi_int_uh2

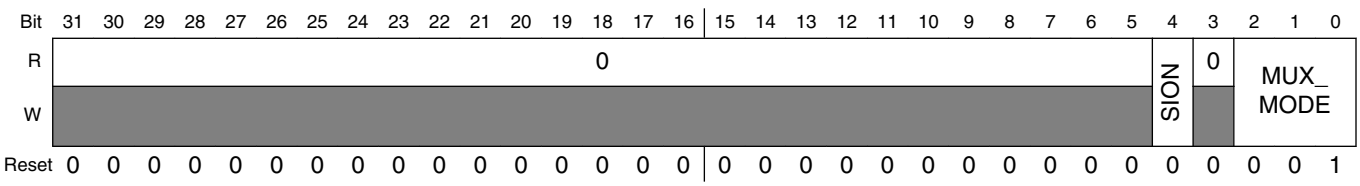
Table continues on the next page...

IOMUXC_OMUX4 field descriptions (continued)

Field	Description
	101011 Select Instance usboh3, Pin ipi_int_uh3 101100 Select Instance usboh3, Pin ipi_int_uotg 101101 Select Instance vpu, Pin ipi_int_vpu 101110 Select Instance vpu, Pin vpu_idle 101111 Select Instance wdog1, Pin ~ipi_wdog_int_b 110000 Select Instance wdog2, Pin ~ipi_wdog_int_b 110001 Select Instance wdog2, Pin wdog_rst_b 110010 Select Instance sahara, Pin ~ipi_int_sahara_host1_b

43.3.9 IOMUXC_SW_MUX_CTL_PAD_GPIO_19 (IOMUXC_GPIO_19)

Address: IOMUXC_GPIO_19 is 53FA_8000h base + 20h offset = 53FA_8020h

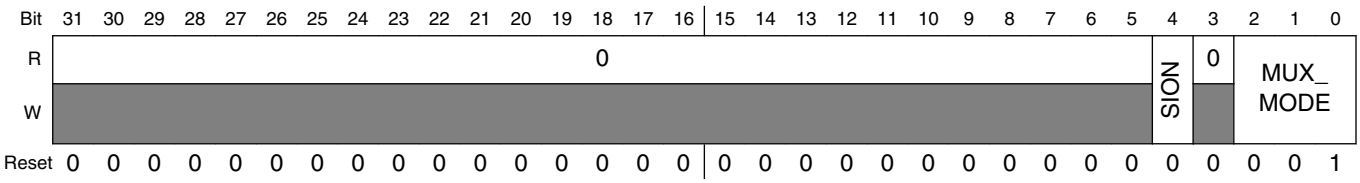


IOMUXC_GPIO_19 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_19. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_19. NOTE: Pad GPIO_19 is involved in Daisy Chain. - Config Register KPP_IPP_IND_COL_5_SELECT_INPUT for mode ALT0. 000 Select mux mode: ALT0 mux port: COL[5] of instance: kpp. 001 Select mux mode: ALT1 mux port: GPIO[5] of instance: gpio4. 010 Select mux mode: ALT2 mux port: CLKO of instance: ccm. 011 Select mux mode: ALT3 mux port: OUT1 of instance: spdif. 100 Select mux mode: ALT4 mux port: CE_RTC_EXT_TRIG2 of instance: rtc. 101 Select mux mode: ALT5 mux port: RDY of instance: ecspi1. 110 Select mux mode: ALT6 mux port: TDATA[3] of instance: fec. 111 Select mux mode: ALT7 mux port: INT_BOOT of instance: src.

43.3.10 IOMUXC_SW_MUX_CTL_PAD_KEY_COL0 (IOMUXC_KEY_COL0)

Address: IOMUXC_KEY_COL0 is 53FA_8000h base + 24h offset = 53FA_8024h

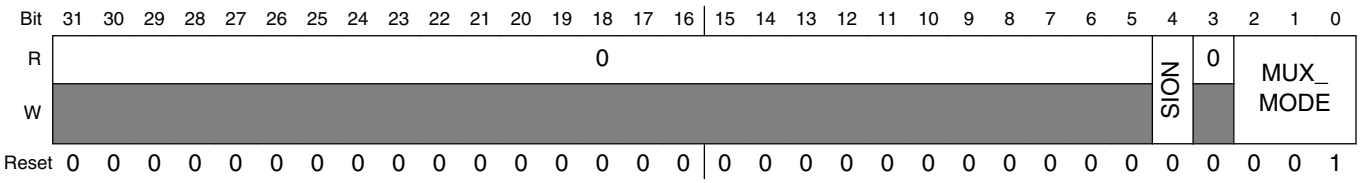


IOMUXC_KEY_COL0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad KEY_COL0. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: KEY_COL0. NOTE: Pad KEY_COL0 is involved in Daisy Chain. - Config Register AUDMUX_P5_INPUT_TXCLK_AMX_SELECT_INPUT for mode ALT2. - Config Register ECSPI1_IPP_CSPI_CLK_IN_SELECT_INPUT for mode ALT5. - Config Register UART4_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: COL[0] of instance: kpp. 001 Select mux mode: ALT1 mux port: GPIO[6] of instance: gpio4. 010 Select mux mode: ALT2 mux port: AUD5_TXC of instance: audmux. 011 Select mux mode: ALT3 mux port: CTI_TRIGIN7 of instance: p_platform_ne_32k_256k. 100 Select mux mode: ALT4 mux port: TXD_MUX of instance: uart4. 101 Select mux mode: ALT5 mux port: SCLK of instance: ecspi1. 110 Select mux mode: ALT6 mux port: RDATA[3] of instance: fec. 111 Select mux mode: ALT7 mux port: ANY_PU_RST of instance: src.

43.3.11 IOMUXC_SW_MUX_CTL_PAD_KEY_ROW0 (IOMUXC_KEY_ROW0)

Address: IOMUXC_KEY_ROW0 is 53FA_8000h base + 28h offset = 53FA_8028h

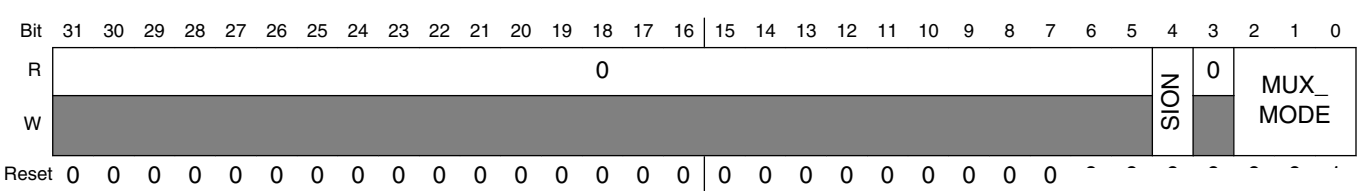


IOMUXC_KEY_ROW0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad KEY_ROW0. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: KEY_ROW0. NOTE: Pad KEY_ROW0 is involved in Daisy Chain. - Config Register AUDMUX_P5_INPUT_DB_AMX_SELECT_INPUT for mode ALT2. - Config Register ECSPI1_IPP_IND_MOSI_SELECT_INPUT for mode ALT5. - Config Register UART4_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: ROW[0] of instance: kpp. 001 Select mux mode: ALT1 mux port: GPIO[7] of instance: gpio4. 010 Select mux mode: ALT2 mux port: AUD5_TXD of instance: audmux. 011 Select mux mode: ALT3 mux port: CTI_TRIGIN_ACK7 of instance: p_platform_ne_32k_256k. 100 Select mux mode: ALT4 mux port: RXD_MUX of instance: uart4. 101 Select mux mode: ALT5 mux port: MOSI of instance: ecspi1. 110 Select mux mode: ALT6 mux port: TX_ER of instance: fec.

43.3.12 IOMUXC_SW_MUX_CTL_PAD_KEY_COL1 (IOMUXC_KEY_COL1)

Address: IOMUXC_KEY_COL1 is 53FA_8000h base + 2Ch offset = 53FA_802Ch



IOMUXC_KEY_COL1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad KEY_COL1. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: KEY_COL1. NOTE: Pad KEY_COL1 is involved in Daisy Chain. - Config Register AUDMUX_P5_INPUT_TXFS_AMX_SELECT_INPUT for mode ALT2. - Config Register ECSPI1_IPP_IND_MISO_SELECT_INPUT for mode ALT5. - Config Register FEC_FEC_RX_CLK_SELECT_INPUT for mode ALT6. - Config Register UART5_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: COL[1] of instance: kpp. 001 Select mux mode: ALT1 mux port: GPIO[8] of instance: gpio4. 010 Select mux mode: ALT2 mux port: AUD5_TXFS of instance: audmux. 011 Select mux mode: ALT3 mux port: CTI_TRIGOUT_ACK6 of instance: p_platform_ne_32k_256k. 100 Select mux mode: ALT4 mux port: TXD_MUX of instance: uart5. 101 Select mux mode: ALT5 mux port: MISO of instance: ecspi1. 110 Select mux mode: ALT6 mux port: RX_CLK of instance: fec. 111 Select mux mode: ALT7 mux port: TXREADY of instance: usbphy1.

43.3.13 IOMUXC_SW_MUX_CTL_PAD_KEY_ROW1 (IOMUXC_KEY_ROW1)

Address: IOMUXC_KEY_ROW1 is 53FA_8000h base + 30h offset = 53FA_8030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_KEY_ROW1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality.

Table continues on the next page...

IOMUXC_KEY_ROW1 field descriptions (continued)

Field	Description
	1 Force input path of pad KEY_ROW1. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: KEY_ROW1. NOTE: Pad KEY_ROW1 is involved in Daisy Chain. - Config Register IOMUXC_.UDMUX_P5_INPUT_DA_AMX_SELECT_INPUT for mode ALT2. - Config Register ECSPI1_IPP_IND_SS_B_0_SELECT_INPUT for mode ALT5. - Config Register FEC_FEC_COL_SELECT_INPUT for mode ALT6. - Config Register UART5_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: ROW[1] of instance: kpp. 001 Select mux mode: ALT1 mux port: GPIO[9] of instance: gpio4. 010 Select mux mode: ALT2 mux port: AUD5_RXD of instance: audmux. 011 Select mux mode: ALT3 mux port: CTI_TRIGOUT_ACK7 of instance: p_platform_ne_32k_256k. 100 Select mux mode: ALT4 mux port: RXD_MUX of instance: uart5. 101 Select mux mode: ALT5 mux port: SS0 of instance: ecspi1. 110 Select mux mode: ALT6 mux port: COL of instance: fec. 111 Select mux mode: ALT7 mux port: RXVALID of instance: usbphy1.

43.3.14 IOMUXC_SW_MUX_CTL_PAD_KEY_COL2 (IOMUXC_KEY_COL2)

Address: IOMUXC_KEY_COL2 is 53FA_8000h base + 34h offset = 53FA_8034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_ MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_KEY_COL2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad KEY_COL2. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved

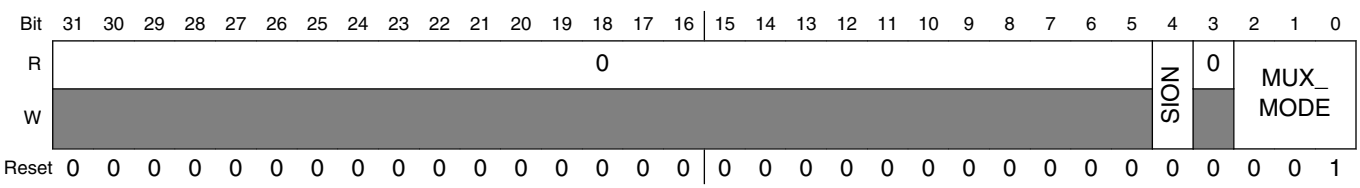
Table continues on the next page...

IOMUXC_KEY_COL2 field descriptions (continued)

Field	Description
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: KEY_COL2. NOTE: Pad KEY_COL2 is involved in Daisy Chain. - Config Register ECSPI1_IPP_IND_SS_B_1_SELECT_INPUT for mode ALT5. - Config Register FEC_FEC_MDI_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: COL[2] of instance: kpp. 001 Select mux mode: ALT1 mux port: GPIO[10] of instance: gpio4. 010 Select mux mode: ALT2 mux port: TXCAN of instance: can1. 011 Select mux mode: ALT3 mux port: CTI_TRIGOUT6 of instance: p_platform_ne_32k_256k. 100 Select mux mode: ALT4 mux port: MDIO of instance: fec. 101 Select mux mode: ALT5 mux port: SS1 of instance: ecspi1. 110 Select mux mode: ALT6 mux port: RDATA[2] of instance: fec. 111 Select mux mode: ALT7 mux port: RXACTIVE of instance: usbphy1.

43.3.15 IOMUXC_SW_MUX_CTL_PAD_KEY_ROW2 (IOMUXC_KEY_ROW2)

Address: IOMUXC_KEY_ROW2 is 53FA_8000h base + 38h offset = 53FA_8038h



IOMUXC_KEY_ROW2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad KEY_ROW2. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: KEY_ROW2. NOTE: Pad KEY_ROW2 is involved in Daisy Chain. - Config Register CAN1_IPP_IND_CANRX_SELECT_INPUT for mode ALT2. - Config Register ECSPI1_IPP_IND_SS_B_2_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: ROW[2] of instance: kpp. 001 Select mux mode: ALT1 mux port: GPIO[11] of instance: gpio4.

Table continues on the next page...

IOMUXC_KEY_ROW2 field descriptions (continued)

Field	Description
	010 Select mux mode: ALT2 mux port: RXCAN of instance: can1. 011 Select mux mode: ALT3 mux port: CTI_TRIGOUT7 of instance: p_platform_ne_32k_256k. 100 Select mux mode: ALT4 mux port: MDC of instance: fec. 101 Select mux mode: ALT5 mux port: SS2 of instance: ecspi1. 110 Select mux mode: ALT6 mux port: TDATA[2] of instance: fec. 111 Select mux mode: ALT7 mux port: RXERROR of instance: usbphy1.

43.3.16 IOMUXC_SW_MUX_CTL_PAD_KEY_COL3 (IOMUXC_KEY_COL3)

Address: IOMUXC_KEY_COL3 is 53FA_8000h base + 3Ch offset = 53FA_803Ch

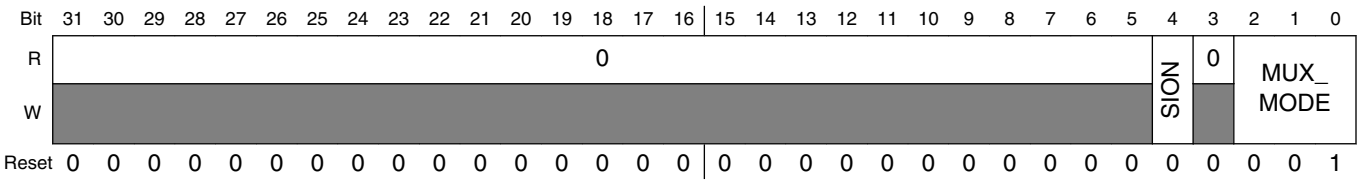
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_KEY_COL3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad KEY_COL3. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: KEY_COL3. NOTE: Pad KEY_COL3 is involved in Daisy Chain. - Config Register ECSPI1_IPP_IND_SS_B_3_SELECT_INPUT for mode ALT5. - Config Register I2C2_IPP_SCL_IN_SELECT_INPUT for mode ALT4. - Config Register SPDIF_SPDIF_IN1_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: COL[3] of instance: kpp. 001 Select mux mode: ALT1 mux port: GPIO[12] of instance: gpio4. 010 Select mux mode: ALT2 mux port: H2_DP of instance: usboh3. 011 Select mux mode: ALT3 mux port: IN1 of instance: spdif. 100 Select mux mode: ALT4 mux port: SCL of instance: i2c2. 101 Select mux mode: ALT5 mux port: SS3 of instance: ecspi1. 110 Select mux mode: ALT6 mux port: CRS of instance: fec. 111 Select mux mode: ALT7 mux port: SIECLOCK of instance: usbphy1.

43.3.17 IOMUXC_SW_MUX_CTL_PAD_KEY_ROW3 (IOMUXC_KEY_ROW3)

Address: IOMUXC_KEY_ROW3 is 53FA_8000h base + 40h offset = 53FA_8040h

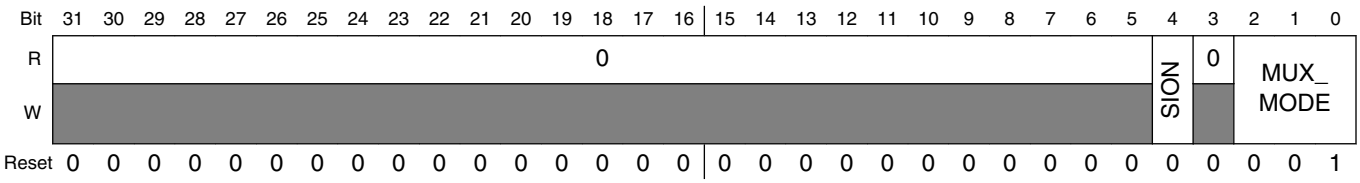


IOMUXC_KEY_ROW3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad KEY_ROW3. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: KEY_ROW3. NOTE: Pad KEY_ROW3 is involved in Daisy Chain. - Config Register CCM_IPP_ASRC_EXT_SELECT_INPUT for mode ALT3. - Config Register CCM_PLL4_BYPASS_CLK_SELECT_INPUT for mode ALT6. - Config Register I2C2_IPP_SDA_IN_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: ROW[3] of instance: kpp. 001 Select mux mode: ALT1 mux port: GPIO[13] of instance: gpio4. 010 Select mux mode: ALT2 mux port: H2_DM of instance: usboh3. 011 Select mux mode: ALT3 mux port: ASRC_EXT_CLK of instance: ccm. 100 Select mux mode: ALT4 mux port: SDA of instance: i2c2. 101 Select mux mode: ALT5 mux port: 32K_OUT of instance: osc32k. 110 Select mux mode: ALT6 mux port: PLL4_BYP of instance: ccm. 111 Select mux mode: ALT7 mux port: LINESTATE[0] of instance: usbphy1.

43.3.18 IOMUXC_SW_MUX_CTL_PAD_KEY_COL4 (IOMUXC_KEY_COL4)

Address: IOMUXC_KEY_COL4 is 53FA_8000h base + 44h offset = 53FA_8044h

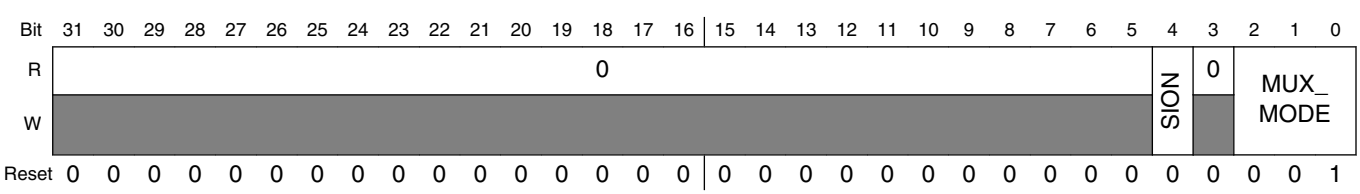


IOMUXC_KEY_COL4 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad KEY_COL4. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: KEY_COL4. NOTE: Pad KEY_COL4 is involved in Daisy Chain. - Config Register UART5_IPP_UART_RTS_B_SELECT_INPUT for mode ALT4. - Config Register USBOH3_IPP_IND_OTG_OC_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: COL[4] of instance: kpp. 001 Select mux mode: ALT1 mux port: GPIO[14] of instance: gpio4. 010 Select mux mode: ALT2 mux port: TXCAN of instance: can2. 011 Select mux mode: ALT3 mux port: SISG[4] of instance: ipu. 100 Select mux mode: ALT4 mux port: RTS of instance: uart5. 101 Select mux mode: ALT5 mux port: USBOTG_OC of instance: usboh3. 111 Select mux mode: ALT7 mux port: LINESTATE[1] of instance: usbphy1.

43.3.19 IOMUXC_SW_MUX_CTL_PAD_KEY_ROW4 (IOMUXC_KEY_ROW4)

Address: IOMUXC_KEY_ROW4 is 53FA_8000h base + 48h offset = 53FA_8048h

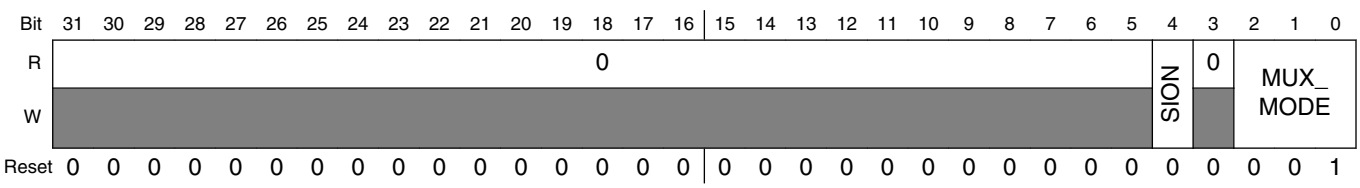


IOMUXC_KEY_ROW4 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad KEY_ROW4. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: KEY_ROW4. NOTE: Pad KEY_ROW4 is involved in Daisy Chain. - Config Register CAN2_IPP_IND_CANRX_SELECT_INPUT for mode ALT2. - Config Register UART5_IPP_UART_RTS_B_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: ROW[4] of instance: kpp. 001 Select mux mode: ALT1 mux port: GPIO[15] of instance: gpio4. 010 Select mux mode: ALT2 mux port: RXCAN of instance: can2. 011 Select mux mode: ALT3 mux port: SISG[5] of instance: ipu. 100 Select mux mode: ALT4 mux port: CTS of instance: uart5. 101 Select mux mode: ALT5 mux port: USBOTG_PWR of instance: usboh3. 111 Select mux mode: ALT7 mux port: VBUSVALID of instance: usbphy1.

43.3.20 IOMUXC_SW_MUX_CTL_PAD_DI0_DISP_CLK (IOMUXC_DI0_DISP_CLK)

Address: IOMUXC_DI0_DISP_CLK is 53FA_8000h base + 4Ch offset = 53FA_804Ch



IOMUXC_DI0_DISP_CLK field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DI0_DISP_CLK. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved

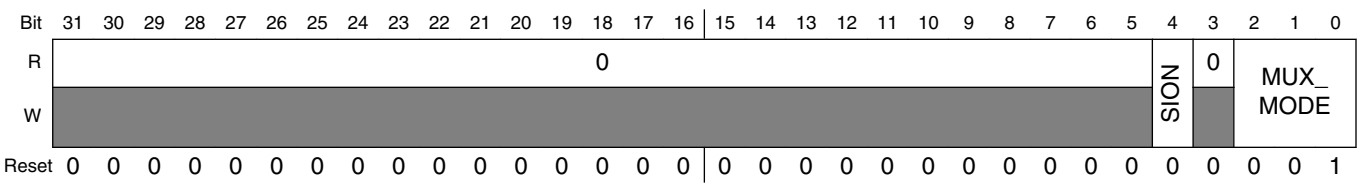
Table continues on the next page...

IOMUXC_DI0_DISP_CLK field descriptions (continued)

Field	Description
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: DI0_DISP_CLK. 000 Select mux mode: ALT0 mux port: DI0_DISP_CLK of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[16] of instance: gpio4. 010 Select mux mode: ALT2 mux port: USBH2_DIR of instance: usboh3. 101 Select mux mode: ALT5 mux port: DEBUG_CORE_STATE[0] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[0] of instance: emi. 111 Select mux mode: ALT7 mux port: AVALID of instance: usbphy1.

43.3.21 IOMUXC_SW_MUX_CTL_PAD_DI0_PIN15 (IOMUXC_DI0_PIN15)

Address: IOMUXC_DI0_PIN15 is 53FA_8000h base + 50h offset = 53FA_8050h

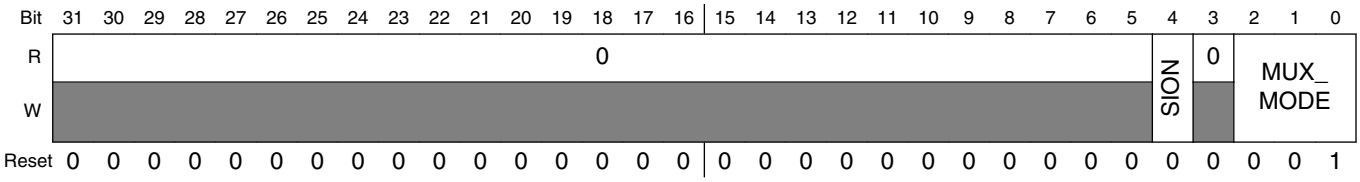


IOMUXC_DI0_PIN15 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DI0_PIN15. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: DI0_PIN15. 000 Select mux mode: ALT0 mux port: DI0_PIN15 of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[17] of instance: gpio4. 010 Select mux mode: ALT2 mux port: AUD6_TXC of instance: audmux. 101 Select mux mode: ALT5 mux port: DEBUG_CORE_STATE[1] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[1] of instance: emi. 111 Select mux mode: ALT7 mux port: BVALID of instance: usbphy1.

43.3.22 IOMUXC_SW_MUX_CTL_PAD_DI0_PIN2 (IOMUXC_DI0_PIN2)

Address: IOMUXC_DI0_PIN2 is 53FA_8000h base + 54h offset = 53FA_8054h

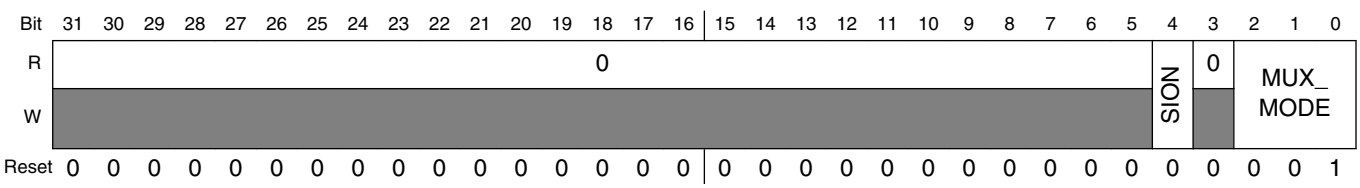


IOMUXC_DI0_PIN2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DI0_PIN2. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: DI0_PIN2. 000 Select mux mode: ALT0 mux port: DI0_PIN2 of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[18] of instance: gpio4. 010 Select mux mode: ALT2 mux port: AUD6_TXD of instance: audmux. 101 Select mux mode: ALT5 mux port: DEBUG_CORE_STATE[2] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[2] of instance: emi. 111 Select mux mode: ALT7 mux port: ENDSESSION of instance: usbphy1.

43.3.23 IOMUXC_SW_MUX_CTL_PAD_DI0_PIN3 (IOMUXC_DI0_PIN3)

Address: IOMUXC_DI0_PIN3 is 53FA_8000h base + 58h offset = 53FA_8058h



IOMUXC_DI0_PIN3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved

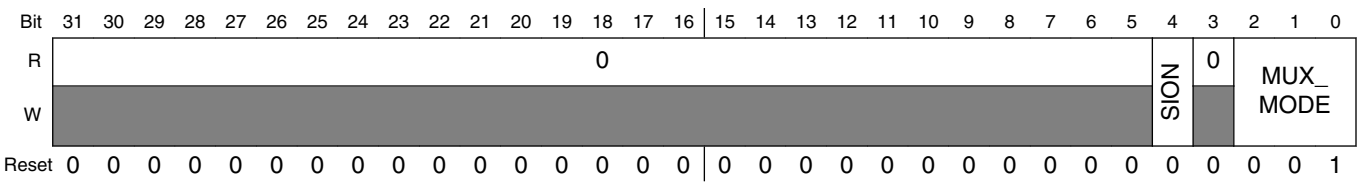
Table continues on the next page...

IOMUXC_DI0_PIN3 field descriptions (continued)

Field	Description
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DI0_PIN3. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: DI0_PIN3. 000 Select mux mode: ALT0 mux port: DI0_PIN3 of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[19] of instance: gpio4. 010 Select mux mode: ALT2 mux port: AUD6_TXFS of instance: audmux. 101 Select mux mode: ALT5 mux port: DEBUG_CORE_STATE[3] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[3] of instance: emi. 111 Select mux mode: ALT7 mux port: IDDIG of instance: usbphy1.

43.3.24 IOMUXC_SW_MUX_CTL_PAD_DI0_PIN4 (IOMUXC_DI0_PIN4)

Address: IOMUXC_DI0_PIN4 is 53FA_8000h base + 5Ch offset = 53FA_805Ch



IOMUXC_DI0_PIN4 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DI0_PIN4. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: DI0_PIN4. NOTE: Pad DI0_PIN4 is involved in Daisy Chain. - Config Register ESDHC1_IPP_WP_ON_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: DI0_PIN4 of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[20] of instance: gpio4. 010 Select mux mode: ALT2 mux port: AUD6_RXD of instance: audmux. 011 Select mux mode: ALT3 mux port: WP of instance: esdhc1.

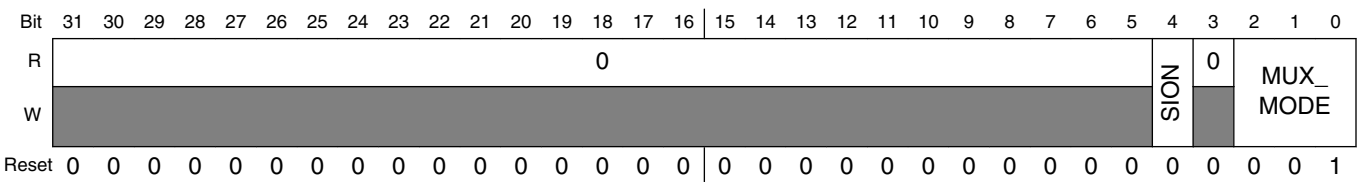
Table continues on the next page...

IOMUXC_DI0_PIN4 field descriptions (continued)

Field	Description
	101 Select mux mode: ALT5 mux port: DEBUG_YIELD of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[4] of instance: emi. 111 Select mux mode: ALT7 mux port: HOSTDISCONNECT of instance: usbphy1.

43.3.25 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT0 (IOMUXC_DISP0_DAT0)

Address: IOMUXC_DISP0_DAT0 is 53FA_8000h base + 60h offset = 53FA_8060h

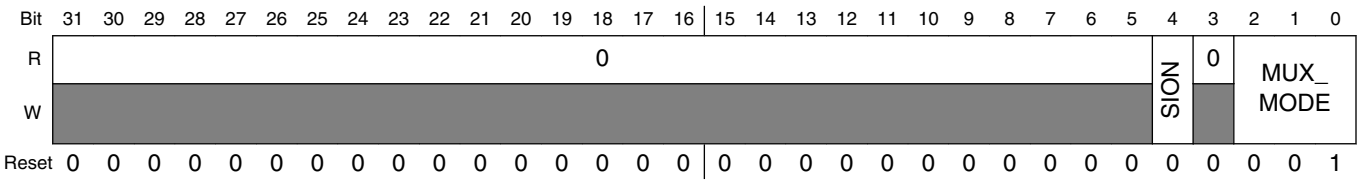


IOMUXC_DISP0_DAT0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT0. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: DISP0_DAT0. NOTE: Pad DISP0_DAT0 is involved in Daisy Chain. - Config Register CSPI_IPP_CSPI_CLK_IN_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: DISP0_DAT[0] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[21] of instance: gpio4. 010 Select mux mode: ALT2 mux port: SCLK of instance: cspi. 011 Select mux mode: ALT3 mux port: USBH2_DATA[0] of instance: usboh3. 101 Select mux mode: ALT5 mux port: DEBUG_CORE_RUN of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[5] of instance: emi. 111 Select mux mode: ALT7 mux port: TXREADY of instance: usbphy2.

43.3.26 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT1 (IOMUXC_DISP0_DAT1)

Address: IOMUXC_DISP0_DAT1 is 53FA_8000h base + 64h offset = 53FA_8064h

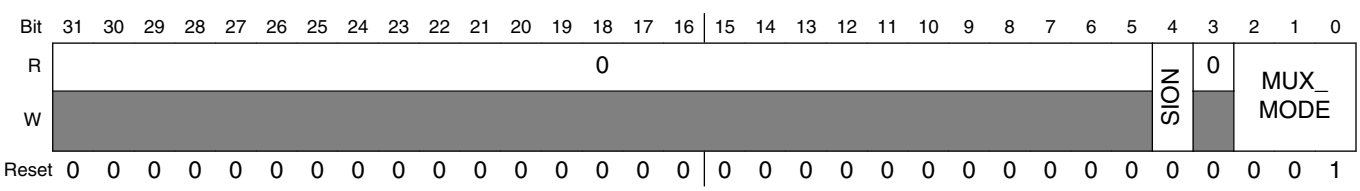


IOMUXC_DISP0_DAT1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT1. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: DISP0_DAT1. NOTE: Pad DISP0_DAT1 is involved in Daisy Chain. - Config Register CSPI_IPP_IND_MOSI_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: DISP0_DAT[1] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[22] of instance: gpio4. 010 Select mux mode: ALT2 mux port: MOSI of instance: cspi. 011 Select mux mode: ALT3 mux port: USBH2_DATA[1] of instance: usboh3. 101 Select mux mode: ALT5 mux port: DEBUG_EVENT_CHANNEL_SEL of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[6] of instance: emi. 111 Select mux mode: ALT7 mux port: RXVALID of instance: usbphy2.

43.3.27 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT2 (IOMUXC_DISP0_DAT2)

Address: IOMUXC_DISP0_DAT2 is 53FA_8000h base + 68h offset = 53FA_8068h

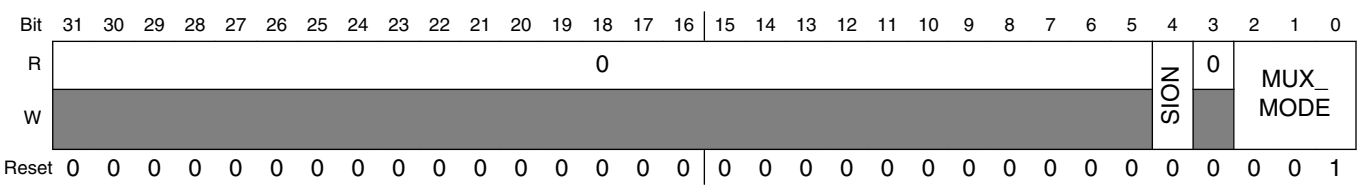


IOMUXC_DISP0_DAT2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT2. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: DISP0_DAT2. NOTE: Pad DISP0_DAT2 is involved in Daisy Chain. - Config Register CSPI_IPP_IND_MISO_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: DISP0_DAT[2] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[23] of instance: gpio4. 010 Select mux mode: ALT2 mux port: MISO of instance: cspi. 011 Select mux mode: ALT3 mux port: USBH2_DATA[2] of instance: usboh3. 101 Select mux mode: ALT5 mux port: DEBUG_MODE of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[7] of instance: emi. 111 Select mux mode: ALT7 mux port: RXACTIVE of instance: usbphy2.

43.3.28 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT3 (IOMUXC_DISP0_DAT3)

Address: IOMUXC_DISP0_DAT3 is 53FA_8000h base + 6Ch offset = 53FA_806Ch



IOMUXC_DISP0_DAT3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT3. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: DISP0_DAT3.

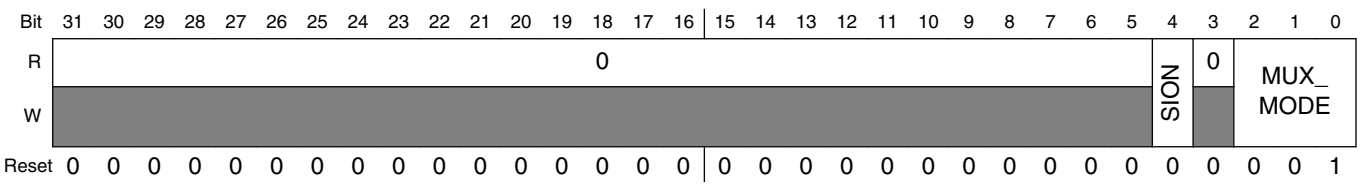
Table continues on the next page...

IOMUXC_DISP0_DAT3 field descriptions (continued)

Field	Description
	NOTE: Pad DISP0_DAT3 is involved in Daisy Chain. - Config Register CSPI_IPP_IND_SS0_B_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: DISP0_DAT[3] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[24] of instance: gpio4. 010 Select mux mode: ALT2 mux port: SS0 of instance: cspi. 011 Select mux mode: ALT3 mux port: USBH2_DATA[3] of instance: usboh3. 101 Select mux mode: ALT5 mux port: DEBUG_BUS_ERROR of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[8] of instance: emi. 111 Select mux mode: ALT7 mux port: RXERROR of instance: usbphy2.

43.3.29 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT4 (IOMUXC_DISP0_DAT4)

Address: IOMUXC_DISP0_DAT4 is 53FA_8000h base + 70h offset = 53FA_8070h

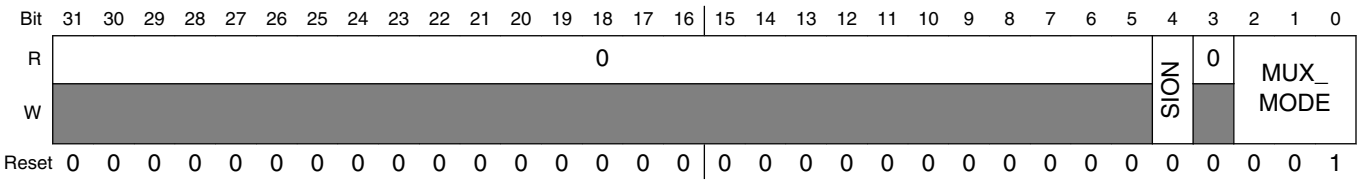


IOMUXC_DISP0_DAT4 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT4. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: DISP0_DAT4. NOTE: Pad DISP0_DAT4 is involved in Daisy Chain. - Config Register CSPI_IPP_IND_SS1_B_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: DISP0_DAT[4] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[25] of instance: gpio4. 010 Select mux mode: ALT2 mux port: SS1 of instance: cspi. 011 Select mux mode: ALT3 mux port: USBH2_DATA[4] of instance: usboh3. 101 Select mux mode: ALT5 mux port: DEBUG_BUS_RWB of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[9] of instance: emi. 111 Select mux mode: ALT7 mux port: SIECLOCK of instance: usbphy2.

43.3.30 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT5 (IOMUXC_DISP0_DAT5)

Address: IOMUXC_DISP0_DAT5 is 53FA_8000h base + 74h offset = 53FA_8074h

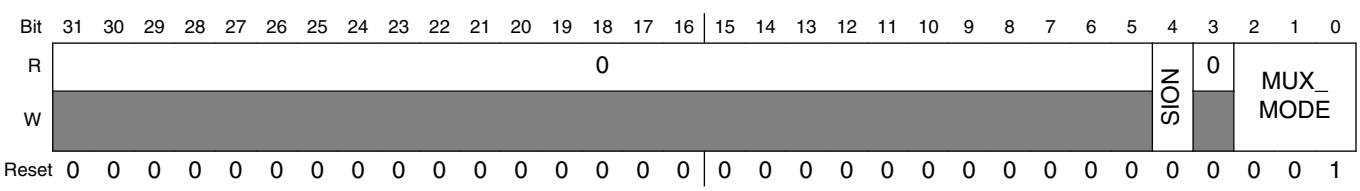


IOMUXC_DISP0_DAT5 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT5. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: DISP0_DAT5. NOTE: Pad DISP0_DAT5 is involved in Daisy Chain. - Config Register CSPI_IPP_IND_SS2_B_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: DISP0_DAT[5] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[26] of instance: gpio4. 010 Select mux mode: ALT2 mux port: SS2 of instance: cspi. 011 Select mux mode: ALT3 mux port: USBH2_DATA[5] of instance: usboh3. 101 Select mux mode: ALT5 mux port: DEBUG_MATCHED_DMBUS of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[10] of instance: emi. 111 Select mux mode: ALT7 mux port: LINESTATE[0] of instance: usbphy2.

43.3.31 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT6 (IOMUXC_DISP0_DAT6)

Address: IOMUXC_DISP0_DAT6 is 53FA_8000h base + 78h offset = 53FA_8078h

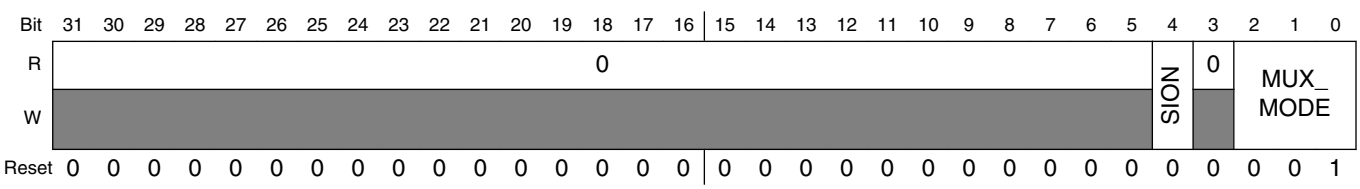


IOMUXC_DISP0_DAT6 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT6. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: DISP0_DAT6. NOTE: Pad DISP0_DAT6 is involved in Daisy Chain. - Config Register CSPI_IPP_IND_SS3_B_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: DISP0_DAT[6] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[27] of instance: gpio4. 010 Select mux mode: ALT2 mux port: SS3 of instance: cspi. 011 Select mux mode: ALT3 mux port: USBH2_DATA[6] of instance: usboh3. 101 Select mux mode: ALT5 mux port: DEBUG_RTBUFFER_WRITE of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[11] of instance: emi. 111 Select mux mode: ALT7 mux port: LINESTATE[1] of instance: usbphy2.

43.3.32 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT7 (IOMUXC_DISP0_DAT7)

Address: IOMUXC_DISP0_DAT7 is 53FA_8000h base + 7Ch offset = 53FA_807Ch



IOMUXC_DISP0_DAT7 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT7. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: DISP0_DAT7.

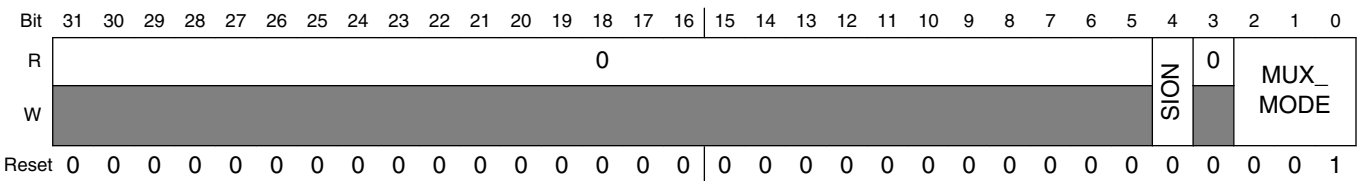
Table continues on the next page...

IOMUXC_DISP0_DAT7 field descriptions (continued)

Field	Description
	DEBUG_EVENT_CHANNEL[0] of instance: sdma. 000 Select mux mode: ALT0 mux port: DISP0_DAT[7] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[28] of instance: gpio4. 010 Select mux mode: ALT2 mux port: RDY of instance: cspi. 011 Select mux mode: ALT3 mux port: USBH2_DATA[7] of instance: usboh3. 101 Select mux mode: ALT5 mux port: 110 Select mux mode: ALT6 mux port: EMI_DEBUG[12] of instance: emi. 111 Select mux mode: ALT7 mux port: VBUSVALID of instance: usbphy2.

43.3.33 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT8 (IOMUXC_DISP0_DAT8)

Address: IOMUXC_DISP0_DAT8 is 53FA_8000h base + 80h offset = 53FA_8080h

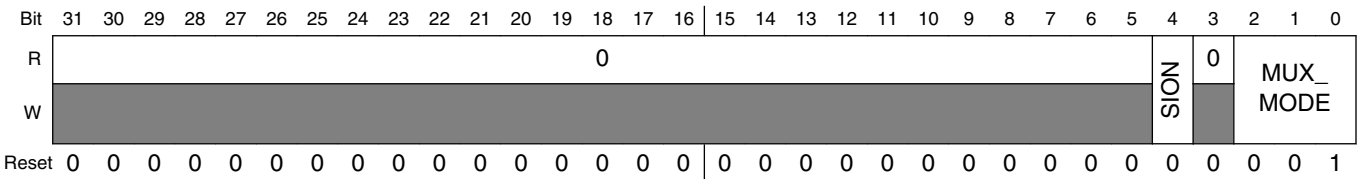


IOMUXC_DISP0_DAT8 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT8. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: DISP0_DAT8. DEBUG_EVENT_CHANNEL[1] of instance: sdma. 000 Select mux mode: ALT0 mux port: DISP0_DAT[8] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[29] of instance: gpio4. 010 Select mux mode: ALT2 mux port: PWMO of instance: pwm1. 011 Select mux mode: ALT3 mux port: WDOG_B of instance: wdog1. 101 Select mux mode: ALT5 mux port: 110 Select mux mode: ALT6 mux port: EMI_DEBUG[13] of instance: emi. 111 Select mux mode: ALT7 mux port: AVALID of instance: usbphy2.

43.3.34 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT9 (IOMUXC_DISP0_DAT9)

Address: IOMUXC_DISP0_DAT9 is 53FA_8000h base + 84h offset = 53FA_8084h

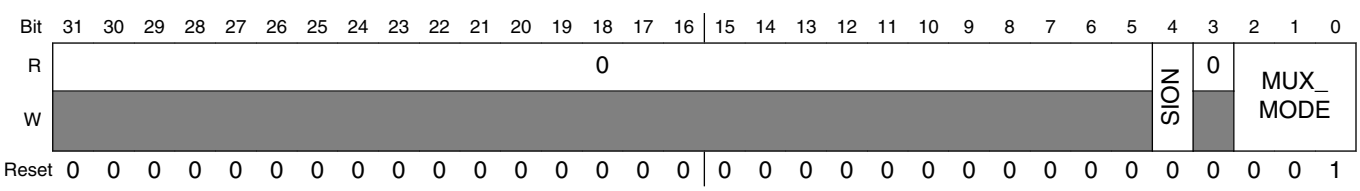


IOMUXC_DISP0_DAT9 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT9. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: DISP0_DAT9. DEBUG_EVENT_CHANNEL[2] of instance: sdma. 000 Select mux mode: ALT0 mux port: DISP0_DAT[9] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[30] of instance: gpio4. 010 Select mux mode: ALT2 mux port: PWMO of instance: pwm2. 011 Select mux mode: ALT3 mux port: WDOG_B of instance: wdog2. 101 Select mux mode: ALT5 mux port: 110 Select mux mode: ALT6 mux port: EMI_DEBUG[14] of instance: emi. 111 Select mux mode: ALT7 mux port: VSTATUS[0] of instance: usbphy2.

43.3.35 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT10 (IOMUXC_DISP0_DAT10)

Address: IOMUXC_DISP0_DAT10 is 53FA_8000h base + 88h offset = 53FA_8088h

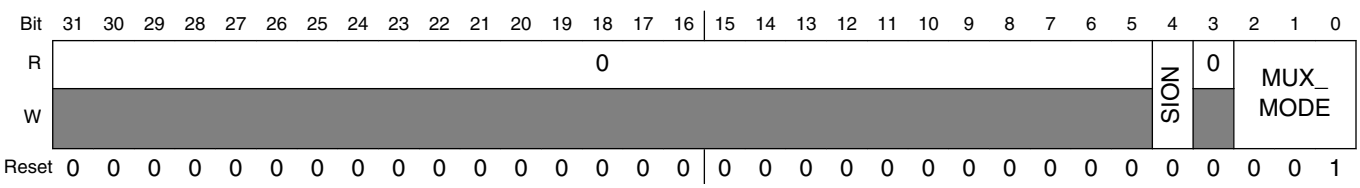


IOMUXC_DISP0_DAT10 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT10. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: DISP0_DAT10. DEBUG_EVENT_CHANNEL[3] of instance: sdma. 000 Select mux mode: ALT0 mux port: DISP0_DAT[10] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[31] of instance: gpio4. 010 Select mux mode: ALT2 mux port: USBH2_STP of instance: usboh3. 101 Select mux mode: ALT5 mux port: 110 Select mux mode: ALT6 mux port: EMI_DEBUG[15] of instance: emi. 111 Select mux mode: ALT7 mux port: VSTATUS[1] of instance: usbphy2.

43.3.36 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT11 (IOMUXC_DISP0_DAT11)

Address: IOMUXC_DISP0_DAT11 is 53FA_8000h base + 8Ch offset = 53FA_808Ch



IOMUXC_DISP0_DAT11 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT11. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: DISP0_DAT11. DEBUG_EVENT_CHANNEL[4] of instance: sdma. 000 Select mux mode: ALT0 mux port: DISP0_DAT[11] of instance: ipu.

Table continues on the next page...

IOMUXC_DISP0_DAT11 field descriptions (continued)

Field	Description
	001 Select mux mode: ALT1 mux port: GPIO[5] of instance: gpio5. 010 Select mux mode: ALT2 mux port: USBH2_NXT of instance: usboh3. 101 Select mux mode: ALT5 mux port: 110 Select mux mode: ALT6 mux port: EMI_DEBUG[16] of instance: emi. 111 Select mux mode: ALT7 mux port: VSTATUS[2] of instance: usbphy2.

43.3.37 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT12 (IOMUXC_DISP0_DAT12)

Address: IOMUXC_DISP0_DAT12 is 53FA_8000h base + 90h offset = 53FA_8090h

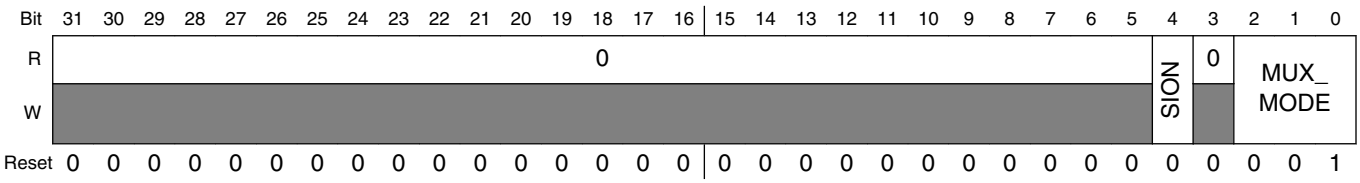
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_ MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_DISP0_DAT12 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT12. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: DISP0_DAT12. 000 Select mux mode: ALT0 mux port: DISP0_DAT[12] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[6] of instance: gpio5. 010 Select mux mode: ALT2 mux port: USBH2_CLK of instance: usboh3. 101 Select mux mode: ALT5 mux port: DEBUG_EVENT_CHANNEL[5] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[17] of instance: emi. 111 Select mux mode: ALT7 mux port: VSTATUS[3] of instance: usbphy2.

43.3.38 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT13 (IOMUXC_DISP0_DAT13)

Address: IOMUXC_DISP0_DAT13 is 53FA_8000h base + 94h offset = 53FA_8094h

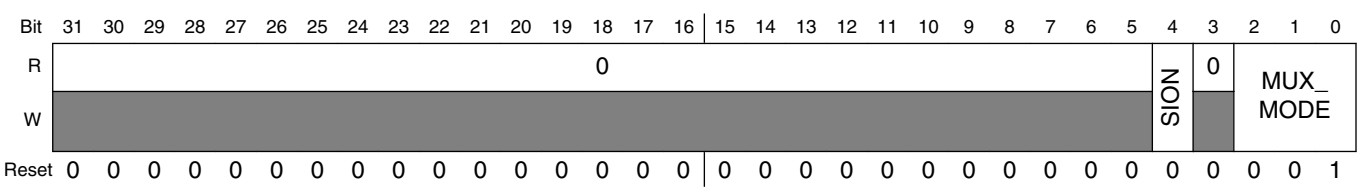


IOMUXC_DISP0_DAT13 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT13. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: DISP0_DAT13. NOTE: Pad DISP0_DAT13 is involved in Daisy Chain. - Config Register AUDMUX_P5_INPUT_RXFS_AMX_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: DISP0_DAT[13] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[7] of instance: gpio5. 011 Select mux mode: ALT3 mux port: AUD5_RXFS of instance: audmux. 101 Select mux mode: ALT5 mux port: DEBUG_EVT_CHN_LINES[0] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[18] of instance: emi. 111 Select mux mode: ALT7 mux port: VSTATUS[4] of instance: usbphy2.

43.3.39 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT14 (IOMUXC_DISP0_DAT14)

Address: IOMUXC_DISP0_DAT14 is 53FA_8000h base + 98h offset = 53FA_8098h



IOMUXC_DISP0_DAT14 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT14. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: DISP0_DAT14. NOTE: Pad DISP0_DAT14 is involved in Daisy Chain. - Config Register AUDMUX_P5_INPUT_RXCLK_AMX_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: DISP0_DAT[14] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[8] of instance: gpio5. 011 Select mux mode: ALT3 mux port: AUD5_RXC of instance: audmux. 101 Select mux mode: ALT5 mux port: DEBUG_EVT_CHN_LINES[1] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[19] of instance: emi. 111 Select mux mode: ALT7 mux port: VSTATUS[5] of instance: usbphy2.

43.3.40 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT15 (IOMUXC_DISP0_DAT15)

Address: IOMUXC_DISP0_DAT15 is 53FA_8000h base + 9Ch offset = 53FA_809Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_DISP0_DAT15 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT15. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: DISP0_DAT15. NOTE: Pad DISP0_DAT15 is involved in Daisy Chain.

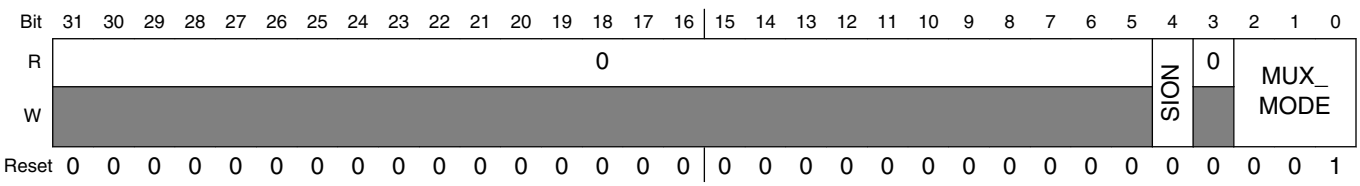
Table continues on the next page...

IOMUXC_DISP0_DAT15 field descriptions (continued)

Field	Description
	- Config Register ECSPI1_IPP_IND_SS_B_1_SELECT_INPUT for mode ALT2. - Config Register ECSPI2_IPP_IND_SS_B_1_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: DISP0_DAT[15] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[9] of instance: gpio5. 010 Select mux mode: ALT2 mux port: SS1 of instance: ecspi1. 011 Select mux mode: ALT3 mux port: SS1 of instance: ecspi2. 101 Select mux mode: ALT5 mux port: DEBUG_EVT_CHN_LINES[2] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[20] of instance: emi. 111 Select mux mode: ALT7 mux port: VSTATUS[6] of instance: usbphy2.

43.3.41 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT16 (IOMUXC_DISP0_DAT16)

Address: IOMUXC_DISP0_DAT16 is 53FA_8000h base + A0h offset = 53FA_80A0h



IOMUXC_DISP0_DAT16 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT16. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: DISP0_DAT16. NOTE: Pad DISP0_DAT16 is involved in Daisy Chain. - Config Register AUDMUX_P5_INPUT_TXCLK_AMX_SELECT_INPUT for mode ALT3. - Config Register ECSPI2_IPP_IND_MOSI_SELECT_INPUT for mode ALT2. - Config Register SDMA_EVENTS_14_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: DISP0_DAT[16] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[10] of instance: gpio5. 010 Select mux mode: ALT2 mux port: MOSI of instance: ecspi2. 011 Select mux mode: ALT3 mux port: AUD5_TXC of instance: audmux. 100 Select mux mode: ALT4 mux port: SDMA_EXT_EVENT[0] of instance: sdma.

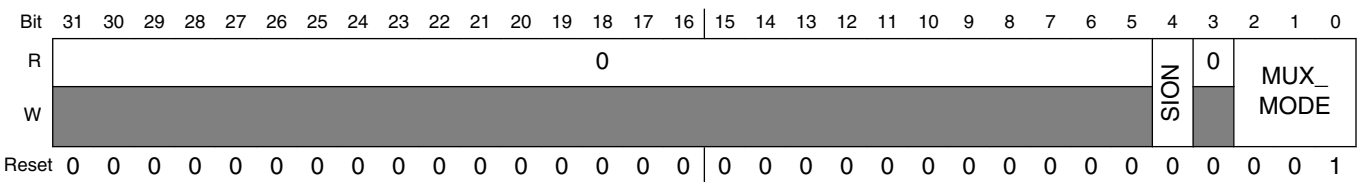
Table continues on the next page...

IOMUXC_DISP0_DAT16 field descriptions (continued)

Field	Description
	101 Select mux mode: ALT5 mux port: DEBUG_EVT_CHN_LINES[3] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[21] of instance: emi. 111 Select mux mode: ALT7 mux port: VSTATUS[7] of instance: usbphy2.

43.3.42 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT17 (IOMUXC_DISP0_DAT17)

Address: IOMUXC_DISP0_DAT17 is 53FA_8000h base + A4h offset = 53FA_80A4h

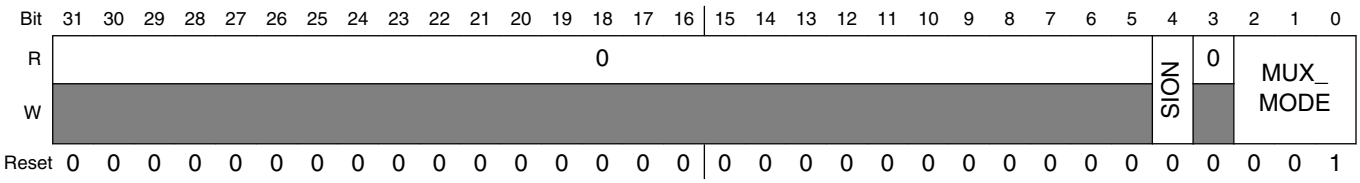


IOMUXC_DISP0_DAT17 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT17. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: DISP0_DAT17. NOTE: Pad DISP0_DAT17 is involved in Daisy Chain. - Config Register AUDMUX_P5_INPUT_DB_AMX_SELECT_INPUT for mode ALT3. - Config Register ECSPI2_IPP_IND_MISO_SELECT_INPUT for mode ALT2. - Config Register SDMA_EVENTS_15_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: DISP0_DAT[17] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[11] of instance: gpio5. 010 Select mux mode: ALT2 mux port: MISO of instance: ecspi2. 011 Select mux mode: ALT3 mux port: AUD5_TXD of instance: audmux. 100 Select mux mode: ALT4 mux port: SDMA_EXT_EVENT[1] of instance: sdma. 101 Select mux mode: ALT5 mux port: DEBUG_EVT_CHN_LINES[4] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[22] of instance: emi.

43.3.43 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT18 (IOMUXC_DISP0_DAT18)

Address: IOMUXC_DISP0_DAT18 is 53FA_8000h base + A8h offset = 53FA_80A8h

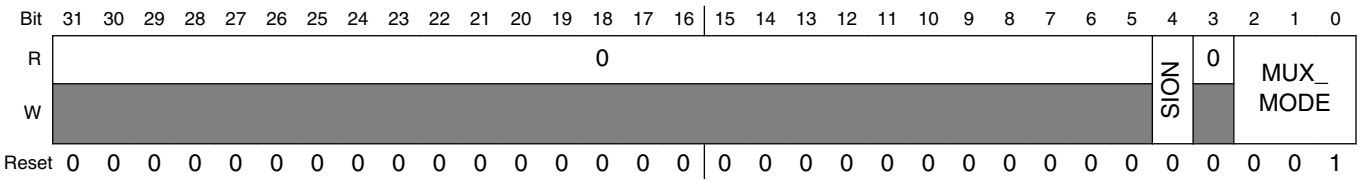


IOMUXC_DISP0_DAT18 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT18. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: DISP0_DAT18. NOTE: Pad DISP0_DAT18 is involved in Daisy Chain. - Config Register AUDMUX_P4_INPUT_RXFS_AMX_SELECT_INPUT for mode ALT4. - Config Register AUDMUX_P5_INPUT_TXFS_AMX_SELECT_INPUT for mode ALT3. - Config Register ECSPI2_IPP_IND_SS_B_0_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: DISP0_DAT[18] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[12] of instance: gpio5. 010 Select mux mode: ALT2 mux port: SS0 of instance: ecspi2. 011 Select mux mode: ALT3 mux port: AUD5_TXFS of instance: audmux. 100 Select mux mode: ALT4 mux port: AUD4_RXFS of instance: audmux. 101 Select mux mode: ALT5 mux port: DEBUG_EVT_CHN_LINES[5] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[23] of instance: emi. 111 Select mux mode: ALT7 mux port: WEIM_CS[2] of instance: emi.

43.3.44 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT19 (IOMUXC_DISP0_DAT19)

Address: IOMUXC_DISP0_DAT19 is 53FA_8000h base + ACh offset = 53FA_80ACh

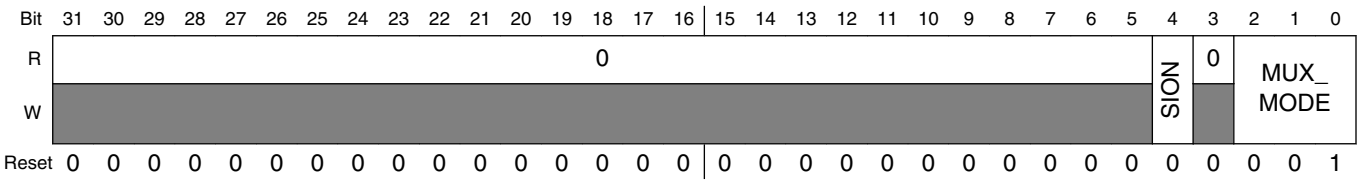


IOMUXC_DISP0_DAT19 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT19. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: DISP0_DAT19. NOTE: Pad DISP0_DAT19 is involved in Daisy Chain. - Config Register AUDMUX_P4_INPUT_RXCLK_AMX_SELECT_INPUT for mode ALT4. - Config Register AUDMUX_P5_INPUT_DA_AMX_SELECT_INPUT for mode ALT3. - Config Register ECSPI2_IPP_CSPI_CLK_IN_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: DISP0_DAT[19] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[13] of instance: gpio5. 010 Select mux mode: ALT2 mux port: SCLK of instance: ecspi2. 011 Select mux mode: ALT3 mux port: AUD5_RXD of instance: audmux. 100 Select mux mode: ALT4 mux port: AUD4_RXC of instance: audmux. 101 Select mux mode: ALT5 mux port: DEBUG_EVT_CHN_LINES[6] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[24] of instance: emi. 111 Select mux mode: ALT7 mux port: WEIM_CS[3] of instance: emi.

43.3.45 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT20 (IOMUXC_DISP0_DAT20)

Address: IOMUXC_DISP0_DAT20 is 53FA_8000h base + B0h offset = 53FA_80B0h

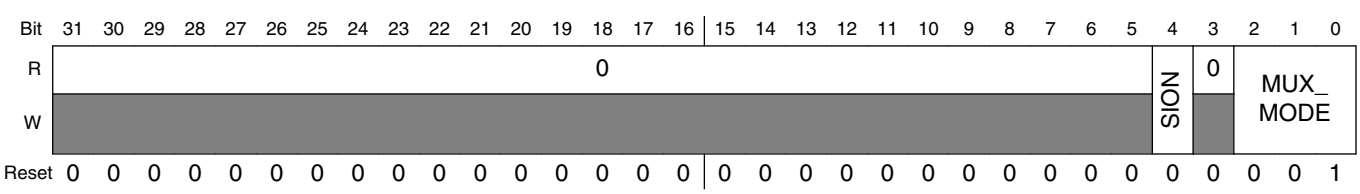


IOMUXC_DISP0_DAT20 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT20. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: DISP0_DAT20. 000 Select mux mode: ALT0 mux port: DISP0_DAT[20] of instance: ipu. NOTE: Pad DISP0_DAT20 is involved in Daisy Chain. - Config Register AUDMUX_P4_INPUT_TXCLK_AMX_SELECT_INPUT for mode ALT3. - Config Register ECSPI1_IPP_CSPI_CLK_IN_SELECT_INPUT for mode ALT2. 001 Select mux mode: ALT1 mux port: GPIO[14] of instance: gpio5. 010 Select mux mode: ALT2 mux port: SCLK of instance: ecspi1. 011 Select mux mode: ALT3 mux port: AUD4_TXC of instance: audmux. 101 Select mux mode: ALT5 mux port: DEBUG_EVT_CHN_LINES[7] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[25] of instance: emi. 111 Select mux mode: ALT7 mux port: TDI of instance: sata_phy.

43.3.46 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT21 (IOMUXC_DISP0_DAT21)

Address: IOMUXC_DISP0_DAT21 is 53FA_8000h base + B4h offset = 53FA_80B4h

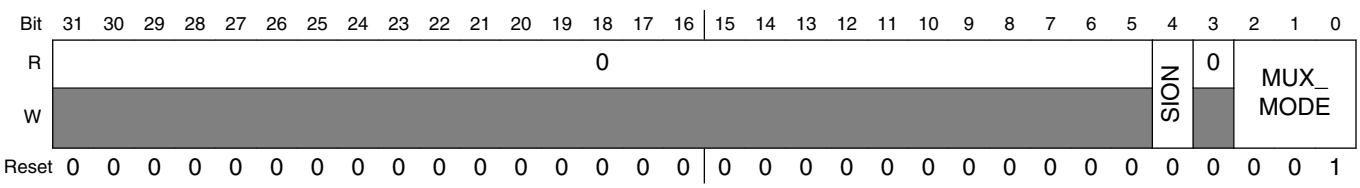


IOMUXC_DISP0_DAT21 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT21. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: DISP0_DAT21. NOTE: Pad DISP0_DAT21 is involved in Daisy Chain. - Config Register AUDMUX_P4_INPUT_DB_AMX_SELECT_INPUT for mode ALT3. - Config Register ECSPI1_IPP_IND_MOSI_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: DISP0_DAT[21] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[15] of instance: gpio5. 010 Select mux mode: ALT2 mux port: MOSI of instance: ecspi1. 011 Select mux mode: ALT3 mux port: AUD4_TXD of instance: audmux. 101 Select mux mode: ALT5 mux port: DEBUG_BUS_DEVICE[0] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[26] of instance: emi. 111 Select mux mode: ALT7 mux port: TDO of instance: sata_phy.

43.3.47 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT22 (IOMUXC_DISP0_DAT22)

Address: IOMUXC_DISP0_DAT22 is 53FA_8000h base + B8h offset = 53FA_80B8h



IOMUXC_DISP0_DAT22 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT22. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved

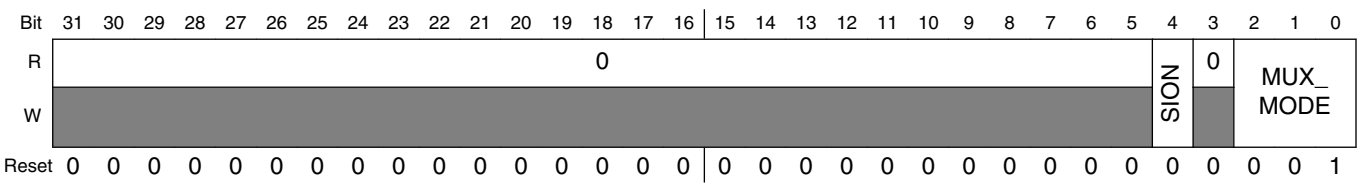
Table continues on the next page...

IOMUXC_DISP0_DAT22 field descriptions (continued)

Field	Description
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: DISP0_DAT22. NOTE: Pad DISP0_DAT22 is involved in Daisy Chain. - Config Register AUDMUX_P4_INPUT_TXFS_AMX_SELECT_INPUT for mode ALT3. - Config Register ECSPI1_IPP_IND_MISO_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: DISP0_DAT[22] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[16] of instance: gpio5. 010 Select mux mode: ALT2 mux port: MISO of instance: ecspi1. 011 Select mux mode: ALT3 mux port: AUD4_TXFS of instance: audmux. 101 Select mux mode: ALT5 mux port: DEBUG_BUS_DEVICE[1] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[27] of instance: emi. 111 Select mux mode: ALT7 mux port: TCK of instance: sata_phy.

43.3.48 IOMUXC_SW_MUX_CTL_PAD_DISP0_DAT23 (IOMUXC_DISP0_DAT23)

Address: IOMUXC_DISP0_DAT23 is 53FA_8000h base + BCh offset = 53FA_80BCh



IOMUXC_DISP0_DAT23 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad DISP0_DAT23. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: DISP0_DAT23. NOTE: Pad DISP0_DAT23 is involved in Daisy Chain. - Config Register AUDMUX_P4_INPUT_DA_AMX_SELECT_INPUT for mode ALT3. - Config Register ECSPI1_IPP_IND_SS_B_0_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: DISP0_DAT[23] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[17] of instance: gpio5. 010 Select mux mode: ALT2 mux port: SS0 of instance: ecspi1.

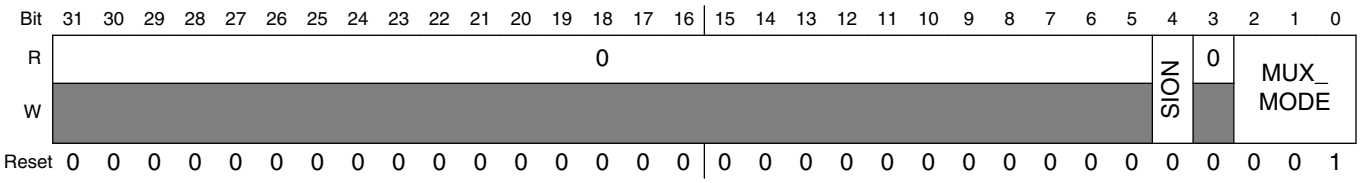
Table continues on the next page...

IOMUXC_DISP0_DAT23 field descriptions (continued)

Field	Description
	011 Select mux mode: ALT3 mux port: AUD4_RXD of instance: audmux. 101 Select mux mode: ALT5 mux port: DEBUG_BUS_DEVICE[2] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[28] of instance: emi. 111 Select mux mode: ALT7 mux port: TMS of instance: sata_phy.

43.3.49 IOMUXC_SW_MUX_CTL_PAD_CSI0_PIXCLK (IOMUXC_CSI0_PIXCLK)

Address: IOMUXC_CSI0_PIXCLK is 53FA_8000h base + C0h offset = 53FA_80C0h

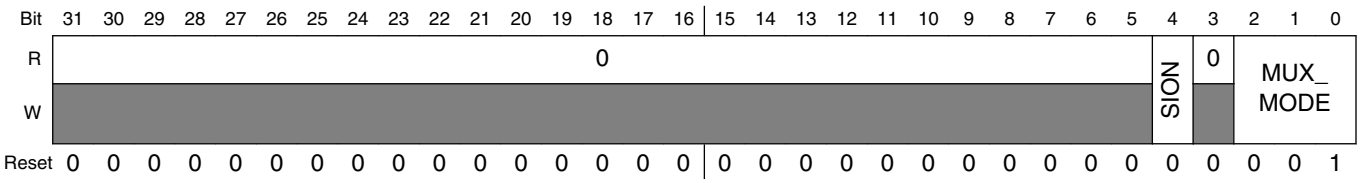


IOMUXC_CSI0_PIXCLK field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_PIXCLK. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: CSI0_PIXCLK. 000 Select mux mode: ALT0 mux port: CSI0_PIXCLK of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[18] of instance: gpio5. 101 Select mux mode: ALT5 mux port: DEBUG_PC[0] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[29] of instance: emi.

43.3.50 IOMUXC_SW_MUX_CTL_PAD_CSI0_MCLK (IOMUXC_CSI0_MCLK)

Address: IOMUXC_CSI0_MCLK is 53FA_8000h base + C4h offset = 53FA_80C4h

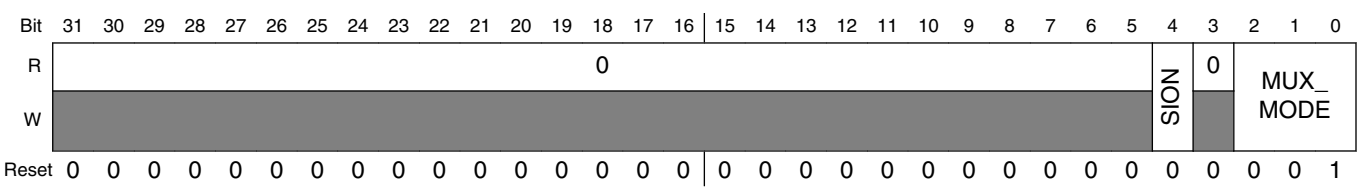


IOMUXC_CSI0_MCLK field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_MCLK. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: CSI0_MCLK. 000 Select mux mode: ALT0 mux port: CSI0_HSYNC of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[19] of instance: gpio5. 010 Select mux mode: ALT2 mux port: CSI0_MCLK of instance: ccm. 101 Select mux mode: ALT5 mux port: DEBUG_PC[1] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[30] of instance: emi. 111 Select mux mode: ALT7 mux port: TRCTL of instance: tpiu.

43.3.51 IOMUXC_SW_MUX_CTL_PAD_CSI0_DATA_EN (IOMUXC_CSI0_DATA_EN)

Address: IOMUXC_CSI0_DATA_EN is 53FA_8000h base + C8h offset = 53FA_80C8h

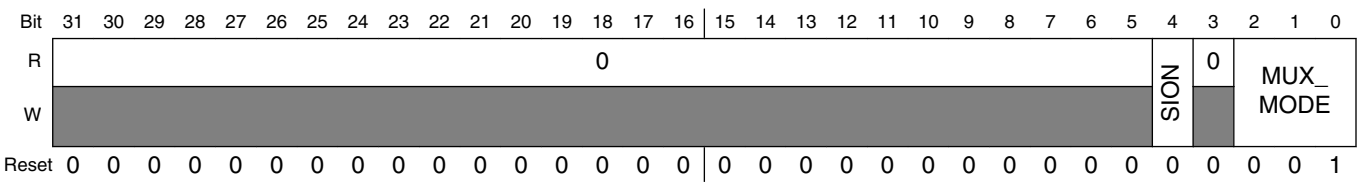


IOMUXC_CSI0_DATA_EN field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_DATA_EN. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: CSI0_DATA_EN. 000 Select mux mode: ALT0 mux port: CSI0_DATA_EN of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[20] of instance: gpio5. 101 Select mux mode: ALT5 mux port: DEBUG_PC[2] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[31] of instance: emi. 111 Select mux mode: ALT7 mux port: TRCLK of instance: tpiu.

43.3.52 IOMUXC_SW_MUX_CTL_PAD_CSI0_VSYNC (IOMUXC_CSI0_VSYNC)

Address: IOMUXC_CSI0_VSYNC is 53FA_8000h base + CCh offset = 53FA_80CCh

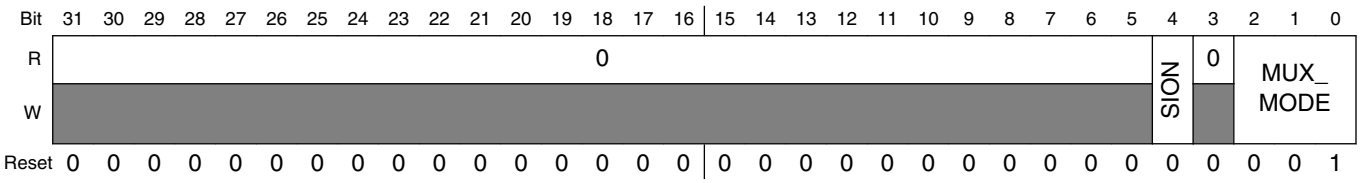


IOMUXC_CSI0_VSYNC field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_VSYNC. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: CSI0_VSYNC. 000 Select mux mode: ALT0 mux port: CSI0_VSYNC of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[21] of instance: gpio5. 101 Select mux mode: ALT5 mux port: DEBUG_PC[3] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[32] of instance: emi. 111 Select mux mode: ALT7 mux port: TRACE[0] of instance: tpiu.

43.3.53 IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT4 (IOMUXC_CSI0_DAT4)

Address: IOMUXC_CSI0_DAT4 is 53FA_8000h base + D0h offset = 53FA_80D0h

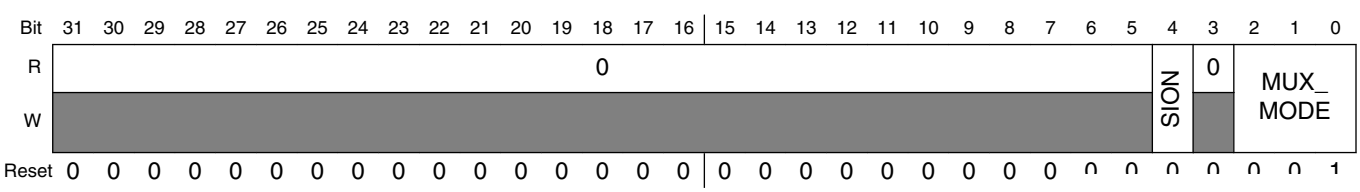


IOMUXC_CSI0_DAT4 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_DAT4. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: CSI0_DAT4. NOTE: Pad CSI0_DAT4 is involved in Daisy Chain. - Config Register ECSPI1_IPP_CSPI_CLK_IN_SELECT_INPUT for mode ALT3. - Config Register KPP_IPP_IND_COL_5_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CSI0_D[4] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[22] of instance: gpio5. 010 Select mux mode: ALT2 mux port: COL[5] of instance: kpp. 011 Select mux mode: ALT3 mux port: SCLK of instance: ecspi1. 100 Select mux mode: ALT4 mux port: USBH3_STP of instance: usboh3. 101 Select mux mode: ALT5 mux port: AUD3_TXC of instance: audmux. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[33] of instance: emi. 111 Select mux mode: ALT7 mux port: TRACE[1] of instance: tpiu.

43.3.54 IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT5 (IOMUXC_CSI0_DAT5)

Address: IOMUXC_CSI0_DAT5 is 53FA_8000h base + D4h offset = 53FA_80D4h



IOMUXC_CSI0_DAT5 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_DAT5. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: CSI0_DAT5. NOTE: Pad CSI0_DAT5 is involved in Daisy Chain. - Config Register ECSPI1_IPP_IND_MOSI_SELECT_INPUT for mode ALT3. - Config Register KPP_IPP_IND_ROW_5_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CSI0_D[5] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[23] of instance: gpio5. 010 Select mux mode: ALT2 mux port: ROW[5] of instance: kpp. 011 Select mux mode: ALT3 mux port: MOSI of instance: ecspi1. 100 Select mux mode: ALT4 mux port: USBH3_NXT of instance: usboh3. 101 Select mux mode: ALT5 mux port: AUD3_TXD of instance: audmux. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[34] of instance: emi. 111 Select mux mode: ALT7 mux port: TRACE[2] of instance: tpiu.

43.3.55 IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT6 (IOMUXC_CSI0_DAT6)

Address: IOMUXC_CSI0_DAT6 is 53FA_8000h base + D8h offset = 53FA_80D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_ MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_CSI0_DAT6 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_DAT6. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved

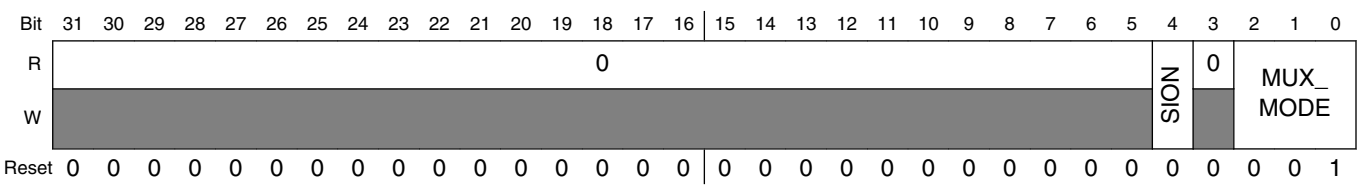
Table continues on the next page...

IOMUXC_CSI0_DAT6 field descriptions (continued)

Field	Description
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: CSI0_DAT6. NOTE: Pad CSI0_DAT6 is involved in Daisy Chain. - Config Register ECSPI1_IPP_IND_MISO_SELECT_INPUT for mode ALT3. - Config Register KPP_IPP_IND_COL_6_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CSI0_D[6] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[24] of instance: gpio5. 010 Select mux mode: ALT2 mux port: COL[6] of instance: kpp. 011 Select mux mode: ALT3 mux port: MISO of instance: ecspi1. 100 Select mux mode: ALT4 mux port: USBH3_CLK of instance: usboh3. 101 Select mux mode: ALT5 mux port: AUD3_TXFS of instance: audmux. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[35] of instance: emi. 111 Select mux mode: ALT7 mux port: TRACE[3] of instance: tpiu.

43.3.56 IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT7 (IOMUXC_CSI0_DAT7)

Address: IOMUXC_CSI0_DAT7 is 53FA_8000h base + DCh offset = 53FA_80DCh



IOMUXC_CSI0_DAT7 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_DAT7. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: CSI0_DAT7. NOTE: Pad CSI0_DAT7 is involved in Daisy Chain. - Config Register ECSPI1_IPP_IND_SS_B_0_SELECT_INPUT for mode ALT3. - Config Register KPP_IPP_IND_ROW_6_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CSI0_D[7] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[25] of instance: gpio5.

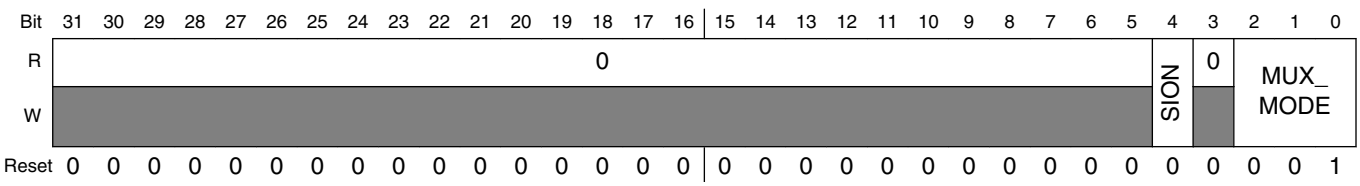
Table continues on the next page...

IOMUXC_CSI0_DAT7 field descriptions (continued)

Field	Description
	010 Select mux mode: ALT2 mux port: ROW[6] of instance: kpp. 011 Select mux mode: ALT3 mux port: SS0 of instance: ecspi1. 100 Select mux mode: ALT4 mux port: USBH3_DIR of instance: usboh3. 101 Select mux mode: ALT5 mux port: AUD3_RXD of instance: audmux. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[36] of instance: emi. 111 Select mux mode: ALT7 mux port: TRACE[4] of instance: tpiu.

43.3.57 IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT8 (IOMUXC_CSI0_DAT8)

Address: IOMUXC_CSI0_DAT8 is 53FA_8000h base + E0h offset = 53FA_80E0h

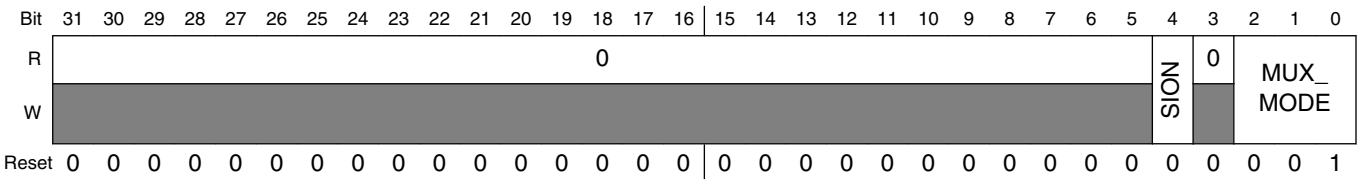


IOMUXC_CSI0_DAT8 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_DAT8. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: CSI0_DAT8. NOTE: Pad CSI0_DAT8 is involved in Daisy Chain. - Config Register ECSPI2_IPP_CSPI_CLK_IN_SELECT_INPUT for mode ALT3. - Config Register I2C1_IPP_SDA_IN_SELECT_INPUT for mode ALT5. - Config Register KPP_IPP_IND_COL_7_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CSI0_D[8] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[26] of instance: gpio5. 010 Select mux mode: ALT2 mux port: COL[7] of instance: kpp. 011 Select mux mode: ALT3 mux port: SCLK of instance: ecspi2. 100 Select mux mode: ALT4 mux port: USBH3_OC of instance: usboh3. 101 Select mux mode: ALT5 mux port: SDA of instance: i2c1. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[37] of instance: emi. 111 Select mux mode: ALT7 mux port: TRACE[5] of instance: tpiu.

43.3.58 IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT9 (IOMUXC_CSI0_DAT9)

Address: IOMUXC_CSI0_DAT9 is 53FA_8000h base + E4h offset = 53FA_80E4h

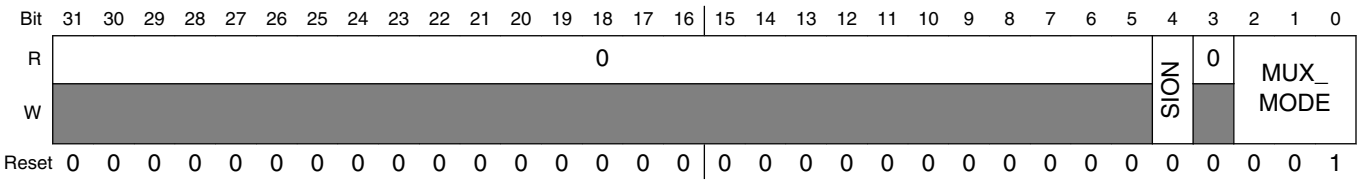


IOMUXC_CSI0_DAT9 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_DAT9. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: CSI0_DAT9. NOTE: Pad CSI0_DAT9 is involved in Daisy Chain. - Config Register ECSPI2_IPP_IND_MOSI_SELECT_INPUT for mode ALT3. - Config Register I2C1_IPP_SCL_IN_SELECT_INPUT for mode ALT5. - Config Register KPP_IPP_IND_ROW_7_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CSI0_D[9] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[27] of instance: gpio5. 010 Select mux mode: ALT2 mux port: ROW[7] of instance: kpp. 011 Select mux mode: ALT3 mux port: MOSI of instance: ecspi2. 100 Select mux mode: ALT4 mux port: USBH3_PWR of instance: usboh3. 101 Select mux mode: ALT5 mux port: SCL of instance: i2c1. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[38] of instance: emi. 111 Select mux mode: ALT7 mux port: TRACE[6] of instance: tpiu.

43.3.59 IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT10 (IOMUXC_CSI0_DAT10)

Address: IOMUXC_CSI0_DAT10 is 53FA_8000h base + E8h offset = 53FA_80E8h

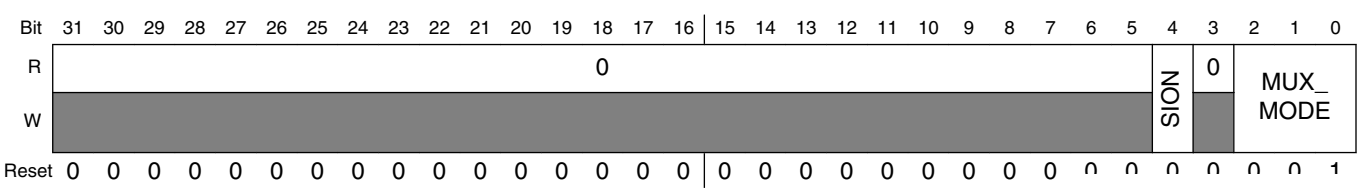


IOMUXC_CSI0_DAT10 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_DAT10. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: CSI0_DAT10. NOTE: Pad CSI0_DAT10 is involved in Daisy Chain. - Config Register ECSPI2_IPP_IND_MISO_SELECT_INPUT for mode ALT3. - Config Register UART1_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CSI0_D[10] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[28] of instance: gpio5. 010 Select mux mode: ALT2 mux port: TXD_MUX of instance: uart1. 011 Select mux mode: ALT3 mux port: MISO of instance: ecspi2. 100 Select mux mode: ALT4 mux port: AUD3_RXC of instance: audmux. 101 Select mux mode: ALT5 mux port: DEBUG_PC[4] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[39] of instance: emi. 111 Select mux mode: ALT7 mux port: TRACE[7] of instance: tpiu.

43.3.60 IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT11 (IOMUXC_CSI0_DAT11)

Address: IOMUXC_CSI0_DAT11 is 53FA_8000h base + ECh offset = 53FA_80ECh



IOMUXC_CSI0_DAT11 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_DAT11. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: CSI0_DAT11. NOTE: Pad CSI0_DAT11 is involved in Daisy Chain. - Config Register ECSPI2_IPP_IND_SS_B_0_SELECT_INPUT for mode ALT3. - Config Register UART1_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CSI0_D[11] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[29] of instance: gpio5. 010 Select mux mode: ALT2 mux port: RXD_MUX of instance: uart1. 011 Select mux mode: ALT3 mux port: SS0 of instance: ecspi2. 100 Select mux mode: ALT4 mux port: AUD3_RXFS of instance: audmux. 101 Select mux mode: ALT5 mux port: DEBUG_PC[5] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[40] of instance: emi. 111 Select mux mode: ALT7 mux port: TRACE[8] of instance: tpiu.

43.3.61 IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT12 (IOMUXC_CSI0_DAT12)

Address: IOMUXC_CSI0_DAT12 is 53FA_8000h base + F0h offset = 53FA_80F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_ MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_CSI0_DAT12 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_DAT12. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_CSI0_DAT12 field descriptions (continued)

Field	Description
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: CSI0_DAT12. NOTE: Pad CSI0_DAT12 is involved in Daisy Chain. - Config Register UART4_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CSI0_D[12] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[30] of instance: gpio5. 010 Select mux mode: ALT2 mux port: TXD_MUX of instance: uart4. 100 Select mux mode: ALT4 mux port: USBH3_DATA[0] of instance: usboh3. 101 Select mux mode: ALT5 mux port: DEBUG_PC[6] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[41] of instance: emi. 111 Select mux mode: ALT7 mux port: TRACE[9] of instance: tpiu.

43.3.62 IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT13 (IOMUXC_CSI0_DAT13)

Address: IOMUXC_CSI0_DAT13 is 53FA_8000h base + F4h offset = 53FA_80F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_CSI0_DAT13 field descriptions

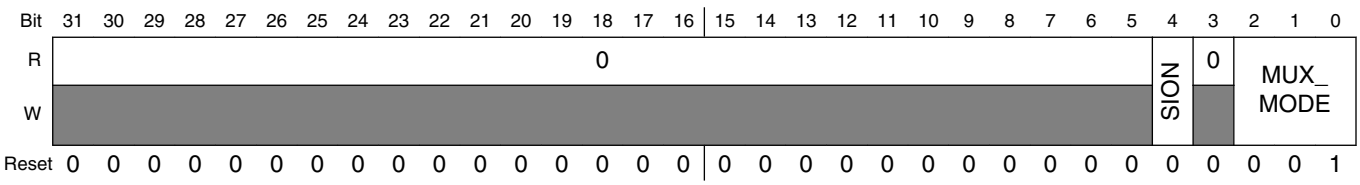
Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_DAT13. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: CSI0_DAT13. NOTE: Pad CSI0_DAT13 is involved in Daisy Chain. - Config Register UART4_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CSI0_D[13] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[31] of instance: gpio5. 010 Select mux mode: ALT2 mux port: RXD_MUX of instance: uart4. 100 Select mux mode: ALT4 mux port: USBH3_DATA[1] of instance: usboh3. 101 Select mux mode: ALT5 mux port: DEBUG_PC[7] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[42] of instance: emi. 111 Select mux mode: ALT7 mux port: TRACE[10] of instance: tpiu.

IOMUXC_CSI0_DAT13 field descriptions (continued)

Field	Description

43.3.63 IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT14 (IOMUXC_CSI0_DAT14)

Address: IOMUXC_CSI0_DAT14 is 53FA_8000h base + F8h offset = 53FA_80F8h

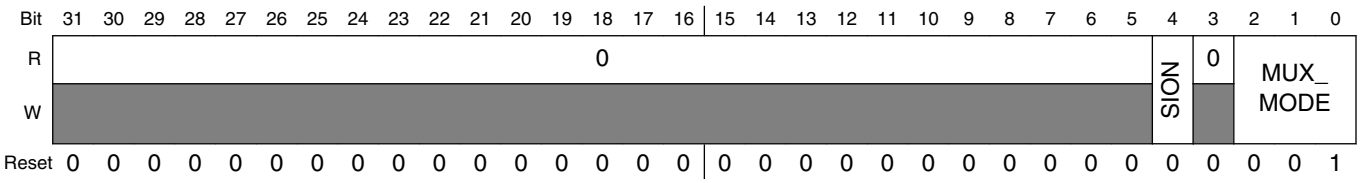


IOMUXC_CSI0_DAT14 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_DAT14. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: CSI0_DAT14. NOTE: Pad CSI0_DAT14 is involved in Daisy Chain. - Config Register UART5_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CSI0_D[14] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[0] of instance: gpio6. 010 Select mux mode: ALT2 mux port: TXD_MUX of instance: uart5. 100 Select mux mode: ALT4 mux port: USBH3_DATA[2] of instance: usboh3. 101 Select mux mode: ALT5 mux port: DEBUG_PC[8] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[43] of instance: emi. 111 Select mux mode: ALT7 mux port: TRACE[11] of instance: tpiu.

43.3.64 IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT15 (IOMUXC_CSI0_DAT15)

Address: IOMUXC_CSI0_DAT15 is 53FA_8000h base + FCh offset = 53FA_80FCh

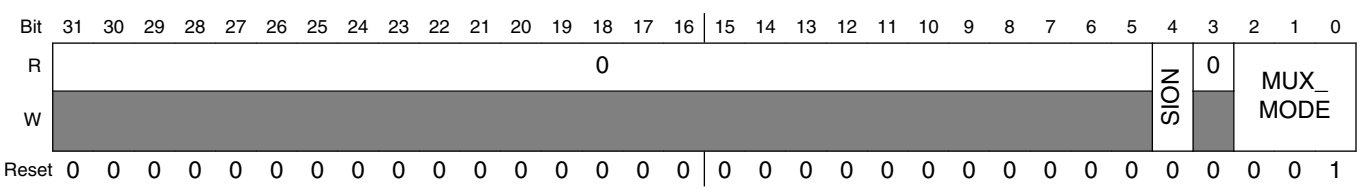


IOMUXC_CSI0_DAT15 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_DAT15. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: CSI0_DAT15. NOTE: Pad CSI0_DAT15 is involved in Daisy Chain. - Config Register UART5_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CSI0_D[15] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[1] of instance: gpio6. 010 Select mux mode: ALT2 mux port: RXD_MUX of instance: uart5. 100 Select mux mode: ALT4 mux port: USBH3_DATA[3] of instance: usboh3. 101 Select mux mode: ALT5 mux port: DEBUG_PC[9] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[44] of instance: emi. 111 Select mux mode: ALT7 mux port: TRACE[12] of instance: tpiu.

43.3.65 IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT16 (IOMUXC_CSI0_DAT16)

Address: IOMUXC_CSI0_DAT16 is 53FA_8000h base + 100h offset = 53FA_8100h

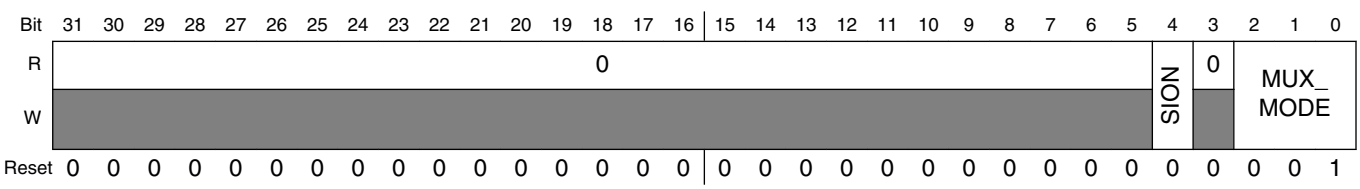


IOMUXC_CSI0_DAT16 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_DAT16. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: CSI0_DAT16. NOTE: Pad CSI0_DAT16 is involved in Daisy Chain. - Config Register UART4_IPP_UART_RTS_B_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CSI0_D[16] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[2] of instance: gpio6. 010 Select mux mode: ALT2 mux port: RTS of instance: uart4. 100 Select mux mode: ALT4 mux port: USBH3_DATA[4] of instance: usboh3. 101 Select mux mode: ALT5 mux port: DEBUG_PC[10] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[45] of instance: emi. 111 Select mux mode: ALT7 mux port: TRACE[13] of instance: tpiu.

43.3.66 IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT17 (IOMUXC_CSI0_DAT17)

Address: IOMUXC_CSI0_DAT17 is 53FA_8000h base + 104h offset = 53FA_8104h



IOMUXC_CSI0_DAT17 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_DAT17. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: CSI0_DAT17.

Table continues on the next page...

IOMUXC_CSI0_DAT17 field descriptions (continued)

Field	Description
	NOTE: Pad CSI0_DAT17 is involved in Daisy Chain. - Config Register UART4_IPP_UART_RTS_B_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CSI0_D[17] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[3] of instance: gpio6. 010 Select mux mode: ALT2 mux port: CTS of instance: uart4. 100 Select mux mode: ALT4 mux port: USBH3_DATA[5] of instance: usboh3. 101 Select mux mode: ALT5 mux port: DEBUG_PC[11] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[46] of instance: emi. 111 Select mux mode: ALT7 mux port: TRACE[14] of instance: tpiu.

43.3.67 IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT18 (IOMUXC_CSI0_DAT18)

Address: IOMUXC_CSI0_DAT18 is 53FA_8000h base + 108h offset = 53FA_8108h

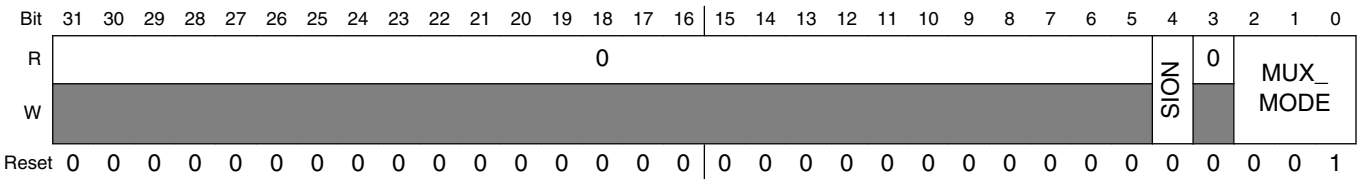
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_ MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_CSI0_DAT18 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_DAT18. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: CSI0_DAT18. NOTE: Pad CSI0_DAT18 is involved in Daisy Chain. - Config Register UART5_IPP_UART_RTS_B_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CSI0_D[18] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[4] of instance: gpio6. 010 Select mux mode: ALT2 mux port: RTS of instance: uart5. 100 Select mux mode: ALT4 mux port: USBH3_DATA[6] of instance: usboh3. 101 Select mux mode: ALT5 mux port: DEBUG_PC[12] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[47] of instance: emi. 111 Select mux mode: ALT7 mux port: TRACE[15] of instance: tpiu.

43.3.68 IOMUXC_SW_MUX_CTL_PAD_CSI0_DAT19 (IOMUXC_CSI0_DAT19)

Address: IOMUXC_CSI0_DAT19 is 53FA_8000h base + 10Ch offset = 53FA_810Ch

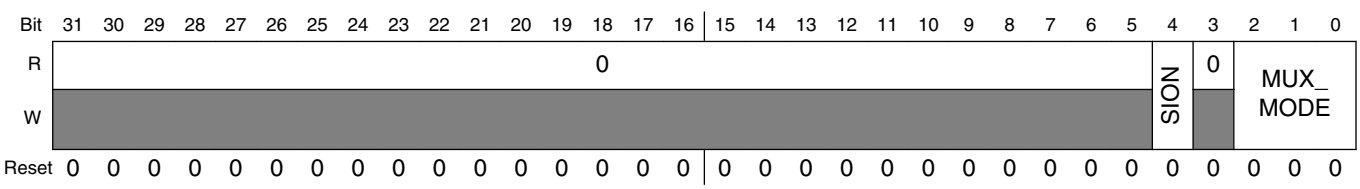


IOMUXC_CSI0_DAT19 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad CSI0_DAT19. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: CSI0_DAT19. NOTE: Pad CSI0_DAT19 is involved in Daisy Chain. - Config Register UART5_IPP_UART_RTS_B_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CSI0_D[19] of instance: ipu. 001 Select mux mode: ALT1 mux port: GPIO[5] of instance: gpio6. 010 Select mux mode: ALT2 mux port: CTS of instance: uart5. 100 Select mux mode: ALT4 mux port: USBH3_DATA[7] of instance: usboh3. 101 Select mux mode: ALT5 mux port: DEBUG_PC[13] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[48] of instance: emi. 111 Select mux mode: ALT7 mux port: BISTOK of instance: usbphy2.

43.3.69 IOMUXC_SW_MUX_CTL_PAD_EIM_A25 (IOMUXC_EIM_A25)

Address: IOMUXC_EIM_A25 is 53FA_8000h base + 110h offset = 53FA_8110h



IOMUXC_EIM_A25 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_A25. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: EIM_A25. NOTE: Pad EIM_A25 is involved in Daisy Chain. - Config Register CSPI_IPP_IND_SS1_B_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: WEIM_A[25] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[2] of instance: gpio5. 010 Select mux mode: ALT2 mux port: RDY of instance: ecspi2. 011 Select mux mode: ALT3 mux port: DI1_PIN12 of instance: ipu. 100 Select mux mode: ALT4 mux port: SS1 of instance: cspi. 110 Select mux mode: ALT6 mux port: DI0_D1_CS of instance: ipu. 111 Select mux mode: ALT7 mux port: BISTOK of instance: usbphy1.

43.3.70 IOMUXC_SW_MUX_CTL_PAD_EIM_EB2 (IOMUXC_EIM_EB2)

Address: IOMUXC_EIM_EB2 is 53FA_8000h base + 114h offset = 53FA_8114h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_ MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_EIM_EB2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_EB2. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_EB2. NOTE: Pad EIM_EB2 is involved in Daisy Chain.

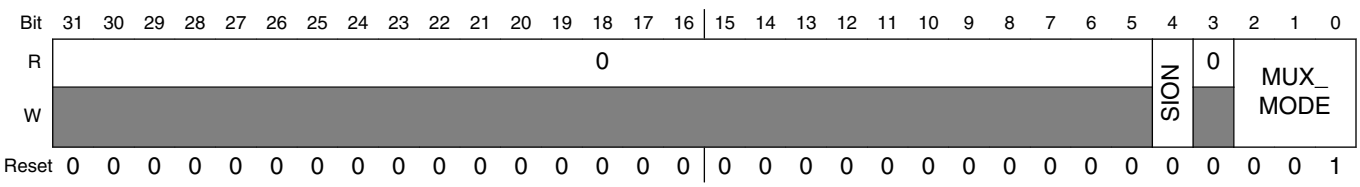
Table continues on the next page...

IOMUXC_EIM_EB2 field descriptions (continued)

Field	Description
	- Config Register CCM_IPP_DI1_CLK_SELECT_INPUT for mode ALT2. - Config Register ECSPI1_IPP_IND_SS_B_0_SELECT_INPUT for mode ALT4. - Config Register I2C2_IPP_SCL_IN_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: WEIM_EB[2] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[30] of instance: gpio2. 010 Select mux mode: ALT2 mux port: DI1_EXT_CLK of instance: ccm. 011 Select mux mode: ALT3 mux port: SER_DISP1_CS of instance: ipu. 100 Select mux mode: ALT4 mux port: SS0 of instance: ecspi1. 101 Select mux mode: ALT5 mux port: SCL of instance: i2c2.

43.3.71 IOMUXC_SW_MUX_CTL_PAD_EIM_D16 (IOMUXC_EIM_D16)

Address: IOMUXC_EIM_D16 is 53FA_8000h base + 118h offset = 53FA_8118h

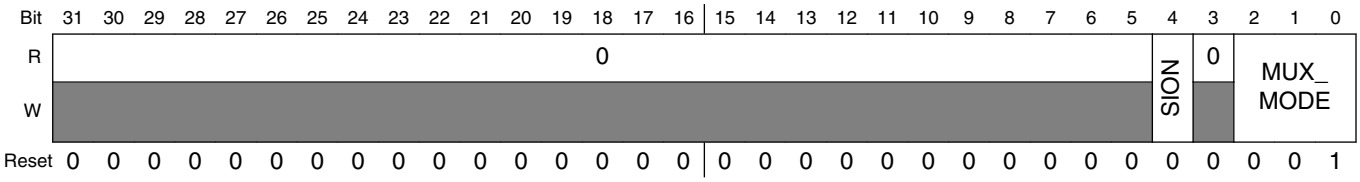


IOMUXC_EIM_D16 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_D16. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_D16. NOTE: Pad EIM_D16 is involved in Daisy Chain. - Config Register ECSPI1_IPP_CSPI_CLK_IN_SELECT_INPUT for mode ALT4. - Config Register I2C2_IPP_SDA_IN_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: WEIM_D[16] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[16] of instance: gpio3. 010 Select mux mode: ALT2 mux port: DI0_PIN5 of instance: ipu. 011 Select mux mode: ALT3 mux port: DISPB1_SER_CLK of instance: ipu. 100 Select mux mode: ALT4 mux port: SCLK of instance: ecspi1. 101 Select mux mode: ALT5 mux port: SDA of instance: i2c2.

43.3.72 IOMUXC_SW_MUX_CTL_PAD_EIM_D17 (IOMUXC_EIM_D17)

Address: IOMUXC_EIM_D17 is 53FA_8000h base + 11Ch offset = 53FA_811Ch

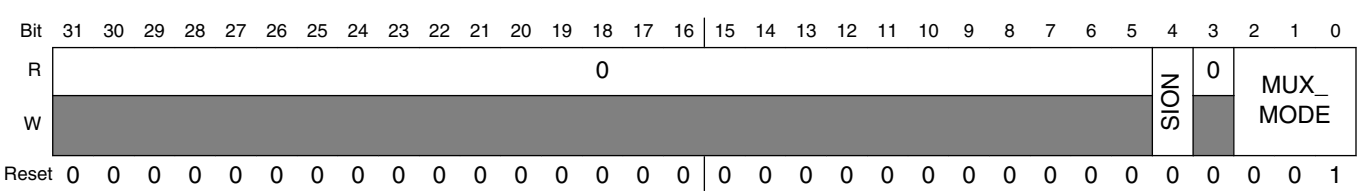


IOMUXC_EIM_D17 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_D17. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_D17. NOTE: Pad EIM_D17 is involved in Daisy Chain. - Config Register ECSPI1_IPP_IND_MISO_SELECT_INPUT for mode ALT4. - Config Register I2C3_IPP_SCL_IN_SELECT_INPUT for mode ALT5. - Config Register IPU_IPP_DI_1_IND_DISPB_SD_D_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: WEIM_D[17] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[17] of instance: gpio3. 010 Select mux mode: ALT2 mux port: DI0_PIN6 of instance: ipu. 011 Select mux mode: ALT3 mux port: DISPB1_SER_DIN of instance: ipu. 100 Select mux mode: ALT4 mux port: MISO of instance: ecspi1. 101 Select mux mode: ALT5 mux port: SCL of instance: i2c3.

43.3.73 IOMUXC_SW_MUX_CTL_PAD_EIM_D18 (IOMUXC_EIM_D18)

Address: IOMUXC_EIM_D18 is 53FA_8000h base + 120h offset = 53FA_8120h



IOMUXC_EIM_D18 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_D18. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: EIM_D18. NOTE: Pad EIM_D18 is involved in Daisy Chain. - Config Register ECSPI1_IPP_IND_MOSI_SELECT_INPUT for mode ALT4. - Config Register I2C3_IPP_SDA_IN_SELECT_INPUT for mode ALT5. - Config Register IPU_IPP_DI_1_IND_DISPB_SD_D_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: WEIM_D[18] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[18] of instance: gpio3. 010 Select mux mode: ALT2 mux port: DI0_PIN7 of instance: ipu. 011 Select mux mode: ALT3 mux port: DISPB1_SER_DIO of instance: ipu. 100 Select mux mode: ALT4 mux port: MOSI of instance: ecspi1. 101 Select mux mode: ALT5 mux port: SDA of instance: i2c3. 110 Select mux mode: ALT6 mux port: DI1_D0_CS of instance: ipu.

43.3.74 IOMUXC_SW_MUX_CTL_PAD_EIM_D19 (IOMUXC_EIM_D19)

Address: IOMUXC_EIM_D19 is 53FA_8000h base + 124h offset = 53FA_8124h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_EIM_D19 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_D19. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_EIM_D19 field descriptions (continued)

Field	Description
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D19. NOTE: Pad EIM_D19 is involved in Daisy Chain. - Config Register ECSPI1_IPP_IND_SS_B_1_SELECT_INPUT for mode ALT4. - Config Register UART1_IPP_UART_RTS_B_SELECT_INPUT for mode ALT6. - Config Register USBOH3_IPP_IND_UH2_OC_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: WEIM_D[19] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[19] of instance: gpio3. 010 Select mux mode: ALT2 mux port: DI0_PIN8 of instance: ipu. 011 Select mux mode: ALT3 mux port: DISPB1_SER_RS of instance: ipu. 100 Select mux mode: ALT4 mux port: SS1 of instance: ecspi1. 101 Select mux mode: ALT5 mux port: EPITO of instance: epit1. 110 Select mux mode: ALT6 mux port: CTS of instance: uart1. 111 Select mux mode: ALT7 mux port: USBH2_OC of instance: usboh3.

43.3.75 IOMUXC_SW_MUX_CTL_PAD_EIM_D20 (IOMUXC_EIM_D20)

Address: IOMUXC_EIM_D20 is 53FA_8000h base + 128h offset = 53FA_8128h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_ MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_EIM_D20 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_D20. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D20. NOTE: Pad EIM_D20 is involved in Daisy Chain. - Config Register CSPI_IPP_IND_SS0_B_SELECT_INPUT for mode ALT4. - Config Register UART1_IPP_UART_RTS_B_SELECT_INPUT for mode ALT6. 000 Select mux mode: ALT0 mux port: WEIM_D[20] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[20] of instance: gpio3.

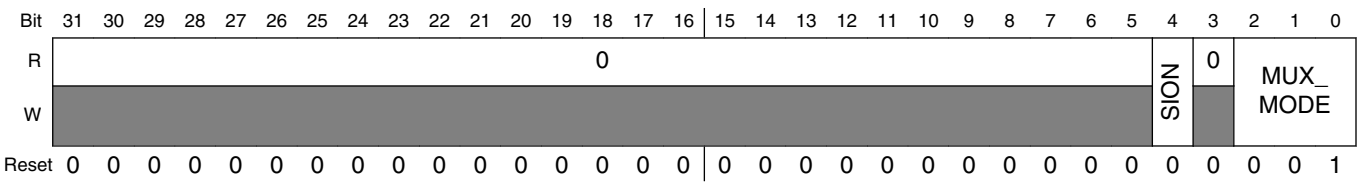
Table continues on the next page...

IOMUXC_EIM_D20 field descriptions (continued)

Field	Description
	010 Select mux mode: ALT2 mux port: DI0_PIN16 of instance: ipu. 011 Select mux mode: ALT3 mux port: SER_DISP0_CS of instance: ipu. 100 Select mux mode: ALT4 mux port: SS0 of instance: cspi. 101 Select mux mode: ALT5 mux port: EPITO of instance: epit2. 110 Select mux mode: ALT6 mux port: RTS of instance: uart1. 111 Select mux mode: ALT7 mux port: USBH2_PWR of instance: usboh3.

43.3.76 IOMUXC_SW_MUX_CTL_PAD_EIM_D21 (IOMUXC_EIM_D21)

Address: IOMUXC_EIM_D21 is 53FA_8000h base + 12Ch offset = 53FA_812Ch

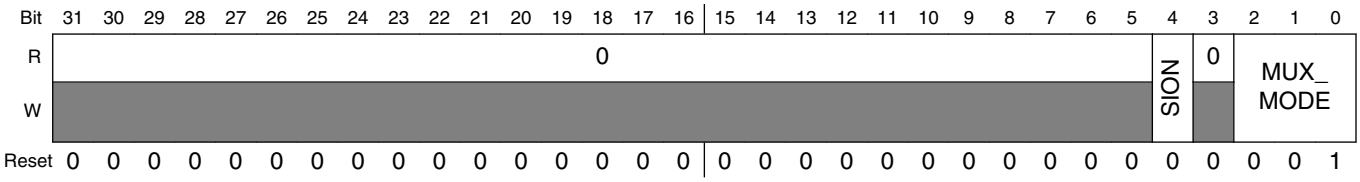


IOMUXC_EIM_D21 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_D21. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: EIM_D21. NOTE: Pad EIM_D21 is involved in Daisy Chain. - Config Register CSPI_IPP_CSPI_CLK_IN_SELECT_INPUT for mode ALT4. - Config Register I2C1_IPP_SCL_IN_SELECT_INPUT for mode ALT5. - Config Register USBOH3_IPP_IND_OTG_OC_SELECT_INPUT for mode ALT6. 000 Select mux mode: ALT0 mux port: WEIM_D[21] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[21] of instance: gpio3. 010 Select mux mode: ALT2 mux port: DI0_PIN17 of instance: ipu. 011 Select mux mode: ALT3 mux port: DISPB0_SER_CLK of instance: ipu. 100 Select mux mode: ALT4 mux port: SCLK of instance: cspi. 101 Select mux mode: ALT5 mux port: SCL of instance: i2c1. 110 Select mux mode: ALT6 mux port: USBOTG_OC of instance: usboh3.

43.3.77 IOMUXC_SW_MUX_CTL_PAD_EIM_D22 (IOMUXC_EIM_D22)

Address: IOMUXC_EIM_D22 is 53FA_8000h base + 130h offset = 53FA_8130h

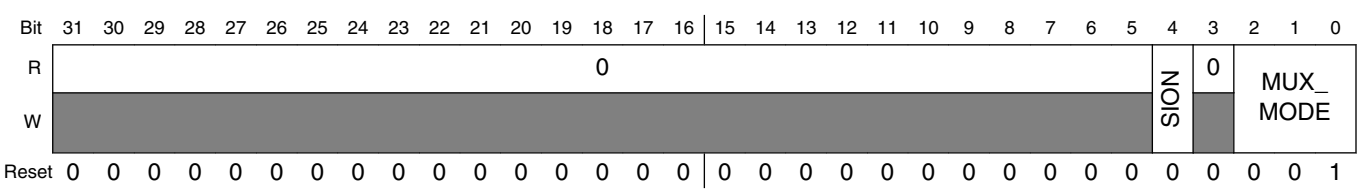


IOMUXC_EIM_D22 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_D22. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_D22. NOTE: Pad EIM_D22 is involved in Daisy Chain. - Config Register CSPI_IPP_IND_MISO_SELECT_INPUT for mode ALT4. - Config Register IPU_IPP_DI_0_IND_DISPB_SD_D_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: WEIM_D[22] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[22] of instance: gpio3. 010 Select mux mode: ALT2 mux port: DI0_PIN1 of instance: ipu. 011 Select mux mode: ALT3 mux port: DISPB0_SER_DIN of instance: ipu. 100 Select mux mode: ALT4 mux port: MISO of instance: cspi. 110 Select mux mode: ALT6 mux port: USBOTG_PWR of instance: usboh3.

43.3.78 IOMUXC_SW_MUX_CTL_PAD_EIM_D23 (IOMUXC_EIM_D23)

Address: IOMUXC_EIM_D23 is 53FA_8000h base + 134h offset = 53FA_8134h

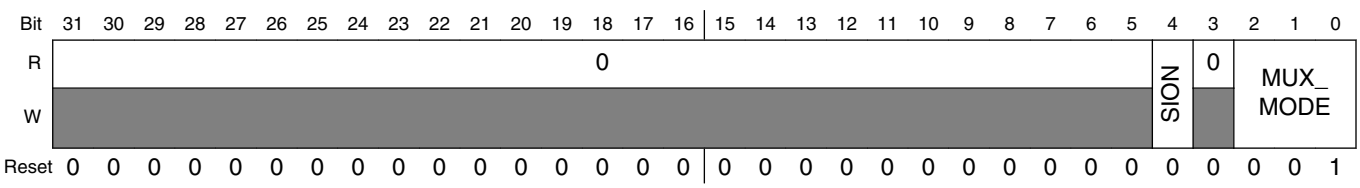


IOMUXC_EIM_D23 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_D23. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D23. NOTE: Pad EIM_D23 is involved in Daisy Chain. - Config Register IPU_IPP_IND_SENS1_DATA_EN_SELECT_INPUT for mode ALT6. - Config Register UART3_IPP_UART_RTS_B_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: WEIM_D[23] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[23] of instance: gpio3. 010 Select mux mode: ALT2 mux port: CTS of instance: uart3. 011 Select mux mode: ALT3 mux port: DCD of instance: uart1. 100 Select mux mode: ALT4 mux port: DI0_D0_CS of instance: ipu. 101 Select mux mode: ALT5 mux port: DI1_PIN2 of instance: ipu. 110 Select mux mode: ALT6 mux port: CSI1_DATA_EN of instance: ipu. 111 Select mux mode: ALT7 mux port: DI1_PIN14 of instance: ipu.

43.3.79 IOMUXC_SW_MUX_CTL_PAD_EIM_EB3 (IOMUXC_EIM_EB3)

Address: IOMUXC_EIM_EB3 is 53FA_8000h base + 138h offset = 53FA_8138h



IOMUXC_EIM_EB3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_EB3. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved

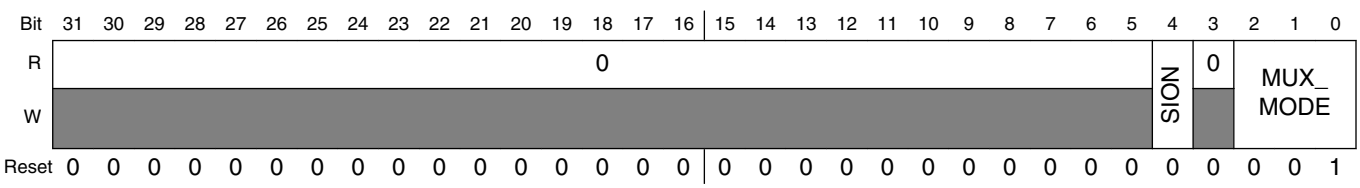
Table continues on the next page...

IOMUXC_EIM_EB3 field descriptions (continued)

Field	Description
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: EIM_EB3. NOTE: Pad EIM_EB3 is involved in Daisy Chain. - Config Register IPU_IPP_IND_SENS1_HSYNC_SELECT_INPUT for mode ALT6. - Config Register UART3_IPP_UART_RTS_B_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: WEIM_EB[3] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[31] of instance: gpio2. 010 Select mux mode: ALT2 mux port: RTS of instance: uart3. 011 Select mux mode: ALT3 mux port: RI of instance: uart1. 101 Select mux mode: ALT5 mux port: DI1_PIN3 of instance: ipu. 110 Select mux mode: ALT6 mux port: CSI1_HSYNC of instance: ipu. 111 Select mux mode: ALT7 mux port: DI1_PIN16 of instance: ipu.

43.3.80 IOMUXC_SW_MUX_CTL_PAD_EIM_D24 (IOMUXC_EIM_D24)

Address: IOMUXC_EIM_D24 is 53FA_8000h base + 13Ch offset = 53FA_813Ch



IOMUXC_EIM_D24 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_D24. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D24. NOTE: Pad EIM_D24 is involved in Daisy Chain. - Config Register AUDMUX_P5_INPUT_RXFS_AMX_SELECT_INPUT for mode ALT5. - Config Register CSPI_IPP_IND_SS2_B_SELECT_INPUT for mode ALT4. - Config Register ECSPI1_IPP_IND_SS_B_2_SELECT_INPUT for mode ALT3. - Config Register UART3_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: WEIM_D[24] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[24] of instance: gpio3.

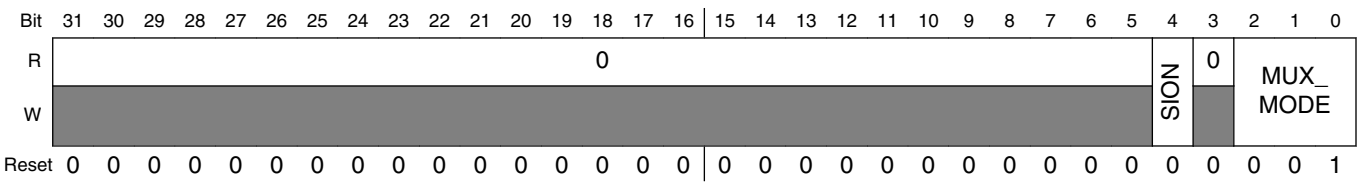
Table continues on the next page...

IOMUXC_EIM_D24 field descriptions (continued)

Field	Description
	010 Select mux mode: ALT2 mux port: TXD_MUX of instance: uart3. 011 Select mux mode: ALT3 mux port: SS2 of instance: ecspi1. 100 Select mux mode: ALT4 mux port: SS2 of instance: cspi. 101 Select mux mode: ALT5 mux port: AUD5_RXFS of instance: audmux. 110 Select mux mode: ALT6 mux port: SS2 of instance: ecspi2. 111 Select mux mode: ALT7 mux port: DTR of instance: uart1.

43.3.81 IOMUXC_SW_MUX_CTL_PAD_EIM_D25 (IOMUXC_EIM_D25)

Address: IOMUXC_EIM_D25 is 53FA_8000h base + 140h offset = 53FA_8140h

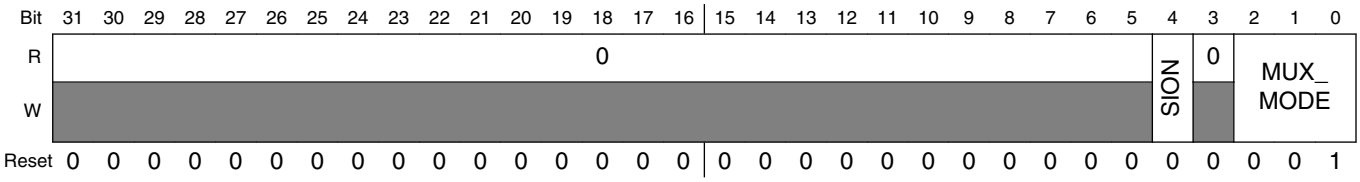


IOMUXC_EIM_D25 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_D25. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D25. NOTE: Pad EIM_D25 is involved in Daisy Chain. - Config Register AUDMUX_P5_INPUT_RXCLK_AMX_SELECT_INPUT for mode ALT5. - Config Register CSPI_IPP_IND_SS3_B_SELECT_INPUT for mode ALT4. - Config Register ECSPI1_IPP_IND_SS_B_3_SELECT_INPUT for mode ALT3. - Config Register UART3_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: WEIM_D[25] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[25] of instance: gpio3. 010 Select mux mode: ALT2 mux port: RXD_MUX of instance: uart3. 011 Select mux mode: ALT3 mux port: SS3 of instance: ecspi1. 100 Select mux mode: ALT4 mux port: SS3 of instance: cspi. 101 Select mux mode: ALT5 mux port: AUD5_RXC of instance: audmux. 110 Select mux mode: ALT6 mux port: SS3 of instance: ecspi2. 111 Select mux mode: ALT7 mux port: DSR of instance: uart1.

43.3.82 IOMUXC_SW_MUX_CTL_PAD_EIM_D26 (IOMUXC_EIM_D26)

Address: IOMUXC_EIM_D26 is 53FA_8000h base + 144h offset = 53FA_8144h

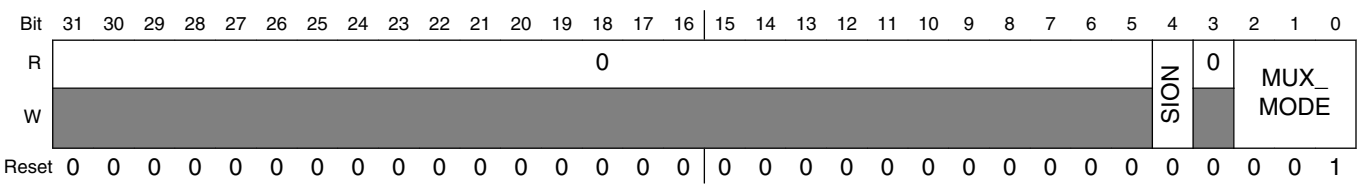


IOMUXC_EIM_D26 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_D26. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D26. NOTE: Pad EIM_D26 is involved in Daisy Chain. - Config Register FIRI_IPP_IND_RXD_SELECT_INPUT for mode ALT3. - Config Register UART2_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: WEIM_D[26] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[26] of instance: gpio3. 010 Select mux mode: ALT2 mux port: TXD_MUX of instance: uart2. 011 Select mux mode: ALT3 mux port: RXD of instance: firi. 100 Select mux mode: ALT4 mux port: CSI0_D[1] of instance: ipu. 101 Select mux mode: ALT5 mux port: DI1_PIN11 of instance: ipu. 110 Select mux mode: ALT6 mux port: SISG[2] of instance: ipu. 111 Select mux mode: ALT7 mux port: DISP1_DAT[22] of instance: ipu.

43.3.83 IOMUXC_SW_MUX_CTL_PAD_EIM_D27 (IOMUXC_EIM_D27)

Address: IOMUXC_EIM_D27 is 53FA_8000h base + 148h offset = 53FA_8148h



IOMUXC_EIM_D27 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_D27. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D27. NOTE: Pad EIM_D27 is involved in Daisy Chain. - Config Register UART2_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: WEIM_D[27] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[27] of instance: gpio3. 010 Select mux mode: ALT2 mux port: RXD_MUX of instance: uart2. 011 Select mux mode: ALT3 mux port: TXD of instance: firi. 100 Select mux mode: ALT4 mux port: CSI0_D[0] of instance: ipu. 101 Select mux mode: ALT5 mux port: DI1_PIN13 of instance: ipu. 110 Select mux mode: ALT6 mux port: SISG[3] of instance: ipu. 111 Select mux mode: ALT7 mux port: DISP1_DAT[23] of instance: ipu.

43.3.84 IOMUXC_SW_MUX_CTL_PAD_EIM_D28 (IOMUXC_EIM_D28)

Address: IOMUXC_EIM_D28 is 53FA_8000h base + 14Ch offset = 53FA_814Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_EIM_D28 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_D28. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D28.

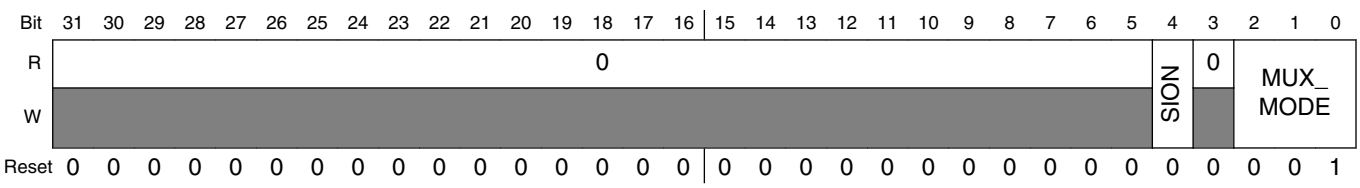
Table continues on the next page...

IOMUXC_EIM_D28 field descriptions (continued)

Field	Description
	NOTE: Pad EIM_D28 is involved in Daisy Chain. - Config Register CSPI_IPP_IND_MOSI_SELECT_INPUT for mode ALT4. - Config Register I2C1_IPP_SDA_IN_SELECT_INPUT for mode ALT5. - Config Register IPU_IPP_DI_0_IND_DISPB_SD_D_SELECT_INPUT for mode ALT3. - Config Register UART2_IPP_UART_RTS_B_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: WEIM_D[28] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[28] of instance: gpio3. 010 Select mux mode: ALT2 mux port: CTS of instance: uart2. 011 Select mux mode: ALT3 mux port: DISPB0_SER_DIO of instance: ipu. 100 Select mux mode: ALT4 mux port: MOSI of instance: cspi. 101 Select mux mode: ALT5 mux port: SDA of instance: i2c1. 110 Select mux mode: ALT6 mux port: EXT_TRIG of instance: ipu. 111 Select mux mode: ALT7 mux port: DI0_PIN13 of instance: ipu.

43.3.85 IOMUXC_SW_MUX_CTL_PAD_EIM_D29 (IOMUXC_EIM_D29)

Address: IOMUXC_EIM_D29 is 53FA_8000h base + 150h offset = 53FA_8150h



IOMUXC_EIM_D29 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_D29. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D29. NOTE: Pad EIM_D29 is involved in Daisy Chain. - Config Register CSPI_IPP_IND_SS0_B_SELECT_INPUT for mode ALT4. - Config Register IPU_IPP_IND_SENS1_VSYNC_SELECT_INPUT for mode ALT6. - Config Register UART2_IPP_UART_RTS_B_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: WEIM_D[29] of instance: emi.

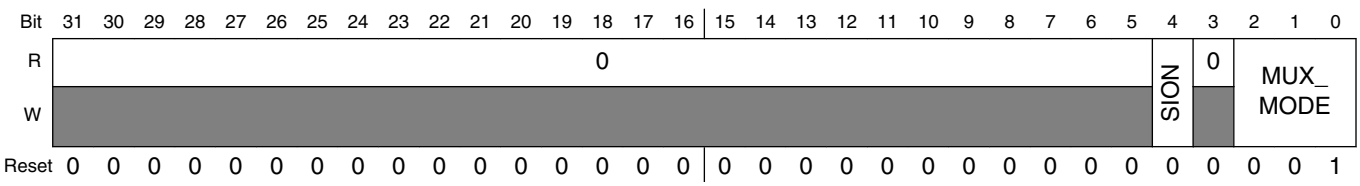
Table continues on the next page...

IOMUXC_EIM_D29 field descriptions (continued)

Field	Description
	001 Select mux mode: ALT1 mux port: GPIO[29] of instance: gpio3. 010 Select mux mode: ALT2 mux port: RTS of instance: uart2. 011 Select mux mode: ALT3 mux port: DISPB0_SER_RS of instance: ipu. 100 Select mux mode: ALT4 mux port: SS0 of instance: cspi. 101 Select mux mode: ALT5 mux port: DI1_PIN15 of instance: ipu. 110 Select mux mode: ALT6 mux port: CSI1_VSYNC of instance: ipu. 111 Select mux mode: ALT7 mux port: DI0_PIN14 of instance: ipu.

43.3.86 IOMUXC_SW_MUX_CTL_PAD_EIM_D30 (IOMUXC_EIM_D30)

Address: IOMUXC_EIM_D30 is 53FA_8000h base + 154h offset = 53FA_8154h



IOMUXC_EIM_D30 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_D30. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D30. NOTE: Pad EIM_D30 is involved in Daisy Chain. - Config Register UART3_IPP_UART_RTS_B_SELECT_INPUT for mode ALT2. - Config Register USBOH3_IPP_IND_UH1_OC_SELECT_INPUT for mode ALT6. - Config Register USBOH3_IPP_IND_UH2_OC_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: WEIM_D[30] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[30] of instance: gpio3. 010 Select mux mode: ALT2 mux port: CTS of instance: uart3. 011 Select mux mode: ALT3 mux port: CSI0_D[3] of instance: ipu. 100 Select mux mode: ALT4 mux port: DI0_PIN11 of instance: ipu. 101 Select mux mode: ALT5 mux port: DISP1_DAT[21] of instance: ipu. 110 Select mux mode: ALT6 mux port: USBH1_OC of instance: usboh3. 111 Select mux mode: ALT7 mux port: USBH2_OC of instance: usboh3.

43.3.87 IOMUXC_SW_MUX_CTL_PAD_EIM_D31 (IOMUXC_EIM_D31)

Address: IOMUXC_EIM_D31 is 53FA_8000h base + 158h offset = 53FA_8158h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_ MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_EIM_D31 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_D31. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: EIM_D31. NOTE: Pad EIM_D31 is involved in Daisy Chain. - Config Register UART3_IPP_UART_RTS_B_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: WEIM_D[31] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[31] of instance: gpio3. 010 Select mux mode: ALT2 mux port: RTS of instance: uart3. 011 Select mux mode: ALT3 mux port: CSI0_D[2] of instance: ipu. 100 Select mux mode: ALT4 mux port: DI0_PIN12 of instance: ipu. 101 Select mux mode: ALT5 mux port: DISP1_DAT[20] of instance: ipu. 110 Select mux mode: ALT6 mux port: USBH1_PWR of instance: usboh3. 111 Select mux mode: ALT7 mux port: USBH2_PWR of instance: usboh3.

43.3.88 IOMUXC_SW_MUX_CTL_PAD_EIM_A24 (IOMUXC_EIM_A24)

Address: IOMUXC_EIM_A24 is 53FA_8000h base + 15Ch offset = 53FA_815Ch

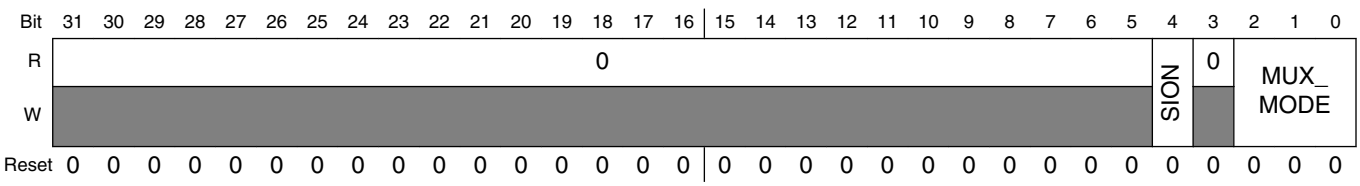
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_ MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_EIM_A24 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_A24. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_A24. 000 Select mux mode: ALT0 mux port: WEIM_A[24] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[4] of instance: gpio5. 010 Select mux mode: ALT2 mux port: DISP1_DAT[19] of instance: ipu. 011 Select mux mode: ALT3 mux port: CSI1_D[19] of instance: ipu. 110 Select mux mode: ALT6 mux port: SISG[2] of instance: ipu. 111 Select mux mode: ALT7 mux port: BVALID of instance: usbphy2.

43.3.89 IOMUXC_SW_MUX_CTL_PAD_EIM_A23 (IOMUXC_EIM_A23)

Address: IOMUXC_EIM_A23 is 53FA_8000h base + 160h offset = 53FA_8160h



IOMUXC_EIM_A23 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_A23. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_A23. 000 Select mux mode: ALT0 mux port: WEIM_A[23] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[6] of instance: gpio6. 010 Select mux mode: ALT2 mux port: DISP1_DAT[18] of instance: ipu. 011 Select mux mode: ALT3 mux port: CSI1_D[18] of instance: ipu.

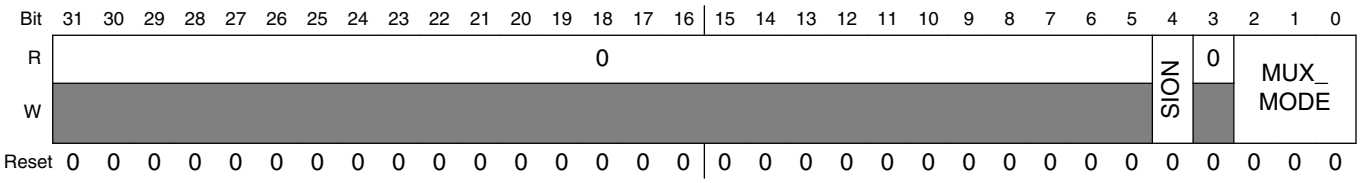
Table continues on the next page...

IOMUXC_EIM_A23 field descriptions (continued)

Field	Description
	110 Select mux mode: ALT6 mux port: SISG[3] of instance: ipu. 111 Select mux mode: ALT7 mux port: ENDSESSION of instance: usbphy2.

43.3.90 IOMUXC_SW_MUX_CTL_PAD_EIM_A22 (IOMUXC_EIM_A22)

Address: IOMUXC_EIM_A22 is 53FA_8000h base + 164h offset = 53FA_8164h

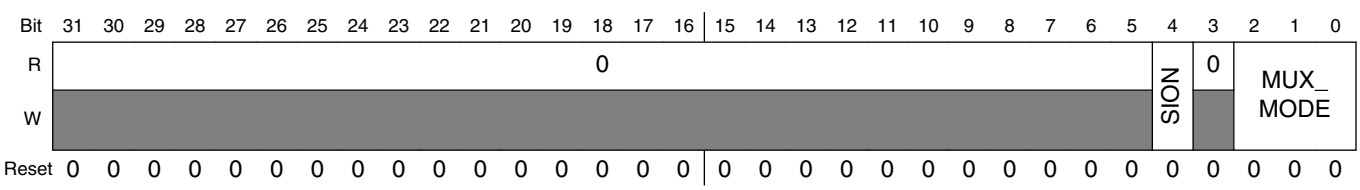


IOMUXC_EIM_A22 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_A22. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_A22. 000 Select mux mode: ALT0 mux port: WEIM_A[22] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[16] of instance: gpio2. 010 Select mux mode: ALT2 mux port: DISP1_DAT[17] of instance: ipu. 011 Select mux mode: ALT3 mux port: CSI1_D[17] of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG1[7] of instance: src.

43.3.91 IOMUXC_SW_MUX_CTL_PAD_EIM_A21 (IOMUXC_EIM_A21)

Address: IOMUXC_EIM_A21 is 53FA_8000h base + 168h offset = 53FA_8168h

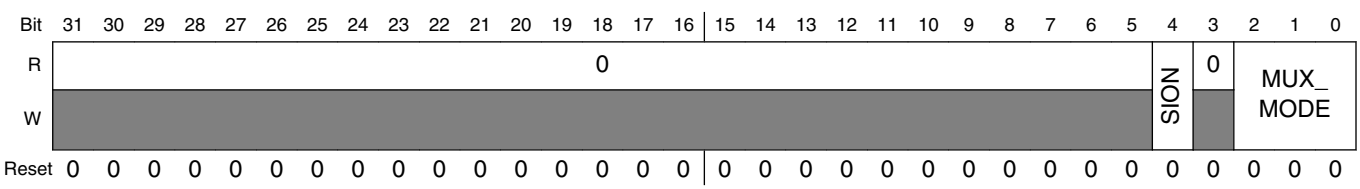


IOMUXC_EIM_A21 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_A21. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_A21. 000 Select mux mode: ALT0 mux port: WEIM_A[21] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[17] of instance: gpio2. 010 Select mux mode: ALT2 mux port: DISP1_DAT[16] of instance: ipu. 011 Select mux mode: ALT3 mux port: CSI1_D[16] of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG1[6] of instance: src.

43.3.92 IOMUXC_SW_MUX_CTL_PAD_EIM_A20 (IOMUXC_EIM_A20)

Address: IOMUXC_EIM_A20 is 53FA_8000h base + 16Ch offset = 53FA_816Ch

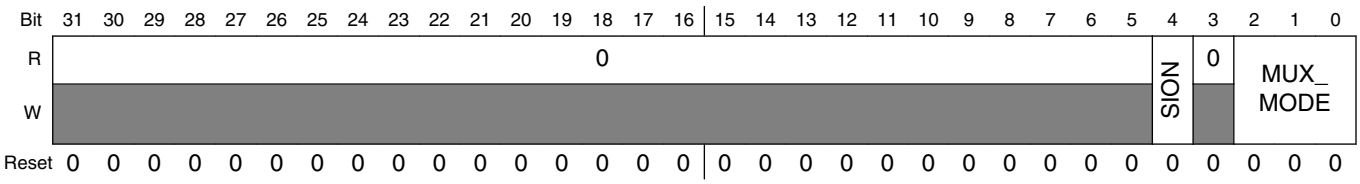


IOMUXC_EIM_A20 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_A20. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_A20. 000 Select mux mode: ALT0 mux port: WEIM_A[20] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[18] of instance: gpio2. 010 Select mux mode: ALT2 mux port: DISP1_DAT[15] of instance: ipu. 011 Select mux mode: ALT3 mux port: CSI1_D[15] of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG1[5] of instance: src.

43.3.93 IOMUXC_SW_MUX_CTL_PAD_EIM_A19 (IOMUXC_EIM_A19)

Address: IOMUXC_EIM_A19 is 53FA_8000h base + 170h offset = 53FA_8170h

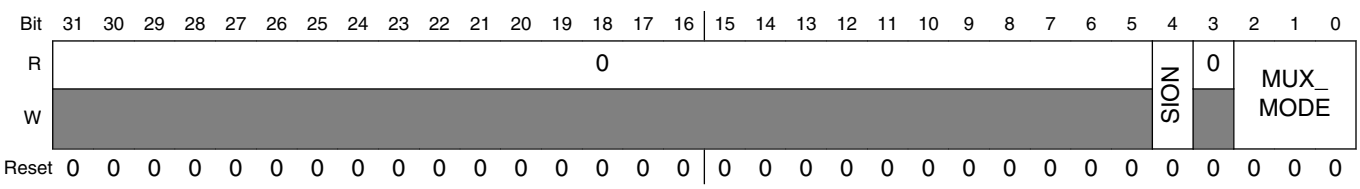


IOMUXC_EIM_A19 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_A19. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_A19. 000 Select mux mode: ALT0 mux port: WEIM_A[19] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[19] of instance: gpio2. 010 Select mux mode: ALT2 mux port: DISP1_DAT[14] of instance: ipu. 011 Select mux mode: ALT3 mux port: CSI1_D[14] of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG1[4] of instance: src.

43.3.94 IOMUXC_SW_MUX_CTL_PAD_EIM_A18 (IOMUXC_EIM_A18)

Address: IOMUXC_EIM_A18 is 53FA_8000h base + 174h offset = 53FA_8174h



IOMUXC_EIM_A18 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality.

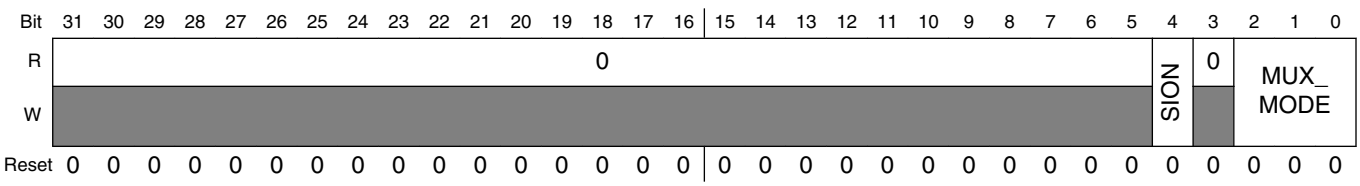
Table continues on the next page...

IOMUXC_EIM_A18 field descriptions (continued)

Field	Description
	1 Force input path of pad EIM_A18. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_A18. 000 Select mux mode: ALT0 mux port: WEIM_A[18] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[20] of instance: gpio2. 010 Select mux mode: ALT2 mux port: DISP1_DAT[13] of instance: ipu. 011 Select mux mode: ALT3 mux port: CSI1_D[13] of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG1[3] of instance: src.

43.3.95 IOMUXC_SW_MUX_CTL_PAD_EIM_A17 (IOMUXC_EIM_A17)

Address: IOMUXC_EIM_A17 is 53FA_8000h base + 178h offset = 53FA_8178h

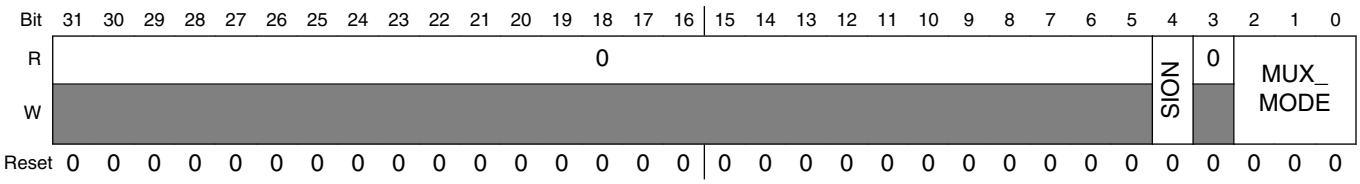


IOMUXC_EIM_A17 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_A17. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_A17. 000 Select mux mode: ALT0 mux port: WEIM_A[17] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[21] of instance: gpio2. 010 Select mux mode: ALT2 mux port: DISP1_DAT[12] of instance: ipu. 011 Select mux mode: ALT3 mux port: CSI1_D[12] of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG1[2] of instance: src.

43.3.96 IOMUXC_SW_MUX_CTL_PAD_EIM_A16 (IOMUXC_EIM_A16)

Address: IOMUXC_EIM_A16 is 53FA_8000h base + 17Ch offset = 53FA_817Ch

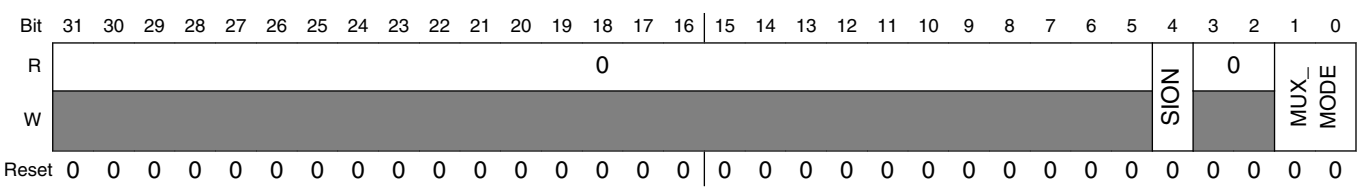


IOMUXC_EIM_A16 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_A16. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_A16. 000 Select mux mode: ALT0 mux port: WEIM_A[16] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[22] of instance: gpio2. 010 Select mux mode: ALT2 mux port: DI1_DISP_CLK of instance: ipu. 011 Select mux mode: ALT3 mux port: CSI1_PIXCLK of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG1[1] of instance: src.

43.3.97 IOMUXC_SW_MUX_CTL_PAD_EIM_CS0 (IOMUXC_EIM_CS0)

Address: IOMUXC_EIM_CS0 is 53FA_8000h base + 180h offset = 53FA_8180h



IOMUXC_EIM_CS0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality.

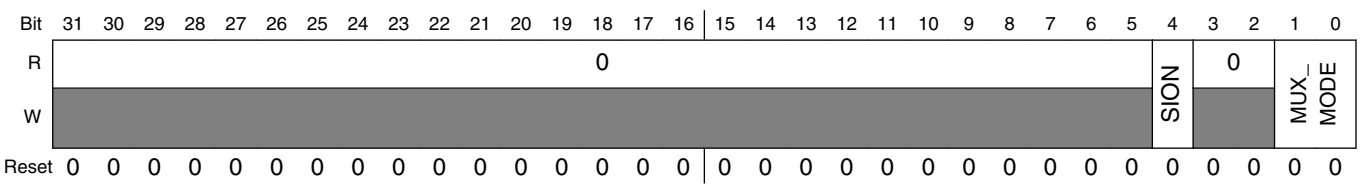
Table continues on the next page...

IOMUXC_EIM_CS0 field descriptions (continued)

Field	Description
	1 Force input path of pad EIM_CS0. 0 Input Path is determined by functionality of the selected mux mode (regular).
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: EIM_CS0. NOTE: Pad EIM_CS0 is involved in Daisy Chain. - Config Register ECSPI2_IPP_CSPI_CLK_IN_SELECT_INPUT for mode ALT2. 00 Select mux mode: ALT0 mux port: WEIM_CS[0] of instance: emi. 01 Select mux mode: ALT1 mux port: GPIO[23] of instance: gpio2. 10 Select mux mode: ALT2 mux port: SCLK of instance: ecspi2. 11 Select mux mode: ALT3 mux port: DI1_PIN5 of instance: ipu.

43.3.98 IOMUXC_SW_MUX_CTL_PAD_EIM_CS1 (IOMUXC_EIM_CS1)

Address: IOMUXC_EIM_CS1 is 53FA_8000h base + 184h offset = 53FA_8184h

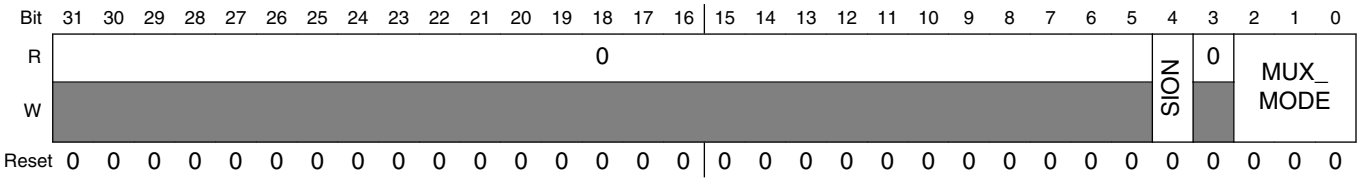


IOMUXC_EIM_CS1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_CS1. 0 Input Path is determined by functionality of the selected mux mode (regular).
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: EIM_CS1. NOTE: Pad EIM_CS1 is involved in Daisy Chain. - Config Register ECSPI2_IPP_IND_MOSI_SELECT_INPUT for mode ALT2. 00 Select mux mode: ALT0 mux port: WEIM_CS[1] of instance: emi. 01 Select mux mode: ALT1 mux port: GPIO[24] of instance: gpio2. 10 Select mux mode: ALT2 mux port: MOSI of instance: ecspi2. 11 Select mux mode: ALT3 mux port: DI1_PIN6 of instance: ipu.

43.3.99 IOMUXC_SW_MUX_CTL_PAD_EIM_OE (IOMUXC_EIM_OE)

Address: IOMUXC_EIM_OE is 53FA_8000h base + 188h offset = 53FA_8188h

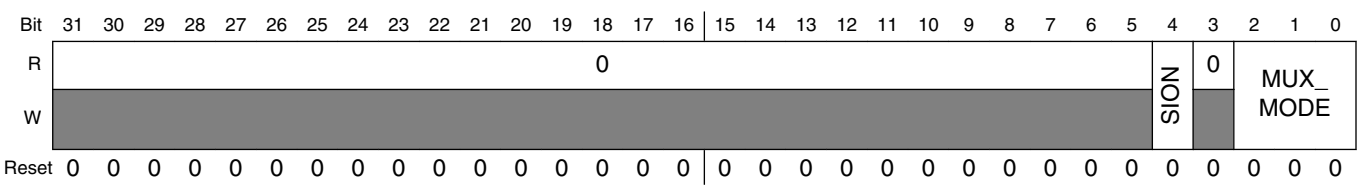


IOMUXC_EIM_OE field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_OE. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_OE. NOTE: Pad EIM_OE is involved in Daisy Chain. - Config Register ECSPI2_IPP_IND_MISO_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: WEIM_OE of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[25] of instance: gpio2. 010 Select mux mode: ALT2 mux port: MISO of instance: ecspi2. 011 Select mux mode: ALT3 mux port: DI1_PIN7 of instance: ipu. 111 Select mux mode: ALT7 mux port: IDDIG of instance: usbphy2.

43.3.100 IOMUXC_SW_MUX_CTL_PAD_EIM_RW (IOMUXC_EIM_RW)

Address: IOMUXC_EIM_RW is 53FA_8000h base + 18Ch offset = 53FA_818Ch



IOMUXC_EIM_RW field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved

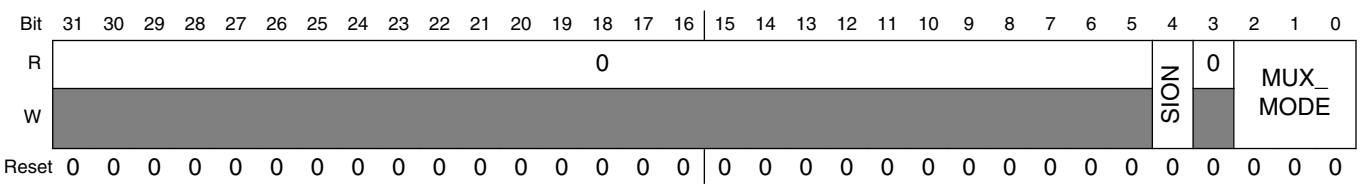
Table continues on the next page...

IOMUXC_EIM_RW field descriptions (continued)

Field	Description
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_RW. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_RW. NOTE: Pad EIM_RW is involved in Daisy Chain. - Config Register ECSPI2_IPP_IND_SS_B_0_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: WEIM_RW of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[26] of instance: gpio2. 010 Select mux mode: ALT2 mux port: SS0 of instance: ecspi2. 011 Select mux mode: ALT3 mux port: DI1_PIN8 of instance: ipu. 111 Select mux mode: ALT7 mux port: HOSTDISCONNECT of instance: usbphy2.

43.3.101 IOMUXC_SW_MUX_CTL_PAD_EIM_LBA (IOMUXC_EIM_LBA)

Address: IOMUXC_EIM_LBA is 53FA_8000h base + 190h offset = 53FA_8190h



IOMUXC_EIM_LBA field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_LBA. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_LBA. NOTE: Pad EIM_LBA is involved in Daisy Chain. - Config Register ECSPI2_IPP_IND_SS_B_1_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: WEIM_LBA of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[27] of instance: gpio2.

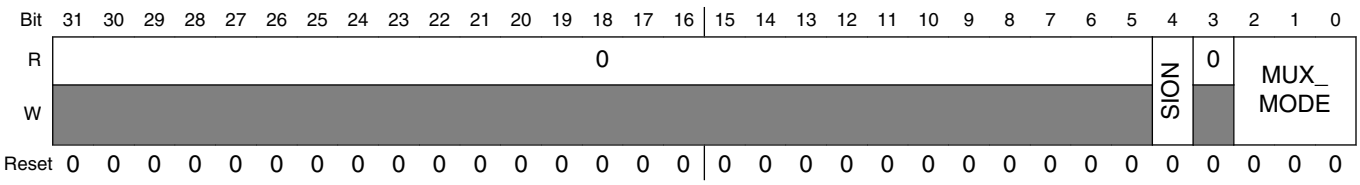
Table continues on the next page...

IOMUXC_EIM_LBA field descriptions (continued)

Field	Description
	010 Select mux mode: ALT2 mux port: SS1 of instance: ecspi2. 011 Select mux mode: ALT3 mux port: DI1_PIN17 of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG1[0] of instance: src.

43.3.102 IOMUXC_SW_MUX_CTL_PAD_EIM_EB0 (IOMUXC_EIM_EB0)

Address: IOMUXC_EIM_EB0 is 53FA_8000h base + 194h offset = 53FA_8194h

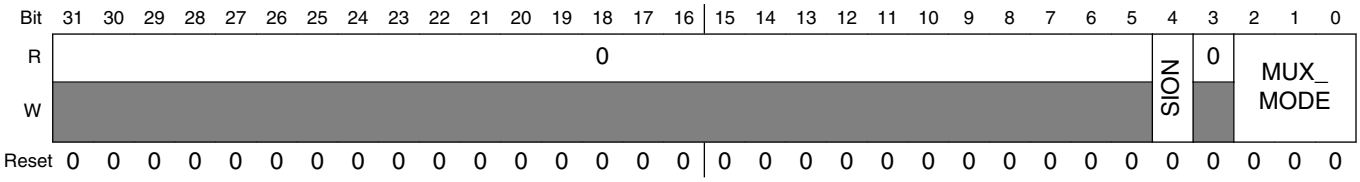


IOMUXC_EIM_EB0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_EB0. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: EIM_EB0. NOTE: Pad EIM_EB0 is involved in Daisy Chain. - Config Register GPC_PMIC_RDY_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: WEIM_EB[0] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[28] of instance: gpio2. 011 Select mux mode: ALT3 mux port: DISP1_DAT[11] of instance: ipu. 100 Select mux mode: ALT4 mux port: CSI1_D[11] of instance: ipu. 101 Select mux mode: ALT5 mux port: PMIC_RDY of instance: gpc. 111 Select mux mode: ALT7 mux port: BT_CFG2[7] of instance: src.

43.3.103 IOMUXC_SW_MUX_CTL_PAD_EIM_EB1 (IOMUXC_EIM_EB1)

Address: IOMUXC_EIM_EB1 is 53FA_8000h base + 198h offset = 53FA_8198h

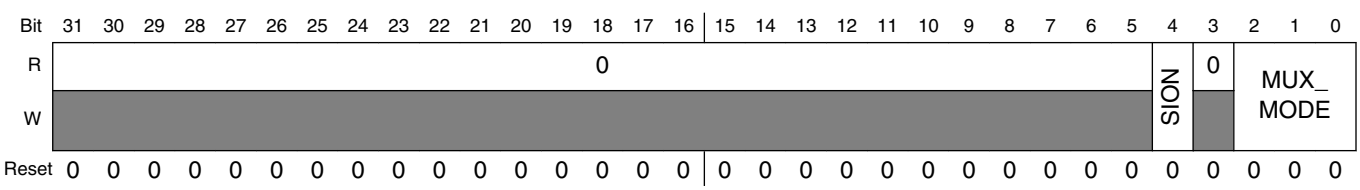


IOMUXC_EIM_EB1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_EB1. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_EB1. 000 Select mux mode: ALT0 mux port: WEIM_EB[1] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[29] of instance: gpio2. 011 Select mux mode: ALT3 mux port: DISP1_DAT[10] of instance: ipu. 100 Select mux mode: ALT4 mux port: CSI1_D[10] of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG2[6] of instance: src.

43.3.104 IOMUXC_SW_MUX_CTL_PAD_EIM_DA0 (IOMUXC_EIM_DA0)

Address: IOMUXC_EIM_DA0 is 53FA_8000h base + 19Ch offset = 53FA_819Ch



IOMUXC_EIM_DA0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved

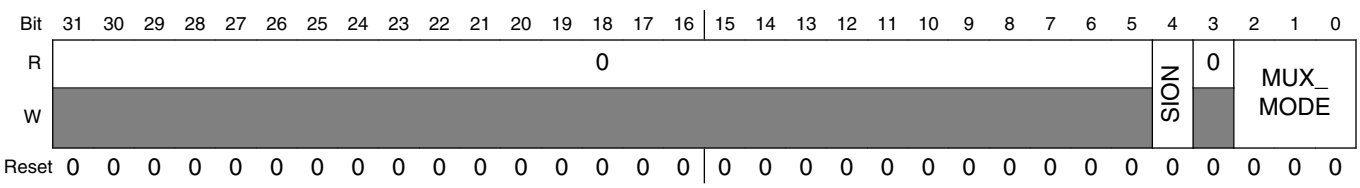
Table continues on the next page...

IOMUXC_EIM_DA0 field descriptions (continued)

Field	Description
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_DA0. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_DA0. 000 Select mux mode: ALT0 mux port: NAND_WEIM_DA[0] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[0] of instance: gpio3. 011 Select mux mode: ALT3 mux port: DISP1_DAT[9] of instance: ipu. 100 Select mux mode: ALT4 mux port: CSI1_D[9] of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG2[5] of instance: src.

43.3.105 IOMUXC_SW_MUX_CTL_PAD_EIM_DA1 (IOMUXC_EIM_DA1)

Address: IOMUXC_EIM_DA1 is 53FA_8000h base + 1A0h offset = 53FA_81A0h

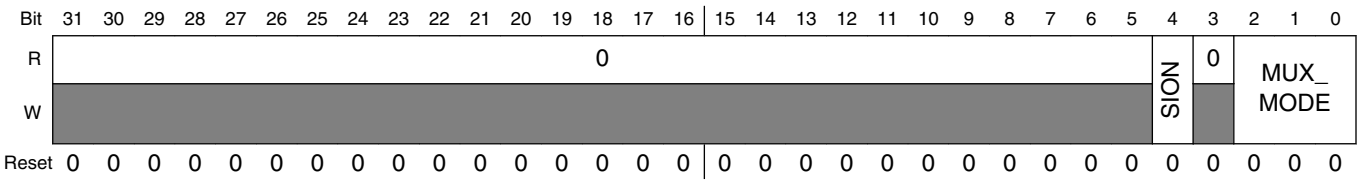


IOMUXC_EIM_DA1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_DA1. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_DA1. 000 Select mux mode: ALT0 mux port: NAND_WEIM_DA[1] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[1] of instance: gpio3. 011 Select mux mode: ALT3 mux port: DISP1_DAT[8] of instance: ipu. 100 Select mux mode: ALT4 mux port: CSI1_D[8] of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG2[4] of instance: src.

43.3.106 IOMUXC_SW_MUX_CTL_PAD_EIM_DA2 (IOMUXC_EIM_DA2)

Address: IOMUXC_EIM_DA2 is 53FA_8000h base + 1A4h offset = 53FA_81A4h

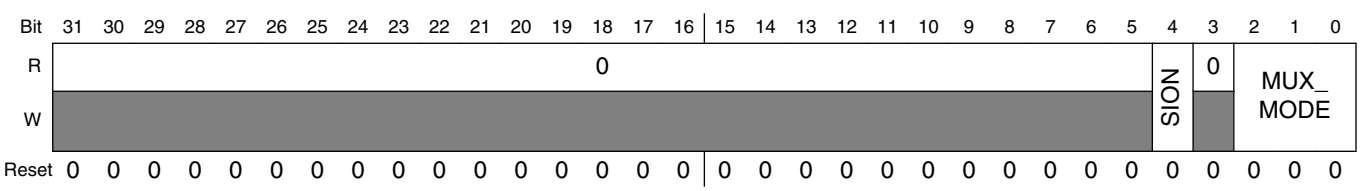


IOMUXC_EIM_DA2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_DA2. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_DA2. 000 Select mux mode: ALT0 mux port: NAND_WEIM_DA[2] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[2] of instance: gpio3. 011 Select mux mode: ALT3 mux port: DISP1_DAT[7] of instance: ipu. 100 Select mux mode: ALT4 mux port: CSI1_D[7] of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG2[3] of instance: src.

43.3.107 IOMUXC_SW_MUX_CTL_PAD_EIM_DA3 (IOMUXC_EIM_DA3)

Address: IOMUXC_EIM_DA3 is 53FA_8000h base + 1A8h offset = 53FA_81A8h



IOMUXC_EIM_DA3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved

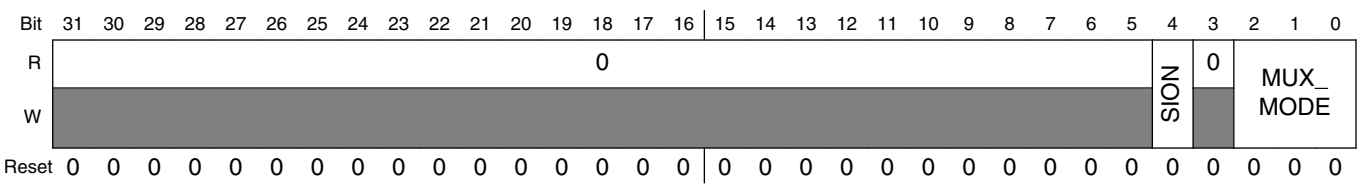
Table continues on the next page...

IOMUXC_EIM_DA3 field descriptions (continued)

Field	Description
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_DA3. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_DA3. 000 Select mux mode: ALT0 mux port: NAND_WEIM_DA[3] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[3] of instance: gpio3. 011 Select mux mode: ALT3 mux port: DISP1_DAT[6] of instance: ipu. 100 Select mux mode: ALT4 mux port: CSI1_D[6] of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG2[2] of instance: src.

43.3.108 IOMUXC_SW_MUX_CTL_PAD_EIM_DA4 (IOMUXC_EIM_DA4)

Address: IOMUXC_EIM_DA4 is 53FA_8000h base + 1ACh offset = 53FA_81ACh

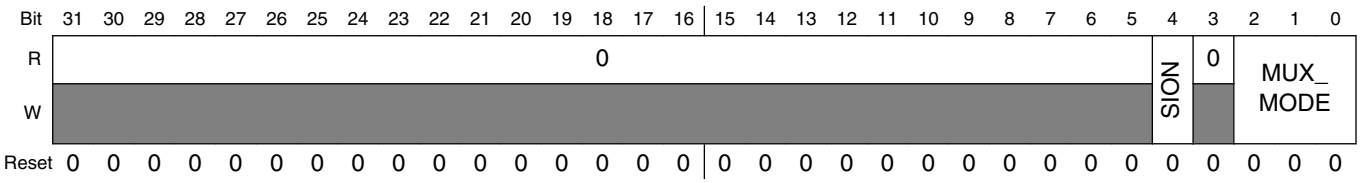


IOMUXC_EIM_DA4 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_DA4. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_DA4. 000 Select mux mode: ALT0 mux port: NAND_WEIM_DA[4] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[4] of instance: gpio3. 011 Select mux mode: ALT3 mux port: DISP1_DAT[5] of instance: ipu. 100 Select mux mode: ALT4 mux port: CSI1_D[5] of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG3[7] of instance: src.

43.3.109 IOMUXC_SW_MUX_CTL_PAD_EIM_DA5 (IOMUXC_EIM_DA5)

Address: IOMUXC_EIM_DA5 is 53FA_8000h base + 1B0h offset = 53FA_81B0h

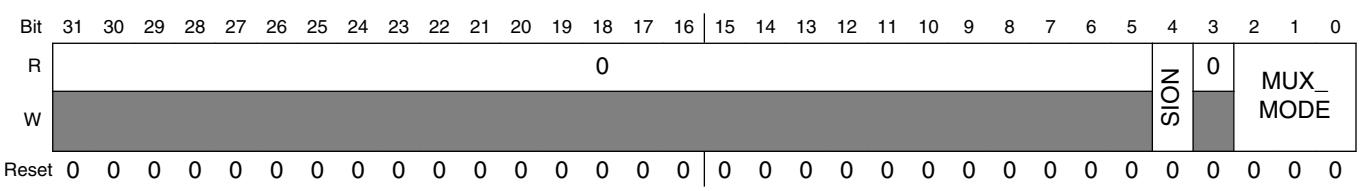


IOMUXC_EIM_DA5 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_DA5. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_DA5. 000 Select mux mode: ALT0 mux port: NAND_WEIM_DA[5] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[5] of instance: gpio3. 011 Select mux mode: ALT3 mux port: DISP1_DAT[4] of instance: ipu. 100 Select mux mode: ALT4 mux port: CSI1_D[4] of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG3[6] of instance: src.

43.3.110 IOMUXC_SW_MUX_CTL_PAD_EIM_DA6 (IOMUXC_EIM_DA6)

Address: IOMUXC_EIM_DA6 is 53FA_8000h base + 1B4h offset = 53FA_81B4h



IOMUXC_EIM_DA6 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved

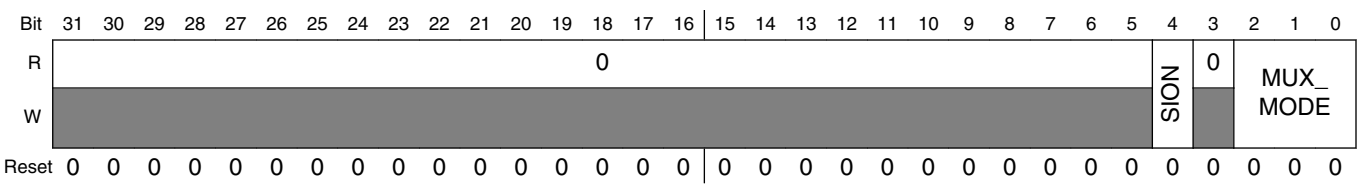
Table continues on the next page...

IOMUXC_EIM_DA6 field descriptions (continued)

Field	Description
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_DA6. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_DA6. 000 Select mux mode: ALT0 mux port: NAND_WEIM_DA[6] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[6] of instance: gpio3. 011 Select mux mode: ALT3 mux port: DISP1_DAT[3] of instance: ipu. 100 Select mux mode: ALT4 mux port: CSI1_D[3] of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG3[5] of instance: src.

43.3.111 IOMUXC_SW_MUX_CTL_PAD_EIM_DA7 (IOMUXC_EIM_DA7)

Address: IOMUXC_EIM_DA7 is 53FA_8000h base + 1B8h offset = 53FA_81B8h

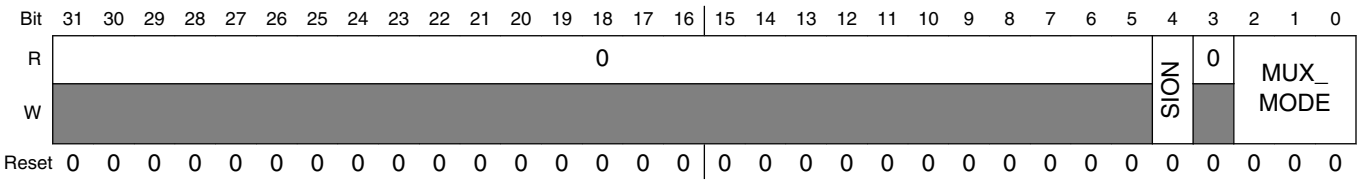


IOMUXC_EIM_DA7 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_DA7. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_DA7. 000 Select mux mode: ALT0 mux port: NAND_WEIM_DA[7] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[7] of instance: gpio3. 011 Select mux mode: ALT3 mux port: DISP1_DAT[2] of instance: ipu. 100 Select mux mode: ALT4 mux port: CSI1_D[2] of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG3[4] of instance: src.

43.3.112 IOMUXC_SW_MUX_CTL_PAD_EIM_DA8 (IOMUXC_EIM_DA8)

Address: IOMUXC_EIM_DA8 is 53FA_8000h base + 1BCh offset = 53FA_81BCh

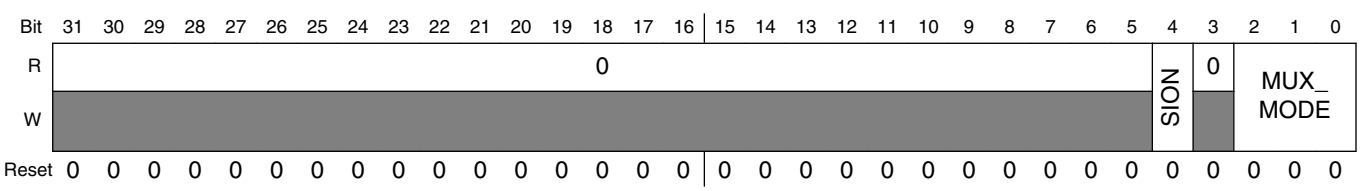


IOMUXC_EIM_DA8 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_DA8. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_DA8. 000 Select mux mode: ALT0 mux port: NAND_WEIM_DA[8] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[8] of instance: gpio3. 011 Select mux mode: ALT3 mux port: DISP1_DAT[1] of instance: ipu. 100 Select mux mode: ALT4 mux port: CSI1_D[1] of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG3[3] of instance: src.

43.3.113 IOMUXC_SW_MUX_CTL_PAD_EIM_DA9 (IOMUXC_EIM_DA9)

Address: IOMUXC_EIM_DA9 is 53FA_8000h base + 1C0h offset = 53FA_81C0h



IOMUXC_EIM_DA9 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved

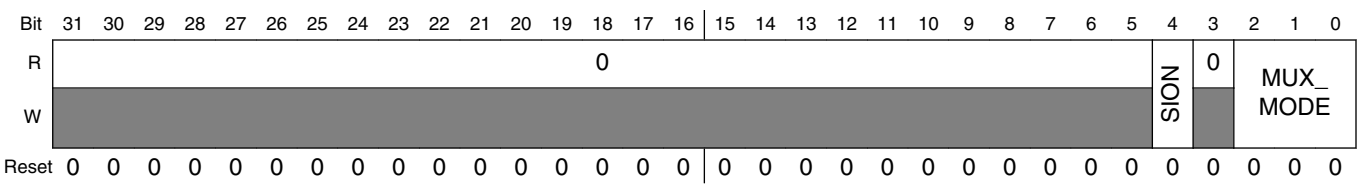
Table continues on the next page...

IOMUXC_EIM_DA9 field descriptions (continued)

Field	Description
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_DA9. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_DA9. 000 Select mux mode: ALT0 mux port: NAND_WEIM_DA[9] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[9] of instance: gpio3. 011 Select mux mode: ALT3 mux port: DISP1_DAT[0] of instance: ipu. 100 Select mux mode: ALT4 mux port: CSI1_D[0] of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG3[2] of instance: src.

43.3.114 IOMUXC_SW_MUX_CTL_PAD_EIM_DA10 (IOMUXC_EIM_DA10)

Address: IOMUXC_EIM_DA10 is 53FA_8000h base + 1C4h offset = 53FA_81C4h

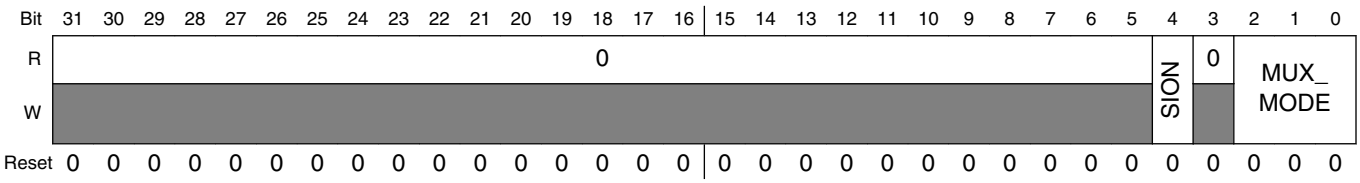


IOMUXC_EIM_DA10 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_DA10. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: EIM_DA10. NOTE: Pad EIM_DA10 is involved in Daisy Chain. - Config Register IPU_IPP_IND_SENS1_DATA_EN_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: NAND_WEIM_DA[10] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[10] of instance: gpio3. 011 Select mux mode: ALT3 mux port: DI1_PIN15 of instance: ipu. 100 Select mux mode: ALT4 mux port: CSI1_DATA_EN of instance: ipu. 111 Select mux mode: ALT7 mux port: BT_CFG3[1] of instance: src.

43.3.115 IOMUXC_SW_MUX_CTL_PAD_EIM_DA11 (IOMUXC_EIM_DA11)

Address: IOMUXC_EIM_DA11 is 53FA_8000h base + 1C8h offset = 53FA_81C8h

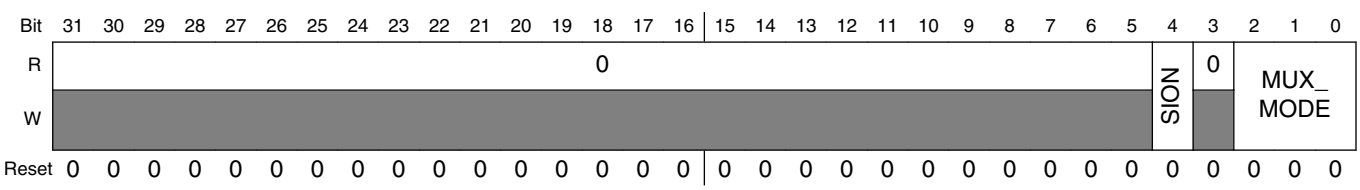


IOMUXC_EIM_DA11 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_DA11. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: EIM_DA11. NOTE: Pad EIM_DA11 is involved in Daisy Chain. - Config Register IPU_IPP_IND_SENS1_HSYNC_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: NAND_WEIM_DA[11] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[11] of instance: gpio3. 011 Select mux mode: ALT3 mux port: DI1_PIN2 of instance: ipu. 100 Select mux mode: ALT4 mux port: CSI1_HSYNC of instance: ipu.

43.3.116 IOMUXC_SW_MUX_CTL_PAD_EIM_DA12 (IOMUXC_EIM_DA12)

Address: IOMUXC_EIM_DA12 is 53FA_8000h base + 1CCh offset = 53FA_81CCh

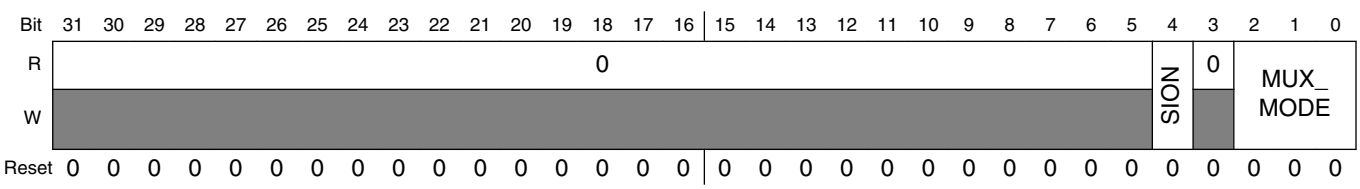


IOMUXC_EIM_DA12 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_DA12. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: EIM_DA12. NOTE: Pad EIM_DA12 is involved in Daisy Chain. - Config Register IPU_IPP_IND_SENS1_VSYNC_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: NAND_WEIM_DA[12] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[12] of instance: gpio3. 011 Select mux mode: ALT3 mux port: DI1_PIN3 of instance: ipu. 100 Select mux mode: ALT4 mux port: CSI1_VSYNC of instance: ipu.

43.3.117 IOMUXC_SW_MUX_CTL_PAD_EIM_DA13 (IOMUXC_EIM_DA13)

Address: IOMUXC_EIM_DA13 is 53FA_8000h base + 1D0h offset = 53FA_81D0h



IOMUXC_EIM_DA13 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_DA13. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: EIM_DA13. NOTE: Pad EIM_DA13 is involved in Daisy Chain. - Config Register CCM_IPP_DI1_CLK_SELECT_INPUT for mode ALT4.

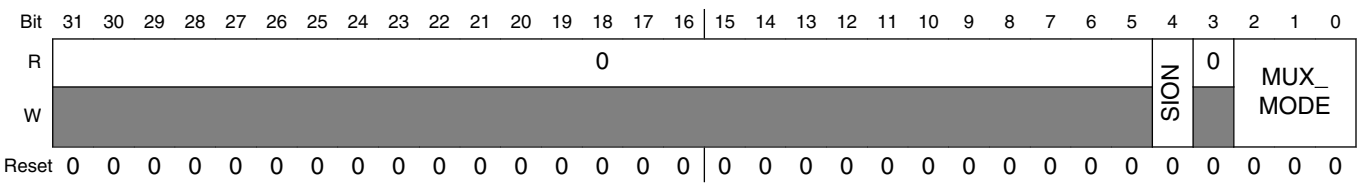
Table continues on the next page...

IOMUXC_EIM_DA13 field descriptions (continued)

Field	Description
	000 Select mux mode: ALT0 mux port: NAND_WEIM_DA[13] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[13] of instance: gpio3. 011 Select mux mode: ALT3 mux port: DI1_D0_CS of instance: ipu. 100 Select mux mode: ALT4 mux port: DI1_EXT_CLK of instance: ccm.

43.3.118 IOMUXC_SW_MUX_CTL_PAD_EIM_DA14 (IOMUXC_EIM_DA14)

Address: IOMUXC_EIM_DA14 is 53FA_8000h base + 1D4h offset = 53FA_81D4h

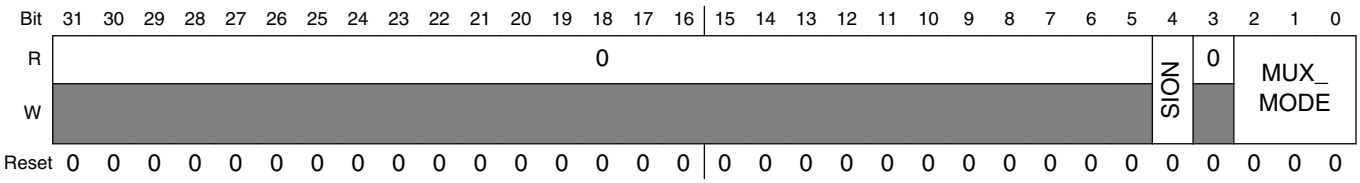


IOMUXC_EIM_DA14 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_DA14. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: EIM_DA14. 000 Select mux mode: ALT0 mux port: NAND_WEIM_DA[14] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[14] of instance: gpio3. 011 Select mux mode: ALT3 mux port: DI1_D1_CS of instance: ipu. 100 Select mux mode: ALT4 mux port: DI0_EXT_CLK of instance: ccm.

43.3.119 IOMUXC_SW_MUX_CTL_PAD_EIM_DA15 (IOMUXC_EIM_DA15)

Address: IOMUXC_EIM_DA15 is 53FA_8000h base + 1D8h offset = 53FA_81D8h

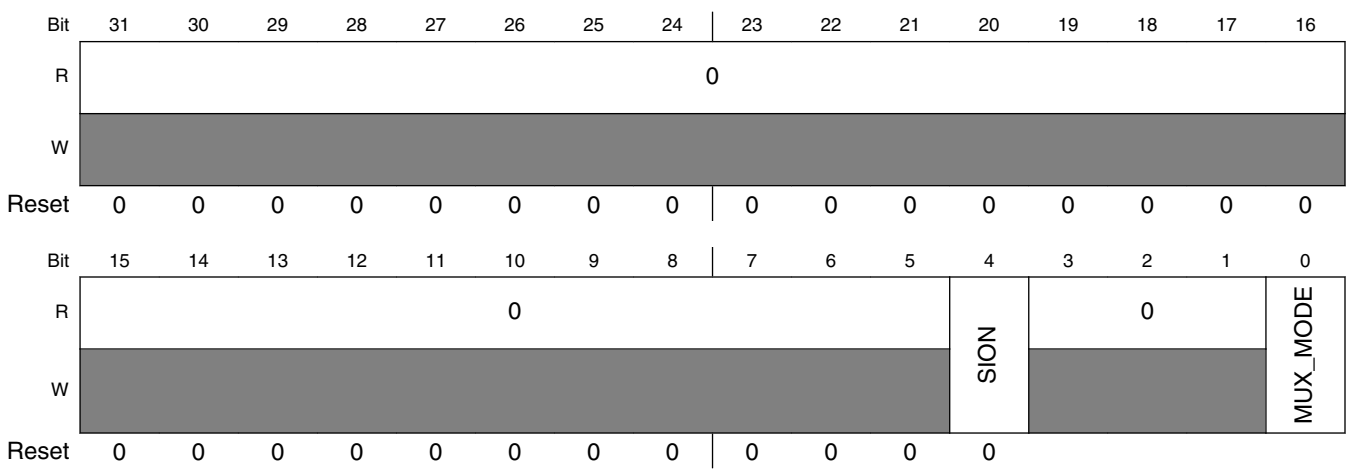


IOMUXC_EIM_DA15 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_DA15. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: EIM_DA15. 000 Select mux mode: ALT0 mux port: NAND_WEIM_DA[15] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[15] of instance: gpio3. 011 Select mux mode: ALT3 mux port: DI1_PIN1 of instance: ipu. 100 Select mux mode: ALT4 mux port: DI1_PIN4 of instance: ipu.

43.3.120 IOMUXC_SW_MUX_CTL_PAD_NANDF_WE_B (IOMUXC_NANDF_WE_B)

Address: IOMUXC_NANDF_WE_B is 53FA_8000h base + 1DCh offset = 53FA_81DCh

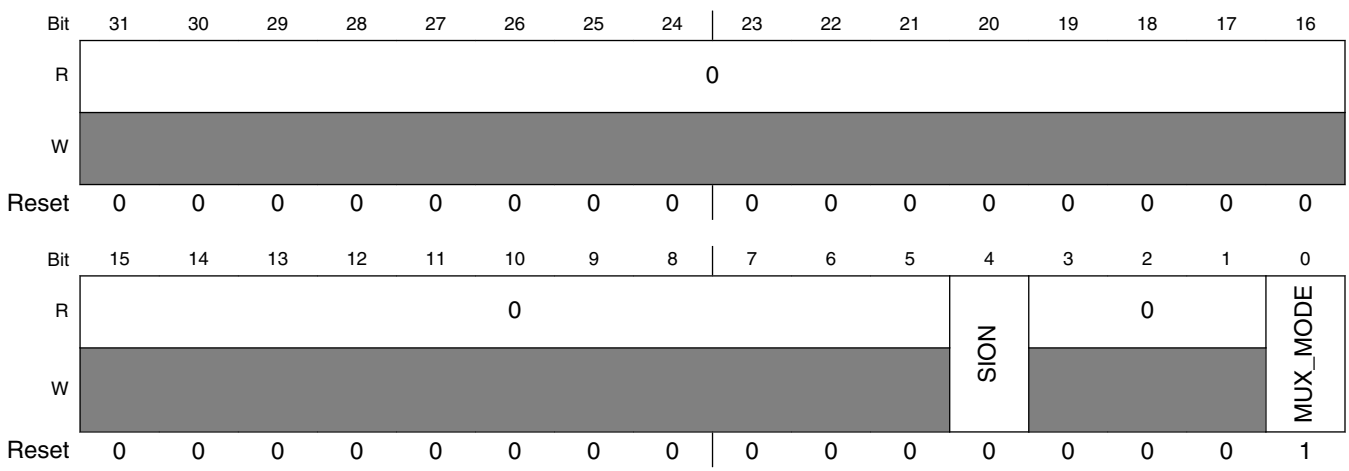


IOMUXC_NANDF_WE_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad NANDF_WE_B. 0 Input Path is determined by functionality of the selected mux mode (regular).
3–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: NANDF_WE_B. 0 Select mux mode: ALT0 mux port: NANDF_WE_B of instance: emi. 1 Select mux mode: ALT1 mux port: GPIO[12] of instance: gpio6.

43.3.121 IOMUXC_SW_MUX_CTL_PAD_NANDF_RE_B (IOMUXC_NANDF_RE_B)

Address: IOMUXC_NANDF_RE_B is 53FA_8000h base + 1E0h offset = 53FA_81E0h



IOMUXC_NANDF_RE_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad NANDF_RE_B. 0 Input Path is determined by functionality of the selected mux mode (regular).
3–1 Reserved	This read-only field is reserved and always has the value zero. Reserved

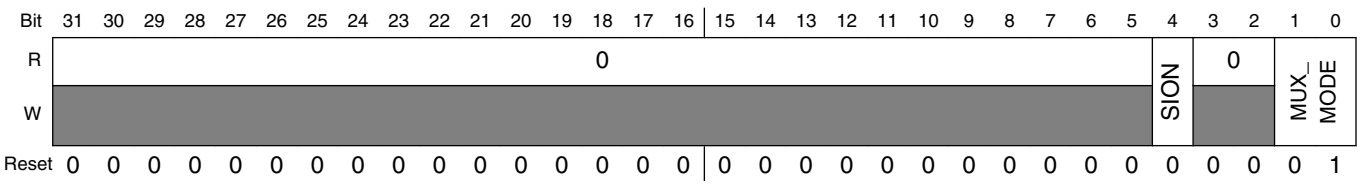
Table continues on the next page...

IOMUXC_NANDF_RE_B field descriptions (continued)

Field	Description
0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: NANDF_RE_B. 0 Select mux mode: ALT0 mux port: NANDF_RE_B of instance: emi. 1 Select mux mode: ALT1 mux port: GPIO[13] of instance: gpio6.

43.3.122 IOMUXC_SW_MUX_CTL_PAD_EIM_WAIT (IOMUXC_EIM_WAIT)

Address: IOMUXC_EIM_WAIT is 53FA_8000h base + 1E4h offset = 53FA_81E4h

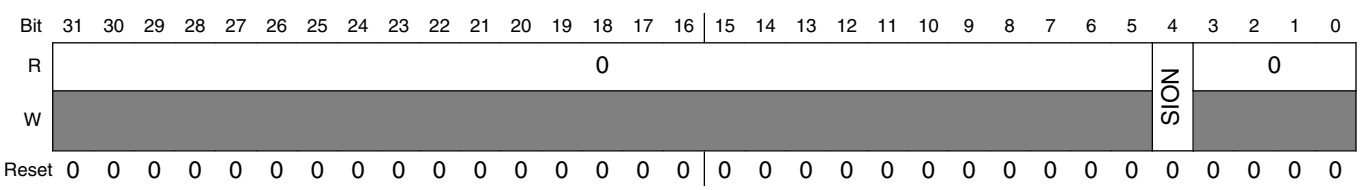


IOMUXC_EIM_WAIT field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_WAIT. 0 Input Path is determined by functionality of the selected mux mode (regular).
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: EIM_WAIT. 00 Select mux mode: ALT0 mux port: WEIM_WAIT of instance: emi. 01 Select mux mode: ALT1 mux port: GPIO[0] of instance: gpio5. 10 Select mux mode: ALT2 mux port: WEIM_DTACK_B of instance: emi.

43.3.123 IOMUXC_SW_MUX_CTL_PAD_EIM_BCLK (IOMUXC_EIM_BCLK)

Address: IOMUXC_EIM_BCLK is 53FA_8000h base + 1E8h offset = 53FA_81E8h



IOMUXC_EIM_BCLK field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad EIM_BCLK. 0 Input Path is determined by functionality of the selected mux mode (regular).
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.124 IOMUXC_SW_MUX_CTL_PAD_LVDS1_TX3_P (IOMUXC_LVDS1_TX3_P)

Address: IOMUXC_LVDS1_TX3_P is 53FA_8000h base + 1ECh offset = 53FA_81ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															MUX_MODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_LVDS1_TX3_P field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: LVDS1_TX3_P. 0 Select mux mode: ALT0 mux port: GPI[22] of instance: gpio6. 1 Select mux mode: ALT1 mux port: LVDS1_TX3 of instance: ldb.

43.3.125 IOMUXC_SW_MUX_CTL_PAD_LVDS1_TX2_P (IOMUXC_LVDS1_TX2_P)

Address: IOMUXC_LVDS1_TX2_P is 53FA_8000h base + 1F0h offset = 53FA_81F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															MUX_MODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_LVDS1_TX2_P field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: LVDS1_TX2_P. 0 Select mux mode: ALT0 mux port: GPI[24] of instance: gpio6. 1 Select mux mode: ALT1 mux port: LVDS1_TX2 of instance: ldb.

43.3.126 IOMUXC_SW_MUX_CTL_PAD_LVDS1_CLK_P (IOMUXC_LVDS1_CLK_P)

Address: IOMUXC_LVDS1_CLK_P is 53FA_8000h base + 1F4h offset = 53FA_81F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															MUX_MODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0				

IOMUXC_LVDS1_CLK_P field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: LVDS1_CLK_P. 0 Select mux mode: ALT0 mux port: GPI[26] of instance: gpio6. 1 Select mux mode: ALT1 mux port: LVDS1_CLK of instance: ldb.

43.3.127 IOMUXC_SW_MUX_CTL_PAD_LVDS1_TX1_P (IOMUXC_LVDS1_TX1_P)

Address: IOMUXC_LVDS1_TX1_P is 53FA_8000h base + 1F8h offset = 53FA_81F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															MUX_MODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_LVDS1_TX1_P field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: LVDS1_TX1_P. 0 Select mux mode: ALT0 mux port: GPI[28] of instance: gpio6. 1 Select mux mode: ALT1 mux port: LVDS1_TX1 of instance: ldb.

43.3.128 IOMUXC_SW_MUX_CTL_PAD_LVDS1_TX0_P (IOMUXC_LVDS1_TX0_P)

Address: IOMUXC_LVDS1_TX0_P is 53FA_8000h base + 1FCh offset = 53FA_81FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															MUX_MODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_LVDS1_TX0_P field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: LVDS1_TX0_P. 0 Select mux mode: ALT0 mux port: GPI[30] of instance: gpio6. 1 Select mux mode: ALT1 mux port: LVDS1_TX0 of instance: ldb.

43.3.129 IOMUXC_SW_MUX_CTL_PAD_LVDS0_TX3_P (IOMUXC_LVDS0_TX3_P)

Address: IOMUXC_LVDS0_TX3_P is 53FA_8000h base + 200h offset = 53FA_8200h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															MUX_MODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0				

IOMUXC_LVDS0_TX3_P field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: LVDS0_TX3_P. 0 Select mux mode: ALT0 mux port: GPI[22] of instance: gpio7. 1 Select mux mode: ALT1 mux port: LVDS0_TX3 of instance: ldb.

43.3.130 IOMUXC_SW_MUX_CTL_PAD_LVDS0_CLK_P (IOMUXC_LVDS0_CLK_P)

Address: IOMUXC_LVDS0_CLK_P is 53FA_8000h base + 204h offset = 53FA_8204h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															MUX_MODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_LVDS0_CLK_P field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: LVDS0_CLK_P. 0 Select mux mode: ALT0 mux port: GPI[24] of instance: gpio7. 1 Select mux mode: ALT1 mux port: LVDS0_CLK of instance: ldb.

43.3.131 IOMUXC_SW_MUX_CTL_PAD_LVDS0_TX2_P (IOMUXC_LVDS0_TX2_P)

Address: IOMUXC_LVDS0_TX2_P is 53FA_8000h base + 208h offset = 53FA_8208h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															MUX_MODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_LVDS0_TX2_P field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: LVDS0_TX2_P. 0 Select mux mode: ALT0 mux port: GPI[26] of instance: gpio7. 1 Select mux mode: ALT1 mux port: LVDS0_TX2 of instance: ldb.

43.3.132 IOMUXC_SW_MUX_CTL_PAD_LVDS0_TX1_P (IOMUXC_LVDS0_TX1_P)

Address: IOMUXC_LVDS0_TX1_P is 53FA_8000h base + 20Ch offset = 53FA_820Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															MUX_MODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0				

IOMUXC_LVDS0_TX1_P field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: LVDS0_TX1_P. 0 Select mux mode: ALT0 mux port: GPI[28] of instance: gpio7. 1 Select mux mode: ALT1 mux port: LVDS0_TX1 of instance: ldb.

43.3.133 IOMUXC_SW_MUX_CTL_PAD_LVDS0_TX0_P (IOMUXC_LVDS0_TX0_P)

Address: IOMUXC_LVDS0_TX0_P is 53FA_8000h base + 210h offset = 53FA_8210h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

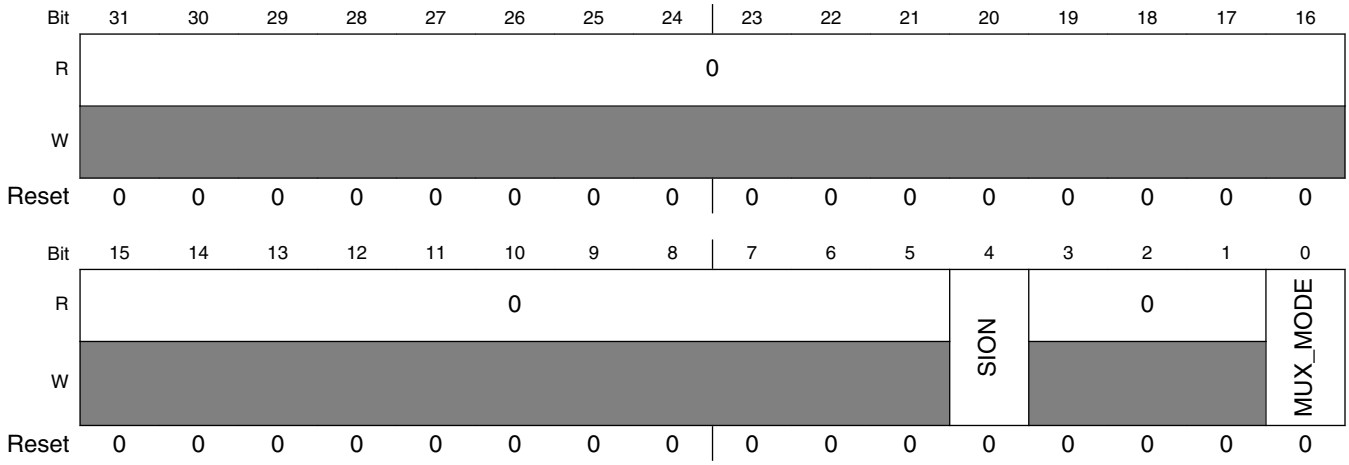
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															MUX_MODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_LVDS0_TX0_P field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: LVDS0_TX0_P. 0 Select mux mode: ALT0 mux port: GPI[30] of instance: gpio7. 1 Select mux mode: ALT1 mux port: LVDS0_TX0 of instance: ldb.

43.3.134 IOMUXC_SW_MUX_CTL_PAD_GPIO_10 (IOMUXC_GPIO_10)

Address: IOMUXC_GPIO_10 is 53FA_8000h base + 214h offset = 53FA_8214h

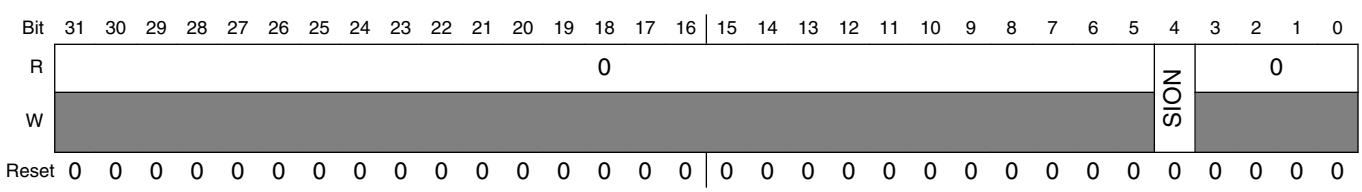


IOMUXC_GPIO_10 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_10. 0 Input Path is determined by functionality of the selected mux mode (regular).
3–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 MUX_MODE	MUX Mode Select Field. Select 1 of 2 iomux modes to be used for pad: GPIO_10. 0 Select mux mode: ALT0 mux port: GPIO[0] of instance: gpio4. 1 Select mux mode: ALT1 mux port: 32K_OUT of instance: osc32k.

43.3.135 IOMUXC_SW_MUX_CTL_PAD_GPIO_11 (IOMUXC_GPIO_11)

Address: IOMUXC_GPIO_11 is 53FA_8000h base + 218h offset = 53FA_8218h

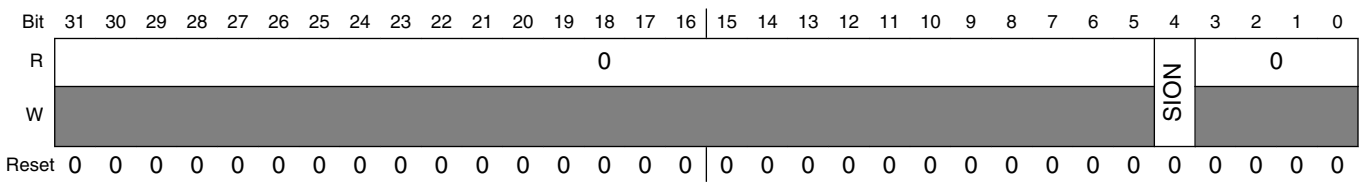


IOMUXC_GPIO_11 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_11. 0 Input Path is determined by functionality of the selected mux mode (regular).
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.136 IOMUXC_SW_MUX_CTL_PAD_GPIO_12 (IOMUXC_GPIO_12)

Address: IOMUXC_GPIO_12 is 53FA_8000h base + 21Ch offset = 53FA_821Ch

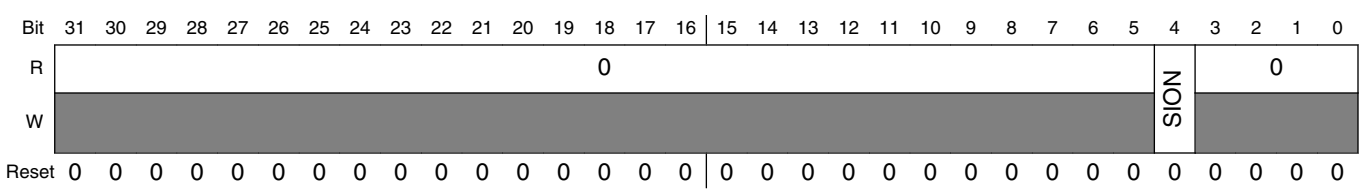


IOMUXC_GPIO_12 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_12. 0 Input Path is determined by functionality of the selected mux mode (regular).
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.137 IOMUXC_SW_MUX_CTL_PAD_GPIO_13 (IOMUXC_GPIO_13)

Address: IOMUXC_GPIO_13 is 53FA_8000h base + 220h offset = 53FA_8220h

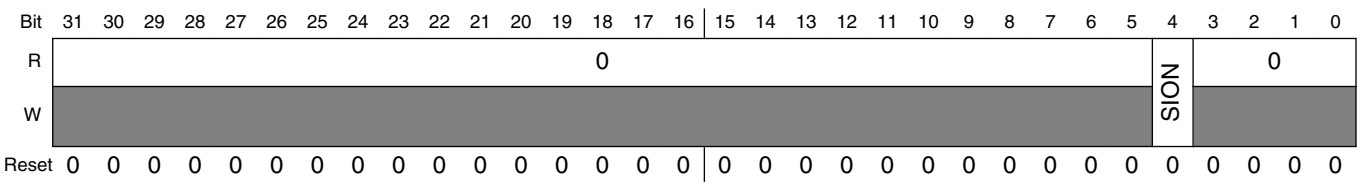


IOMUXC_GPIO_13 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_13. 0 Input Path is determined by functionality of the selected mux mode (regular).
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.138 IOMUXC_SW_MUX_CTL_PAD_GPIO_14 (IOMUXC_GPIO_14)

Address: IOMUXC_GPIO_14 is 53FA_8000h base + 224h offset = 53FA_8224h

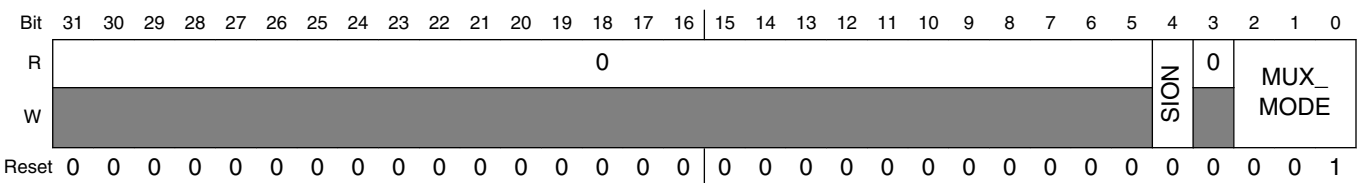


IOMUXC_GPIO_14 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_14. 0 Input Path is determined by functionality of the selected mux mode (regular).
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.139 IOMUXC_SW_MUX_CTL_PAD_NANDF_CLE (IOMUXC_NANDF_CLE)

Address: IOMUXC_NANDF_CLE is 53FA_8000h base + 228h offset = 53FA_8228h



IOMUXC_NANDF_CLE field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad NANDF_CLE. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: NANDF_CLE. 000 Select mux mode: ALT0 mux port: NANDF_CLE of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[7] of instance: gpio6. 111 Select mux mode: ALT7 mux port: VSTATUS[0] of instance: usbphy1.

43.3.140 IOMUXC_SW_MUX_CTL_PAD_NANDF_ALE (IOMUXC_NANDF_ALE)

Address: IOMUXC_NANDF_ALE is 53FA_8000h base + 22Ch offset = 53FA_822Ch

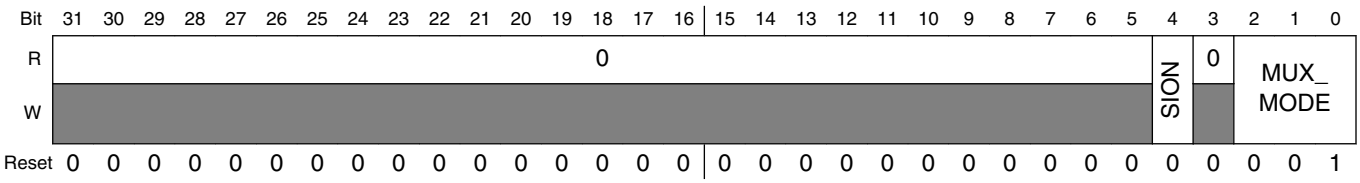
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_ MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_NANDF_ALE field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad NANDF_ALE. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: NANDF_ALE. 000 Select mux mode: ALT0 mux port: NANDF_ALE of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[8] of instance: gpio6. 111 Select mux mode: ALT7 mux port: VSTATUS[1] of instance: usbphy1.

43.3.141 IOMUXC_SW_MUX_CTL_PAD_NANDF_WP_B (IOMUXC_NANDF_WP_B)

Address: IOMUXC_NANDF_WP_B is 53FA_8000h base + 230h offset = 53FA_8230h

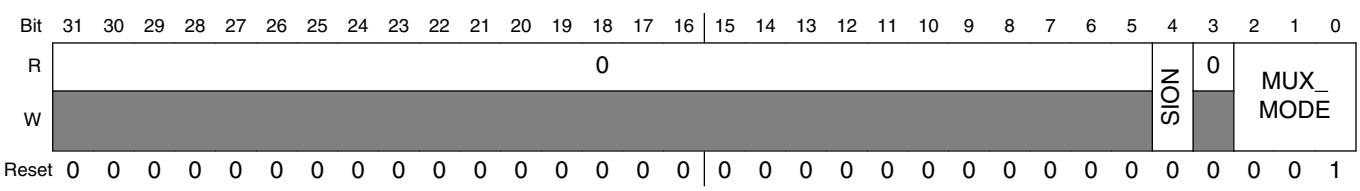


IOMUXC_NANDF_WP_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad NANDF_WP_B. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: NANDF_WP_B. 000 Select mux mode: ALT0 mux port: NANDF_WP_B of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[9] of instance: gpio6. 111 Select mux mode: ALT7 mux port: VSTATUS[2] of instance: usbphy1.

43.3.142 IOMUXC_SW_MUX_CTL_PAD_NANDF_RB0 (IOMUXC_NANDF_RB0)

Address: IOMUXC_NANDF_RB0 is 53FA_8000h base + 234h offset = 53FA_8234h



IOMUXC_NANDF_RB0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality.

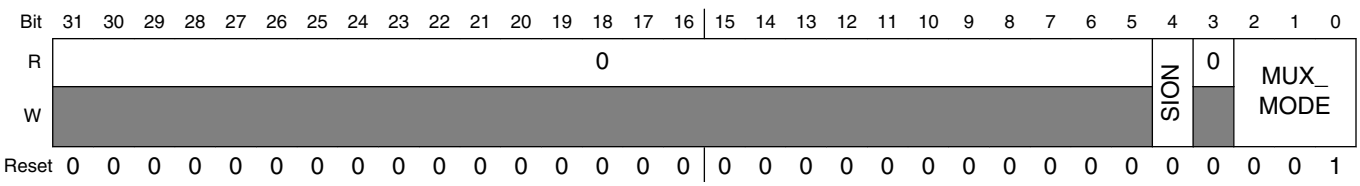
Table continues on the next page...

IOMUXC_NANDF_RB0 field descriptions (continued)

Field	Description
	1 Force input path of pad NANDF_RB0. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: NANDF_RB0. 000 Select mux mode: ALT0 mux port: NANDF_RB[0] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[10] of instance: gpio6. 111 Select mux mode: ALT7 mux port: VSTATUS[3] of instance: usbphy1.

43.3.143 IOMUXC_SW_MUX_CTL_PAD_NANDF_CS0 (IOMUXC_NANDF_CS0)

Address: IOMUXC_NANDF_CS0 is 53FA_8000h base + 238h offset = 53FA_8238h

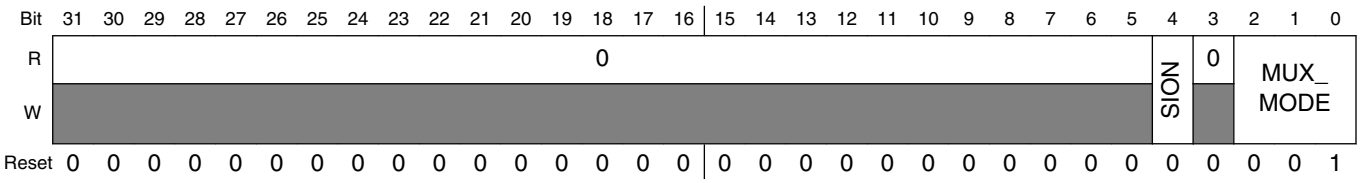


IOMUXC_NANDF_CS0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad NANDF_CS0. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: NANDF_CS0. 000 Select mux mode: ALT0 mux port: NANDF_CS[0] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[11] of instance: gpio6. 111 Select mux mode: ALT7 mux port: VSTATUS[4] of instance: usbphy1.

43.3.144 IOMUXC_SW_MUX_CTL_PAD_NANDF_CS1 (IOMUXC_NANDF_CS1)

Address: IOMUXC_NANDF_CS1 is 53FA_8000h base + 23Ch offset = 53FA_823Ch

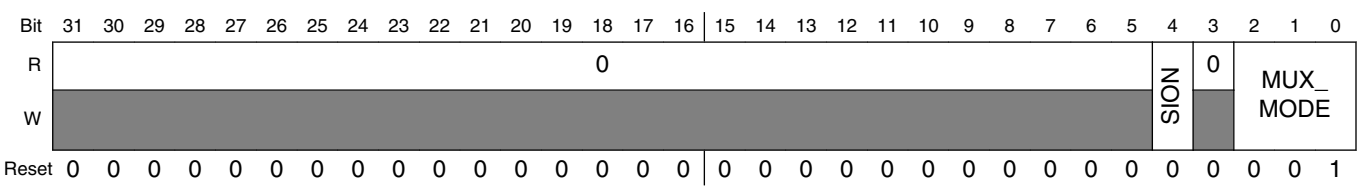


IOMUXC_NANDF_CS1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad NANDF_CS1. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: NANDF_CS1. NOTE: Pad NANDF_CS1 is involved in Daisy Chain. - Config Register MLB_MLBCLK_IN_SELECT_INPUT for mode ALT6. 000 Select mux mode: ALT0 mux port: NANDF_CS[1] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[14] of instance: gpio6. 110 Select mux mode: ALT6 mux port: MLBCLK of instance: mlb. 111 Select mux mode: ALT7 mux port: VSTATUS[5] of instance: usbphy1.

43.3.145 IOMUXC_SW_MUX_CTL_PAD_NANDF_CS2 (IOMUXC_NANDF_CS2)

Address: IOMUXC_NANDF_CS2 is 53FA_8000h base + 240h offset = 53FA_8240h

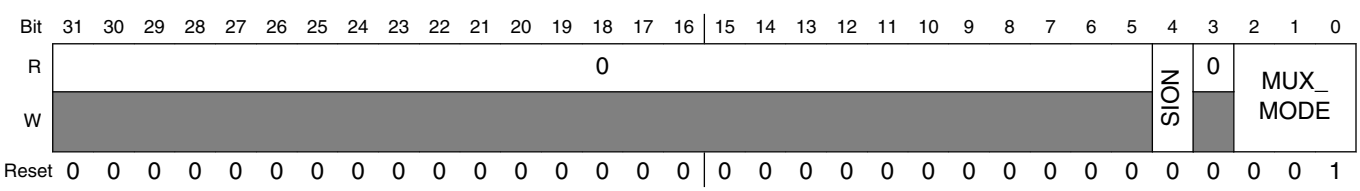


IOMUXC_NANDF_CS2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad NANDF_CS2. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: NANDF_CS2. NOTE: Pad NANDF_CS2 is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_SDO0_SELECT_INPUT for mode ALT3. - Config Register MLB_MLBSIG_IN_SELECT_INPUT for mode ALT6. 000 Select mux mode: ALT0 mux port: NANDF_CS[2] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[15] of instance: gpio6. 010 Select mux mode: ALT2 mux port: SISG[0] of instance: ipu. 011 Select mux mode: ALT3 mux port: TX0 of instance: esai1. 100 Select mux mode: ALT4 mux port: WEIM_CRE of instance: emi. 101 Select mux mode: ALT5 mux port: CSI0_MCLK of instance: ccm. 110 Select mux mode: ALT6 mux port: MLBSIG of instance: mlb. 111 Select mux mode: ALT7 mux port: VSTATUS[6] of instance: usbphy1.

43.3.146 IOMUXC_SW_MUX_CTL_PAD_NANDF_CS3 (IOMUXC_NANDF_CS3)

Address: IOMUXC_NANDF_CS3 is 53FA_8000h base + 244h offset = 53FA_8244h



IOMUXC_NANDF_CS3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad NANDF_CS3. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved

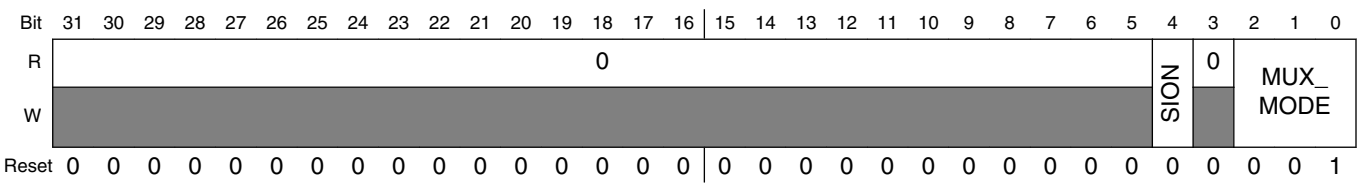
Table continues on the next page...

IOMUXC_NANDF_CS3 field descriptions (continued)

Field	Description
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: NANDF_CS3. NOTE: Pad NANDF_CS3 is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_SDO1_SELECT_INPUT for mode ALT3. - Config Register MLB_MLBDAT_IN_SELECT_INPUT for mode ALT6. 000 Select mux mode: ALT0 mux port: NANDF_CS[3] of instance: emi. 001 Select mux mode: ALT1 mux port: GPIO[16] of instance: gpio6. 010 Select mux mode: ALT2 mux port: SISG[1] of instance: ipu. 011 Select mux mode: ALT3 mux port: TX1 of instance: esai1. 100 Select mux mode: ALT4 mux port: WEIM_A[26] of instance: emi. 110 Select mux mode: ALT6 mux port: MLBDAT of instance: mlb. 111 Select mux mode: ALT7 mux port: VSTATUS[7] of instance: usbphy1.

43.3.147 IOMUXC_SW_MUX_CTL_PAD_FEC_MDIO (IOMUXC_FEC_MDIO)

Address: IOMUXC_FEC_MDIO is 53FA_8000h base + 248h offset = 53FA_8248h



IOMUXC_FEC_MDIO field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad FEC_MDIO. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: FEC_MDIO. NOTE: Pad FEC_MDIO is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_SCKR_SELECT_INPUT for mode ALT2. - Config Register FEC_FEC_COL_SELECT_INPUT for mode ALT3. - Config Register FEC_FEC_MDI_SELECT_INPUT for mode ALT0. 000 Select mux mode: ALT0 mux port: MDIO of instance: fec. 001 Select mux mode: ALT1 mux port: GPIO[22] of instance: gpio1.

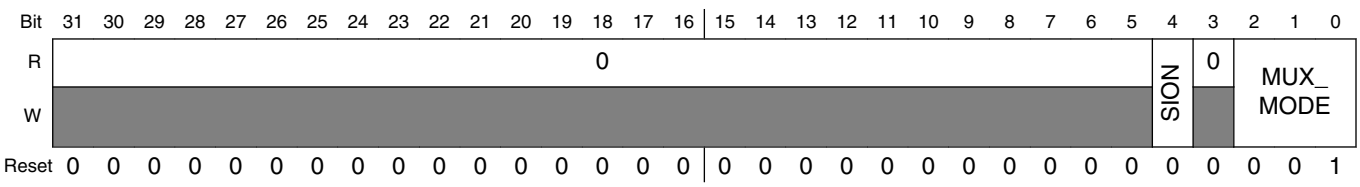
Table continues on the next page...

IOMUXC_FEC_MDIO field descriptions (continued)

Field	Description
	010 Select mux mode: ALT2 mux port: SCKR of instance: esai1. 011 Select mux mode: ALT3 mux port: COL of instance: fec. 100 Select mux mode: ALT4 mux port: CE_RTC_PS2 of instance: rtc. 101 Select mux mode: ALT5 mux port: DEBUG_BUS_DEVICE[3] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[49] of instance: emi.

43.3.148 IOMUXC_SW_MUX_CTL_PAD_FEC_REF_CLK (IOMUXC_FEC_REF_CLK)

Address: IOMUXC_FEC_REF_CLK is 53FA_8000h base + 24Ch offset = 53FA_824Ch

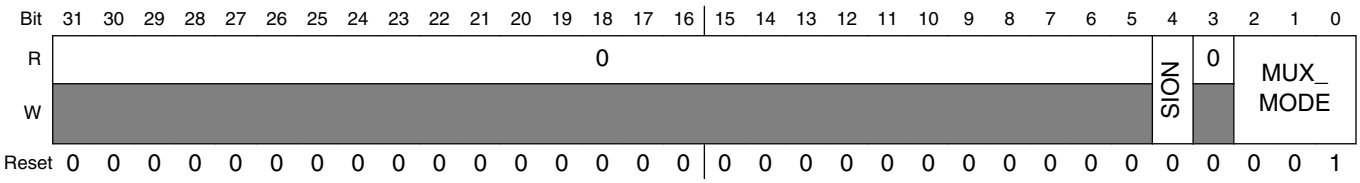


IOMUXC_FEC_REF_CLK field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad FEC_REF_CLK. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: FEC_REF_CLK. NOTE: Pad FEC_REF_CLK is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_FSR_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: TX_CLK of instance: fec. 001 Select mux mode: ALT1 mux port: GPIO[23] of instance: gpio1. 010 Select mux mode: ALT2 mux port: FSR of instance: esai1. 101 Select mux mode: ALT5 mux port: DEBUG_BUS_DEVICE[4] of instance: sdma. 110 Select mux mode: ALT6 mux port: EMI_DEBUG[50] of instance: emi.

43.3.149 IOMUXC_SW_MUX_CTL_PAD_FEC_RX_ER (IOMUXC_FEC_RX_ER)

Address: IOMUXC_FEC_RX_ER is 53FA_8000h base + 250h offset = 53FA_8250h

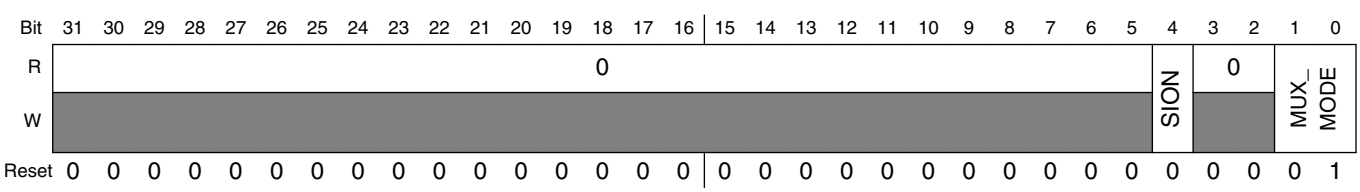


IOMUXC_FEC_RX_ER field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad FEC_RX_ER. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: FEC_RX_ER. NOTE: Pad FEC_RX_ER is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_HCKR_SELECT_INPUT for mode ALT2. - Config Register FEC_FEC_RX_CLK_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: RX_ER of instance: fec. 001 Select mux mode: ALT1 mux port: GPIO[24] of instance: gpio1. 010 Select mux mode: ALT2 mux port: HCKR of instance: esai1. 011 Select mux mode: ALT3 mux port: RX_CLK of instance: fec. 100 Select mux mode: ALT4 mux port: CE_RTC_PS3 of instance: rtc.

43.3.150 IOMUXC_SW_MUX_CTL_PAD_FEC_CRS_DV (IOMUXC_FEC_CRS_DV)

Address: IOMUXC_FEC_CRS_DV is 53FA_8000h base + 254h offset = 53FA_8254h

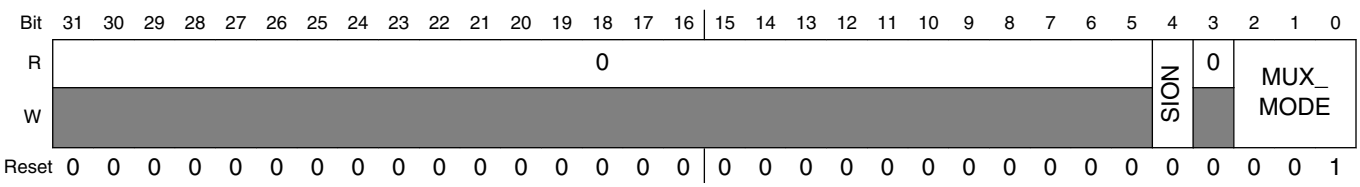


IOMUXC_FEC_CRS_DV field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad FEC_CRS_DV. 0 Input Path is determined by functionality of the selected mux mode (regular).
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: FEC_CRS_DV. NOTE: Pad FEC_CRS_DV is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_SCKT_SELECT_INPUT for mode ALT2. 00 Select mux mode: ALT0 mux port: RX_DV of instance: fec. 01 Select mux mode: ALT1 mux port: GPIO[25] of instance: gpio1. 10 Select mux mode: ALT2 mux port: SCKT of instance: esai1.

43.3.151 IOMUXC_SW_MUX_CTL_PAD_FEC_RXD1 (IOMUXC_FEC_RXD1)

Address: IOMUXC_FEC_RXD1 is 53FA_8000h base + 258h offset = 53FA_8258h



IOMUXC_FEC_RXD1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad FEC_RXD1. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: FEC_RXD1. NOTE: Pad FEC_RXD1 is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_FST_SELECT_INPUT for mode ALT2. - Config Register MLB_MLBSIG_IN_SELECT_INPUT for mode ALT3.

Table continues on the next page...

IOMUXC_FEC_RXD1 field descriptions (continued)

Field	Description
	000 Select mux mode: ALT0 mux port: RDATA[1] of instance: fec. 001 Select mux mode: ALT1 mux port: GPIO[26] of instance: gpio1. 010 Select mux mode: ALT2 mux port: FST of instance: esai1. 011 Select mux mode: ALT3 mux port: MLBSIG of instance: mlb. 100 Select mux mode: ALT4 mux port: CE_RTC_PS1 of instance: rtc.

43.3.152 IOMUXC_SW_MUX_CTL_PAD_FEC_RXD0 (IOMUXC_FEC_RXD0)

Address: IOMUXC_FEC_RXD0 is 53FA_8000h base + 25Ch offset = 53FA_825Ch

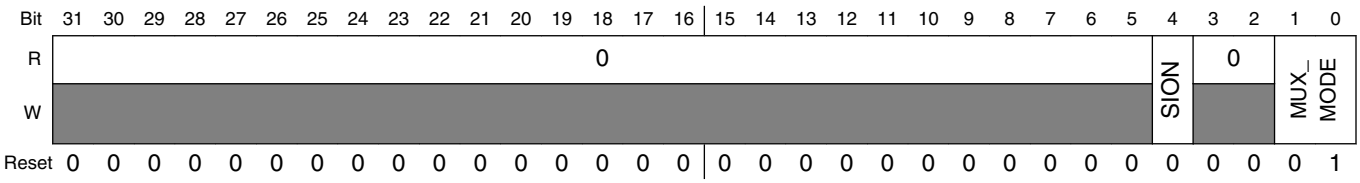
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0		MUX_MODE	
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_FEC_RXD0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad FEC_RXD0. 0 Input Path is determined by functionality of the selected mux mode (regular).
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: FEC_RXD0. NOTE: Pad FEC_RXD0 is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_HCKT_SELECT_INPUT for mode ALT2. 00 Select mux mode: ALT0 mux port: RDATA[0] of instance: fec. 01 Select mux mode: ALT1 mux port: GPIO[27] of instance: gpio1. 10 Select mux mode: ALT2 mux port: HCKT of instance: esai1. 11 Select mux mode: ALT3 mux port: 32K_OUT of instance: osc32k.

43.3.153 IOMUXC_SW_MUX_CTL_PAD_FEC_TX_EN (IOMUXC_FEC_TX_EN)

Address: IOMUXC_FEC_TX_EN is 53FA_8000h base + 260h offset = 53FA_8260h

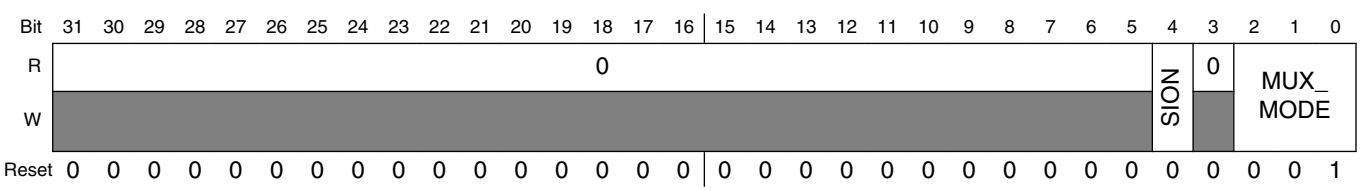


IOMUXC_FEC_TX_EN field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad FEC_TX_EN. 0 Input Path is determined by functionality of the selected mux mode (regular).
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 MUX_MODE	MUX Mode Select Field. Select 1 of 3 iomux modes to be used for pad: FEC_TX_EN. NOTE: Pad FEC_TX_EN is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_SDO3_SDI2_SELECT_INPUT for mode ALT2. 00 Select mux mode: ALT0 mux port: TX_EN of instance: fec. 01 Select mux mode: ALT1 mux port: GPIO[28] of instance: gpio1. 10 Select mux mode: ALT2 mux port: TX3_RX2 of instance: esai1.

43.3.154 IOMUXC_SW_MUX_CTL_PAD_FEC_TXD1 (IOMUXC_FEC_TXD1)

Address: IOMUXC_FEC_TXD1 is 53FA_8000h base + 264h offset = 53FA_8264h

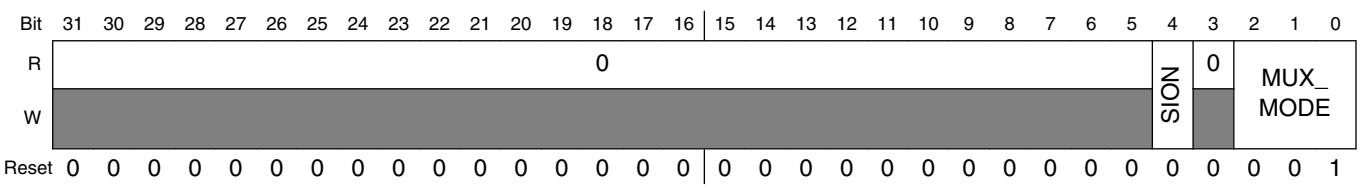


IOMUXC_FEC_TXD1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad FEC_TXD1. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: FEC_TXD1. NOTE: Pad FEC_TXD1 is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_SDO2_SDI3_SELECT_INPUT for mode ALT2. - Config Register MLB_MLBCLK_IN_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: TDATA[1] of instance: fec. 001 Select mux mode: ALT1 mux port: GPIO[29] of instance: gpio1. 010 Select mux mode: ALT2 mux port: TX2_RX3 of instance: esai1. 011 Select mux mode: ALT3 mux port: MLBCLK of instance: mlb. 100 Select mux mode: ALT4 mux port: CE_RTC_PRSC_CLK of instance: rtc.

43.3.155 IOMUXC_SW_MUX_CTL_PAD_FEC_TXD0 (IOMUXC_FEC_TXD0)

Address: IOMUXC_FEC_TXD0 is 53FA_8000h base + 268h offset = 53FA_8268h



IOMUXC_FEC_TXD0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad FEC_TXD0. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: FEC_TXD0. NOTE: Pad FEC_TXD0 is involved in Daisy Chain.

Table continues on the next page...

IOMUXC_FEC_TXD0 field descriptions (continued)

Field	Description
	- Config Register ESAI1_IPP_IND_SDO4_SDI1_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: TDATA[0] of instance: fec. 001 Select mux mode: ALT1 mux port: GPIO[30] of instance: gpio1. 010 Select mux mode: ALT2 mux port: TX4_RX1 of instance: esai1. 111 Select mux mode: ALT7 mux port: DATAOUT[0] of instance: usbphy2.

43.3.156 IOMUXC_SW_MUX_CTL_PAD_FEC_MDC (IOMUXC_FEC_MDC)

Address: IOMUXC_FEC_MDC is 53FA_8000h base + 26Ch offset = 53FA_826Ch

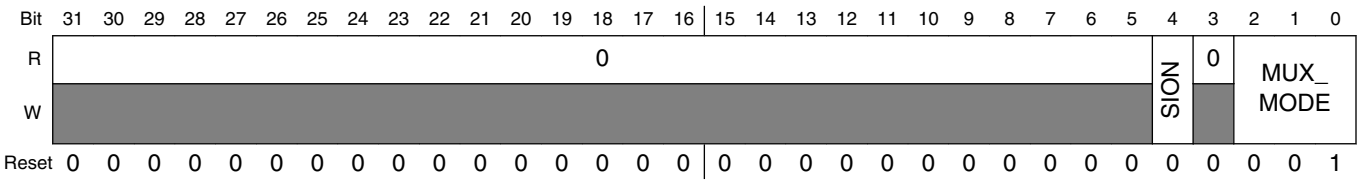
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_ MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_FEC_MDC field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad FEC_MDC. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: FEC_MDC. NOTE: Pad FEC_MDC is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_SDO5_SDI0_SELECT_INPUT for mode ALT2. - Config Register MLB_MLBDAT_IN_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: MDC of instance: fec. 001 Select mux mode: ALT1 mux port: GPIO[31] of instance: gpio1. 010 Select mux mode: ALT2 mux port: TX5_RX0 of instance: esai1. 011 Select mux mode: ALT3 mux port: MLBDAT of instance: mlb. 100 Select mux mode: ALT4 mux port: CE_RTC_ALARM1_TRIG of instance: rtc. 111 Select mux mode: ALT7 mux port: DATAOUT[1] of instance: usbphy2.

43.3.157 IOMUXC_SW_MUX_CTL_PAD_PATA_DIOW (IOMUXC_PATA_DIOW)

Address: IOMUXC_PATA_DIOW is 53FA_8000h base + 270h offset = 53FA_8270h

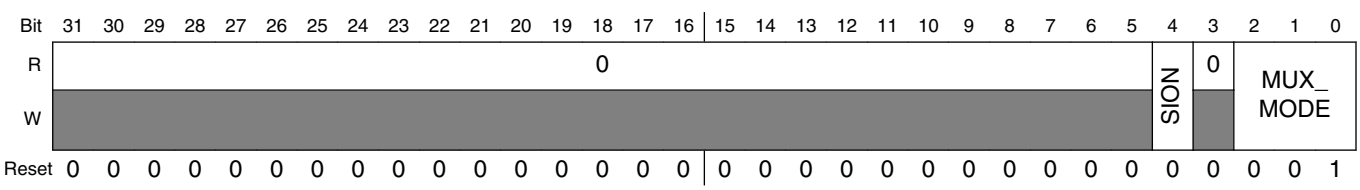


IOMUXC_PATA_DIOW field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DIOW. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: PATA_DIOW. NOTE: Pad PATA_DIOW is involved in Daisy Chain. - Config Register UART1_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: DIOW of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[17] of instance: gpio6. 011 Select mux mode: ALT3 mux port: TXD_MUX of instance: uart1. 111 Select mux mode: ALT7 mux port: DATAOUT[2] of instance: usbphy2.

43.3.158 IOMUXC_SW_MUX_CTL_PAD_PATA_DMACK (IOMUXC_PATA_DMACK)

Address: IOMUXC_PATA_DMACK is 53FA_8000h base + 274h offset = 53FA_8274h

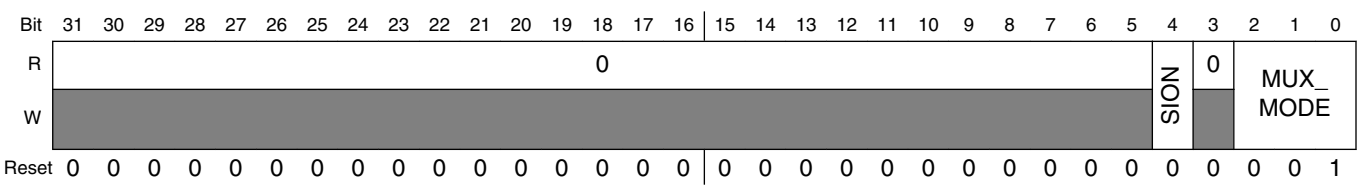


IOMUXC_PATA_DMACK field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DMACK. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: PATA_DMACK. NOTE: Pad PATA_DMACK is involved in Daisy Chain. - Config Register UART1_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: DMACK of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[18] of instance: gpio6. 011 Select mux mode: ALT3 mux port: RXD_MUX of instance: uart1. 111 Select mux mode: ALT7 mux port: DATAOUT[3] of instance: usbphy2.

43.3.159 IOMUXC_SW_MUX_CTL_PAD_PATA_DMARQ (IOMUXC_PATA_DMARQ)

Address: IOMUXC_PATA_DMARQ is 53FA_8000h base + 278h offset = 53FA_8278h



IOMUXC_PATA_DMARQ field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DMARQ. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: PATA_DMARQ. NOTE: Pad PATA_DMARQ is involved in Daisy Chain. - Config Register UART2_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT3.

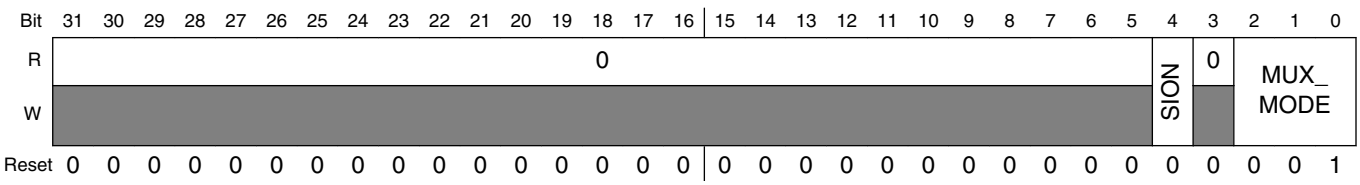
Table continues on the next page...

IOMUXC_PATA_DMARQ field descriptions (continued)

Field	Description
	000 Select mux mode: ALT0 mux port: DMARQ of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[0] of instance: gpio7. 011 Select mux mode: ALT3 mux port: TXD_MUX of instance: uart2. 101 Select mux mode: ALT5 mux port: CCM_OUT_0 of instance: ccm. 111 Select mux mode: ALT7 mux port: DATAOUT[4] of instance: usbphy2.

43.3.160 IOMUXC_SW_MUX_CTL_PAD_PATA_BUFFER_EN (IOMUXC_PATA_BUFFER_EN)

Address: IOMUXC_PATA_BUFFER_EN is 53FA_8000h base + 27Ch offset = 53FA_827Ch

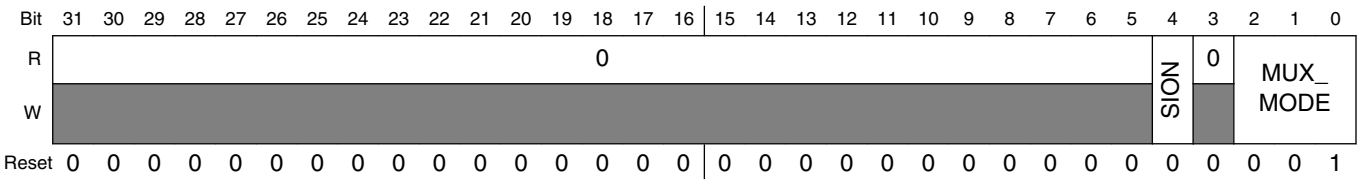


IOMUXC_PATA_BUFFER_EN field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_BUFFER_EN. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: PATA_BUFFER_EN. NOTE: Pad PATA_BUFFER_EN is involved in Daisy Chain. - Config Register UART2_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: BUFFER_EN of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[1] of instance: gpio7. 011 Select mux mode: ALT3 mux port: RXD_MUX of instance: uart2. 101 Select mux mode: ALT5 mux port: CCM_OUT_1 of instance: ccm. 111 Select mux mode: ALT7 mux port: DATAOUT[5] of instance: usbphy2.

43.3.161 IOMUXC_SW_MUX_CTL_PAD_PATA_INTRQ (IOMUXC_PATA_INTRQ)

Address: IOMUXC_PATA_INTRQ is 53FA_8000h base + 280h offset = 53FA_8280h

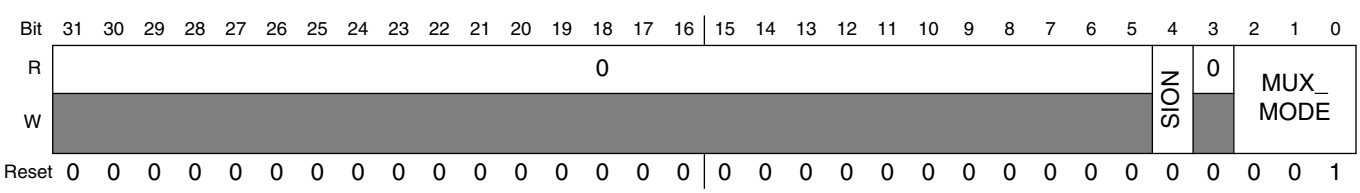


IOMUXC_PATA_INTRQ field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_INTRQ. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: PATA_INTRQ. NOTE: Pad PATA_INTRQ is involved in Daisy Chain. - Config Register UART2_IPP_UART_RTS_B_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: INTRQ of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[2] of instance: gpio7. 011 Select mux mode: ALT3 mux port: CTS of instance: uart2. 100 Select mux mode: ALT4 mux port: TXCAN of instance: can1. 101 Select mux mode: ALT5 mux port: CCM_OUT_2 of instance: ccm. 111 Select mux mode: ALT7 mux port: DATAOUT[6] of instance: usbphy2.

43.3.162 IOMUXC_SW_MUX_CTL_PAD_PATA_DIOR (IOMUXC_PATA_DIOR)

Address: IOMUXC_PATA_DIOR is 53FA_8000h base + 284h offset = 53FA_8284h



IOMUXC_PATA_DIOR field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DIOR. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: PATA_DIOR. NOTE: Pad PATA_DIOR is involved in Daisy Chain. - Config Register CAN1_IPP_IND_CANRX_SELECT_INPUT for mode ALT4. - Config Register UART2_IPP_UART_RTS_B_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: DIOR of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[3] of instance: gpio7. 011 Select mux mode: ALT3 mux port: RTS of instance: uart2. 100 Select mux mode: ALT4 mux port: RXCAN of instance: can1. 111 Select mux mode: ALT7 mux port: DATAOUT[7] of instance: usbphy2.

43.3.163 IOMUXC_SW_MUX_CTL_PAD_PATA_RESET_B (IOMUXC_PATA_RESET_B)

Address: IOMUXC_PATA_RESET_B is 53FA_8000h base + 288h offset = 53FA_8288h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_ MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_PATA_RESET_B field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_RESET_B. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: PATA_RESET_B. NOTE: Pad PATA_RESET_B is involved in Daisy Chain.

Table continues on the next page...

IOMUXC_PATA_RESET_B field descriptions (continued)

Field	Description
	- Config Register UART1_IPP_UART_RTS_B_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: PATA_RESET_B of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[4] of instance: gpio7. 010 Select mux mode: ALT2 mux port: CMD of instance: esdhc3. 011 Select mux mode: ALT3 mux port: CTS of instance: uart1. 100 Select mux mode: ALT4 mux port: TXCAN of instance: can2. 111 Select mux mode: ALT7 mux port: DATAOUT[0] of instance: usbphy1.

43.3.164 IOMUXC_SW_MUX_CTL_PAD_PATA_IORDY (IOMUXC_PATA_IORDY)

Address: IOMUXC_PATA_IORDY is 53FA_8000h base + 28Ch offset = 53FA_828Ch

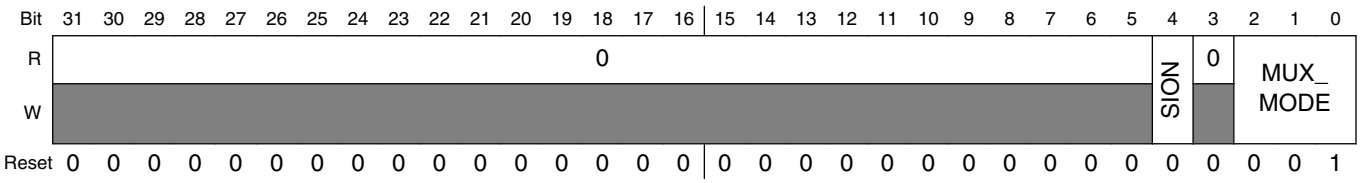
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_PATA_IORDY field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_IORDY. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: PATA_IORDY. NOTE: Pad PATA_IORDY is involved in Daisy Chain. - Config Register CAN2_IPP_IND_CANRX_SELECT_INPUT for mode ALT4. - Config Register UART1_IPP_UART_RTS_B_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: IORDY of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[5] of instance: gpio7. 010 Select mux mode: ALT2 mux port: CLK of instance: esdhc3. 011 Select mux mode: ALT3 mux port: RTS of instance: uart1. 100 Select mux mode: ALT4 mux port: RXCAN of instance: can2. 111 Select mux mode: ALT7 mux port: DATAOUT[1] of instance: usbphy1.

43.3.165 IOMUXC_SW_MUX_CTL_PAD_PATA_DA_0 (IOMUXC_PATA_DA_0)

Address: IOMUXC_PATA_DA_0 is 53FA_8000h base + 290h offset = 53FA_8290h

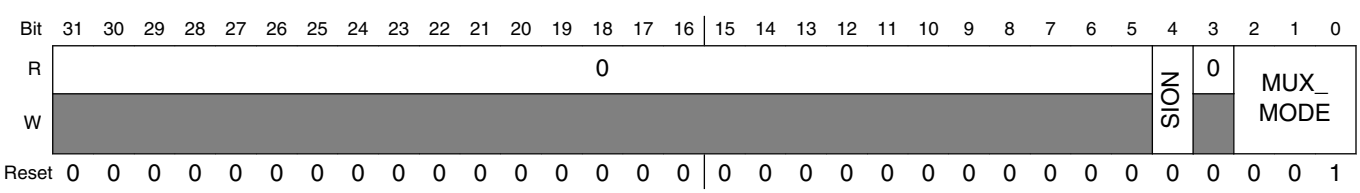


IOMUXC_PATA_DA_0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DA_0. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: PATA_DA_0. NOTE: Pad PATA_DA_0 is involved in Daisy Chain. - Config Register OWIRE_BATTERY_LINE_IN_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: DA_0 of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[6] of instance: gpio7. 010 Select mux mode: ALT2 mux port: RST of instance: esdhc3. 100 Select mux mode: ALT4 mux port: LINE of instance: owire. 111 Select mux mode: ALT7 mux port: DATAOUT[2] of instance: usbphy1.

43.3.166 IOMUXC_SW_MUX_CTL_PAD_PATA_DA_1 (IOMUXC_PATA_DA_1)

Address: IOMUXC_PATA_DA_1 is 53FA_8000h base + 294h offset = 53FA_8294h

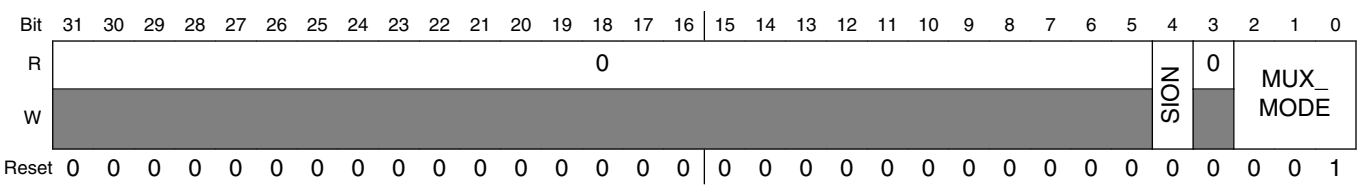


IOMUXC_PATA_DA_1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DA_1. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: PATA_DA_1. NOTE: Pad PATA_DA_1 is involved in Daisy Chain. - Config Register UART3_IPP_UART_RTS_B_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: DA_1 of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[7] of instance: gpio7. 010 Select mux mode: ALT2 mux port: CMD of instance: esdhc4. 100 Select mux mode: ALT4 mux port: CTS of instance: uart3. 111 Select mux mode: ALT7 mux port: DATAOUT[3] of instance: usbphy1.

43.3.167 IOMUXC_SW_MUX_CTL_PAD_PATA_DA_2 (IOMUXC_PATA_DA_2)

Address: IOMUXC_PATA_DA_2 is 53FA_8000h base + 298h offset = 53FA_8298h



IOMUXC_PATA_DA_2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DA_2. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: PATA_DA_2. NOTE: Pad PATA_DA_2 is involved in Daisy Chain. - Config Register UART3_IPP_UART_RTS_B_SELECT_INPUT for mode ALT4.

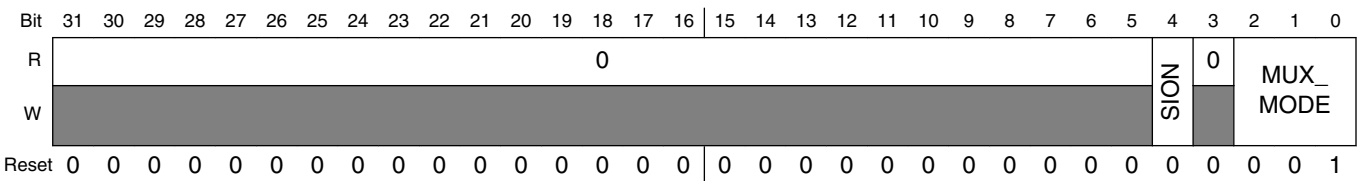
Table continues on the next page...

IOMUXC_PATA_DA_2 field descriptions (continued)

Field	Description
	000 Select mux mode: ALT0 mux port: DA_2 of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[8] of instance: gpio7. 010 Select mux mode: ALT2 mux port: CLK of instance: esdhc4. 100 Select mux mode: ALT4 mux port: RTS of instance: uart3. 111 Select mux mode: ALT7 mux port: DATAOUT[4] of instance: usbphy1.

43.3.168 IOMUXC_SW_MUX_CTL_PAD_PATA_CS_0 (IOMUXC_PATA_CS_0)

Address: IOMUXC_PATA_CS_0 is 53FA_8000h base + 29Ch offset = 53FA_829Ch

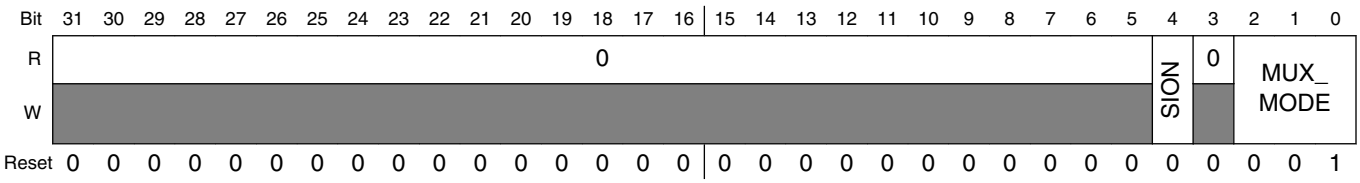


IOMUXC_PATA_CS_0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_CS_0. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: PATA_CS_0. NOTE: Pad PATA_CS_0 is involved in Daisy Chain. - Config Register UART3_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: CS_0 of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[9] of instance: gpio7. 100 Select mux mode: ALT4 mux port: TXD_MUX of instance: uart3. 111 Select mux mode: ALT7 mux port: DATAOUT[5] of instance: usbphy1.

43.3.169 IOMUXC_SW_MUX_CTL_PAD_PATA_DATA14 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA14)

Address: IOMUXC_SW_MUX_CTL_PAD_PATA_DATA14 is 53FA_8000h base + 29Ch offset = 53FA_829Ch

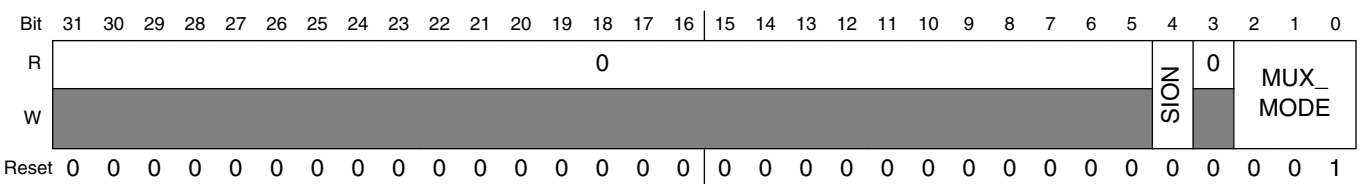


IOMUXC_SW_MUX_CTL_PAD_PATA_DATA14 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DATA14. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: PATA_DATA14. 000 Select mux mode: ALT0 mux port: PATA_DATA[14] of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[14] of instance: gpio2. 010 Select mux mode: ALT2 mux port: DAT6 of instance: esdhc2. 011 Select mux mode: ALT3 mux port: NANDF_D[14] of instance: emi. 100 Select mux mode: ALT4 mux port: DAT2 of instance: esdhc4. 101 Select mux mode: ALT5 mux port: GPU_DEBUG_OUT[14] of instance: gpu3d. 110 Select mux mode: ALT6 mux port: IPU_DIAG_BUS[14] of instance: ipu.

43.3.170 IOMUXC_SW_MUX_CTL_PAD_PATA_CS_1 (IOMUXC_SW_MUX_CTL_PAD_PATA_CS_1)

Address: IOMUXC_SW_MUX_CTL_PAD_PATA_CS_1 is 53FA_8000h base + 2A0h offset = 53FA_82A0h

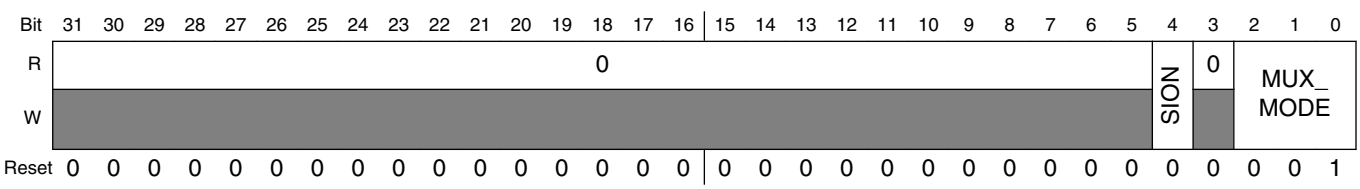


IOMUXC_SW_MUX_CTL_PAD_PATA_CS_1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_CS_1. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 4 iomux modes to be used for pad: PATA_CS_1. NOTE: Pad PATA_CS_1 is involved in Daisy Chain. - Config Register UART3_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: CS_1 of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[10] of instance: gpio7. 100 Select mux mode: ALT4 mux port: RXD_MUX of instance: uart3. 111 Select mux mode: ALT7 mux port: DATAOUT[6] of instance: usbphy1.

43.3.171 IOMUXC_SW_MUX_CTL_PAD_PATA_DATA0 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA0)

Address: IOMUXC_SW_MUX_CTL_PAD_PATA_DATA0 is 53FA_8000h base + 2A4h offset = 53FA_82A4h



IOMUXC_SW_MUX_CTL_PAD_PATA_DATA0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DATA0. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: PATA_DATA0. 000 Select mux mode: ALT0 mux port: PATA_DATA[0] of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[0] of instance: gpio2. 011 Select mux mode: ALT3 mux port: NANDF_D[0] of instance: emi.

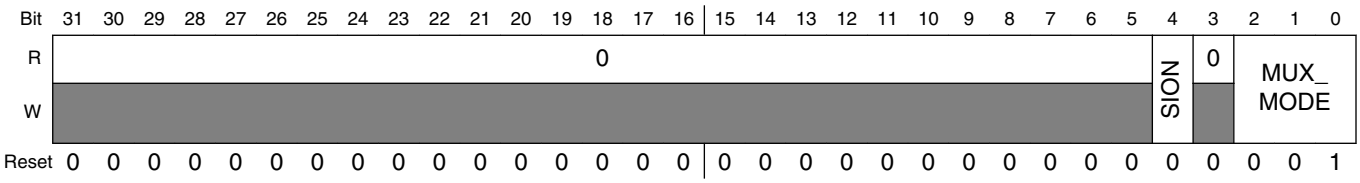
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_PATA_DATA0 field descriptions (continued)

Field	Description
	100 Select mux mode: ALT4 mux port: DAT4 of instance: esdhc3. 101 Select mux mode: ALT5 mux port: GPU_DEBUG_OUT[0] of instance: gpu3d. 110 Select mux mode: ALT6 mux port: IPU_DIAG_BUS[0] of instance: ipu. 111 Select mux mode: ALT7 mux port: DATAOUT[7] of instance: usbphy1.

43.3.172 IOMUXC_SW_MUX_CTL_PAD_PATA_DATA1 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA1)

Address: IOMUXC_SW_MUX_CTL_PAD_PATA_DATA1 is 53FA_8000h base + 2A8h offset = 53FA_82A8h

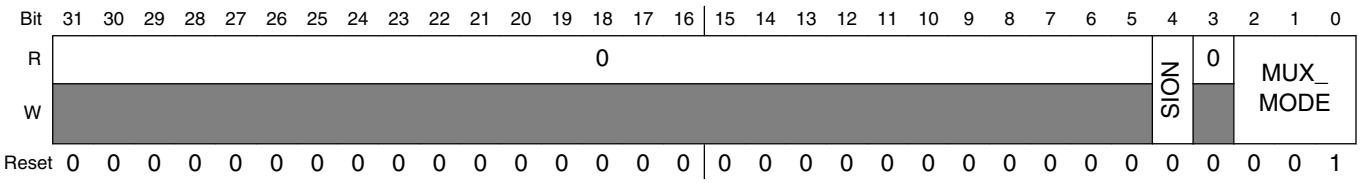


IOMUXC_SW_MUX_CTL_PAD_PATA_DATA1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DATA1. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: PATA_DATA1. 000 Select mux mode: ALT0 mux port: PATA_DATA[1] of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[1] of instance: gpio2. 011 Select mux mode: ALT3 mux port: NANDF_D[1] of instance: emi. 100 Select mux mode: ALT4 mux port: DAT5 of instance: esdhc3. 101 Select mux mode: ALT5 mux port: GPU_DEBUG_OUT[1] of instance: gpu3d. 110 Select mux mode: ALT6 mux port: IPU_DIAG_BUS[1] of instance: ipu.

43.3.173 IOMUXC_SW_MUX_CTL_PAD_PATA_DATA2 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA2)

Address: IOMUXC_SW_MUX_CTL_PAD_PATA_DATA2 is 53FA_8000h base + 2ACh offset = 53FA_82ACh

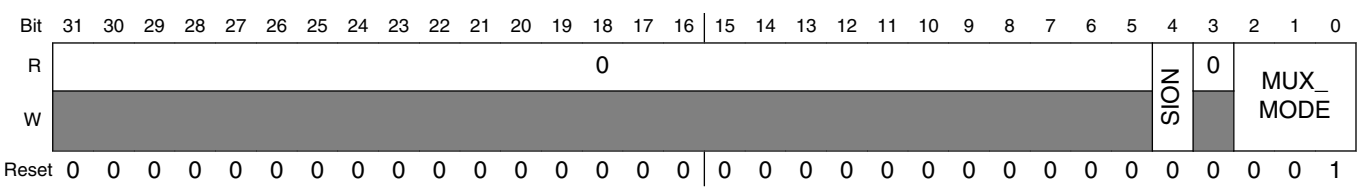


IOMUXC_SW_MUX_CTL_PAD_PATA_DATA2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DATA2. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: PATA_DATA2. 000 Select mux mode: ALT0 mux port: PATA_DATA[2] of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[2] of instance: gpio2. 011 Select mux mode: ALT3 mux port: NANDF_D[2] of instance: emi. 100 Select mux mode: ALT4 mux port: DAT6 of instance: esdhc3. 101 Select mux mode: ALT5 mux port: GPU_DEBUG_OUT[2] of instance: gpu3d. 110 Select mux mode: ALT6 mux port: IPU_DIAG_BUS[2] of instance: ipu.

43.3.174 IOMUXC_SW_MUX_CTL_PAD_PATA_DATA3 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA3)

Address: IOMUXC_SW_MUX_CTL_PAD_PATA_DATA3 is 53FA_8000h base + 2B0h offset = 53FA_82B0h

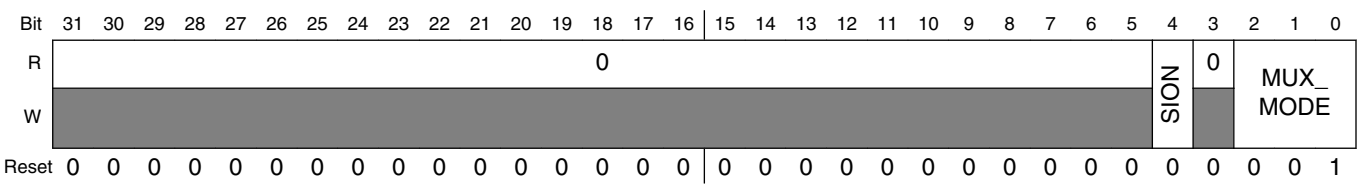


IOMUXC_SW_MUX_CTL_PAD_PATA_DATA3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DATA3. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: PATA_DATA3. 000 Select mux mode: ALT0 mux port: PATA_DATA[3] of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[3] of instance: gpio2. 011 Select mux mode: ALT3 mux port: NANDF_D[3] of instance: emi. 100 Select mux mode: ALT4 mux port: DAT7 of instance: esdhc3. 101 Select mux mode: ALT5 mux port: GPU_DEBUG_OUT[3] of instance: gpu3d. 110 Select mux mode: ALT6 mux port: IPU_DIAG_BUS[3] of instance: ipu.

43.3.175 IOMUXC_SW_MUX_CTL_PAD_PATA_DATA4 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA4)

Address: IOMUXC_SW_MUX_CTL_PAD_PATA_DATA4 is 53FA_8000h base + 2B4h offset = 53FA_82B4h



IOMUXC_SW_MUX_CTL_PAD_PATA_DATA4 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DATA4. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: PATA_DATA4. 000 Select mux mode: ALT0 mux port: PATA_DATA[4] of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[4] of instance: gpio2. 011 Select mux mode: ALT3 mux port: NANDF_D[4] of instance: emi.

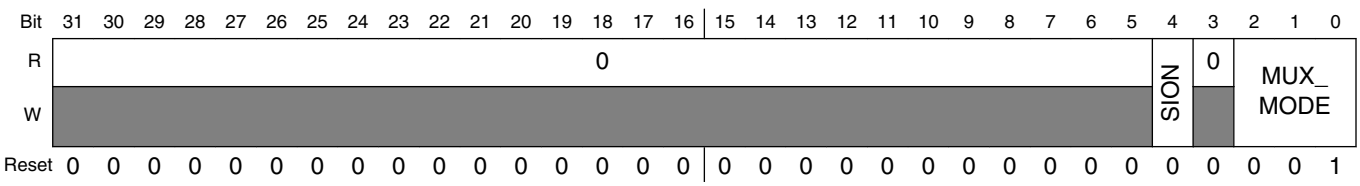
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_PATA_DATA4 field descriptions (continued)

Field	Description
	100 Select mux mode: ALT4 mux port: DAT4 of instance: esdhc4. 101 Select mux mode: ALT5 mux port: GPU_DEBUG_OUT[4] of instance: gpu3d. 110 Select mux mode: ALT6 mux port: IPU_DIAG_BUS[4] of instance: ipu.

43.3.176 IOMUXC_SW_MUX_CTL_PAD_PATA_DATA5 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA5)

Address: IOMUXC_SW_MUX_CTL_PAD_PATA_DATA5 is 53FA_8000h base + 2B8h offset = 53FA_82B8h

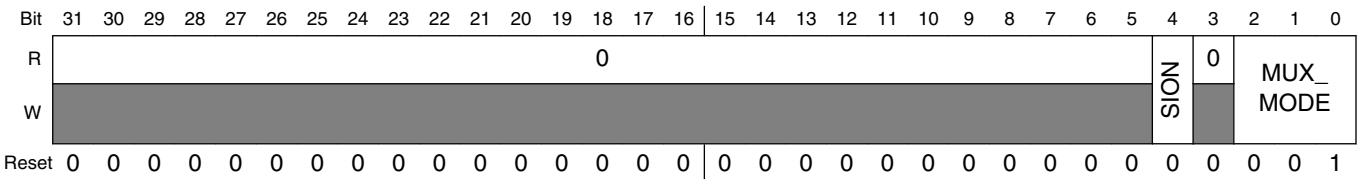


IOMUXC_SW_MUX_CTL_PAD_PATA_DATA5 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DATA5. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: PATA_DATA5. 000 Select mux mode: ALT0 mux port: PATA_DATA[5] of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[5] of instance: gpio2. 011 Select mux mode: ALT3 mux port: NANDF_D[5] of instance: emi. 100 Select mux mode: ALT4 mux port: DAT5 of instance: esdhc4. 101 Select mux mode: ALT5 mux port: GPU_DEBUG_OUT[5] of instance: gpu3d. 110 Select mux mode: ALT6 mux port: IPU_DIAG_BUS[5] of instance: ipu.

43.3.177 IOMUXC_SW_MUX_CTL_PAD_PATA_DATA6 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA6)

Address: IOMUXC_SW_MUX_CTL_PAD_PATA_DATA6 is 53FA_8000h base + 2BCh offset = 53FA_82BCh

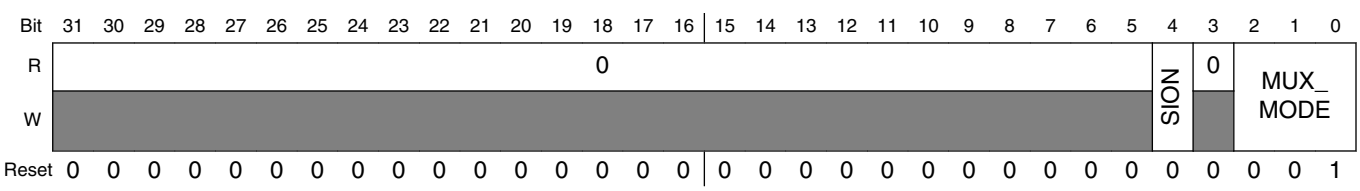


IOMUXC_SW_MUX_CTL_PAD_PATA_DATA6 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DATA6. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: PATA_DATA6. 000 Select mux mode: ALT0 mux port: PATA_DATA[6] of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[6] of instance: gpio2. 011 Select mux mode: ALT3 mux port: NANDF_D[6] of instance: emi. 100 Select mux mode: ALT4 mux port: DAT6 of instance: esdhc4. 101 Select mux mode: ALT5 mux port: GPU_DEBUG_OUT[6] of instance: gpu3d. 110 Select mux mode: ALT6 mux port: IPU_DIAG_BUS[6] of instance: ipu.

43.3.178 IOMUXC_SW_MUX_CTL_PAD_PATA_DATA7 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA7)

Address: IOMUXC_SW_MUX_CTL_PAD_PATA_DATA7 is 53FA_8000h base + 2C0h offset = 53FA_82C0h

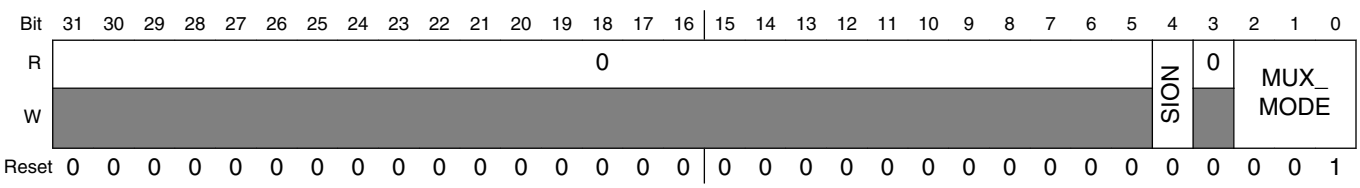


IOMUXC_SW_MUX_CTL_PAD_PATA_DATA7 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DATA7. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: PATA_DATA7. 000 Select mux mode: ALT0 mux port: PATA_DATA[7] of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[7] of instance: gpio2. 011 Select mux mode: ALT3 mux port: NANDF_D[7] of instance: emi. 100 Select mux mode: ALT4 mux port: DAT7 of instance: esdhc4. 101 Select mux mode: ALT5 mux port: GPU_DEBUG_OUT[7] of instance: gpu3d. 110 Select mux mode: ALT6 mux port: IPU_DIAG_BUS[7] of instance: ipu.

43.3.179 IOMUXC_SW_MUX_CTL_PAD_PATA_DATA8 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA8)

Address: IOMUXC_SW_MUX_CTL_PAD_PATA_DATA8 is 53FA_8000h base + 2C4h offset = 53FA_82C4h



IOMUXC_SW_MUX_CTL_PAD_PATA_DATA8 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DATA8. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: PATA_DATA8. 000 Select mux mode: ALT0 mux port: PATA_DATA[8] of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[8] of instance: gpio2. 010 Select mux mode: ALT2 mux port: DAT4 of instance: esdhc1.

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_PATA_DATA8 field descriptions (continued)

Field	Description
	011 Select mux mode: ALT3 mux port: NANDF_D[8] of instance: emi. 100 Select mux mode: ALT4 mux port: DAT0 of instance: esdhc3. 101 Select mux mode: ALT5 mux port: GPU_DEBUG_OUT[8] of instance: gpu3d. 110 Select mux mode: ALT6 mux port: IPU_DIAG_BUS[8] of instance: ipu.

43.3.180 IOMUXC_SW_MUX_CTL_PAD_PATA_DATA9 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA9)

Address: IOMUXC_SW_MUX_CTL_PAD_PATA_DATA9 is 53FA_8000h base + 2C8h offset = 53FA_82C8h

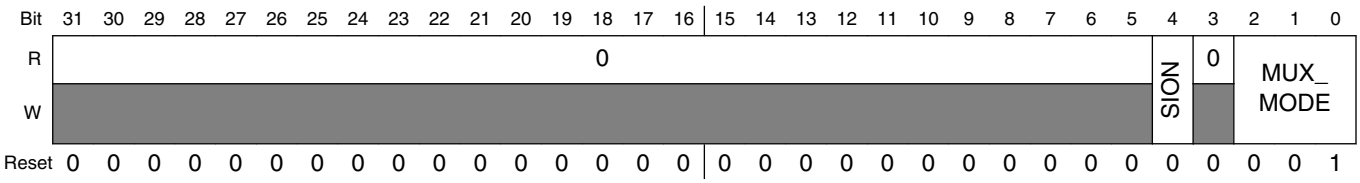
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_SW_MUX_CTL_PAD_PATA_DATA9 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DATA9. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: PATA_DATA9. 000 Select mux mode: ALT0 mux port: PATA_DATA[9] of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[9] of instance: gpio2. 010 Select mux mode: ALT2 mux port: DAT5 of instance: esdhc1. 011 Select mux mode: ALT3 mux port: NANDF_D[9] of instance: emi. 100 Select mux mode: ALT4 mux port: DAT1 of instance: esdhc3. 101 Select mux mode: ALT5 mux port: GPU_DEBUG_OUT[9] of instance: gpu3d. 110 Select mux mode: ALT6 mux port: IPU_DIAG_BUS[9] of instance: ipu.

43.3.181 IOMUXC_SW_MUX_CTL_PAD_PATA_DATA10 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA10)

Address: IOMUXC_SW_MUX_CTL_PAD_PATA_DATA10 is 53FA_8000h base + 2CCh offset = 53FA_82CCh

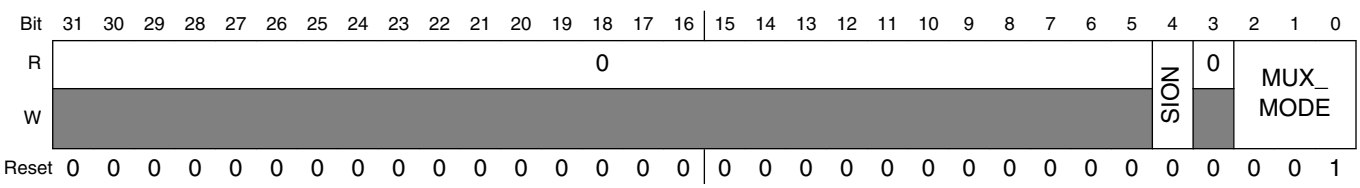


IOMUXC_SW_MUX_CTL_PAD_PATA_DATA10 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DATA10. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: PATA_DATA10. 000 Select mux mode: ALT0 mux port: PATA_DATA[10] of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[10] of instance: gpio2. 010 Select mux mode: ALT2 mux port: DAT6 of instance: esdhc1. 011 Select mux mode: ALT3 mux port: NANDF_D[10] of instance: emi. 100 Select mux mode: ALT4 mux port: DAT2 of instance: esdhc3. 101 Select mux mode: ALT5 mux port: GPU_DEBUG_OUT[10] of instance: gpu3d. 110 Select mux mode: ALT6 mux port: IPU_DIAG_BUS[10] of instance: ipu.

43.3.182 IOMUXC_SW_MUX_CTL_PAD_PATA_DATA11 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA11)

Address: IOMUXC_SW_MUX_CTL_PAD_PATA_DATA11 is 53FA_8000h base + 2D0h offset = 53FA_82D0h

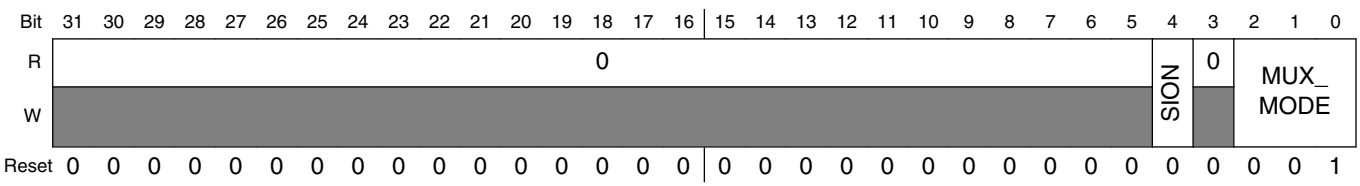


IOMUXC_SW_MUX_CTL_PAD_PATA_DATA11 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DATA11. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: PATA_DATA11. 000 Select mux mode: ALT0 mux port: PATA_DATA[11] of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[11] of instance: gpio2. 010 Select mux mode: ALT2 mux port: DAT7 of instance: esdhc1. 011 Select mux mode: ALT3 mux port: NANDF_D[11] of instance: emi. 100 Select mux mode: ALT4 mux port: DAT3 of instance: esdhc3. 101 Select mux mode: ALT5 mux port: GPU_DEBUG_OUT[11] of instance: gpu3d. 110 Select mux mode: ALT6 mux port: IPU_DIAG_BUS[11] of instance: ipu.

43.3.183 IOMUXC_SW_MUX_CTL_PAD_PATA_DATA12 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA12)

Address: IOMUXC_SW_MUX_CTL_PAD_PATA_DATA12 is 53FA_8000h base + 2D4h offset = 53FA_82D4h



IOMUXC_SW_MUX_CTL_PAD_PATA_DATA12 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DATA12. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: PATA_DATA12. 000 Select mux mode: ALT0 mux port: PATA_DATA[12] of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[12] of instance: gpio2.

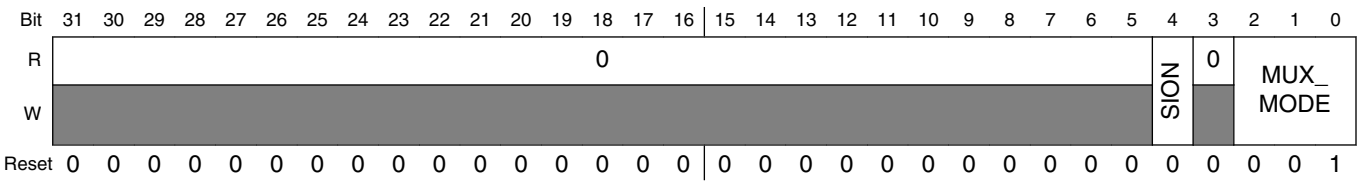
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_PATA_DATA12 field descriptions (continued)

Field	Description
	010 Select mux mode: ALT2 mux port: DAT4 of instance: esdhc2. 011 Select mux mode: ALT3 mux port: NANDF_D[12] of instance: emi. 100 Select mux mode: ALT4 mux port: DAT0 of instance: esdhc4. 101 Select mux mode: ALT5 mux port: GPU_DEBUG_OUT[12] of instance: gpu3d. 110 Select mux mode: ALT6 mux port: IPU_DIAG_BUS[12] of instance: ipu.

43.3.184 IOMUXC_SW_MUX_CTL_PAD_PATA_DATA13 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA13)

Address: IOMUXC_SW_MUX_CTL_PAD_PATA_DATA13 is 53FA_8000h base + 2D8h offset = 53FA_82D8h

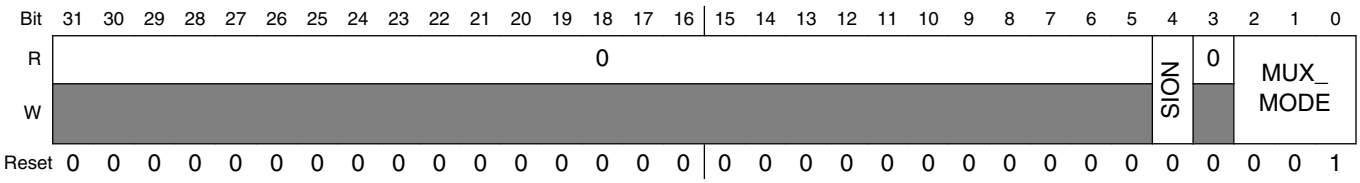


IOMUXC_SW_MUX_CTL_PAD_PATA_DATA13 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DATA13. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: PATA_DATA13. 000 Select mux mode: ALT0 mux port: PATA_DATA[13] of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[13] of instance: gpio2. 010 Select mux mode: ALT2 mux port: DAT5 of instance: esdhc2. 011 Select mux mode: ALT3 mux port: NANDF_D[13] of instance: emi. 100 Select mux mode: ALT4 mux port: DAT1 of instance: esdhc4. 101 Select mux mode: ALT5 mux port: GPU_DEBUG_OUT[13] of instance: gpu3d. 110 Select mux mode: ALT6 mux port: IPU_DIAG_BUS[13] of instance: ipu.

43.3.185 IOMUXC_SW_MUX_CTL_PAD_PATA_DATA15 (IOMUXC_SW_MUX_CTL_PAD_PATA_DATA15)

Address: IOMUXC_SW_MUX_CTL_PAD_PATA_DATA15 is 53FA_8000h base + 2E0h offset = 53FA_82E0h

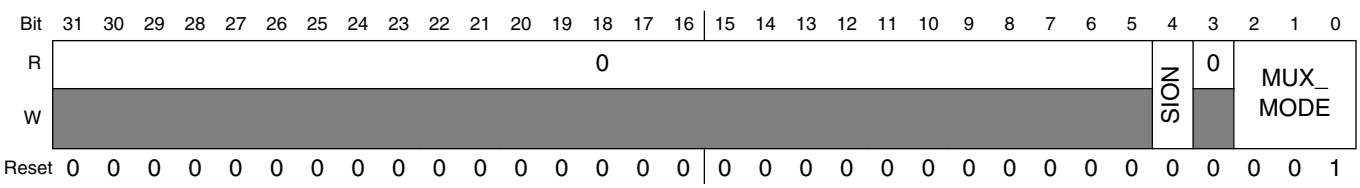


IOMUXC_SW_MUX_CTL_PAD_PATA_DATA15 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad PATA_DATA15. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: PATA_DATA15. 000 Select mux mode: ALT0 mux port: PATA_DATA[15] of instance: pata. 001 Select mux mode: ALT1 mux port: GPIO[15] of instance: gpio2. 010 Select mux mode: ALT2 mux port: DAT7 of instance: esdhc2. 011 Select mux mode: ALT3 mux port: NANDF_D[15] of instance: emi. 100 Select mux mode: ALT4 mux port: DAT3 of instance: esdhc4. 101 Select mux mode: ALT5 mux port: GPU_DEBUG_OUT[15] of instance: gpu3d. 110 Select mux mode: ALT6 mux port: IPU_DIAG_BUS[15] of instance: ipu.

43.3.186 IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0 (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0)

Address: IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0 is 53FA_8000h base + 2E4h offset = 53FA_82E4h

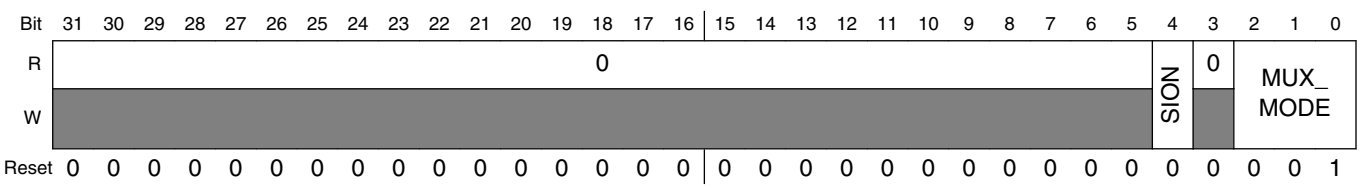


IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad SD1_DATA0. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SD1_DATA0. NOTE: Pad SD1_DATA0 is involved in Daisy Chain. - Config Register CCM_PLL3_BYPASS_CLK_SELECT_INPUT for mode ALT7. - Config Register CSPI_IPP_IND_MISO_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: DAT0 of instance: esdhc1. 001 Select mux mode: ALT1 mux port: GPIO[16] of instance: gpio1. 011 Select mux mode: ALT3 mux port: IND_CAPIN1 of instance: gpt. 101 Select mux mode: ALT5 mux port: MISO of instance: cspi. 111 Select mux mode: ALT7 mux port: PLL3_BYP of instance: ccm.

43.3.187 IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1 (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1)

Address: IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1 is 53FA_8000h base + 2E8h offset = 53FA_82E8h



IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad SD1_DATA1. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SD1_DATA1. NOTE: Pad SD1_DATA1 is involved in Daisy Chain.

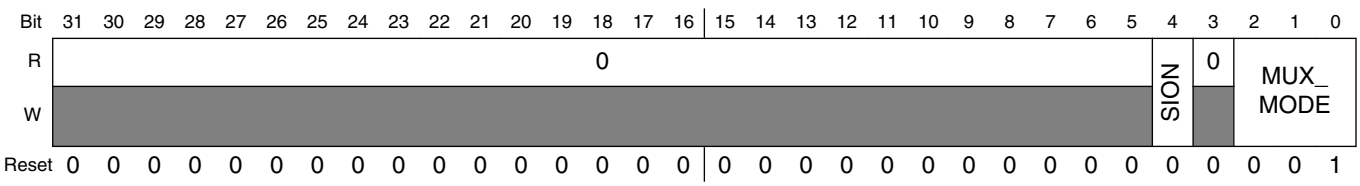
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1 field descriptions (continued)

Field	Description
	- Config Register CCM_PLL4_BYPASS_CLK_SELECT_INPUT for mode ALT7. - Config Register CSPI_IPP_IND_SS0_B_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: DAT1 of instance: esdhc1. 001 Select mux mode: ALT1 mux port: GPIO[17] of instance: gpio1. 011 Select mux mode: ALT3 mux port: IND_CAPIN2 of instance: gpt. 101 Select mux mode: ALT5 mux port: SS0 of instance: cspi. 111 Select mux mode: ALT7 mux port: PLL4_BYP of instance: ccm.

43.3.188 IOMUXC_SW_MUX_CTL_PAD_SD1_CMD (IOMUXC_SW_MUX_CTL_PAD_SD1_CMD)

Address: IOMUXC_SW_MUX_CTL_PAD_SD1_CMD is 53FA_8000h base + 2ECh offset = 53FA_82ECh

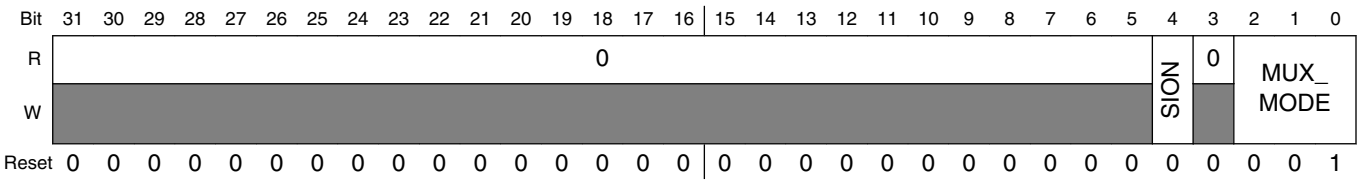


IOMUXC_SW_MUX_CTL_PAD_SD1_CMD field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad SD1_CMD. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 5 iomux modes to be used for pad: SD1_CMD. NOTE: Pad SD1_CMD is involved in Daisy Chain. - Config Register CCM_PLL1_BYPASS_CLK_SELECT_INPUT for mode ALT7. - Config Register CSPI_IPP_IND_MOSI_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: CMD of instance: esdhc1. 001 Select mux mode: ALT1 mux port: GPIO[18] of instance: gpio1. 011 Select mux mode: ALT3 mux port: DO_CMPOUT1 of instance: gpt. 101 Select mux mode: ALT5 mux port: MOSI of instance: cspi. 111 Select mux mode: ALT7 mux port: PLL1_BYP of instance: ccm.

43.3.189 IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2 (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2)

Address: IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2 is 53FA_8000h base + 2F0h offset = 53FA_82F0h

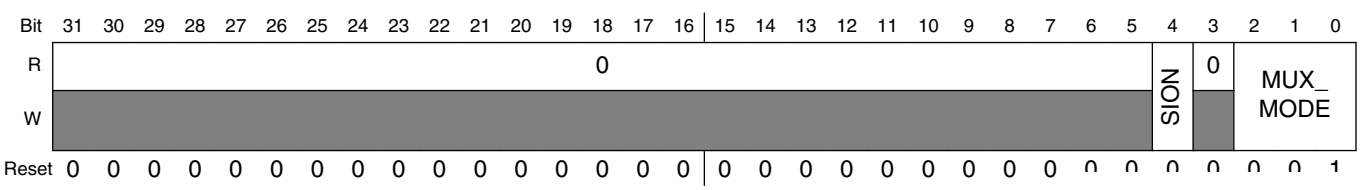


IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad SD1_DATA2. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: SD1_DATA2. NOTE: Pad SD1_DATA2 is involved in Daisy Chain. - Config Register CCM_PLL2_BYPASS_CLK_SELECT_INPUT for mode ALT7. - Config Register CSPI_IPP_IND_SS1_B_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: DAT2 of instance: esdhc1. 001 Select mux mode: ALT1 mux port: GPIO[19] of instance: gpio1. 010 Select mux mode: ALT2 mux port: DO_CMPOUT2 of instance: gpt. 011 Select mux mode: ALT3 mux port: PWMO of instance: pwm2. 100 Select mux mode: ALT4 mux port: WDOG_B of instance: wdog1. 101 Select mux mode: ALT5 mux port: SS1 of instance: cspi. 110 Select mux mode: ALT6 mux port: WDOG_RST_B_DEB of instance: wdog1. 111 Select mux mode: ALT7 mux port: PLL2_BYP of instance: ccm.

43.3.190 IOMUXC_SW_MUX_CTL_PAD_SD1_CLK (IOMUXC_SW_MUX_CTL_PAD_SD1_CLK)

Address: IOMUXC_SW_MUX_CTL_PAD_SD1_CLK is 53FA_8000h base + 2F4h offset = 53FA_82F4h

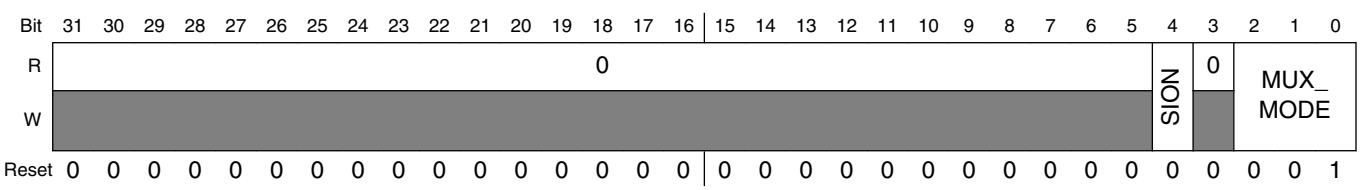


IOMUXC_SW_MUX_CTL_PAD_SD1_CLK field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad SD1_CLK. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD1_CLK. NOTE: Pad SD1_CLK is involved in Daisy Chain. - Config Register CSPI_IPP_CSPI_CLK_IN_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: CLK of instance: esdhc1. 001 Select mux mode: ALT1 mux port: GPIO[20] of instance: gpio1. 010 Select mux mode: ALT2 mux port: 32K_OUT of instance: osc32k. 011 Select mux mode: ALT3 mux port: IND_CLKIN of instance: gpt. 101 Select mux mode: ALT5 mux port: SCLK of instance: cspi. 111 Select mux mode: ALT7 mux port: DTB[0] of instance: sata_phy.

43.3.191 IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3 (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3)

Address: IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3 is 53FA_8000h base + 2F8h offset = 53FA_82F8h



IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad SD1_DATA3. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: SD1_DATA3. NOTE: Pad SD1_DATA3 is involved in Daisy Chain.

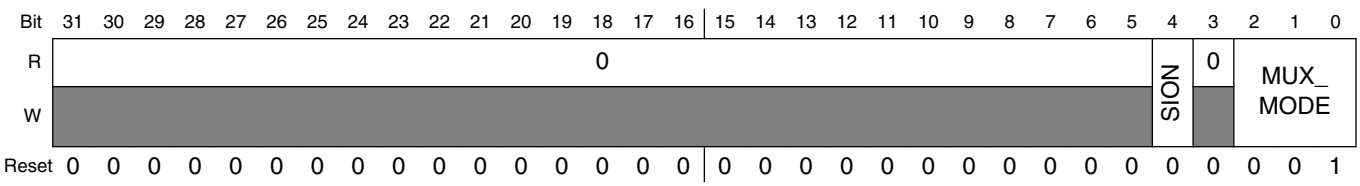
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3 field descriptions (continued)

Field	Description
	- Config Register CSPI_IPP_IND_SS2_B_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: DAT3 of instance: esdhc1. 001 Select mux mode: ALT1 mux port: GPIO[21] of instance: gpio1. 010 Select mux mode: ALT2 mux port: DO_CMPOUT3 of instance: gpt. 011 Select mux mode: ALT3 mux port: PWMO of instance: pwm1. 100 Select mux mode: ALT4 mux port: WDOG_B of instance: wdog2. 101 Select mux mode: ALT5 mux port: SS2 of instance: cspi. 110 Select mux mode: ALT6 mux port: WDOG_RST_B_DEB of instance: wdog2. 111 Select mux mode: ALT7 mux port: DTB[1] of instance: sata_phy.

43.3.192 IOMUXC_SW_MUX_CTL_PAD_SD2_CLK (IOMUXC_SW_MUX_CTL_PAD_SD2_CLK)

Address: IOMUXC_SW_MUX_CTL_PAD_SD2_CLK is 53FA_8000h base + 2FCh offset = 53FA_82FCh



IOMUXC_SW_MUX_CTL_PAD_SD2_CLK field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad SD2_CLK. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD2_CLK. NOTE: Pad SD2_CLK is involved in Daisy Chain. - Config Register AUDMUX_P4_INPUT_RXFS_AMX_SELECT_INPUT for mode ALT3. - Config Register CSPI_IPP_CSPI_CLK_IN_SELECT_INPUT for mode ALT5. - Config Register KPP_IPP_IND_COL_5_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CLK of instance: esdhc2. 001 Select mux mode: ALT1 mux port: GPIO[10] of instance: gpio1. 010 Select mux mode: ALT2 mux port: COL[5] of instance: kpp. 011 Select mux mode: ALT3 mux port: AUD4_RXFS of instance: audmux. 101 Select mux mode: ALT5 mux port: SCLK of instance: cspi. 111 Select mux mode: ALT7 mux port: RANDOM_V of instance: scc.

43.3.193 IOMUXC_SW_MUX_CTL_PAD_SD2_CMD (IOMUXC_SW_MUX_CTL_PAD_SD2_CMD)

Address: IOMUXC_SW_MUX_CTL_PAD_SD2_CMD is 53FA_8000h base + 300h offset = 53FA_8300h

Bit	31–5	4	3	2–0
R	0	SION	0	MUX_MODE
W				
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	0	0 0 1

IOMUXC_SW_MUX_CTL_PAD_SD2_CMD field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad SD2_CMD. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD2_CMD. NOTE: Pad SD2_CMD is involved in Daisy Chain. - Config Register AUDMUX_P4_INPUT_RXCLK_AMX_SELECT_INPUT for mode ALT3. - Config Register CSPI_IPP_IND_MOSI_SELECT_INPUT for mode ALT5. - Config Register KPP_IPP_IND_ROW_5_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CMD of instance: esdhc2. 001 Select mux mode: ALT1 mux port: GPIO[11] of instance: gpio1. 010 Select mux mode: ALT2 mux port: ROW[5] of instance: kpp. 011 Select mux mode: ALT3 mux port: AUD4_RXC of instance: audmux. 101 Select mux mode: ALT5 mux port: MOSI of instance: cspi. 111 Select mux mode: ALT7 mux port: RANDOM of instance: scc.

43.3.194 IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3 (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3)

Address: IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3 is 53FA_8000h base + 304h offset = 53FA_8304h

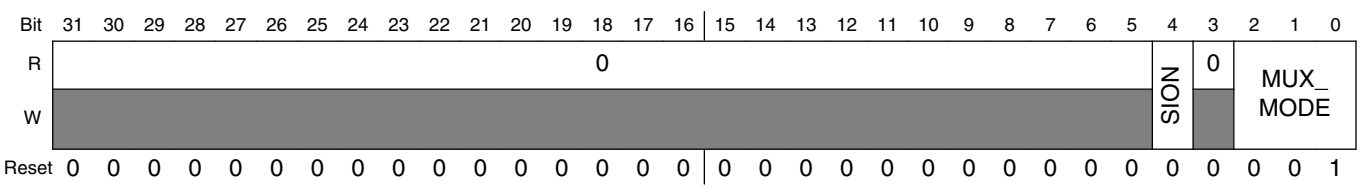
Bit	31–5	4	3	2–0
R	0	SION	0	MUX_MODE
W				
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	0	0 0 1

IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad SD2_DATA3. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD2_DATA3. NOTE: Pad SD2_DATA3 is involved in Daisy Chain. - Config Register AUDMUX_P4_INPUT_TXCLK_AMX_SELECT_INPUT for mode ALT3. - Config Register CSPI_IPP_IND_SS2_B_SELECT_INPUT for mode ALT5. - Config Register KPP_IPP_IND_COL_6_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: DAT3 of instance: esdhc2. 001 Select mux mode: ALT1 mux port: GPIO[12] of instance: gpio1. 010 Select mux mode: ALT2 mux port: COL[6] of instance: kpp. 011 Select mux mode: ALT3 mux port: AUD4_TXC of instance: audmux. 101 Select mux mode: ALT5 mux port: SS2 of instance: cspi. 111 Select mux mode: ALT7 mux port: DONE of instance: sjc.

43.3.195 IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2 (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2)

Address: IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2 is 53FA_8000h base + 308h offset = 53FA_8308h



IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad SD2_DATA2. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved

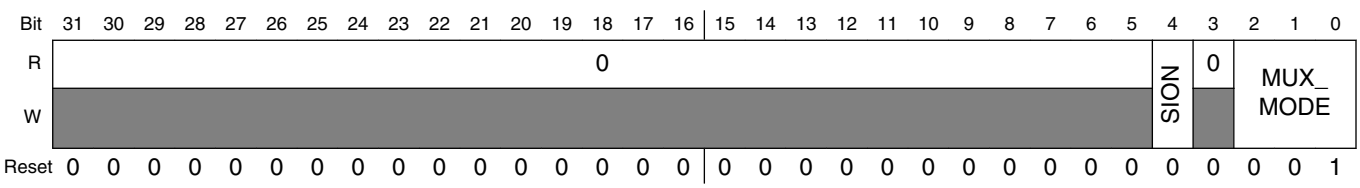
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2 field descriptions (continued)

Field	Description
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD2_DATA2. NOTE: Pad SD2_DATA2 is involved in Daisy Chain. - Config Register AUDMUX_P4_INPUT_DB_AMX_SELECT_INPUT for mode ALT3. - Config Register CSPI_IPP_IND_SS1_B_SELECT_INPUT for mode ALT5. - Config Register KPP_IPP_IND_ROW_6_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: DAT2 of instance: esdhc2. 001 Select mux mode: ALT1 mux port: GPIO[13] of instance: gpio1. 010 Select mux mode: ALT2 mux port: ROW[6] of instance: kpp. 011 Select mux mode: ALT3 mux port: AUD4_TXD of instance: audmux. 101 Select mux mode: ALT5 mux port: SS1 of instance: cspi. 111 Select mux mode: ALT7 mux port: FAIL of instance: sjc.

43.3.196 IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1 (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1)

Address: IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1 is 53FA_8000h base + 30Ch offset = 53FA_830Ch



IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad SD2_DATA1. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD2_DATA1. NOTE: Pad SD2_DATA1 is involved in Daisy Chain. - Config Register AUDMUX_P4_INPUT_TXFS_AMX_SELECT_INPUT for mode ALT3. - Config Register CSPI_IPP_IND_SS0_B_SELECT_INPUT for mode ALT5. - Config Register KPP_IPP_IND_COL_7_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: DAT1 of instance: esdhc2. 001 Select mux mode: ALT1 mux port: GPIO[14] of instance: gpio1.

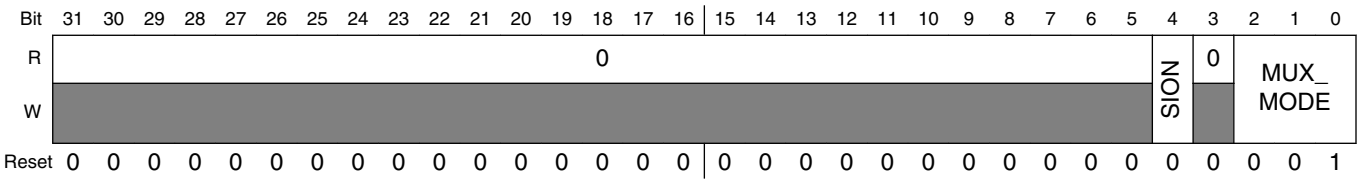
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1 field descriptions (continued)

Field	Description
	010 Select mux mode: ALT2 mux port: COL[7] of instance: kpp. 011 Select mux mode: ALT3 mux port: AUD4_TXFS of instance: audmux. 101 Select mux mode: ALT5 mux port: SS0 of instance: cspi. 111 Select mux mode: ALT7 mux port: RTIC_SEC_VIO of instance: rtic.

43.3.197 IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0 (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0)

Address: IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0 is 53FA_8000h base + 310h offset = 53FA_8310h



IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad SD2_DATA0. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 6 iomux modes to be used for pad: SD2_DATA0. NOTE: Pad SD2_DATA0 is involved in Daisy Chain. - Config Register AUDMUX_P4_INPUT_DA_AMX_SELECT_INPUT for mode ALT3. - Config Register CSPI_IPP_IND_MISO_SELECT_INPUT for mode ALT5. - Config Register KPP_IPP_IND_ROW_7_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: DAT0 of instance: esdhc2. 001 Select mux mode: ALT1 mux port: GPIO[15] of instance: gpio1. 010 Select mux mode: ALT2 mux port: ROW[7] of instance: kpp. 011 Select mux mode: ALT3 mux port: AUD4_RXD of instance: audmux. 101 Select mux mode: ALT5 mux port: MISO of instance: cspi. 111 Select mux mode: ALT7 mux port: RTIC_DONE_INT of instance: rtic.

43.3.198 IOMUXC_SW_MUX_CTL_PAD_GPIO_0 (IOMUXC_SW_MUX_CTL_PAD_GPIO_0)

Address: IOMUXC_SW_MUX_CTL_PAD_GPIO_0 is 53FA_8000h base + 314h offset = 53FA_8314h

Bit	31–5	4	3	2–0
R	0	SION	0	MUX_MODE
W				
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	0	0 0 1

IOMUXC_SW_MUX_CTL_PAD_GPIO_0 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_0. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_0. NOTE: Pad GPIO_0 is involved in Daisy Chain. - Config Register KPP_IPP_IND_COL_5_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: CLKO of instance: ccm. 001 Select mux mode: ALT1 mux port: GPIO[0] of instance: gpio1. 010 Select mux mode: ALT2 mux port: COL[5] of instance: kpp. 011 Select mux mode: ALT3 mux port: SSI_EXT1_CLK of instance: ccm. 100 Select mux mode: ALT4 mux port: EPITO of instance: epit1. 101 Select mux mode: ALT5 mux port: SRTC_ALARM_DEB of instance: srtc. 110 Select mux mode: ALT6 mux port: USBH1_PWR of instance: usboh3. 111 Select mux mode: ALT7 mux port: TD of instance: csu.

43.3.199 IOMUXC_SW_MUX_CTL_PAD_GPIO_1 (IOMUXC_SW_MUX_CTL_PAD_GPIO_1)

Address: IOMUXC_SW_MUX_CTL_PAD_GPIO_1 is 53FA_8000h base + 318h offset = 53FA_8318h

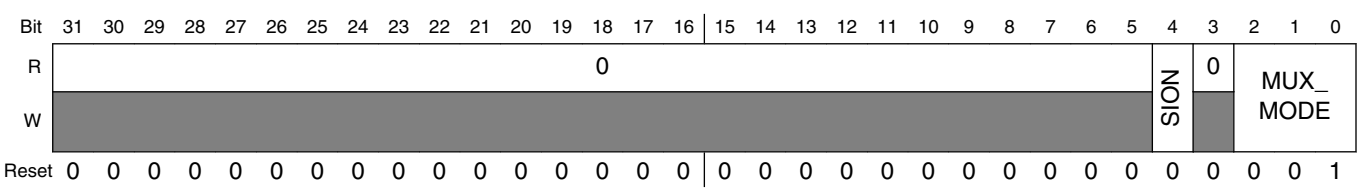
Bit	31–5	4	3	2–0
R	0	SION	0	MUX_MODE
W				
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	0	0 0 1

IOMUXC_SW_MUX_CTL_PAD_GPIO_1 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_1. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_1. NOTE: Pad GPIO_1 is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_SCKR_SELECT_INPUT for mode ALT0. - Config Register KPP_IPP_IND_ROW_5_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: SCKR of instance: esai1. 001 Select mux mode: ALT1 mux port: GPIO[1] of instance: gpio1. 010 Select mux mode: ALT2 mux port: ROW[5] of instance: kpp. 011 Select mux mode: ALT3 mux port: SSI_EXT2_CLK of instance: ccm. 100 Select mux mode: ALT4 mux port: PWMO of instance: pwm2. 101 Select mux mode: ALT5 mux port: WDOG_B of instance: wdog2. 110 Select mux mode: ALT6 mux port: CD of instance: esdhc1. 111 Select mux mode: ALT7 mux port: TESTER_ACK of instance: src.

43.3.200 IOMUXC_SW_MUX_CTL_PAD_GPIO_9 (IOMUXC_SW_MUX_CTL_PAD_GPIO_9)

Address: IOMUXC_SW_MUX_CTL_PAD_GPIO_9 is 53FA_8000h base + 31Ch offset = 53FA_831Ch



IOMUXC_SW_MUX_CTL_PAD_GPIO_9 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_9. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved

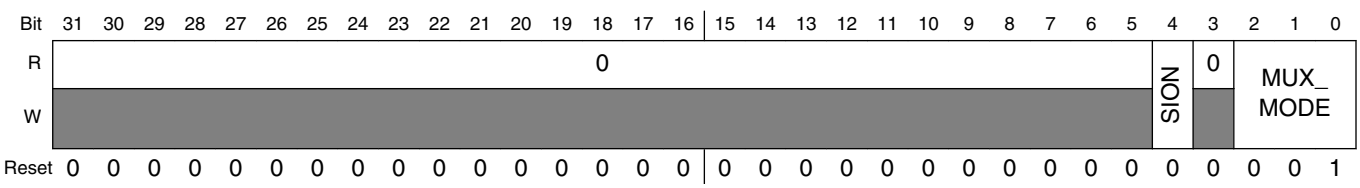
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_GPIO_9 field descriptions (continued)

Field	Description
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_9. NOTE: Pad GPIO_9 is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_FSR_SELECT_INPUT for mode ALT0. - Config Register ESDHC1_IPP_WP_ON_SELECT_INPUT for mode ALT6. - Config Register KPP_IPP_IND_COL_6_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: FSR of instance: esai1. 001 Select mux mode: ALT1 mux port: GPIO[9] of instance: gpio1. 010 Select mux mode: ALT2 mux port: COL[6] of instance: kpp. 011 Select mux mode: ALT3 mux port: REF_EN_B of instance: ccm. 100 Select mux mode: ALT4 mux port: PWMO of instance: pwm1. 101 Select mux mode: ALT5 mux port: WDOG_B of instance: wdog1. 110 Select mux mode: ALT6 mux port: WP of instance: esdhc1. 111 Select mux mode: ALT7 mux port: FAIL_STATE of instance: scc.

43.3.201 IOMUXC_SW_MUX_CTL_PAD_GPIO_3 (IOMUXC_SW_MUX_CTL_PAD_GPIO_3)

Address: IOMUXC_SW_MUX_CTL_PAD_GPIO_3 is 53FA_8000h base + 320h offset = 53FA_8320h



IOMUXC_SW_MUX_CTL_PAD_GPIO_3 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_3. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_3. NOTE: Pad GPIO_3 is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_HCKR_SELECT_INPUT for mode ALT0. - Config Register I2C3_IPP_SCL_IN_SELECT_INPUT for mode ALT2. - Config Register MLB_MLBCLK_IN_SELECT_INPUT for mode ALT7.

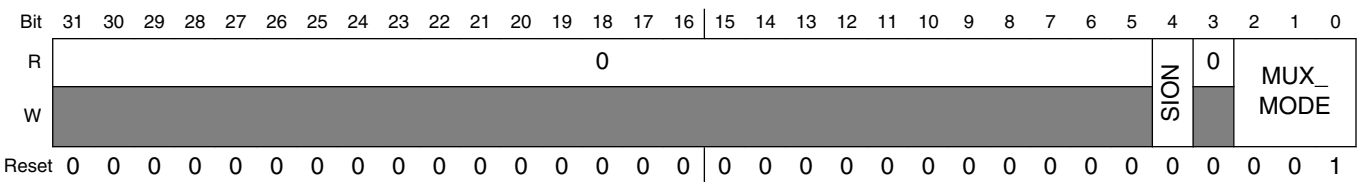
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_GPIO_3 field descriptions (continued)

Field	Description
	- Config Register USBOH3_IPP_IND_UH1_OC_SELECT_INPUT for mode ALT6. 000 Select mux mode: ALT0 mux port: HCKR of instance: esai1. 001 Select mux mode: ALT1 mux port: GPIO[3] of instance: gpio1. 010 Select mux mode: ALT2 mux port: SCL of instance: i2c3. 011 Select mux mode: ALT3 mux port: TOG_EN of instance: dpllip1. 100 Select mux mode: ALT4 mux port: CLKO2 of instance: ccm. 101 Select mux mode: ALT5 mux port: OBSRV_INT_OUT0 of instance: observe_mux. 110 Select mux mode: ALT6 mux port: USBH1_OC of instance: usboh3. 111 Select mux mode: ALT7 mux port: MLBCLK of instance: mlb.

43.3.202 IOMUXC_SW_MUX_CTL_PAD_GPIO_6 (IOMUXC_SW_MUX_CTL_PAD_GPIO_6)

Address: IOMUXC_SW_MUX_CTL_PAD_GPIO_6 is 53FA_8000h base + 324h offset = 53FA_8324h



IOMUXC_SW_MUX_CTL_PAD_GPIO_6 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_6. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_6. NOTE: Pad GPIO_6 is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_SCKT_SELECT_INPUT for mode ALT0. - Config Register I2C3_IPP_SDA_IN_SELECT_INPUT for mode ALT2. - Config Register MLB_MLBSIG_IN_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: SCKT of instance: esai1. 001 Select mux mode: ALT1 mux port: GPIO[6] of instance: gpio1. 010 Select mux mode: ALT2 mux port: SDA of instance: i2c3. 011 Select mux mode: ALT3 mux port: CCM_OUT_0 of instance: ccm. 100 Select mux mode: ALT4 mux port: CSU_INT_DEB of instance: csu.

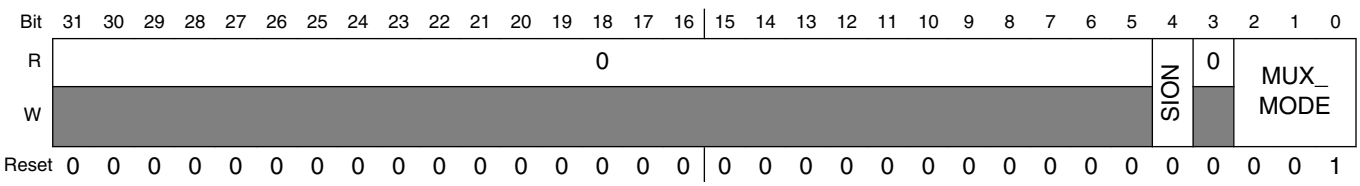
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_GPIO_6 field descriptions (continued)

Field	Description
	101 Select mux mode: ALT5 mux port: OBSRV_INT_OUT1 of instance: observe_mux. 110 Select mux mode: ALT6 mux port: LCTL of instance: esdhc2. 111 Select mux mode: ALT7 mux port: MLBSIG of instance: mlb.

43.3.203 IOMUXC_SW_MUX_CTL_PAD_GPIO_2 (IOMUXC_SW_MUX_CTL_PAD_GPIO_2)

Address: IOMUXC_SW_MUX_CTL_PAD_GPIO_2 is 53FA_8000h base + 328h offset = 53FA_8328h



IOMUXC_SW_MUX_CTL_PAD_GPIO_2 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_2. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_2. NOTE: Pad GPIO_2 is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_FST_SELECT_INPUT for mode ALT0. - Config Register KPP_IPP_IND_ROW_6_SELECT_INPUT for mode ALT2. - Config Register MLB_MLBDAT_IN_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: FST of instance: esai1. 001 Select mux mode: ALT1 mux port: GPIO[2] of instance: gpio1. 010 Select mux mode: ALT2 mux port: ROW[6] of instance: kpp. 011 Select mux mode: ALT3 mux port: CCM_OUT_1 of instance: ccm. 100 Select mux mode: ALT4 mux port: CSU_ALARM_AUT[0] of instance: csu. 101 Select mux mode: ALT5 mux port: OBSRV_INT_OUT2 of instance: observe_mux. 110 Select mux mode: ALT6 mux port: WP of instance: esdhc2. 111 Select mux mode: ALT7 mux port: MLBDAT of instance: mlb.

43.3.204 IOMUXC_SW_MUX_CTL_PAD_GPIO_4 (IOMUXC_SW_MUX_CTL_PAD_GPIO_4)

Address: IOMUXC_SW_MUX_CTL_PAD_GPIO_4 is 53FA_8000h base + 32Ch offset = 53FA_832Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_SW_MUX_CTL_PAD_GPIO_4 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_4. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_4. NOTE: Pad GPIO_4 is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_HCKT_SELECT_INPUT for mode ALT0. - Config Register KPP_IPP_IND_COL_7_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: HCKT of instance: esai1. 001 Select mux mode: ALT1 mux port: GPIO[4] of instance: gpio1. 010 Select mux mode: ALT2 mux port: COL[7] of instance: kpp. 011 Select mux mode: ALT3 mux port: CCM_OUT_2 of instance: ccm. 100 Select mux mode: ALT4 mux port: CSU_ALARM_AUT[1] of instance: csu. 101 Select mux mode: ALT5 mux port: OBSRV_INT_OUT3 of instance: observe_mux. 110 Select mux mode: ALT6 mux port: CD of instance: esdhc2. 111 Select mux mode: ALT7 mux port: SEC_STATE of instance: scc.

43.3.205 IOMUXC_SW_MUX_CTL_PAD_GPIO_5 (IOMUXC_SW_MUX_CTL_PAD_GPIO_5)

Address: IOMUXC_SW_MUX_CTL_PAD_GPIO_5 is 53FA_8000h base + 330h offset = 53FA_8330h

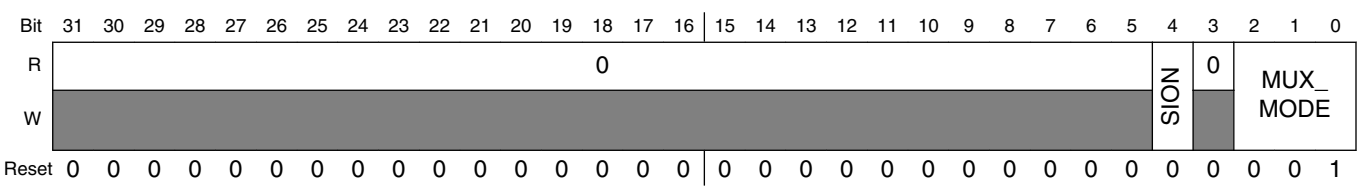
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											SION	0	MUX_MODE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

IOMUXC_SW_MUX_CTL_PAD_GPIO_5 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_5. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_5. NOTE: Pad GPIO_5 is involved in Daisy Chain. - Config Register CCM_PLL1_BYPASS_CLK_SELECT_INPUT for mode ALT7. - Config Register ESAI1_IPP_IND_SDO2_SDI3_SELECT_INPUT for mode ALT0. - Config Register I2C3_IPP_SCL_IN_SELECT_INPUT for mode ALT6. - Config Register KPP_IPP_IND_ROW_7_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: TX2_RX3 of instance: esai1. 001 Select mux mode: ALT1 mux port: GPIO[5] of instance: gpio1. 010 Select mux mode: ALT2 mux port: ROW[7] of instance: kpp. 011 Select mux mode: ALT3 mux port: CLKO of instance: ccm. 100 Select mux mode: ALT4 mux port: CSU_ALARM_AUT[2] of instance: csu. 101 Select mux mode: ALT5 mux port: OBSRV_INT_OUT4 of instance: observe_mux. 110 Select mux mode: ALT6 mux port: SCL of instance: i2c3. 111 Select mux mode: ALT7 mux port: PLL1_BYP of instance: ccm.

43.3.206 IOMUXC_SW_MUX_CTL_PAD_GPIO_7 (IOMUXC_SW_MUX_CTL_PAD_GPIO_7)

Address: IOMUXC_SW_MUX_CTL_PAD_GPIO_7 is 53FA_8000h base + 334h offset = 53FA_8334h



IOMUXC_SW_MUX_CTL_PAD_GPIO_7 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality.

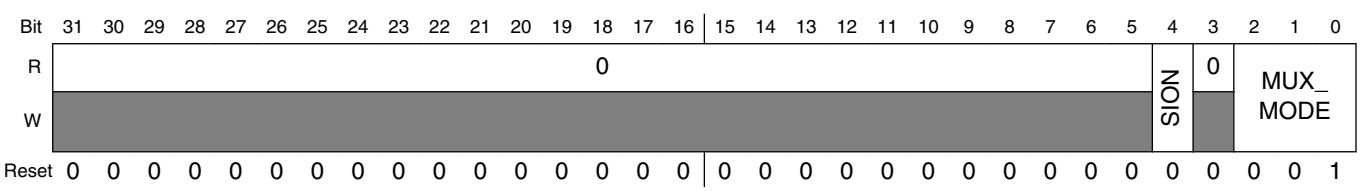
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_GPIO_7 field descriptions (continued)

Field	Description
	1 Force input path of pad GPIO_7. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_7. NOTE: Pad GPIO_7 is involved in Daisy Chain. - Config Register CCM_PLL2_BYPASS_CLK_SELECT_INPUT for mode ALT7. - Config Register ESAI1_IPP_IND_SDO4_SDI1_SELECT_INPUT for mode ALT0. - Config Register FIRI_IPP_IND_RXD_SELECT_INPUT for mode ALT5. - Config Register UART2_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: TX4_RX1 of instance: esai1. 001 Select mux mode: ALT1 mux port: GPIO[7] of instance: gpio1. 010 Select mux mode: ALT2 mux port: EPITO of instance: epit1. 011 Select mux mode: ALT3 mux port: TXCAN of instance: can1. 100 Select mux mode: ALT4 mux port: TXD_MUX of instance: uart2. 101 Select mux mode: ALT5 mux port: RXD of instance: firi. 110 Select mux mode: ALT6 mux port: PLOCK of instance: spdif. 111 Select mux mode: ALT7 mux port: PLL2_BYP of instance: ccm.

43.3.207 IOMUXC_SW_MUX_CTL_PAD_GPIO_8 (IOMUXC_SW_MUX_CTL_PAD_GPIO_8)

Address: IOMUXC_SW_MUX_CTL_PAD_GPIO_8 is 53FA_8000h base + 338h offset = 53FA_8338h



IOMUXC_SW_MUX_CTL_PAD_GPIO_8 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_8. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved

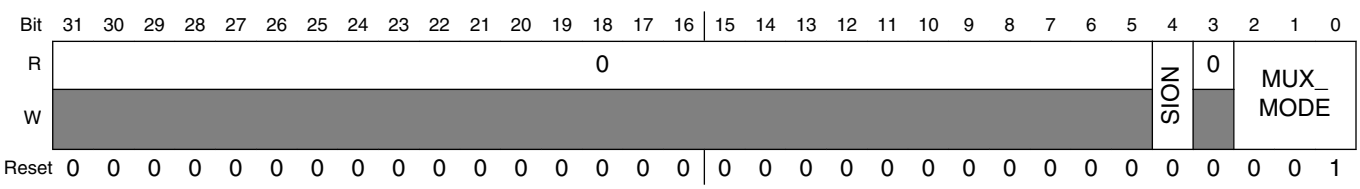
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_GPIO_8 field descriptions (continued)

Field	Description
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_8. NOTE: Pad GPIO_8 is involved in Daisy Chain. - Config Register CAN1_IPP_IND_CANRX_SELECT_INPUT for mode ALT3. - Config Register CCM_PLL3_BYPASS_CLK_SELECT_INPUT for mode ALT7. - Config Register ESAI1_IPP_IND_SDO5_SDI0_SELECT_INPUT for mode ALT0. - Config Register UART2_IPP_UART_RXD_MUX_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: TX5_RX0 of instance: esai1. 001 Select mux mode: ALT1 mux port: GPIO[8] of instance: gpio1. 010 Select mux mode: ALT2 mux port: EPITO of instance: epit2. 011 Select mux mode: ALT3 mux port: RXCAN of instance: can1. 100 Select mux mode: ALT4 mux port: RXD_MUX of instance: uart2. 101 Select mux mode: ALT5 mux port: TXD of instance: firi. 110 Select mux mode: ALT6 mux port: SRCLK of instance: spdif. 111 Select mux mode: ALT7 mux port: PLL3_BYP of instance: ccm.

43.3.208 IOMUXC_SW_MUX_CTL_PAD_GPIO_16 (IOMUXC_SW_MUX_CTL_PAD_GPIO_16)

Address: IOMUXC_SW_MUX_CTL_PAD_GPIO_16 is 53FA_8000h base + 33Ch offset = 53FA_833Ch



IOMUXC_SW_MUX_CTL_PAD_GPIO_16 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_16. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_16. NOTE: Pad GPIO_16 is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_SDO3_SDI2_SELECT_INPUT for mode ALT0. - Config Register I2C3_IPP_SDA_IN_SELECT_INPUT for mode ALT6.

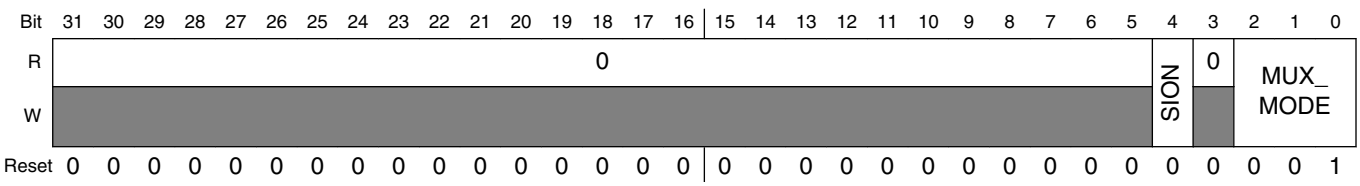
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_GPIO_16 field descriptions (continued)

Field	Description
	- Config Register SPDIF_SPDIF_IN1_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: TX3_RX2 of instance: esai1. 001 Select mux mode: ALT1 mux port: GPIO[11] of instance: gpio7. 010 Select mux mode: ALT2 mux port: PWRFAIL_INT of instance: tzic. 011 Select mux mode: ALT3 mux port: PMU_IRQ_B of instance: p_platform_ne_32k_256k. 100 Select mux mode: ALT4 mux port: CE_RTC_EXT_TRIG1 of instance: rtc. 101 Select mux mode: ALT5 mux port: IN1 of instance: spdif. 110 Select mux mode: ALT6 mux port: SDA of instance: i2c3. 111 Select mux mode: ALT7 mux port: DE_B of instance: sjc.

43.3.209 IOMUXC_SW_MUX_CTL_PAD_GPIO_17 (IOMUXC_SW_MUX_CTL_PAD_GPIO_17)

Address: IOMUXC_SW_MUX_CTL_PAD_GPIO_17 is 53FA_8000h base + 340h offset = 53FA_8340h



IOMUXC_SW_MUX_CTL_PAD_GPIO_17 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_17. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_17. NOTE: Pad GPIO_17 is involved in Daisy Chain. - Config Register ESAI1_IPP_IND_SDO0_SELECT_INPUT for mode ALT0. - Config Register GPC_PMIC_RDY_SELECT_INPUT for mode ALT3. - Config Register SDMA_EVENTS_14_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: TX0 of instance: esai1. 001 Select mux mode: ALT1 mux port: GPIO[12] of instance: gpio7. 010 Select mux mode: ALT2 mux port: SDMA_EXT_EVENT[0] of instance: sdma. 011 Select mux mode: ALT3 mux port: PMIC_RDY of instance: gpc. 100 Select mux mode: ALT4 mux port: CE_RTC_FSV_TRIG of instance: rtc.

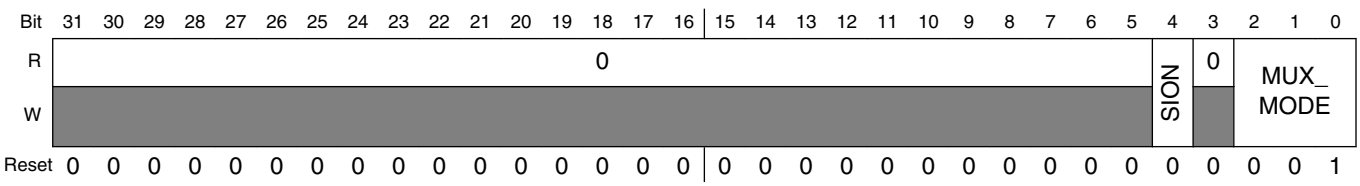
Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_GPIO_17 field descriptions (continued)

Field	Description
	101 Select mux mode: ALT5 mux port: OUT1 of instance: spdif. 110 Select mux mode: ALT6 mux port: SNOOP2 of instance: ipu. 111 Select mux mode: ALT7 mux port: JTAG_ACT of instance: sjc.

43.3.210 IOMUXC_SW_MUX_CTL_PAD_GPIO_18 (IOMUXC_SW_MUX_CTL_PAD_GPIO_18)

Address: IOMUXC_SW_MUX_CTL_PAD_GPIO_18 is 53FA_8000h base + 344h offset = 53FA_8344h

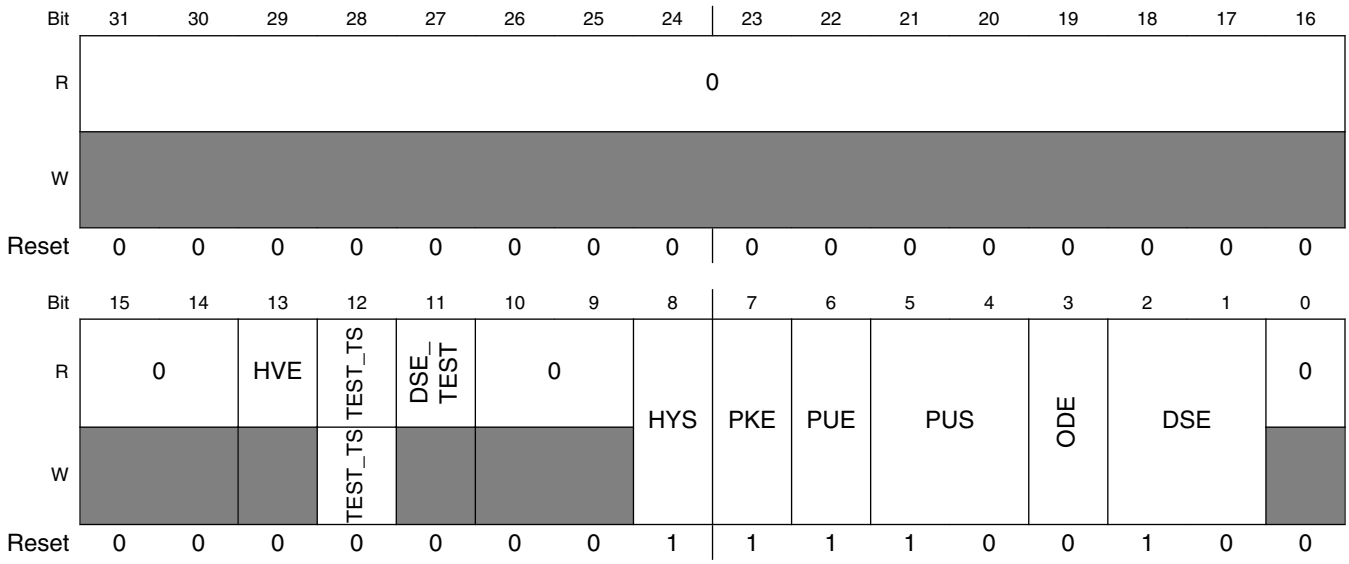


IOMUXC_SW_MUX_CTL_PAD_GPIO_18 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 SION	Software Input On Field. Force the selected mux mode Input path no matter of MUX_MODE functionality. 1 Force input path of pad GPIO_18. 0 Input Path is determined by functionality of the selected mux mode (regular).
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 MUX_MODE	MUX Mode Select Field. Select 1 of 8 iomux modes to be used for pad: GPIO_18. NOTE: Pad GPIO_18 is involved in Daisy Chain. - Config Register CCM_IPP_ASRC_EXT_SELECT_INPUT for mode ALT5. - Config Register ESAI1_IPP_IND_SDO1_SELECT_INPUT for mode ALT0. - Config Register OWIRE_BATTERY_LINE_IN_SELECT_INPUT for mode ALT3. - Config Register SDMA_EVENTS_15_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: TX1 of instance: esai1. 001 Select mux mode: ALT1 mux port: GPIO[13] of instance: gpio7. 010 Select mux mode: ALT2 mux port: SDMA_EXT_EVENT[1] of instance: sdma. 011 Select mux mode: ALT3 mux port: LINE of instance: owire. 100 Select mux mode: ALT4 mux port: CE_RTC_ALARM2_TRIG of instance: rtc. 101 Select mux mode: ALT5 mux port: ASRC_EXT_CLK of instance: ccm. 110 Select mux mode: ALT6 mux port: LCTL of instance: esdhc1. 111 Select mux mode: ALT7 mux port: SYSTEM_RST of instance: src.

43.3.211 IOMUXC_SW_PAD_CTL_PAD_GPIO_19 (IOMUXC_SW_PAD_CTL_PAD_GPIO_19)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_19 is 53FA_8000h base + 348h offset = 53FA_8348h



IOMUXC_SW_PAD_CTL_PAD_GPIO_19 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_19. 0 Hysteresis Disabled 1 Hysteresis Enabled

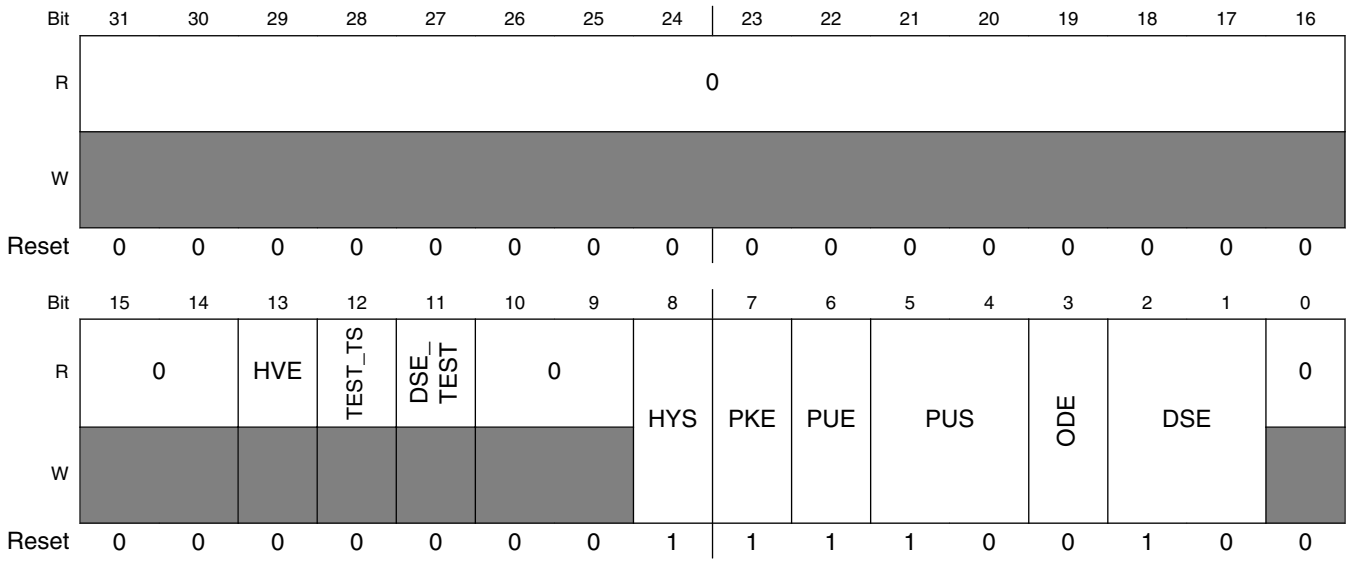
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_19 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_19. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_19. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_19. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_19. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_19. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.212 IOMUXC_SW_PAD_CTL_PAD_KEY_COL0 (IOMUXC_SW_PAD_CTL_PAD_KEY_COL0)

Address: IOMUXC_SW_PAD_CTL_PAD_KEY_COL0 is 53FA_8000h base + 34Ch offset = 53FA_834Ch



IOMUXC_SW_PAD_CTL_PAD_KEY_COL0 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: KEY_COL0. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_COL0 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: KEY_COL0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: KEY_COL0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: KEY_COL0. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: KEY_COL0. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: KEY_COL0. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.213 IOMUXC_SW_PAD_CTL_PAD_KEY_ROW0 (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW0)

Address: IOMUXC_SW_PAD_CTL_PAD_KEY_ROW0 is 53FA_8000h base + 350h offset = 53FA_8350h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW0 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: KEY_ROW0. 0 Hysteresis Disabled 1 Hysteresis Enabled

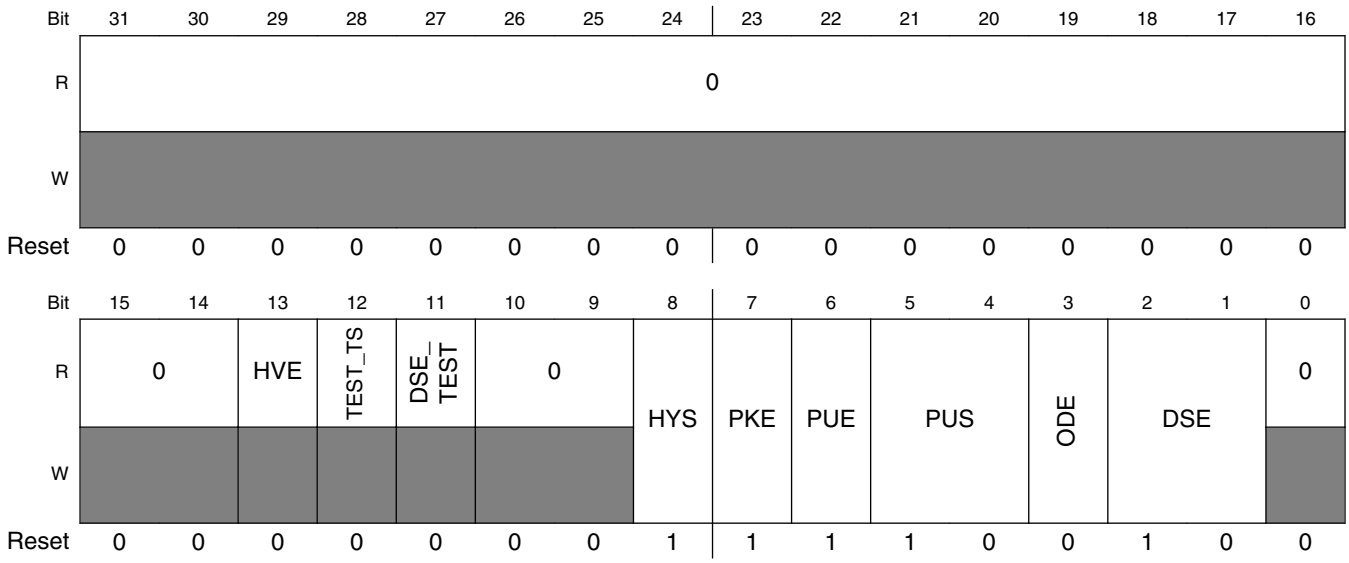
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW0 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: KEY_ROW0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: KEY_ROW0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: KEY_ROW0. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: KEY_ROW0. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: KEY_ROW0. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.214 IOMUXC_SW_PAD_CTL_PAD_KEY_COL1 (IOMUXC_SW_PAD_CTL_PAD_KEY_COL1)

Address: IOMUXC_SW_PAD_CTL_PAD_KEY_COL1 is 53FA_8000h base + 354h offset = 53FA_8354h



IOMUXC_SW_PAD_CTL_PAD_KEY_COL1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: KEY_COL1. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_COL1 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: KEY_COL1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: KEY_COL1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: KEY_COL1. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: KEY_COL1. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: KEY_COL1. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.215 IOMUXC_SW_PAD_CTL_PAD_KEY_ROW1 (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW1)

Address: IOMUXC_SW_PAD_CTL_PAD_KEY_ROW1 is 53FA_8000h base + 358h offset = 53FA_8358h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: KEY_ROW1. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW1 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: KEY_ROW1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: KEY_ROW1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: KEY_ROW1. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: KEY_ROW1. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: KEY_ROW1. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.216 IOMUXC_SW_PAD_CTL_PAD_KEY_COL2 (IOMUXC_SW_PAD_CTL_PAD_KEY_COL2)

Address: IOMUXC_SW_PAD_CTL_PAD_KEY_COL2 is 53FA_8000h base + 35Ch offset = 53FA_835Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_KEY_COL2 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: KEY_COL2. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_COL2 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: KEY_COL2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: KEY_COL2. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: KEY_COL2. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: KEY_COL2. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: KEY_COL2. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.217 IOMUXC_SW_PAD_CTL_PAD_KEY_ROW2 (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW2)

Address: IOMUXC_SW_PAD_CTL_PAD_KEY_ROW2 is 53FA_8000h base + 360h offset = 53FA_8360h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW2 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: KEY_ROW2. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW2 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: KEY_ROW2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: KEY_ROW2. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: KEY_ROW2. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: KEY_ROW2. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: KEY_ROW2. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.218 IOMUXC_SW_PAD_CTL_PAD_KEY_COL3 (IOMUXC_SW_PAD_CTL_PAD_KEY_COL3)

Address: IOMUXC_SW_PAD_CTL_PAD_KEY_COL3 is 53FA_8000h base + 364h offset = 53FA_8364h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_KEY_COL3 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: KEY_COL3. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_COL3 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: KEY_COL3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: KEY_COL3. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: KEY_COL3. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: KEY_COL3. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: KEY_COL3. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.219 IOMUXC_SW_PAD_CTL_PAD_KEY_ROW3 (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW3)

Address: IOMUXC_SW_PAD_CTL_PAD_KEY_ROW3 is 53FA_8000h base + 368h offset = 53FA_8368h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW3 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: KEY_ROW3. 0 Hysteresis Disabled 1 Hysteresis Enabled

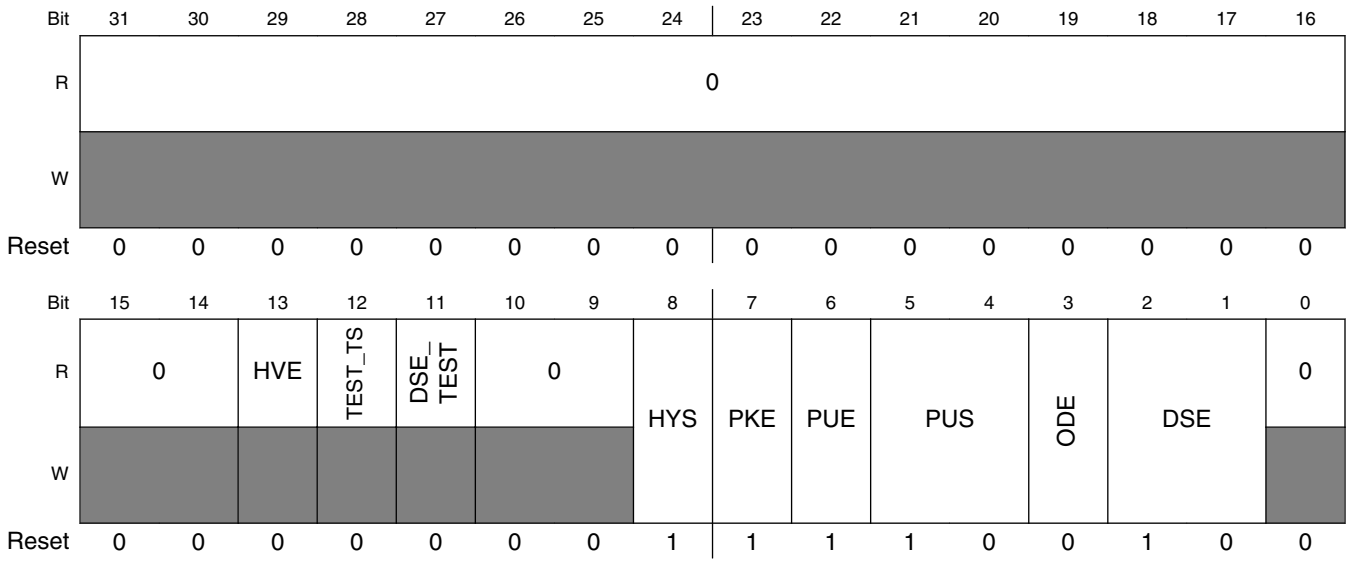
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW3 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: KEY_ROW3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: KEY_ROW3. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: KEY_ROW3. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: KEY_ROW3. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: KEY_ROW3. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.220 IOMUXC_SW_PAD_CTL_PAD_KEY_COL4 (IOMUXC_SW_PAD_CTL_PAD_KEY_COL4)

Address: IOMUXC_SW_PAD_CTL_PAD_KEY_COL4 is 53FA_8000h base + 36Ch offset = 53FA_836Ch



IOMUXC_SW_PAD_CTL_PAD_KEY_COL4 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: KEY_COL4. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_COL4 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: KEY_COL4. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: KEY_COL4. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: KEY_COL4. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: KEY_COL4. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: KEY_COL4. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.221 IOMUXC_SW_PAD_CTL_PAD_KEY_ROW4 (IOMUXC_SW_PAD_CTL_PAD_KEY_ROW4)

Address: IOMUXC_SW_PAD_CTL_PAD_KEY_ROW4 is 53FA_8000h base + 370h offset = 53FA_8370h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW4 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: KEY_ROW4. 0 Hysteresis Disabled 1 Hysteresis Enabled

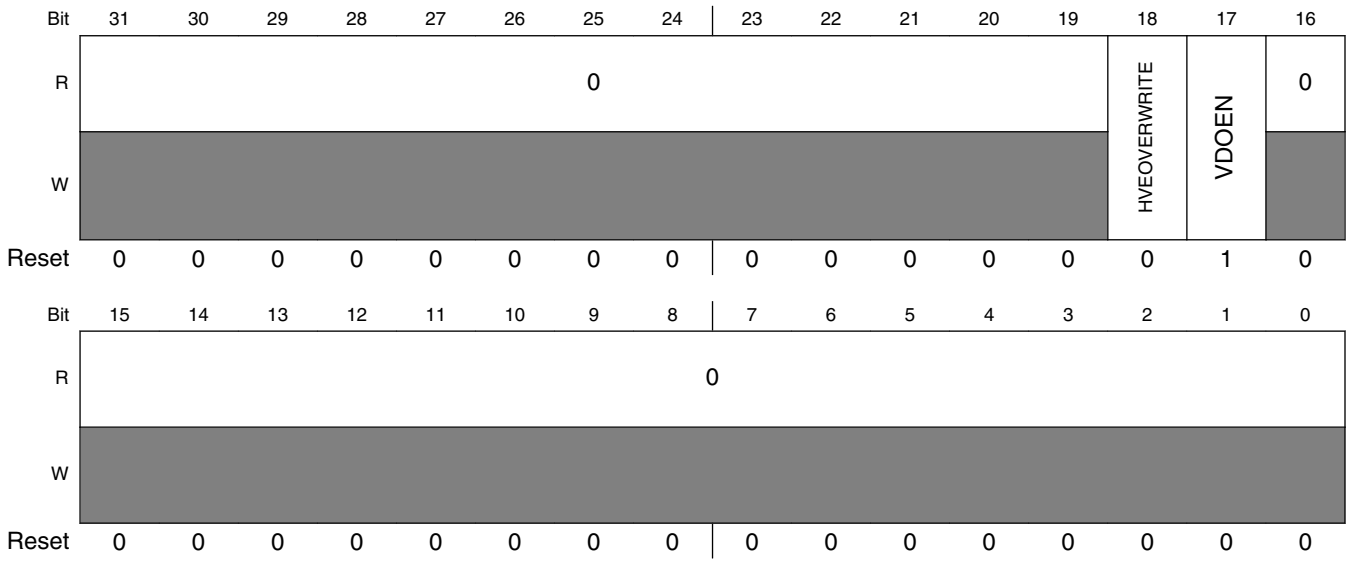
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_KEY_ROW4 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: KEY_ROW4. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: KEY_ROW4. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: KEY_ROW4. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: KEY_ROW4. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: KEY_ROW4. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.222 IOMUXC_SW_PAD_CTL_PAD_NVCC_KEYPAD (IOMUXC_SW_PAD_CTL_PAD_NVCC_KEYPAD)

Address: IOMUXC_SW_PAD_CTL_PAD_NVCC_KEYPAD is 53FA_8000h base + 374h offset = 53FA_8374h

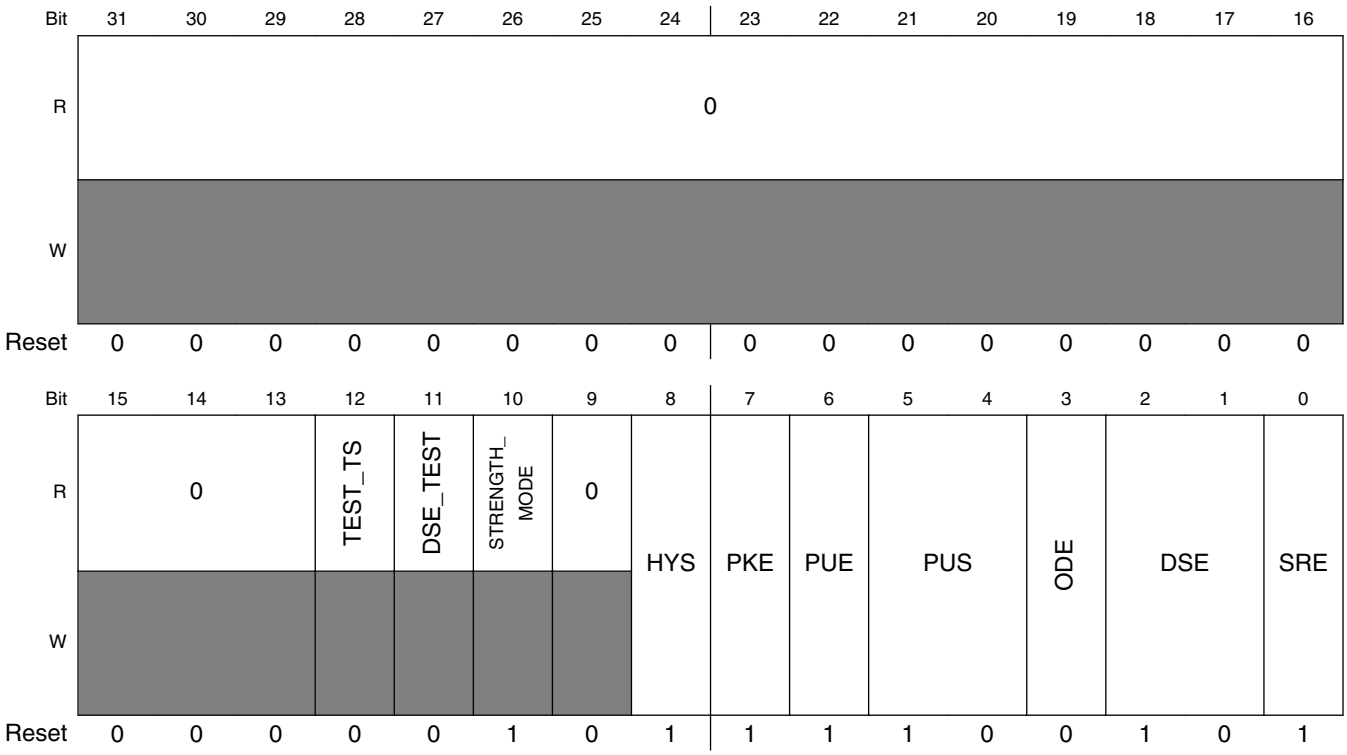


IOMUXC_SW_PAD_CTL_PAD_NVCC_KEYPAD field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18 HVEOVERWRITE	HVEOVERWRITE (ipp_owr) Select one out of next values for pad: NVCC_KEYPAD. This field affects the voltage levels for the following PADS: KEY_ROW4, KEY_COL4, KEY_ROW3, KEY_COL3, KEY_ROW2, KEY_COL2, KEY_ROW1, KEY_COL1, KEY_ROW0, KEY_COL0, GPIO_19 0 PAD support high voltage (3.0V-3.6V). Out of reset value is 0. The HVEOVERWRITE field should be updated before automatic voltage detector is disabled. 1 PAD support low voltage (1.65V-3.1V)
17 VDOEN	VDOEN (ipp_oen) Select one out of next values for pad: NVCC_KEYPAD. 0 Voltage detector output disable and HVEOVERWRITE field will be used. It is recommended to disable the automatic voltage detector after boot and update the HVEOVERWRITE field with the actual I/O supply value. 1 Voltage detector output enable (default 1)
16–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.223 IOMUXC_SW_PAD_CTL_PAD_DI0_DISP_CLK (IOMUXC_SW_PAD_CTL_PAD_DI0_DISP_CLK)

Address: IOMUXC_SW_PAD_CTL_PAD_DI0_DISP_CLK is 53FA_8000h base + 378h offset = 53FA_8378h



IOMUXC_SW_PAD_CTL_PAD_DI0_DISP_CLK field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DI0_DISP_CLK field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DI0_DISP_CLK. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DI0_DISP_CLK. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DI0_DISP_CLK. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DI0_DISP_CLK. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DI0_DISP_CLK. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DI0_DISP_CLK. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DI0_DISP_CLK. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.224 IOMUXC_SW_PAD_CTL_PAD_DI0_PIN15 (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN15)

Address: IOMUXC_SW_PAD_CTL_PAD_DI0_PIN15 is 53FA_8000h base + 37Ch offset = 53FA_837Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	DSE_TEST	STRENGTH_MODE	0	HYS	PKE	PUE	PUS		ODE	DSE		SRE
W																
Reset	0	0	0	0	0	1	0	1	1	1	1	0	0	1	0	1

IOMUXC_SW_PAD_CTL_PAD_DI0_PIN15 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DI0_PIN15 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DI0_PIN15. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DI0_PIN15. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DI0_PIN15. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DI0_PIN15. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DI0_PIN15. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DI0_PIN15. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DI0_PIN15. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.225 IOMUXC_SW_PAD_CTL_PAD_DI0_PIN2 (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN2)

Address: IOMUXC_SW_PAD_CTL_PAD_DI0_PIN2 is 53FA_8000h base + 380h offset = 53FA_8380h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	DSE_TEST	STRENGTH_MODE	0	HYS	PKE	PUE	PUS		ODE	DSE		SRE
W																
Reset	0	0	0	0	0	1	0	1	1	1	1	0	0	1	0	1

IOMUXC_SW_PAD_CTL_PAD_DI0_PIN2 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DI0_PIN2 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DI0_PIN2. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DI0_PIN2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DI0_PIN2. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DI0_PIN2. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DI0_PIN2. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DI0_PIN2. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DI0_PIN2. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.226 IOMUXC_SW_PAD_CTL_PAD_DI0_PIN3 (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN3)

Address: IOMUXC_SW_PAD_CTL_PAD_DI0_PIN3 is 53FA_8000h base + 384h offset = 53FA_8384h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	DSE_TEST	STRENGTH_MODE	0	HYS	PKE	PUE	PUS		ODE	DSE		SRE
W																
Reset	0	0	0	0	0	1	0	1	1	1	1	0	0	1	0	1

IOMUXC_SW_PAD_CTL_PAD_DI0_PIN3 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DI0_PIN3 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DI0_PIN3. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DI0_PIN3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DI0_PIN3. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DI0_PIN3. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DI0_PIN3. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DI0_PIN3. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DI0_PIN3. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.227 IOMUXC_SW_PAD_CTL_PAD_DI0_PIN4 (IOMUXC_SW_PAD_CTL_PAD_DI0_PIN4)

Address: IOMUXC_SW_PAD_CTL_PAD_DI0_PIN4 is 53FA_8000h base + 388h offset = 53FA_8388h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	DSE_TEST	STRENGTH_MODE	0	HYS	PKE	PUE	PUS		ODE	DSE		SRE
W																
Reset	0	0	0	0	0	1	0	1	1	1	1	0	0	1	0	1

IOMUXC_SW_PAD_CTL_PAD_DI0_PIN4 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

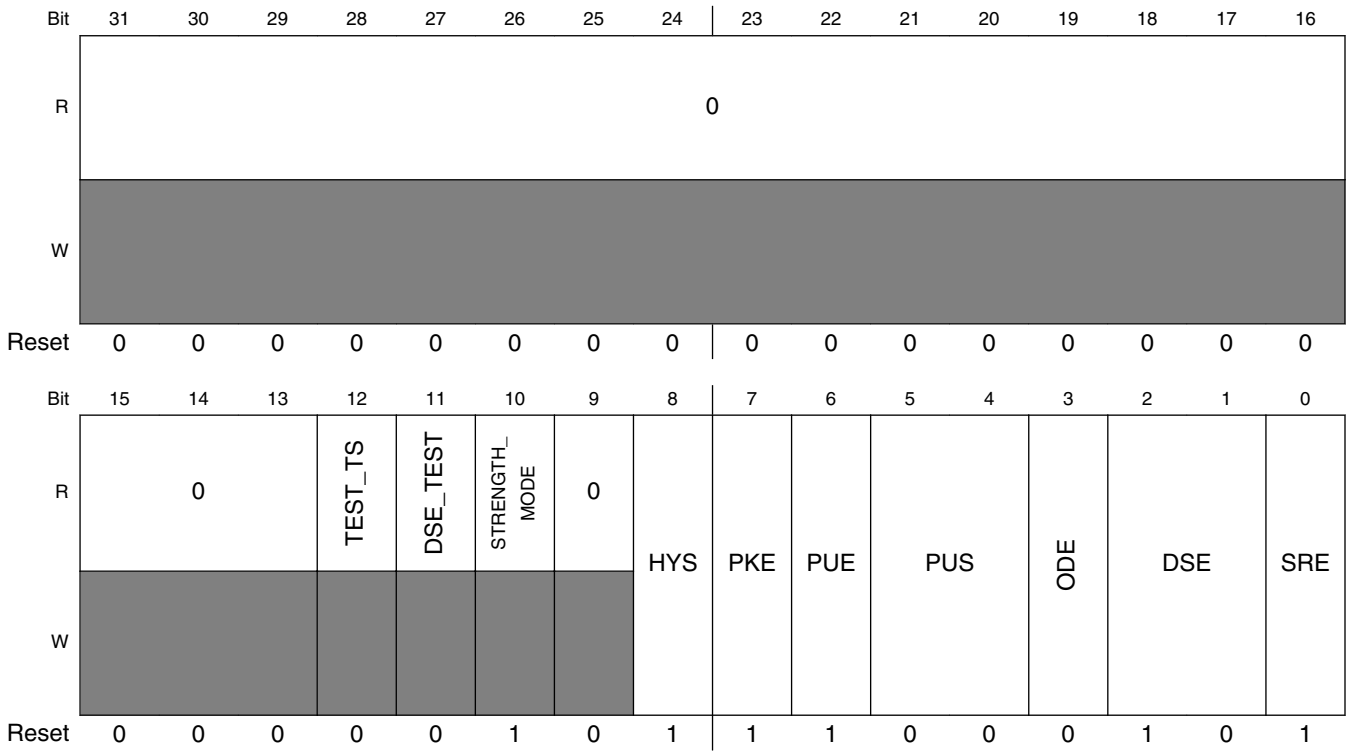
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DI0_PIN4 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DI0_PIN4. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DI0_PIN4. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DI0_PIN4. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DI0_PIN4. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DI0_PIN4. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DI0_PIN4. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DI0_PIN4. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.228 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT0 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT0)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT0 is 53FA_8000h base + 38Ch offset = 53FA_838Ch



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT0 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

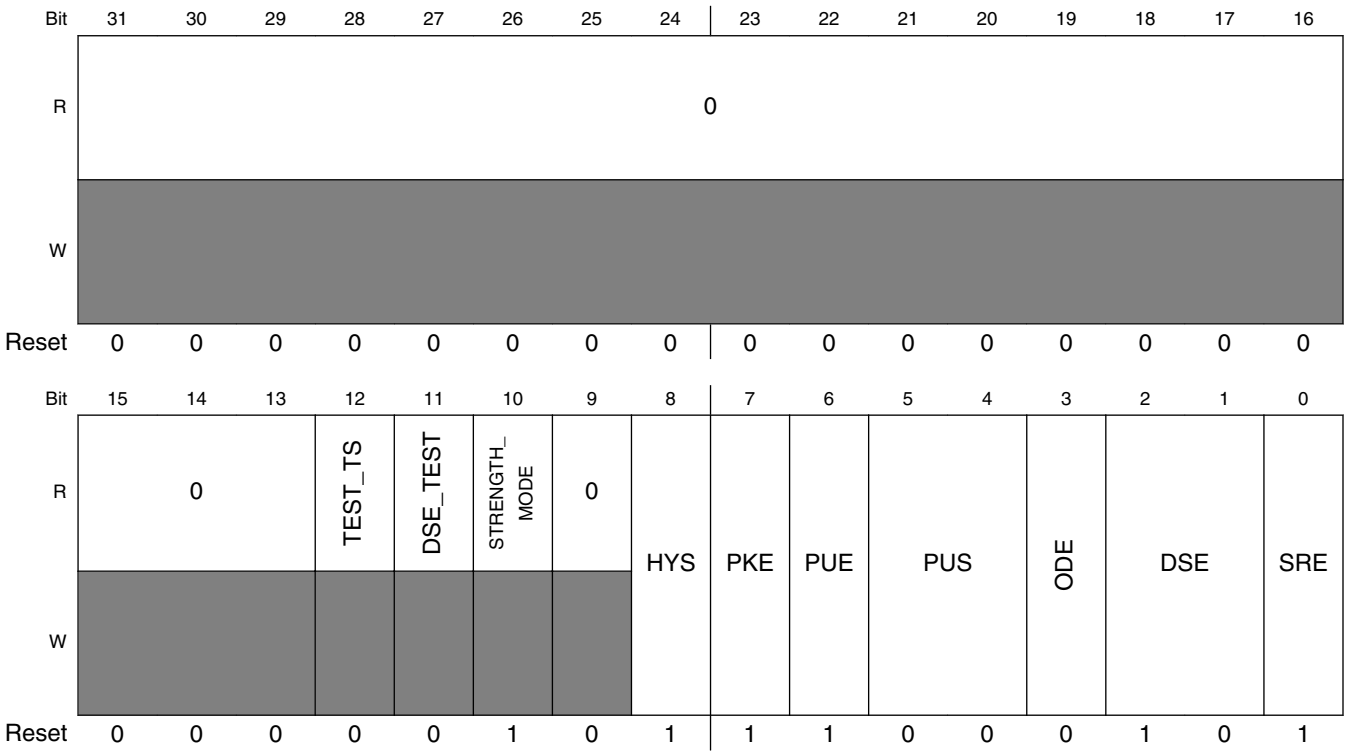
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT0 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT0. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT0. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT0. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT0. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT0. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.229 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT1 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT1)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT1 is 53FA_8000h base + 390h offset = 53FA_8390h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT1 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

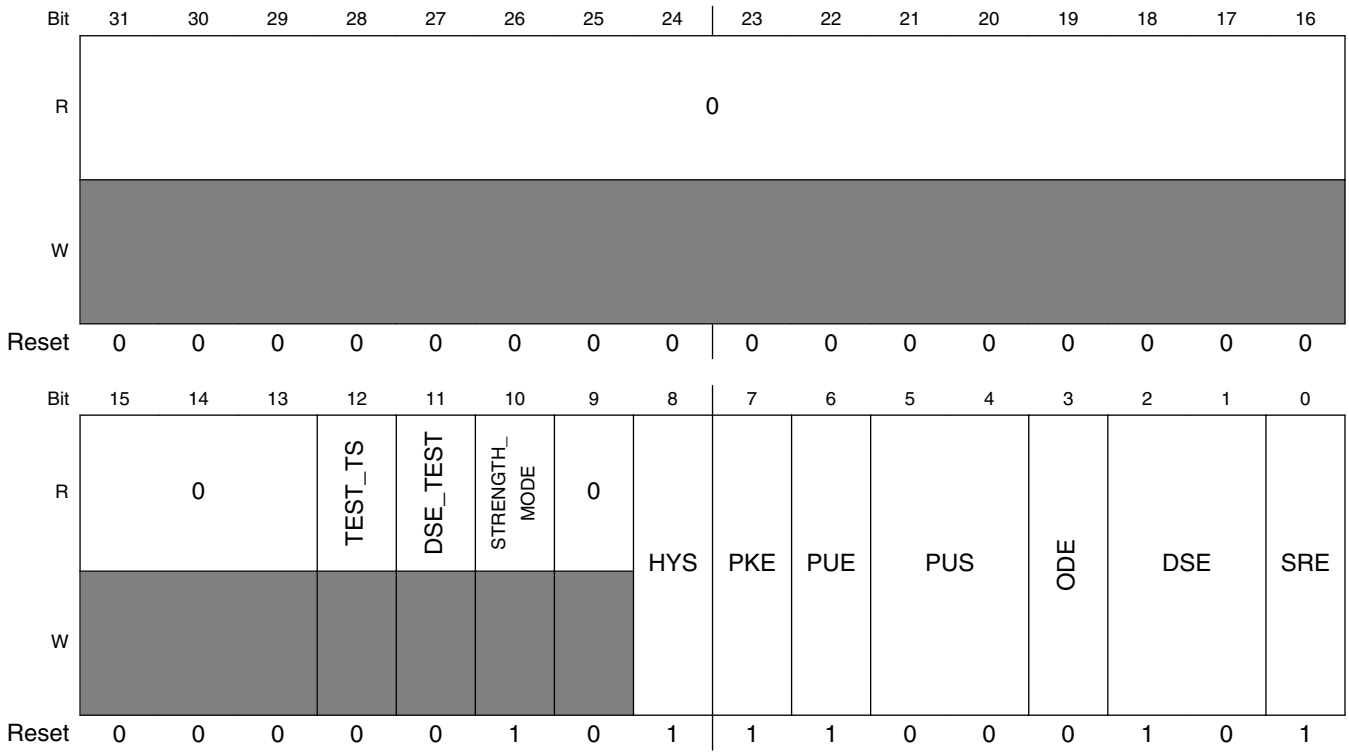
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT1 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT1. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT1. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT1. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT1. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT1. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.230 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT2 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT2)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT2 is 53FA_8000h base + 394h offset = 53FA_8394h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT2 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

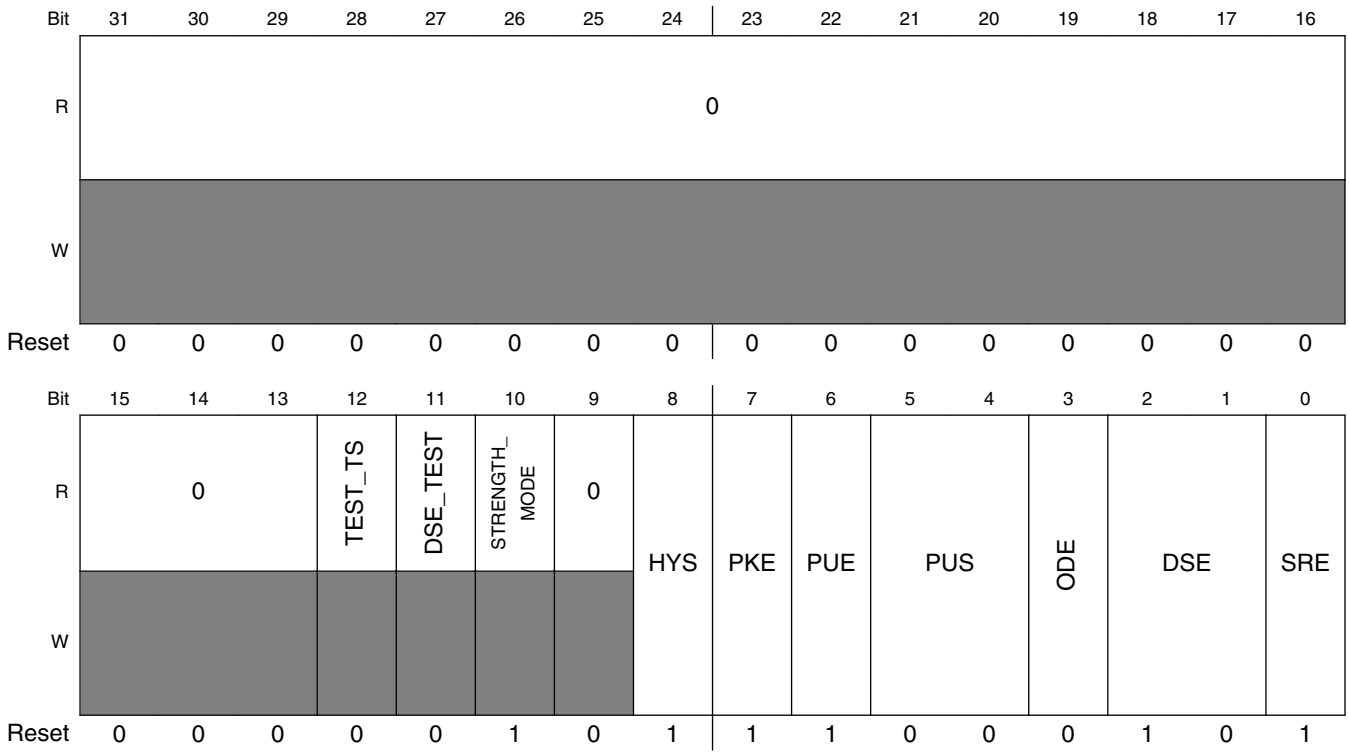
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT2 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT2. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT2. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT2. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT2. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT2. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT2. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.231 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT3 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT3)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT3 is 53FA_8000h base + 398h offset = 53FA_8398h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT3 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

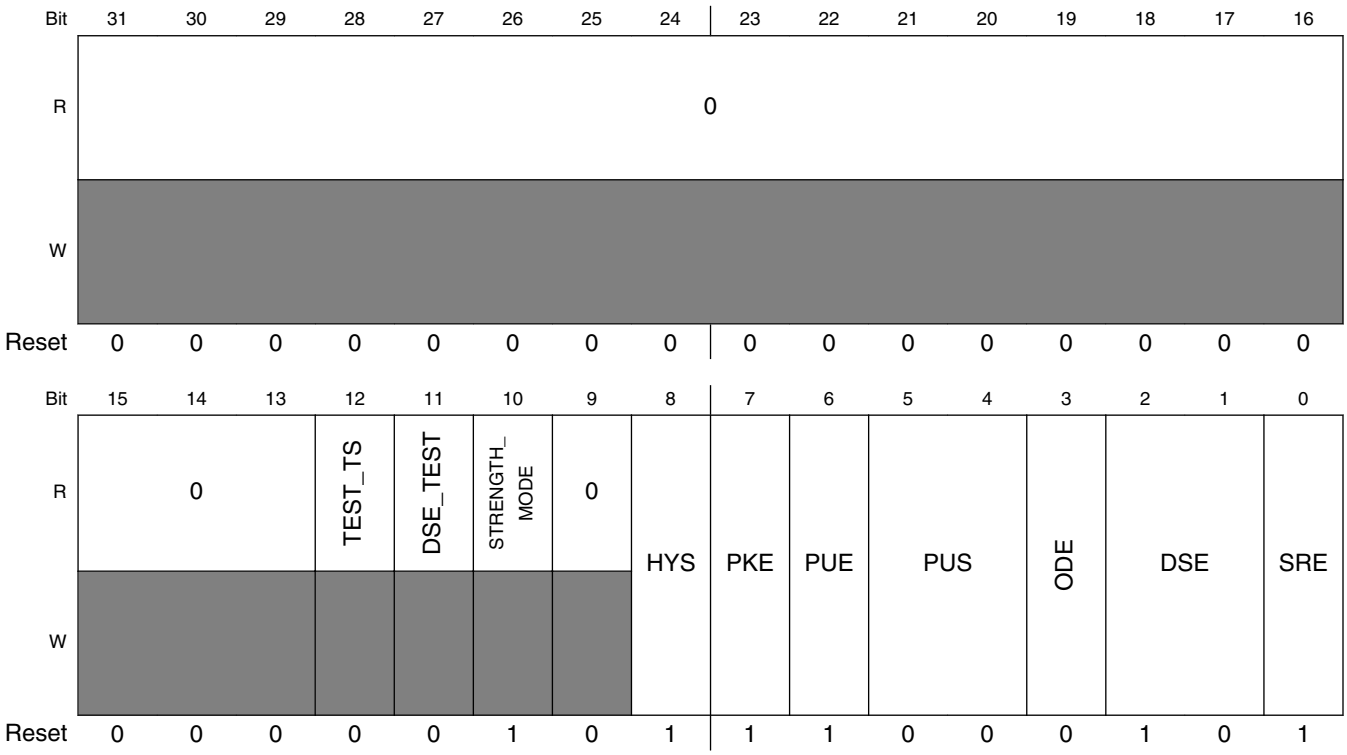
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT3 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT3. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT3. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT3. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT3. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT3. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT3. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.232 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT4 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT4)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT4 is 53FA_8000h base + 39Ch offset = 53FA_839Ch



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT4 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

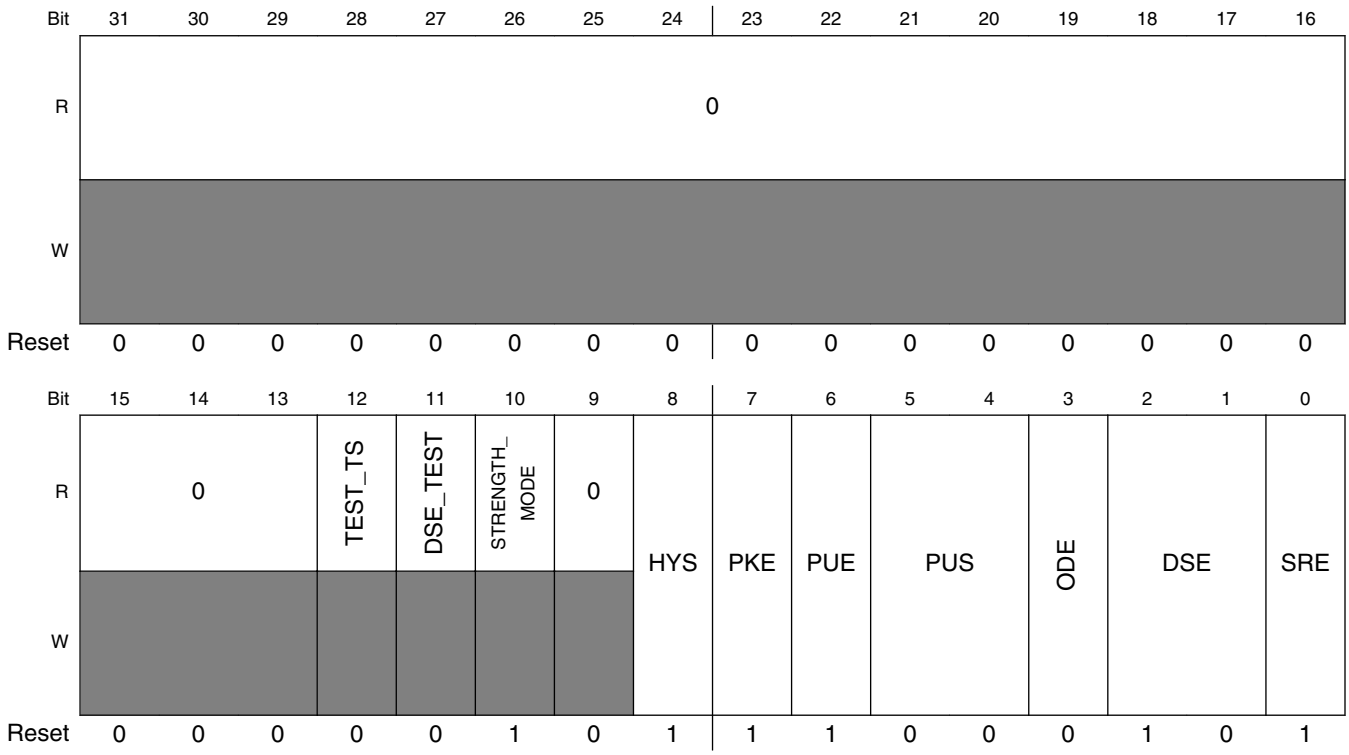
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT4 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT4. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT4. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT4. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT4. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT4. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT4. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT4. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.233 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT5 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT5)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT5 is 53FA_8000h base + 3A0h offset = 53FA_83A0h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT5 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

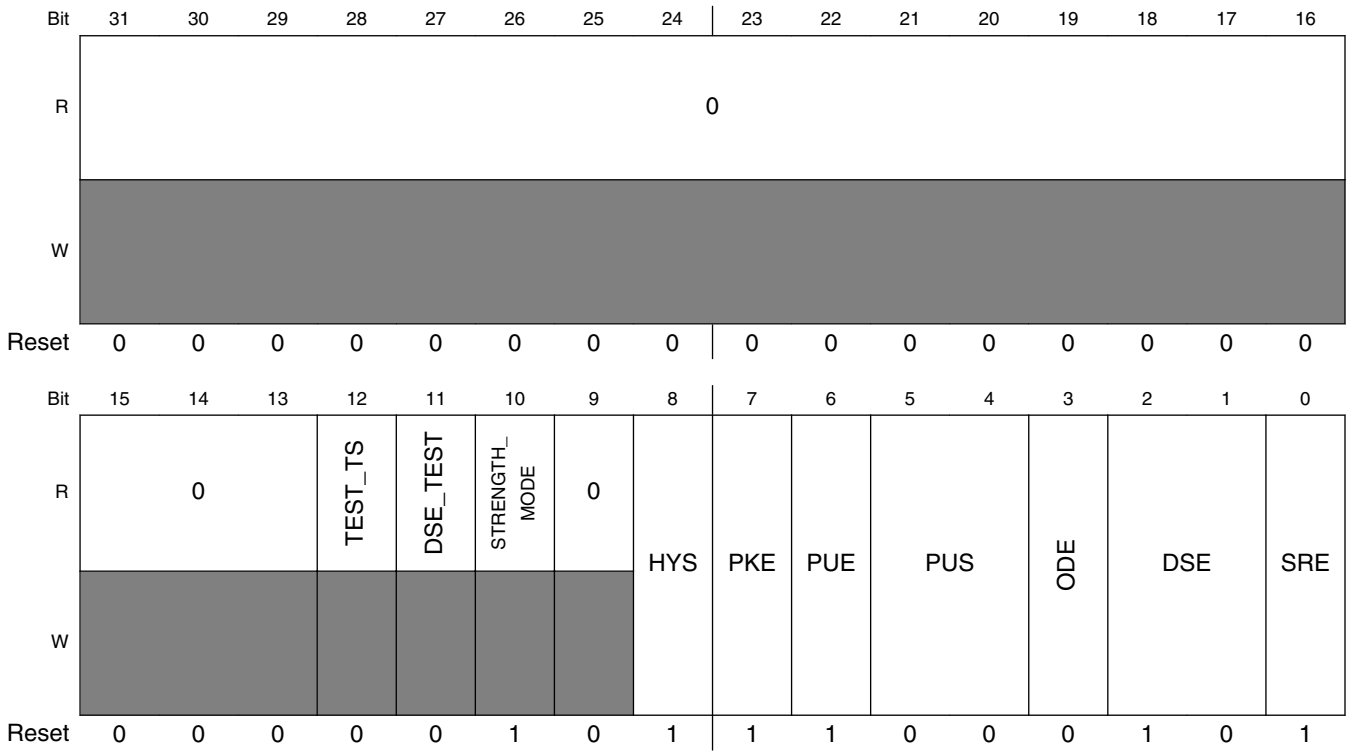
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT5 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT5. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT5. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT5. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT5. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT5. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT5. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT5. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.234 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT6 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT6)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT6 is 53FA_8000h base + 3A4h offset = 53FA_83A4h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT6 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

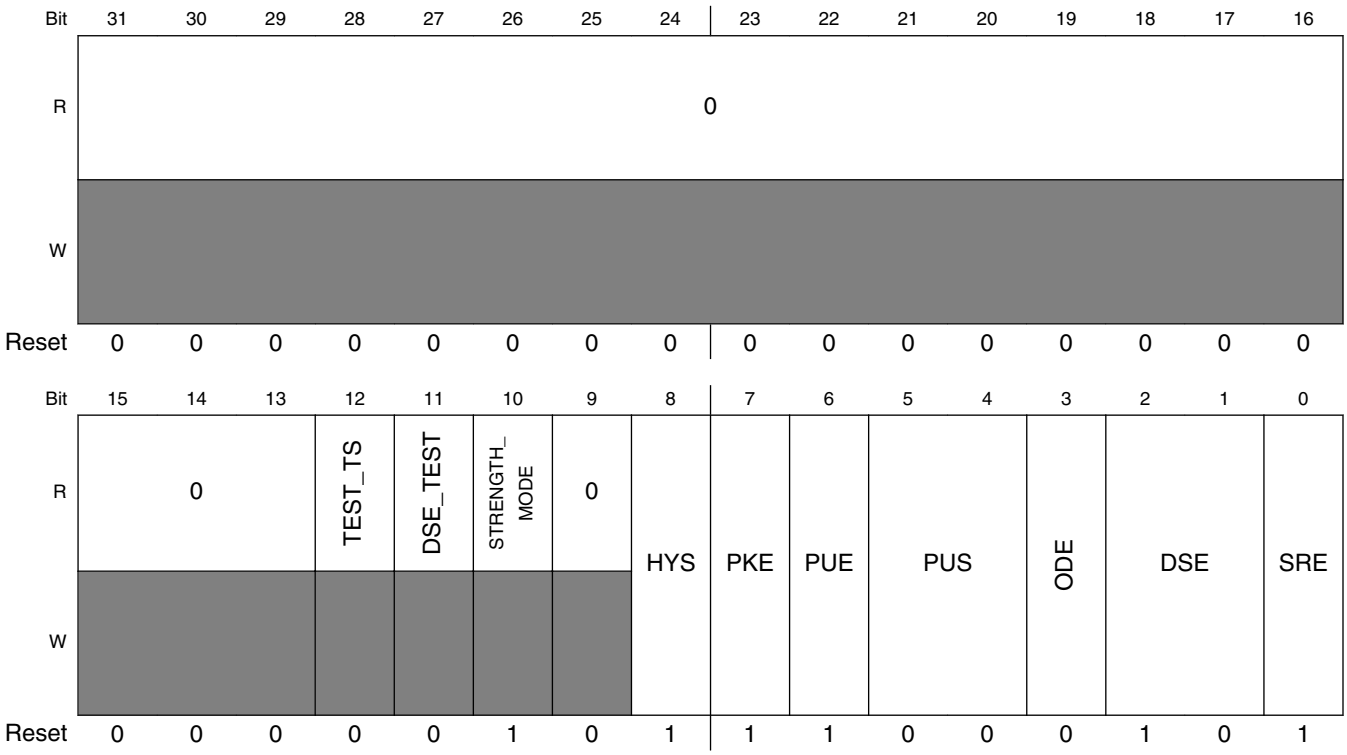
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT6 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT6. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT6. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT6. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT6. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT6. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT6. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT6. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.235 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT7 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT7)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT7 is 53FA_8000h base + 3A8h offset = 53FA_83A8h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT7 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

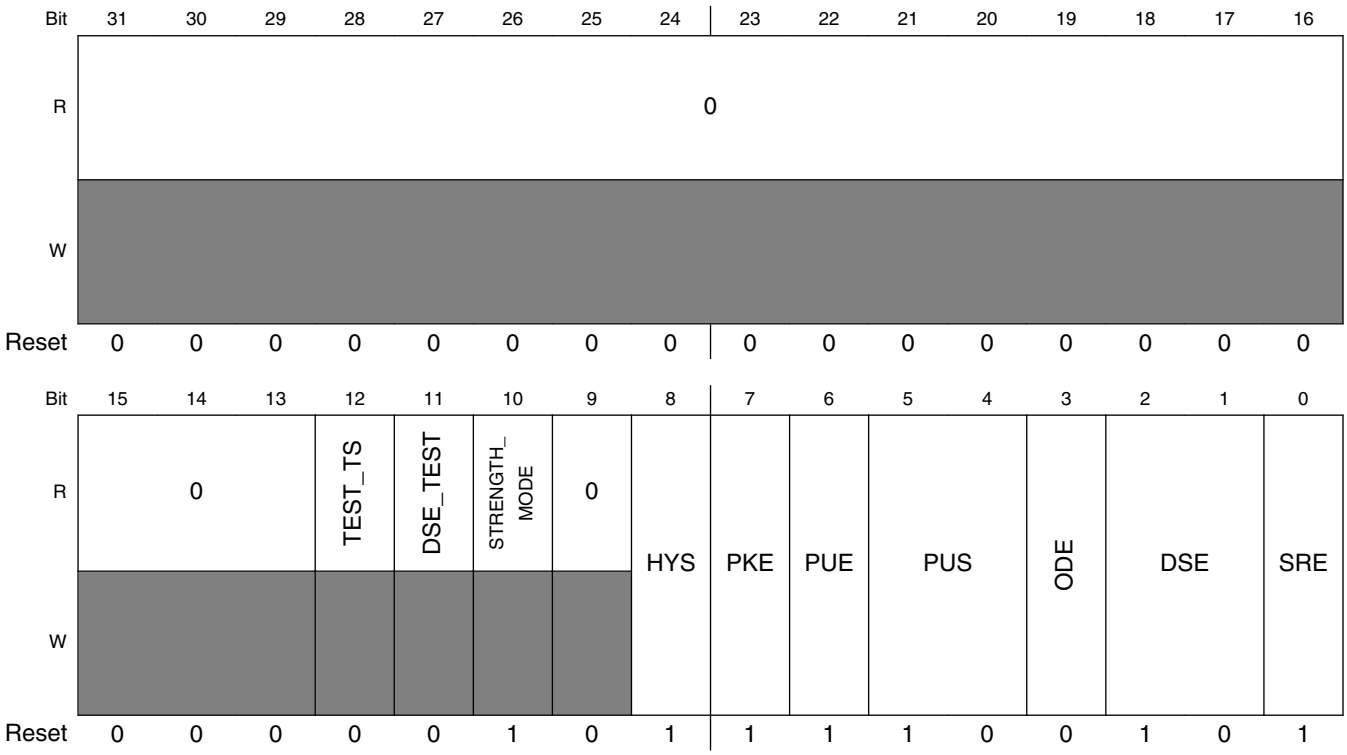
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT7 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT7. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT7. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT7. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT7. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT7. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT7. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT7. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.236 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT8 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT8)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT8 is 53FA_8000h base + 3ACh offset = 53FA_83ACh



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT8 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

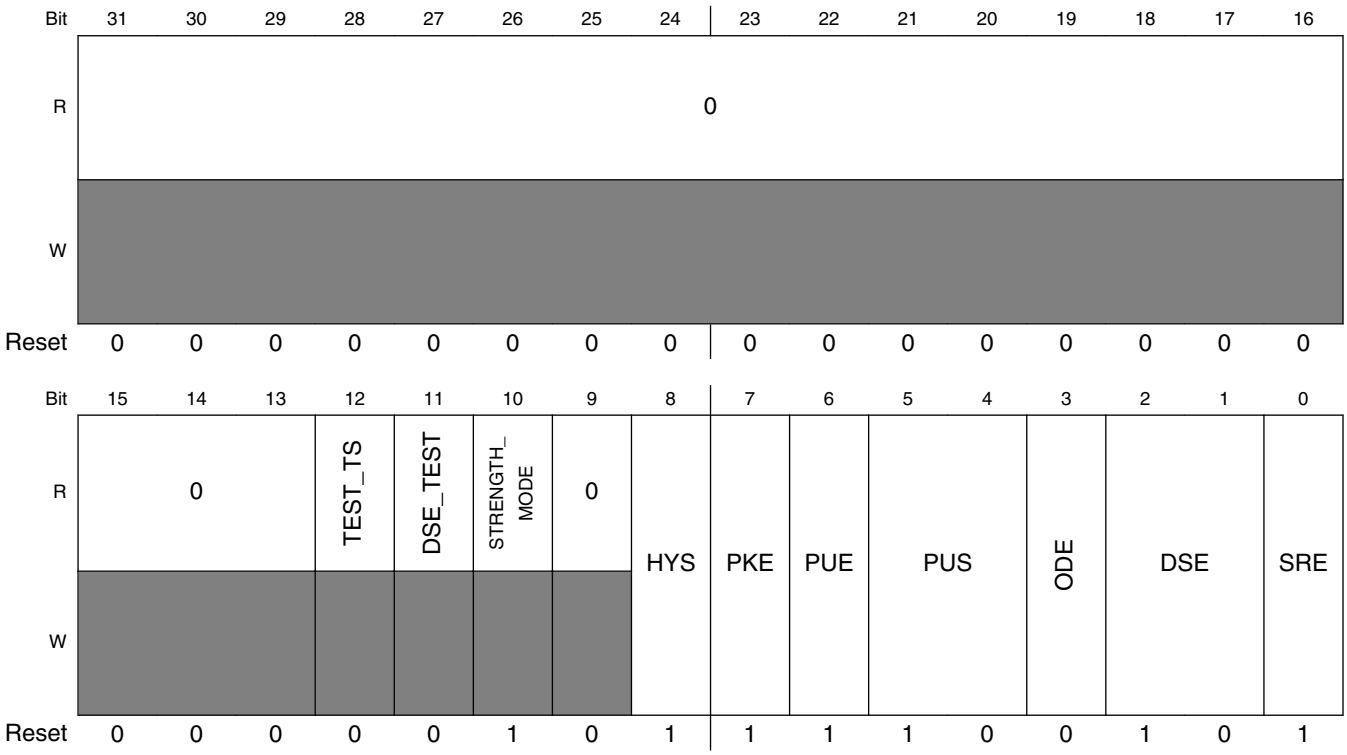
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT8 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT8. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT8. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT8. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT8. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT8. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT8. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT8. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.237 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT9 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT9)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT9 is 53FA_8000h base + 3B0h offset = 53FA_83B0h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT9 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

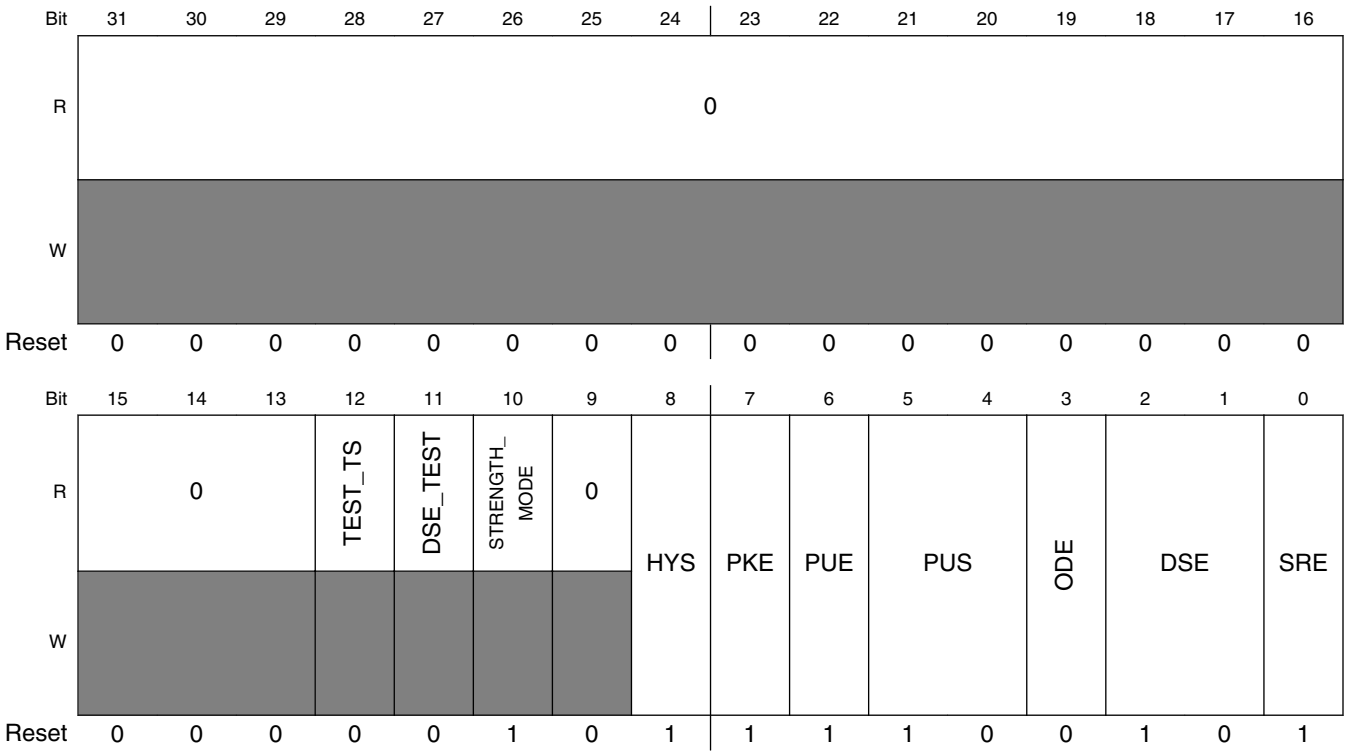
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT9 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT9. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT9. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT9. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT9. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT9. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT9. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT9. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.238 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT10 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT10)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT10 is 53FA_8000h base + 3B4h offset = 53FA_83B4h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT10 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

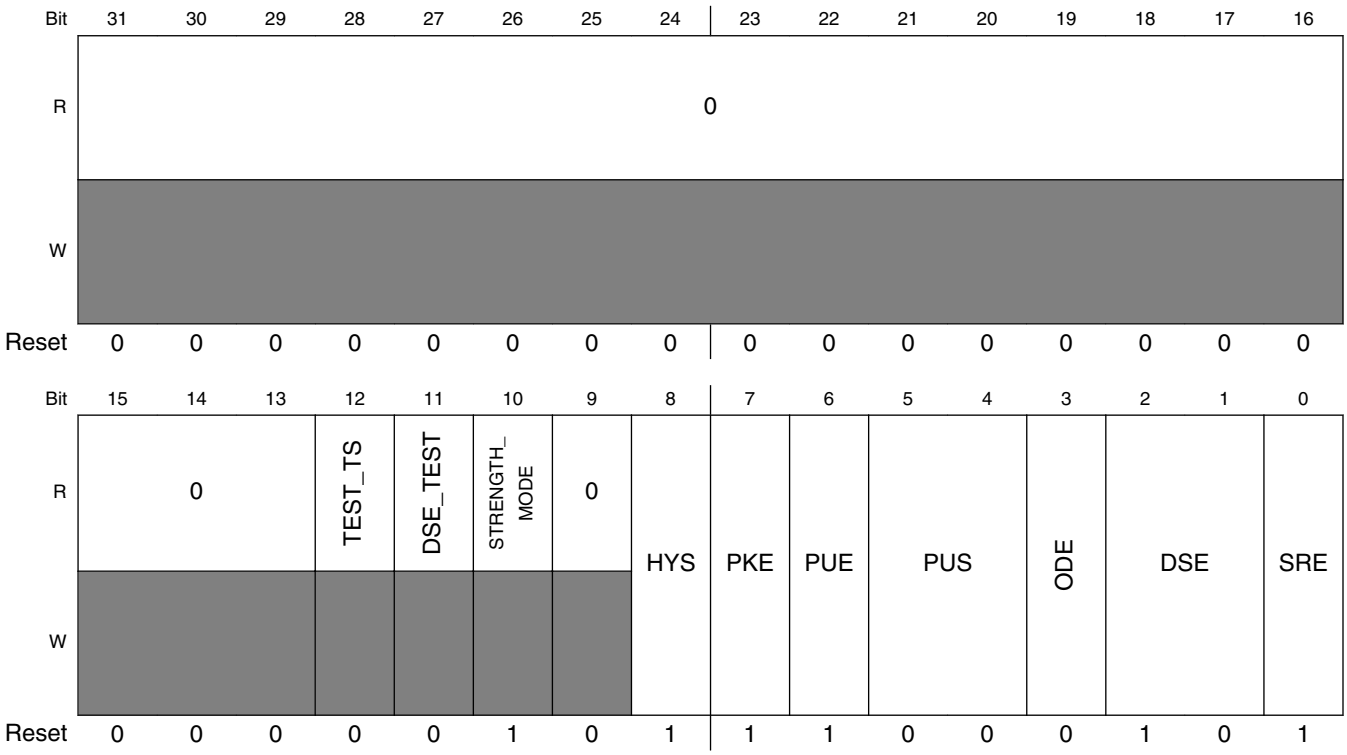
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT10 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT10. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT10. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT10. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT10. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT10. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT10. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT10. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.239 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT11 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT11)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT11 is 53FA_8000h base + 3B8h offset = 53FA_83B8h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT11 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

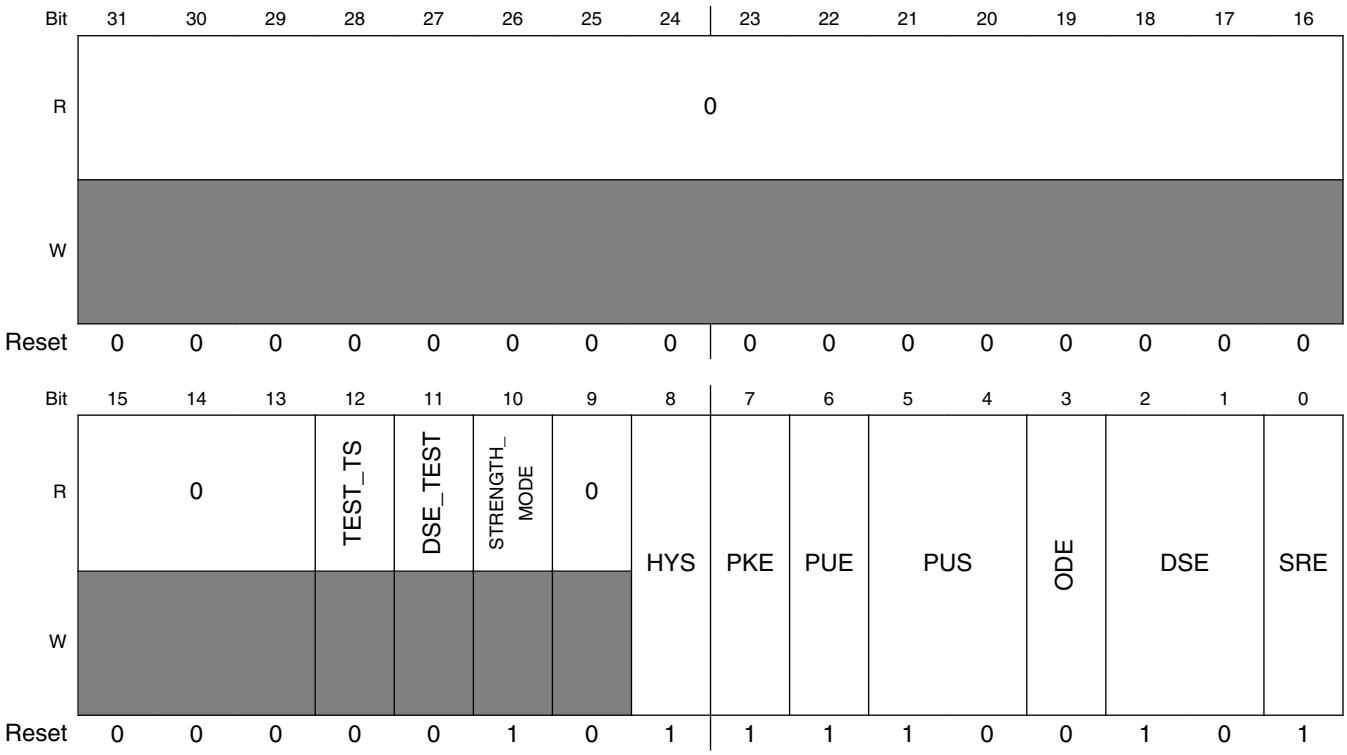
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT11 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT11. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT11. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT11. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT11. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT11. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT11. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT11. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.240 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT12 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT12)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT12 is 53FA_8000h base + 3BCh offset = 53FA_83BCh



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT12 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

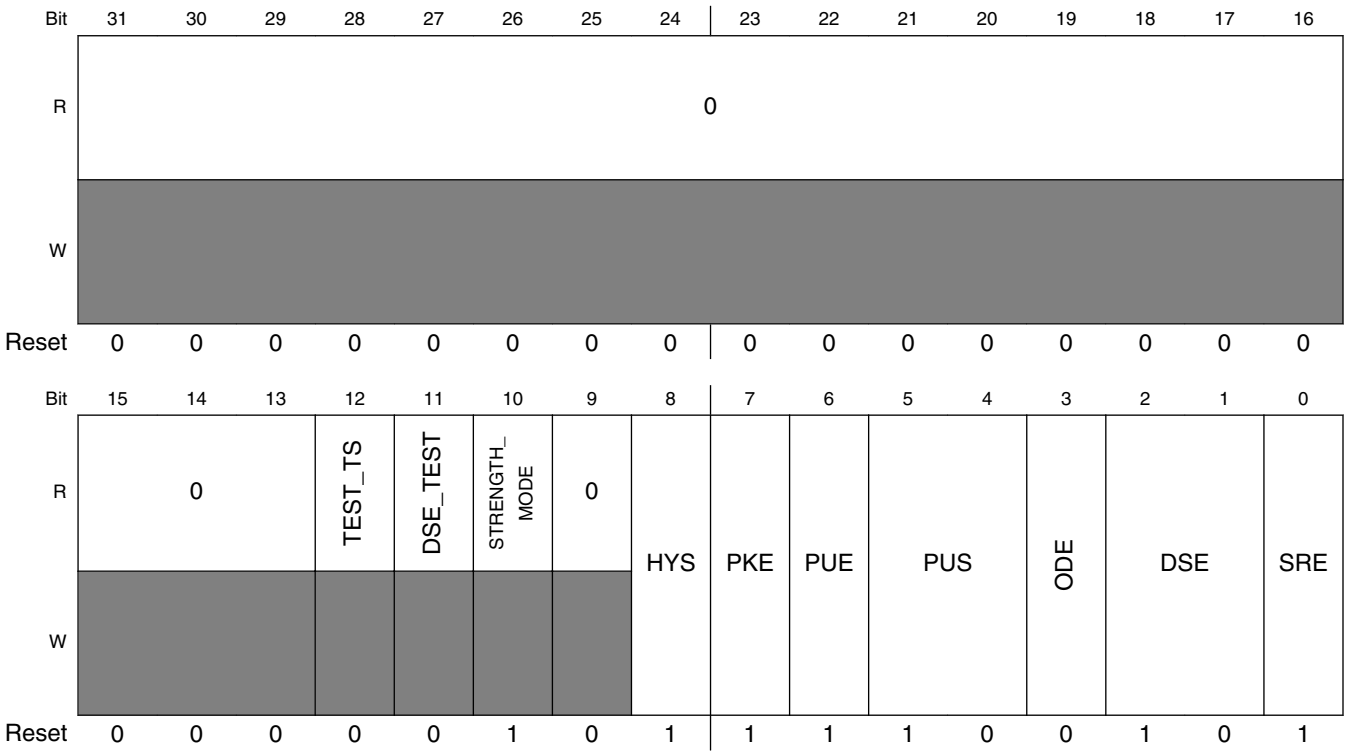
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT12 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT12. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT12. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT12. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT12. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT12. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT12. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT12. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.241 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT13 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT13)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT13 is 53FA_8000h base + 3C0h offset = 53FA_83C0h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT13 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

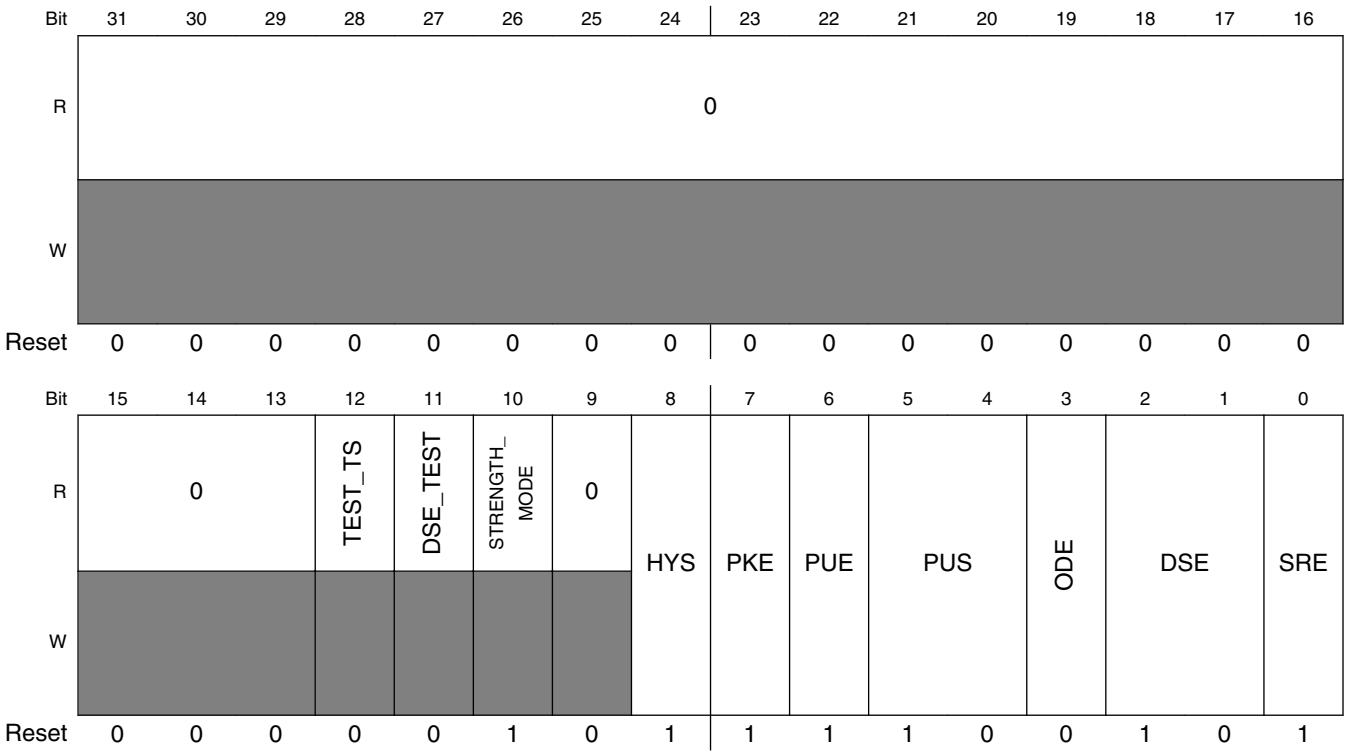
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT13 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT13. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT13. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT13. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT13. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT13. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT13. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT13. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.242 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT14 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT14)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT14 is 53FA_8000h base + 3C4h offset = 53FA_83C4h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT14 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

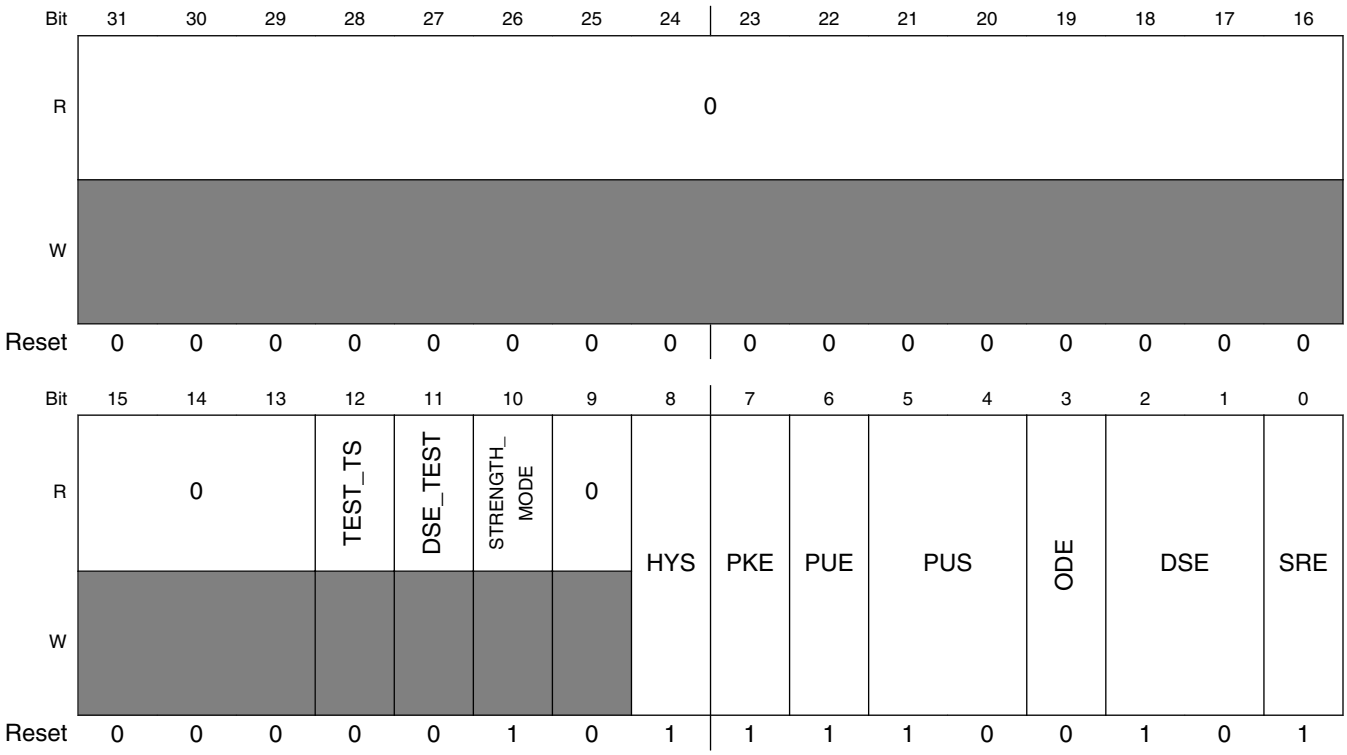
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT14 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT14. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT14. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT14. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT14. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT14. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT14. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT14. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.243 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT15 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT15)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT15 is 53FA_8000h base + 3C8h offset = 53FA_83C8h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT15 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

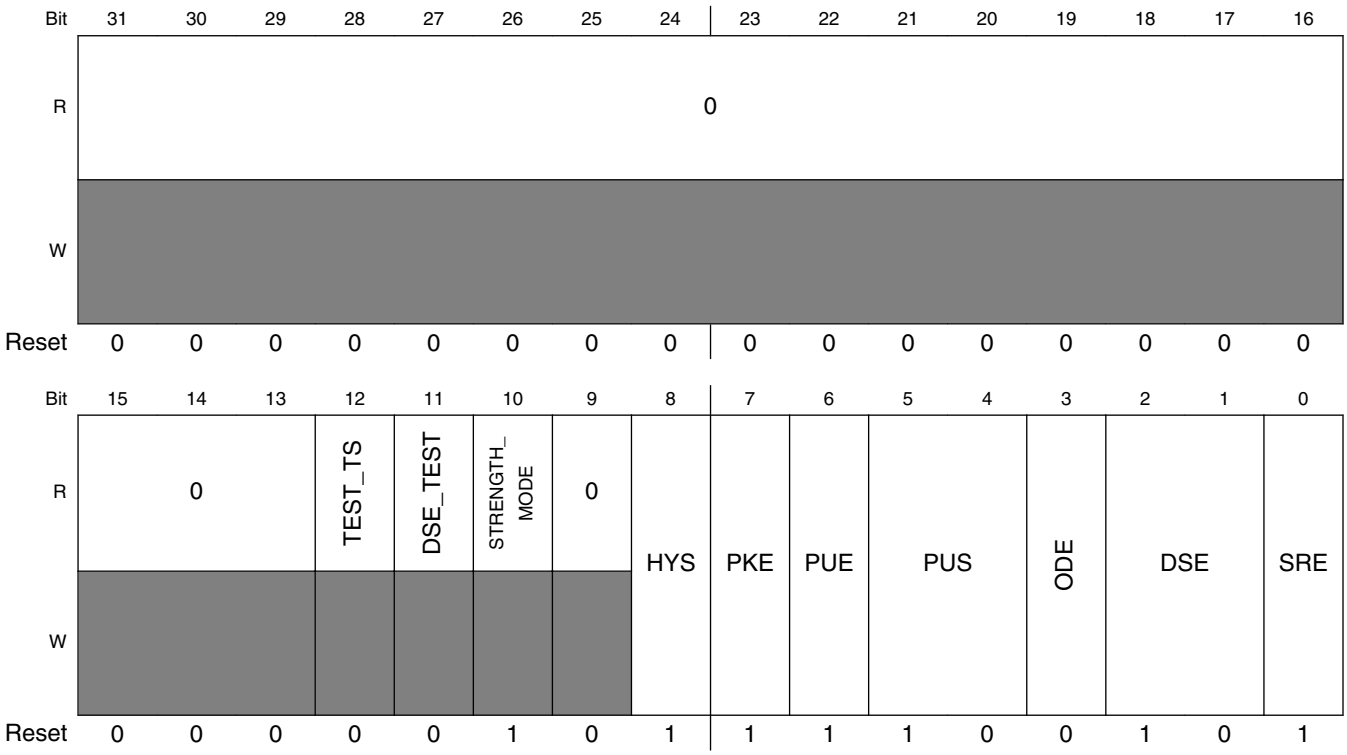
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT15 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT15. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT15. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT15. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT15. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT15. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT15. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT15. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.244 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT16 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT16)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT16 is 53FA_8000h base + 3CCh offset = 53FA_83CCh



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT16 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

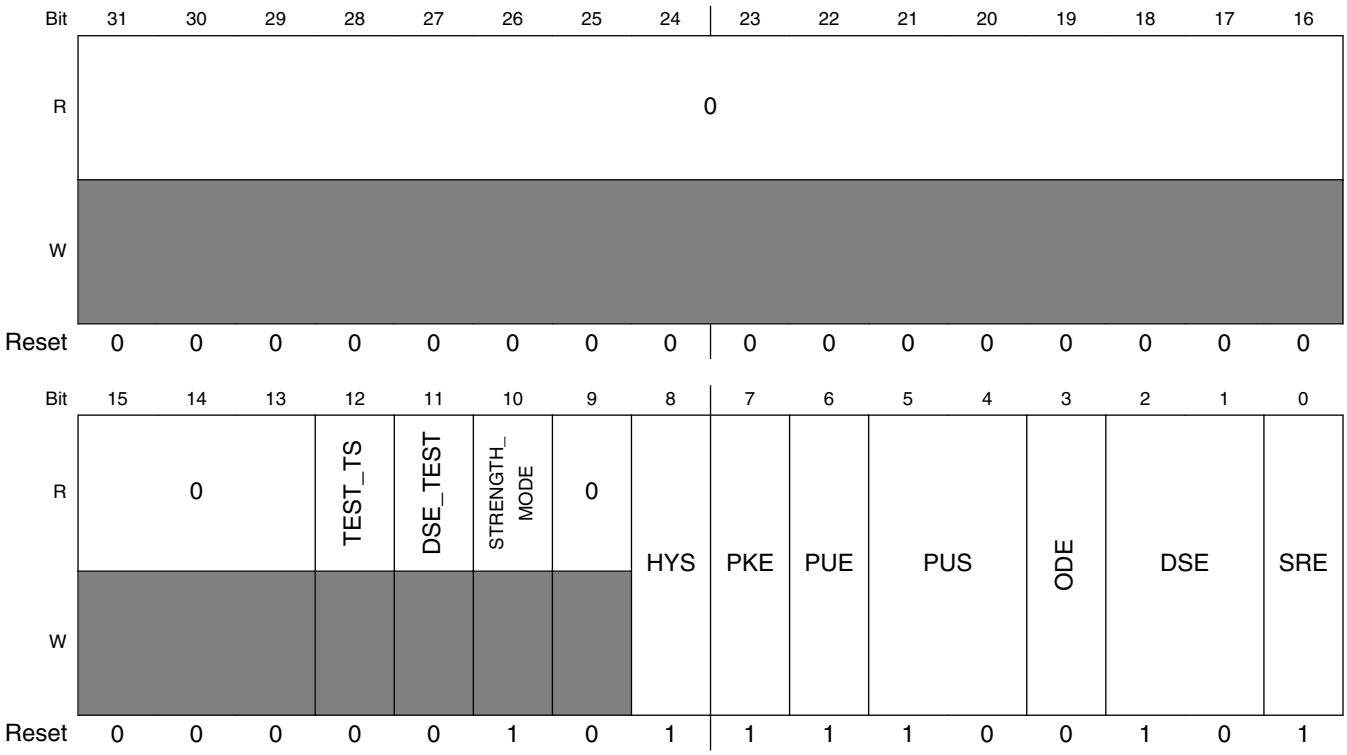
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT16 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT16. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT16. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT16. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT16. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT16. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT16. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT16. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.245 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT17 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT17)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT17 is 53FA_8000h base + 3D0h offset = 53FA_83D0h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT17 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

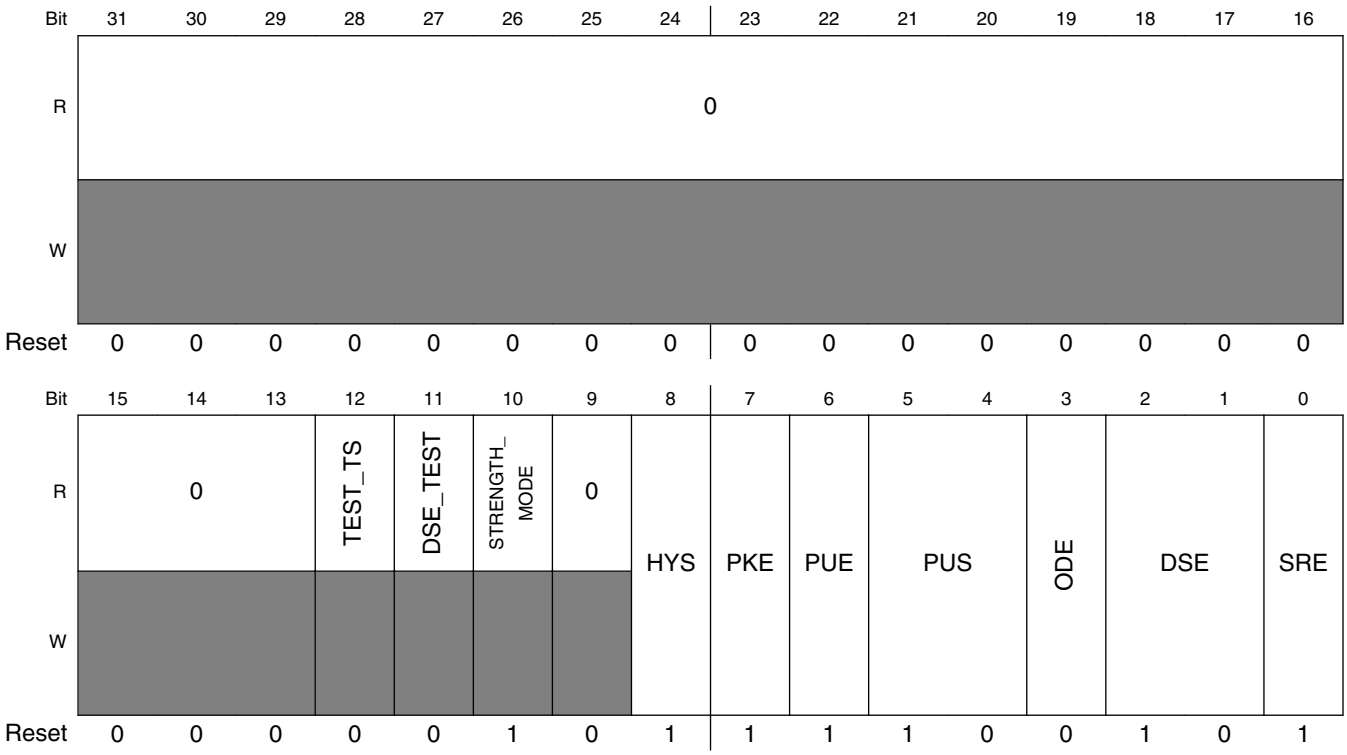
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT17 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT17. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT17. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT17. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT17. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT17. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT17. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT17. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.246 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT18 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT18)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT18 is 53FA_8000h base + 3D4h offset = 53FA_83D4h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT18 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

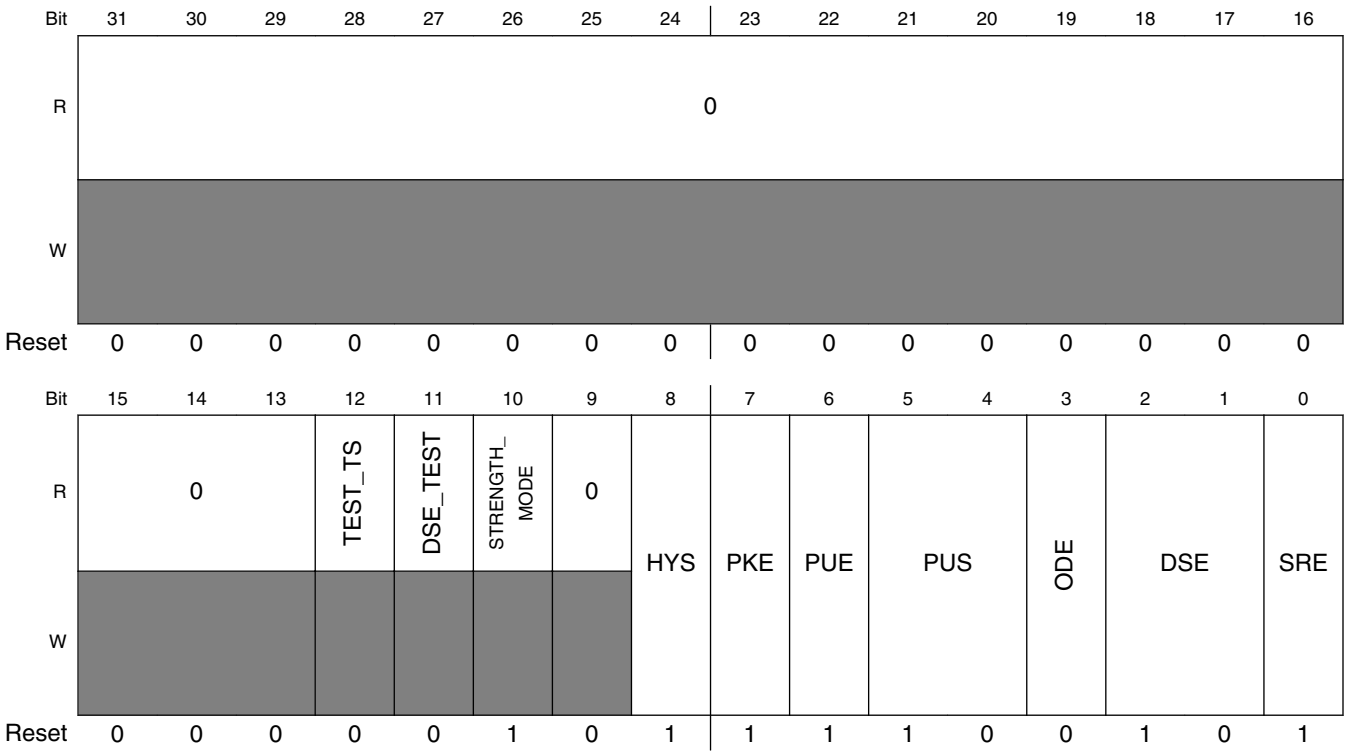
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT18 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT18. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT18. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT18. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT18. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT18. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT18. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT18. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.247 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT19 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT19)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT19 is 53FA_8000h base + 3D8h offset = 53FA_83D8h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT19 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

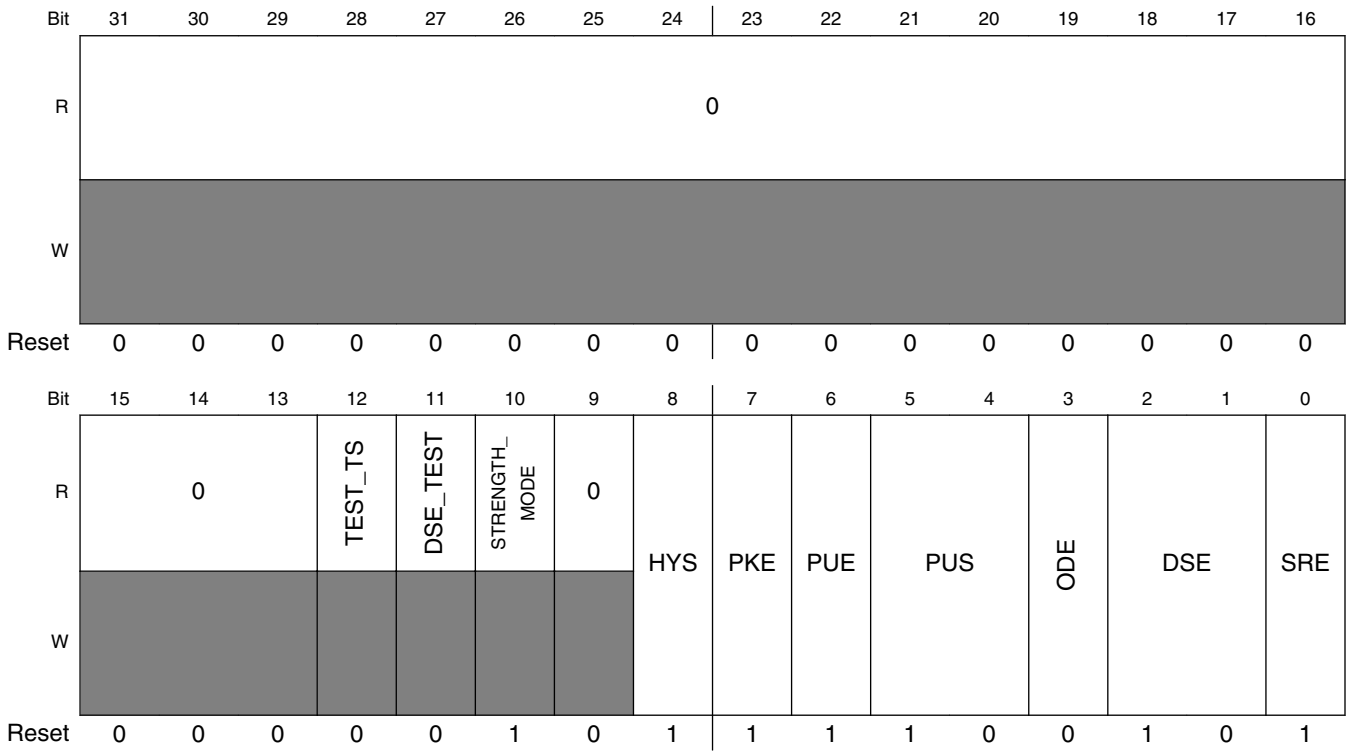
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT19 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT19. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT19. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT19. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT19. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT19. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT19. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT19. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.248 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT20 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT20)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT20 is 53FA_8000h base + 3DCh offset = 53FA_83DCh



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT20 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

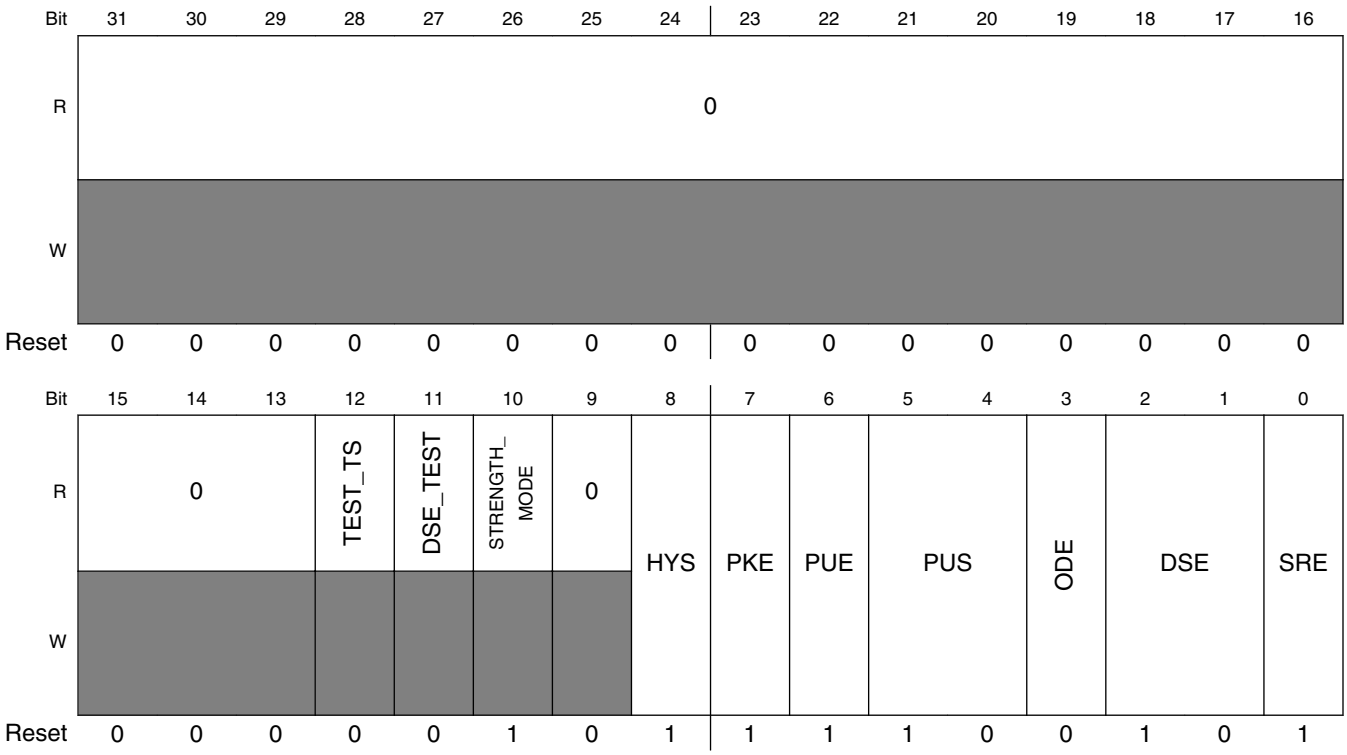
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT20 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT20. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT20. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT20. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT20. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT20. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT20. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT20. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.249 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT21 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT21)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT21 is 53FA_8000h base + 3E0h offset = 53FA_83E0h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT21 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

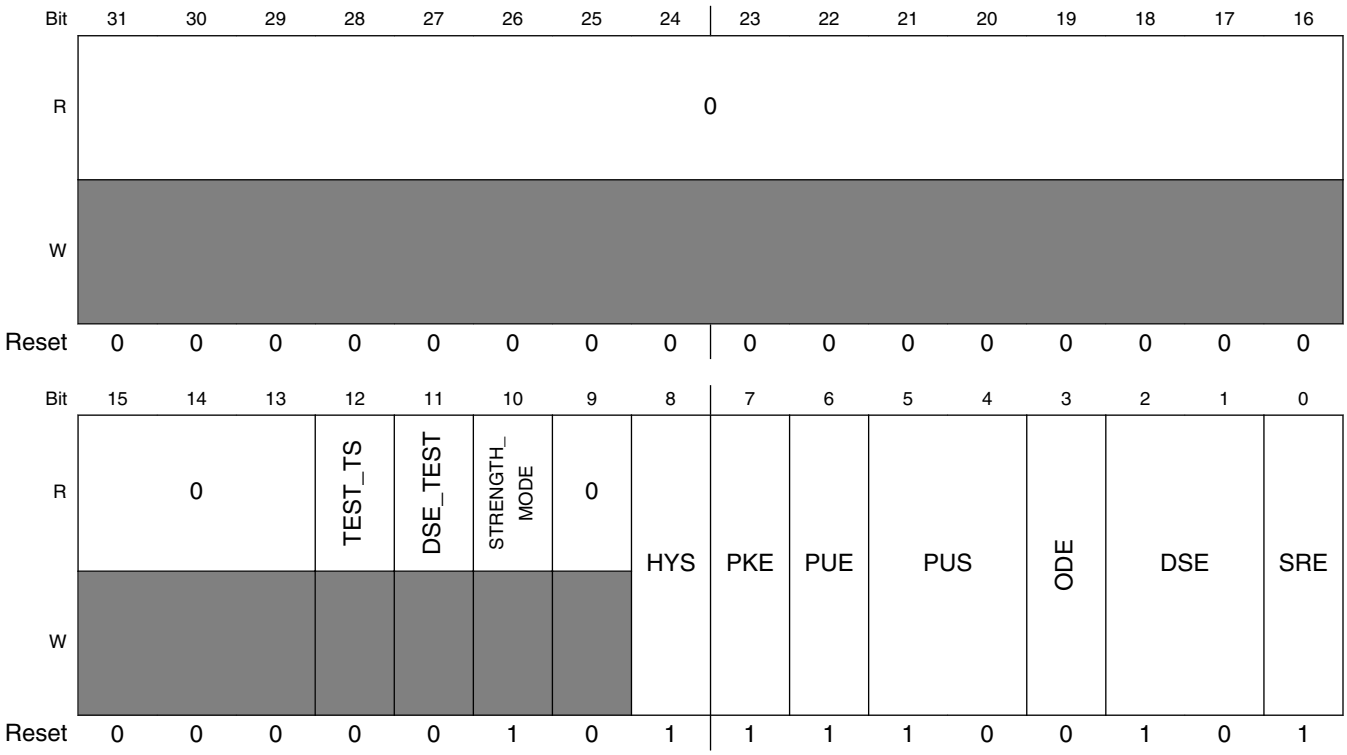
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT21 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT21. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT21. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT21. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT21. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT21. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT21. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT21. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.250 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT22 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT22)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT22 is 53FA_8000h base + 3E4h offset = 53FA_83E4h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT22 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

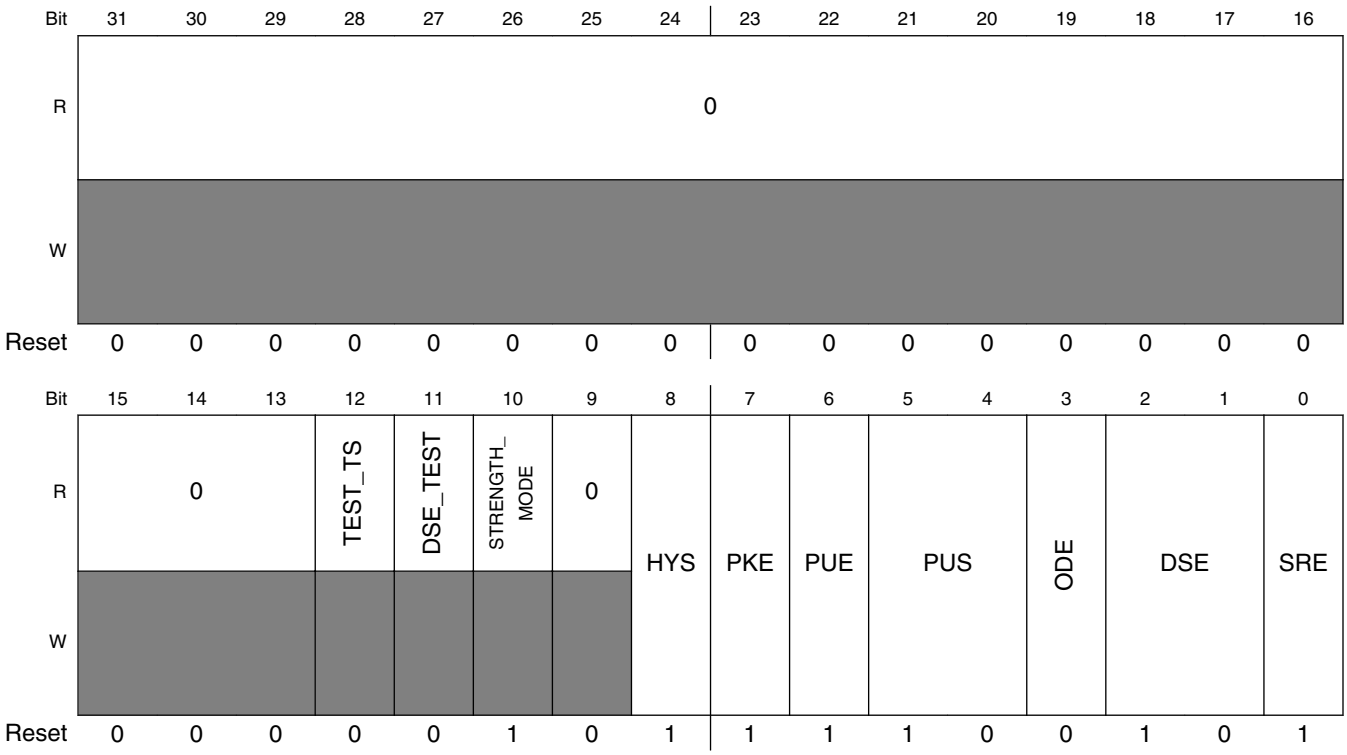
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT22 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT22. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT22. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT22. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT22. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT22. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT22. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT22. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.251 IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT23 (IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT23)

Address: IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT23 is 53FA_8000h base + 3E8h offset = 53FA_83E8h



IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT23 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

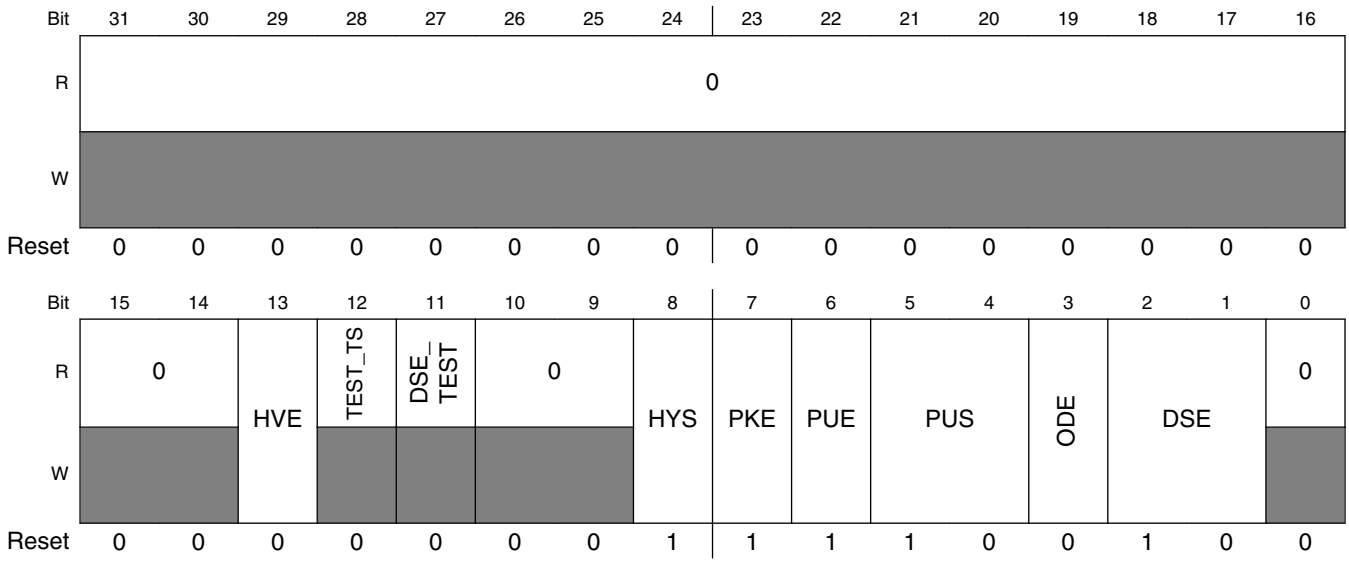
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DISP0_DAT23 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: DISP0_DAT23. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DISP0_DAT23. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DISP0_DAT23. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DISP0_DAT23. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: DISP0_DAT23. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: DISP0_DAT23. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: DISP0_DAT23. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.252 IOMUXC_SW_PAD_CTL_PAD_CSI0_PIXCLK (IOMUXC_SW_PAD_CTL_PAD_CSI0_PIXCLK)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_PIXCLK is 53FA_8000h base + 3ECh offset = 53FA_83ECh



IOMUXC_SW_PAD_CTL_PAD_CSI0_PIXCLK field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_PIXCLK. 0 Hysteresis Disabled 1 Hysteresis Enabled

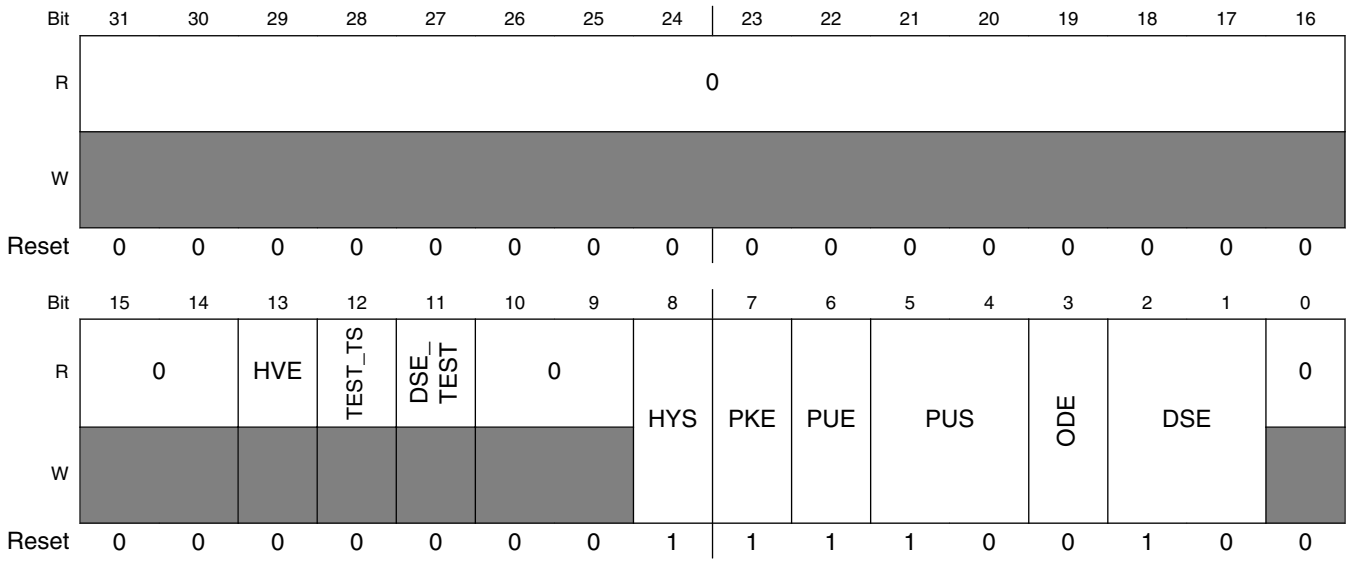
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_PIXCLK field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_PIXCLK. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_PIXCLK. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_PIXCLK. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_PIXCLK. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_PIXCLK. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.253 IOMUXC_SW_PAD_CTL_PAD_CSI0_MCLK (IOMUXC_SW_PAD_CTL_PAD_CSI0_MCLK)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_MCLK is 53FA_8000h base + 3F0h offset = 53FA_83F0h



IOMUXC_SW_PAD_CTL_PAD_CSI0_MCLK field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_MCLK. 0 Hysteresis Disabled 1 Hysteresis Enabled

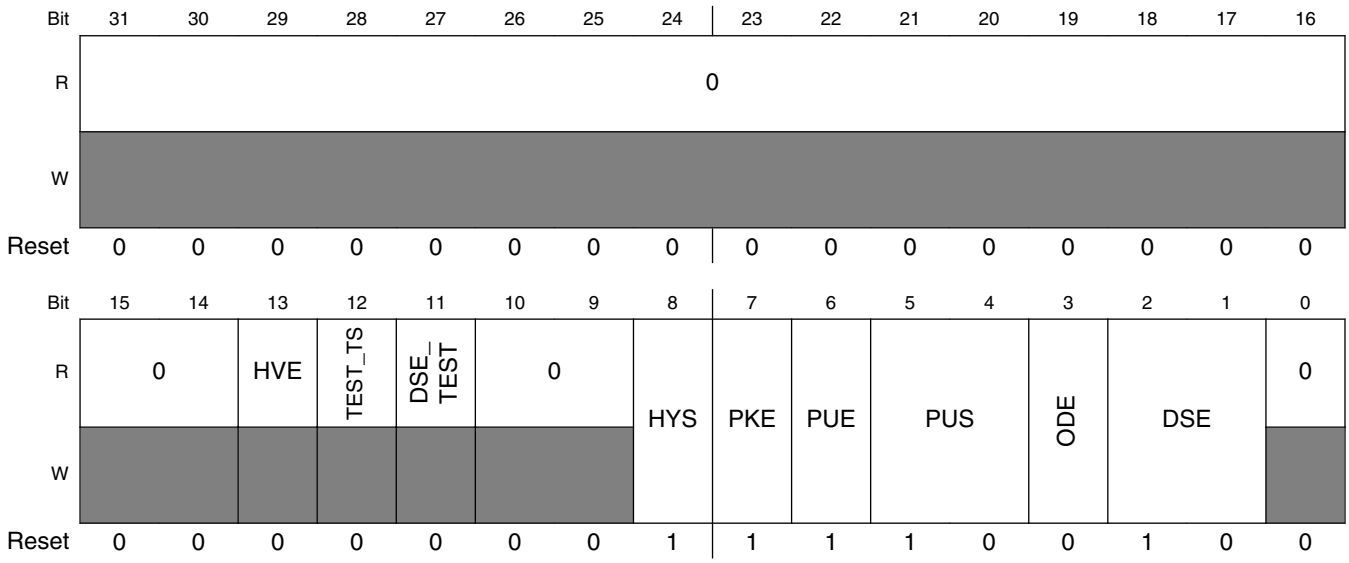
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_MCLK field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_MCLK. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_MCLK. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_MCLK. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_MCLK. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_MCLK. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.254 IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA_EN (IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA_EN)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA_EN is 53FA_8000h base + 3F4h offset = 53FA_83F4h



IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA_EN field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_DATA_EN. 0 Hysteresis Disabled 1 Hysteresis Enabled

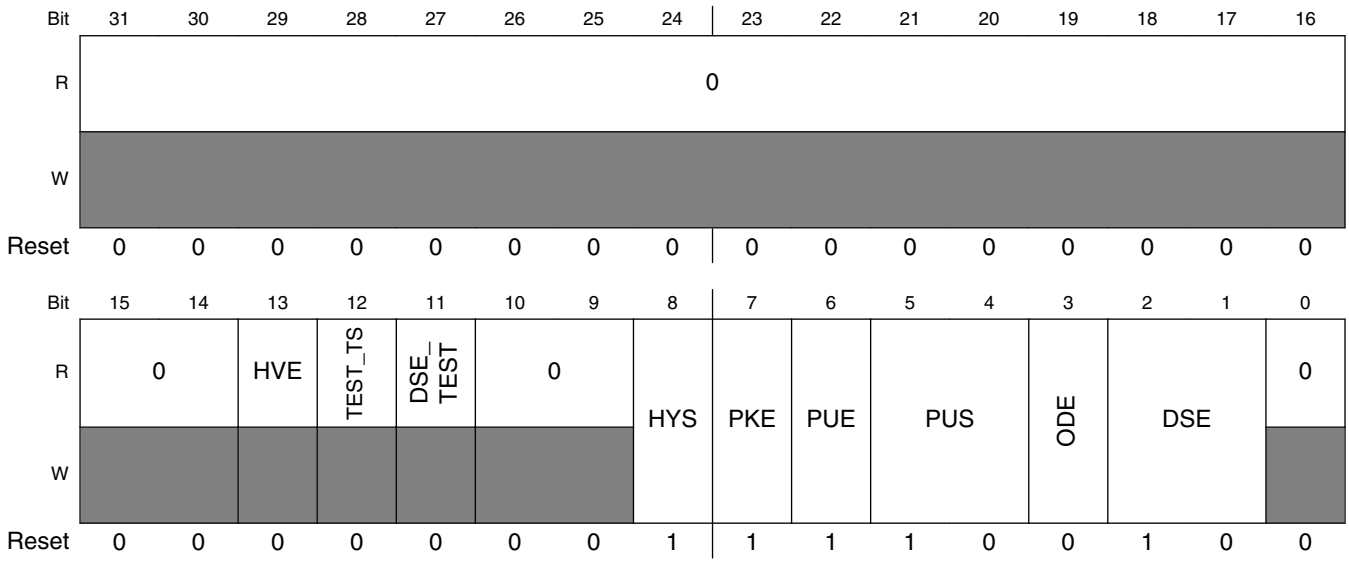
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DATA_EN field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_DATA_EN. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_DATA_EN. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_DATA_EN. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_DATA_EN. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_DATA_EN. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.255 IOMUXC_SW_PAD_CTL_PAD_CSI0_VSYNC (IOMUXC_SW_PAD_CTL_PAD_CSI0_VSYNC)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_VSYNC is 53FA_8000h base + 3F8h offset = 53FA_83F8h



IOMUXC_SW_PAD_CTL_PAD_CSI0_VSYNC field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_VSYNC. 0 Hysteresis Disabled 1 Hysteresis Enabled

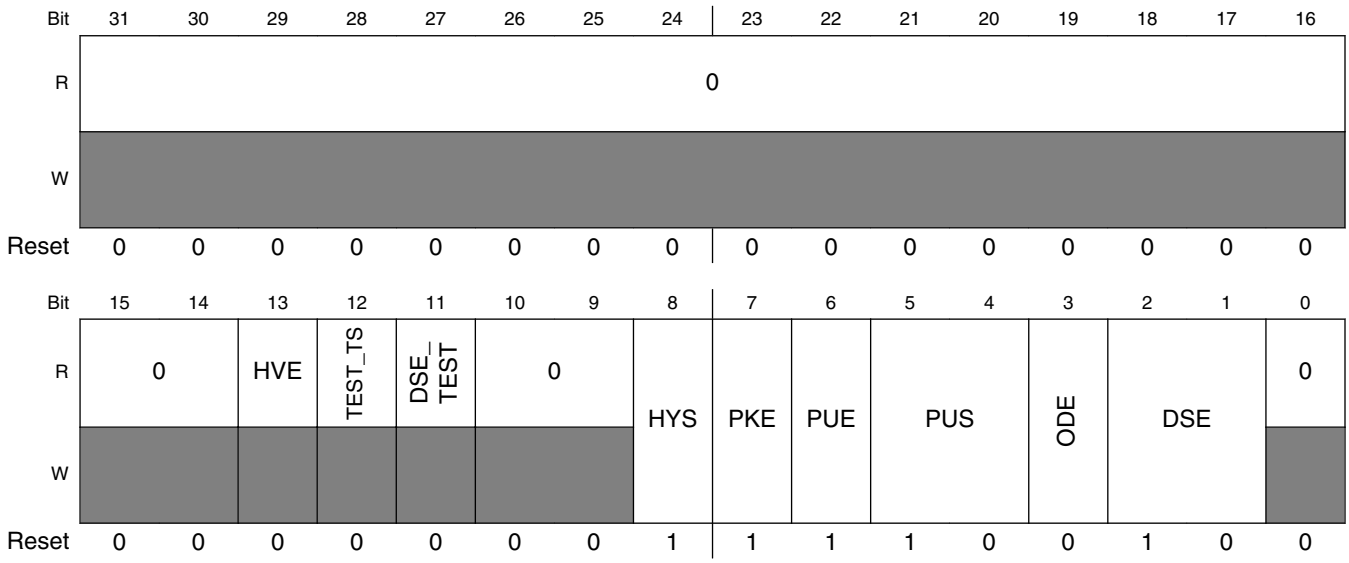
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_VSYNC field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_VSYNC. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_VSYNC. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_VSYNC. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_VSYNC. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_VSYNC. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.256 IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT4 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT4)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT4 is 53FA_8000h base + 3FCh offset = 53FA_83FCh



IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT4 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_DAT4. 0 Hysteresis Disabled 1 Hysteresis Enabled

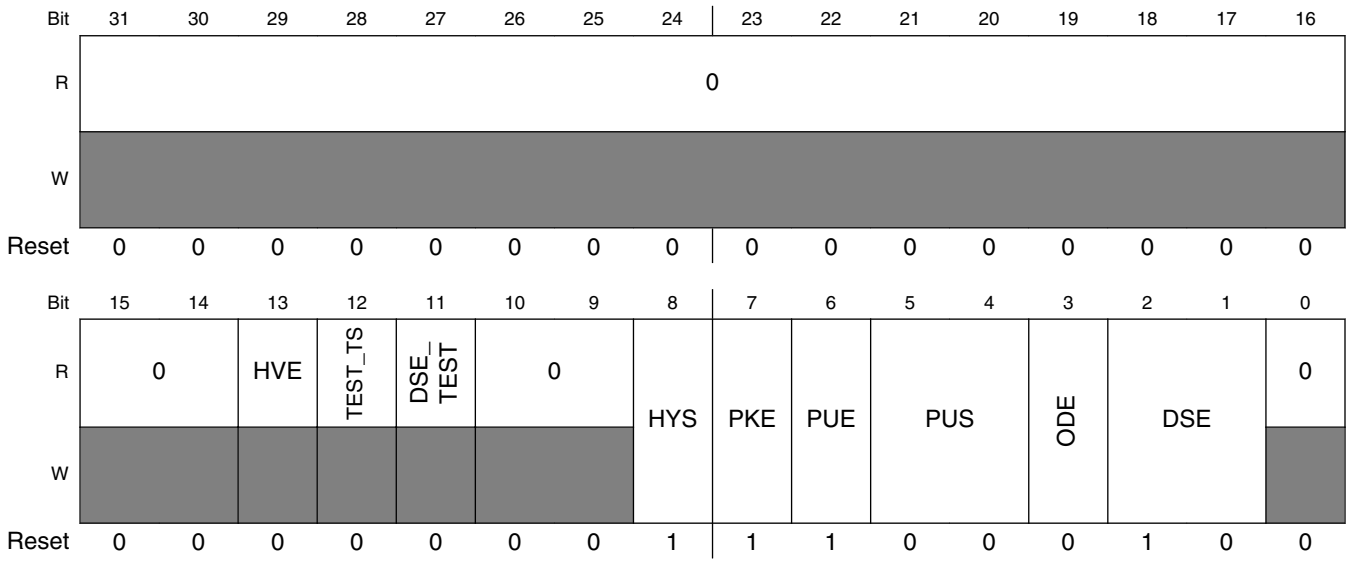
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT4 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_DAT4. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_DAT4. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_DAT4. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_DAT4. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_DAT4. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.257 IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT5 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT5)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT5 is 53FA_8000h base + 400h offset = 53FA_8400h



IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT5 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_DAT5. 0 Hysteresis Disabled 1 Hysteresis Enabled

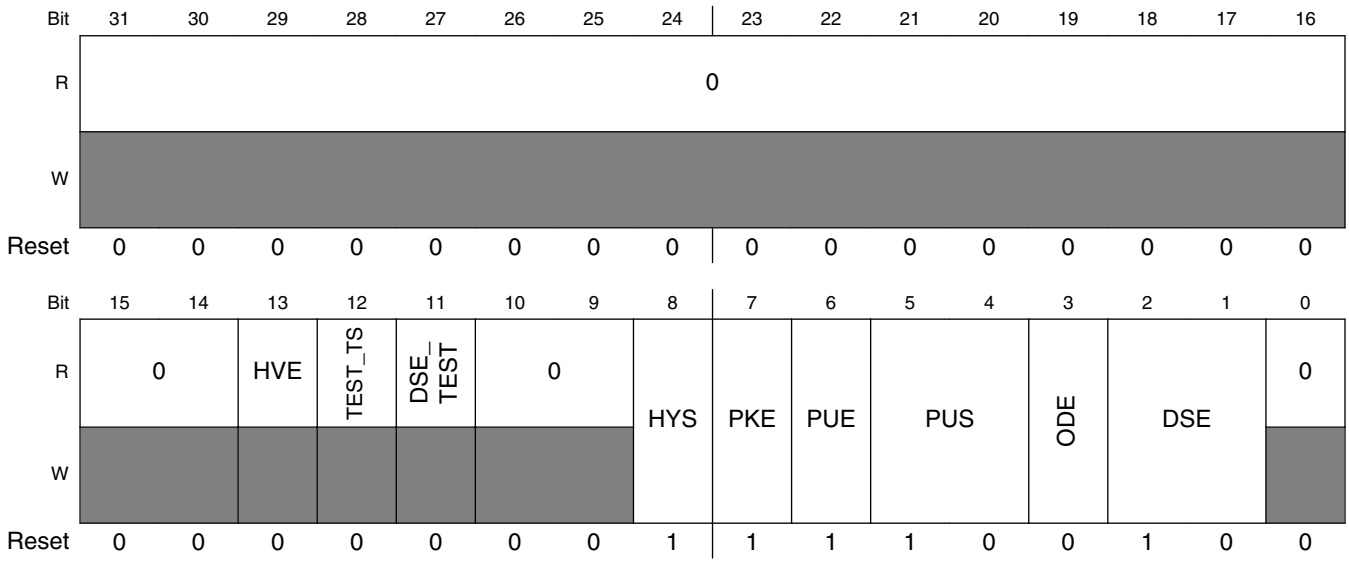
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT5 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_DAT5. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_DAT5. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_DAT5. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_DAT5. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_DAT5. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.258 IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT6 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT6)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT6 is 53FA_8000h base + 404h offset = 53FA_8404h



IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT6 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_DAT6. 0 Hysteresis Disabled 1 Hysteresis Enabled

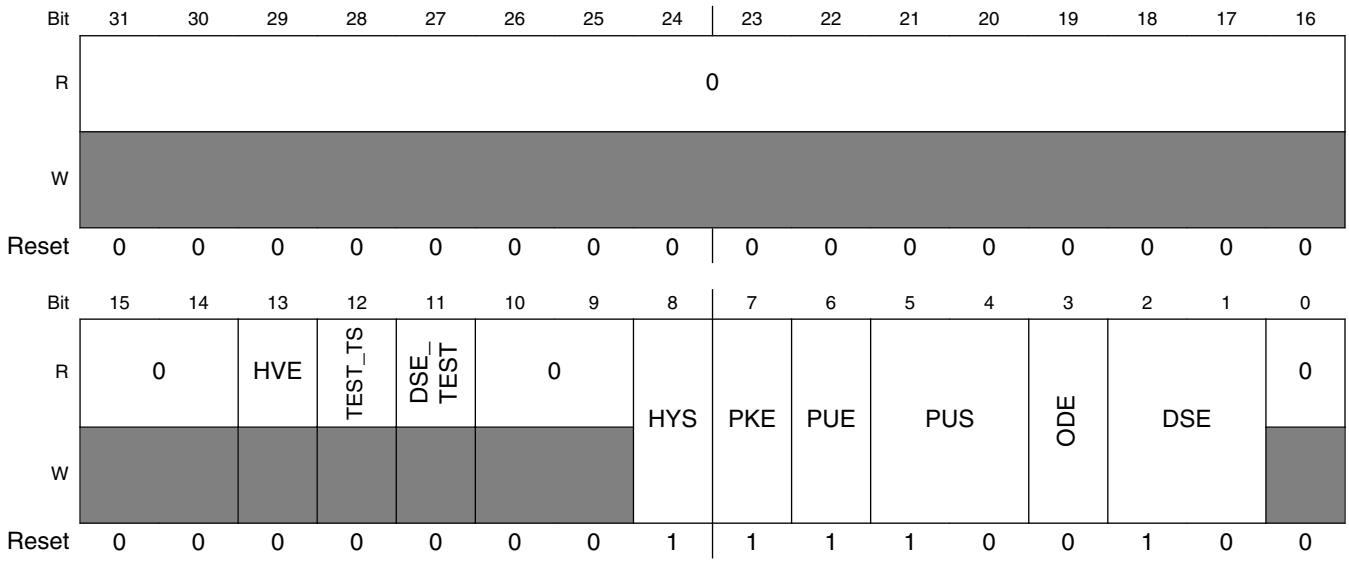
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT6 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_DAT6. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_DAT6. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_DAT6. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_DAT6. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_DAT6. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.259 IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT7 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT7)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT7 is 53FA_8000h base + 408h offset = 53FA_8408h



IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT7 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_DAT7. 0 Hysteresis Disabled 1 Hysteresis Enabled

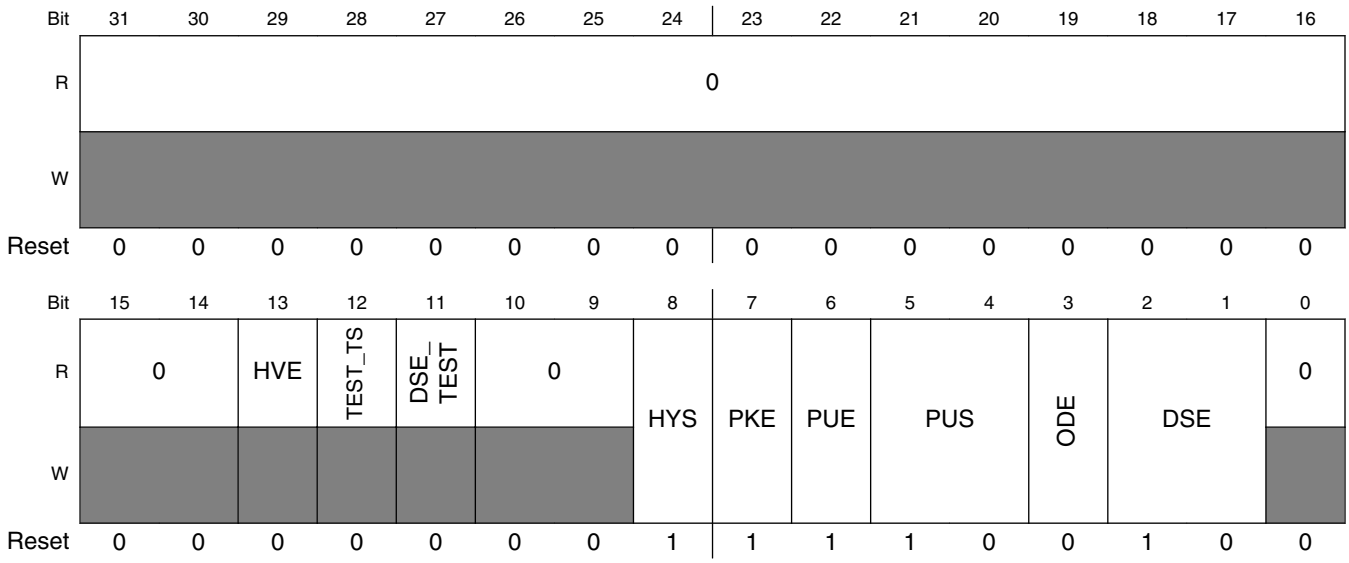
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT7 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_DAT7. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_DAT7. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_DAT7. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_DAT7. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_DAT7. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.260 IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT8 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT8)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT8 is 53FA_8000h base + 40Ch offset = 53FA_840Ch



IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT8 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_DAT8. 0 Hysteresis Disabled 1 Hysteresis Enabled

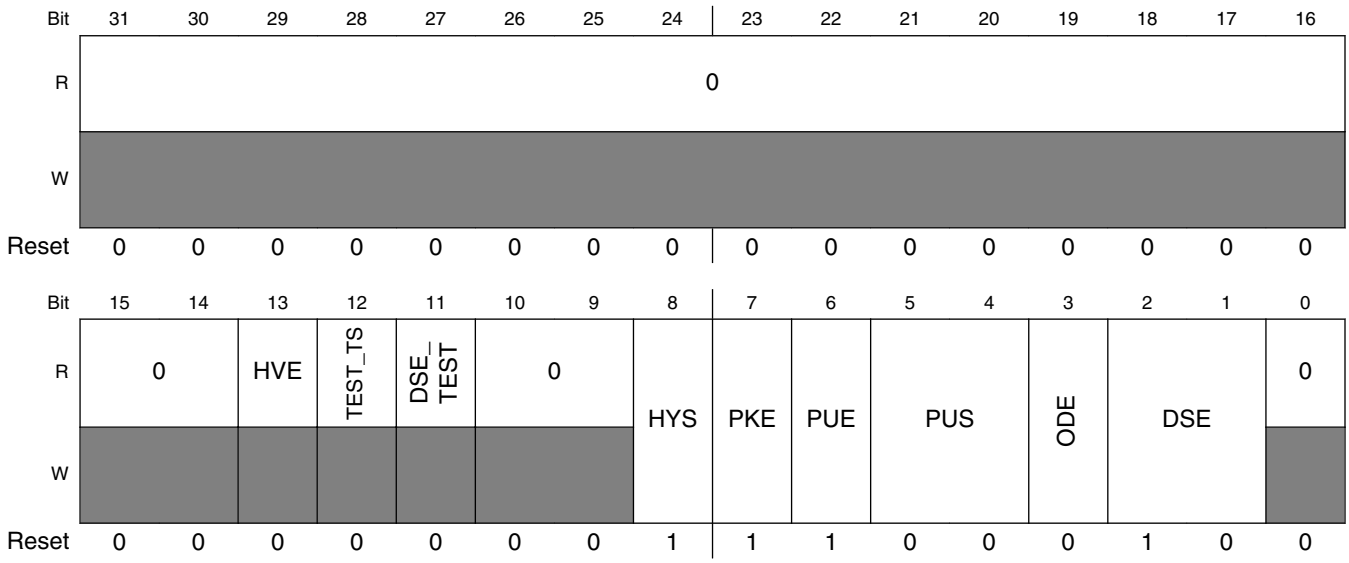
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT8 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_DAT8. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_DAT8. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_DAT8. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_DAT8. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_DAT8. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.261 IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT9 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT9)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT9 is 53FA_8000h base + 410h offset = 53FA_8410h



IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT9 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_DAT9. 0 Hysteresis Disabled 1 Hysteresis Enabled

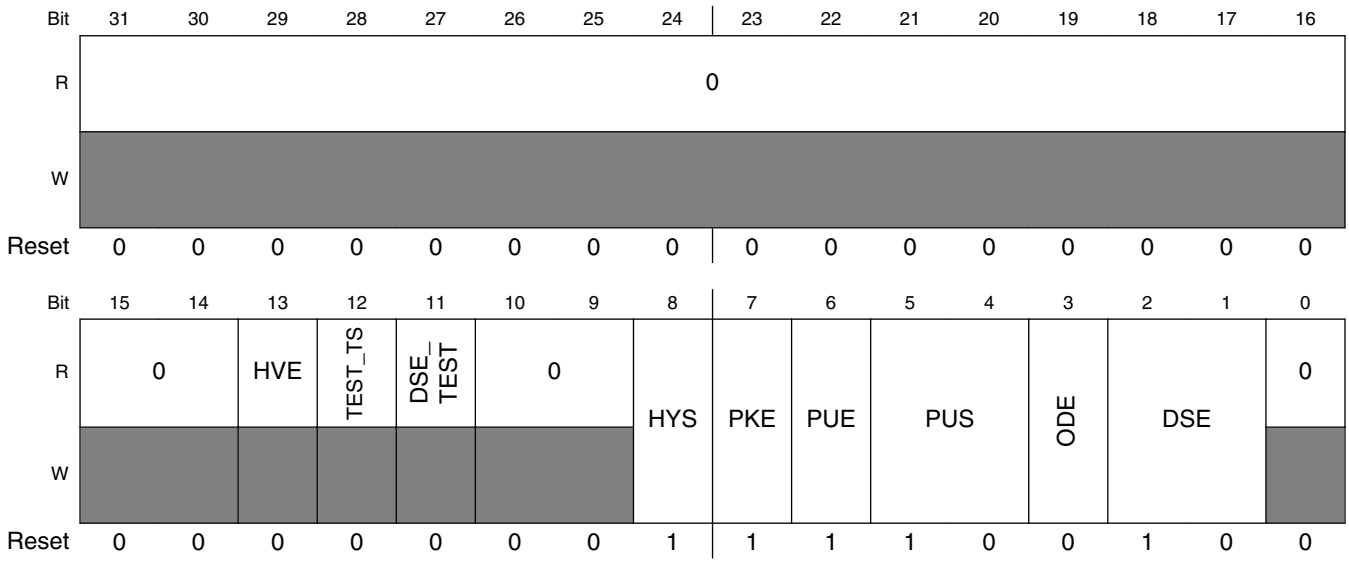
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT9 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_DAT9. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_DAT9. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_DAT9. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_DAT9. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_DAT9. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.262 IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT10 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT10)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT10 is 53FA_8000h base + 414h offset = 53FA_8414h



IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT10 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_DAT10. 0 Hysteresis Disabled 1 Hysteresis Enabled

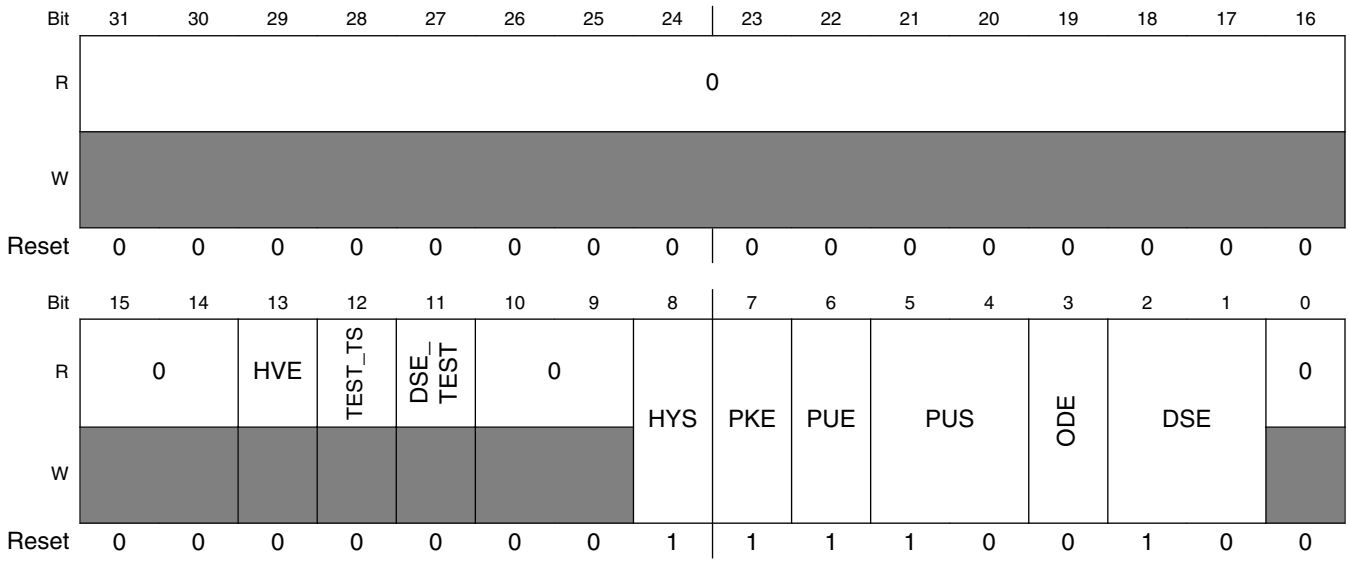
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT10 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_DAT10. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_DAT10. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_DAT10. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_DAT10. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_DAT10. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.263 IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT11 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT11)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT11 is 53FA_8000h base + 418h offset = 53FA_8418h



IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT11 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_DAT11. 0 Hysteresis Disabled 1 Hysteresis Enabled

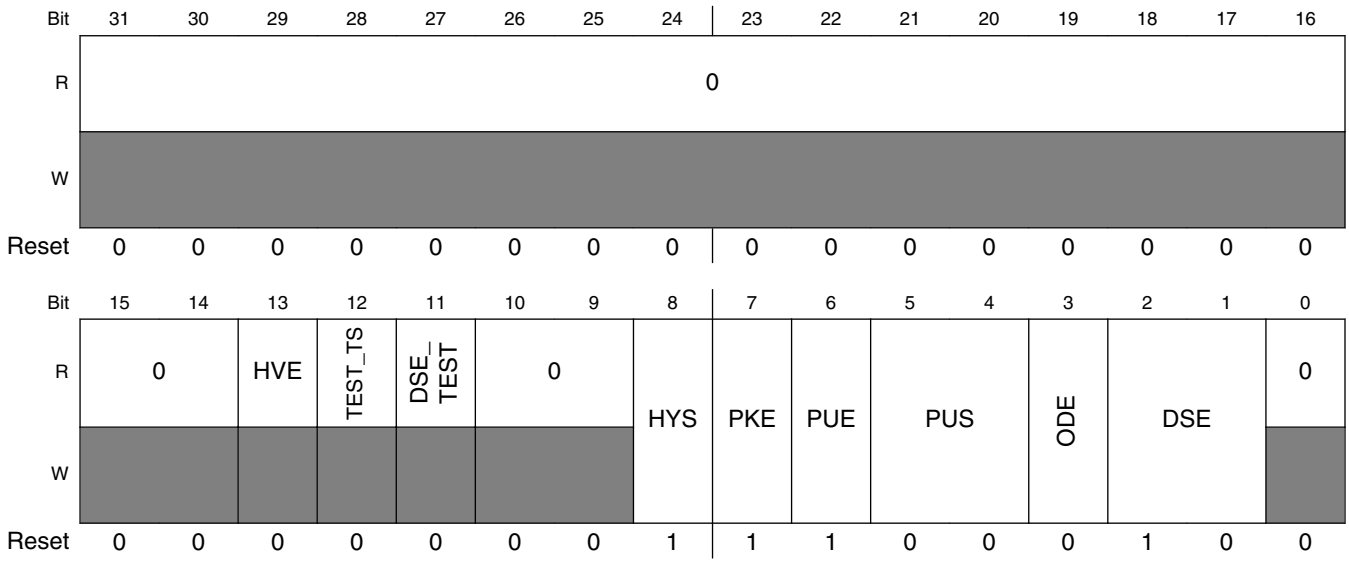
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT11 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_DAT11. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_DAT11. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_DAT11. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_DAT11. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_DAT11. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.264 IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT12 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT12)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT12 is 53FA_8000h base + 41Ch offset = 53FA_841Ch



IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT12 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_DAT12. 0 Hysteresis Disabled 1 Hysteresis Enabled

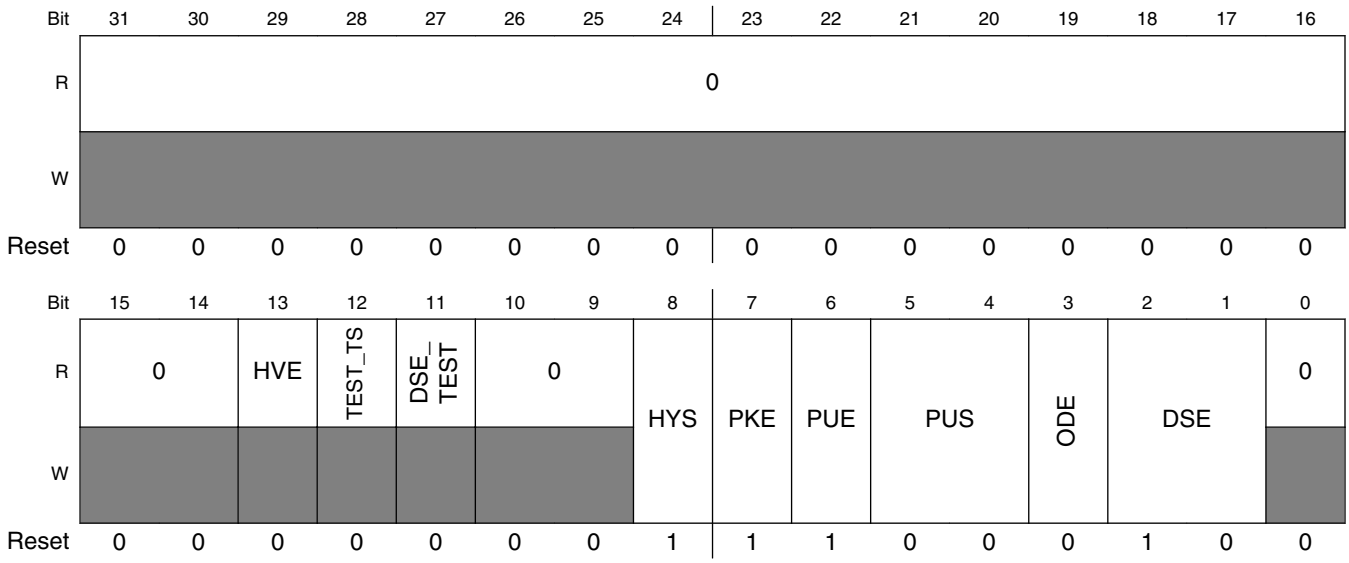
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT12 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_DAT12. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_DAT12. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_DAT12. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_DAT12. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_DAT12. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.265 IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT13 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT13)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT13 is 53FA_8000h base + 420h offset = 53FA_8420h



IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT13 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_DAT13. 0 Hysteresis Disabled 1 Hysteresis Enabled

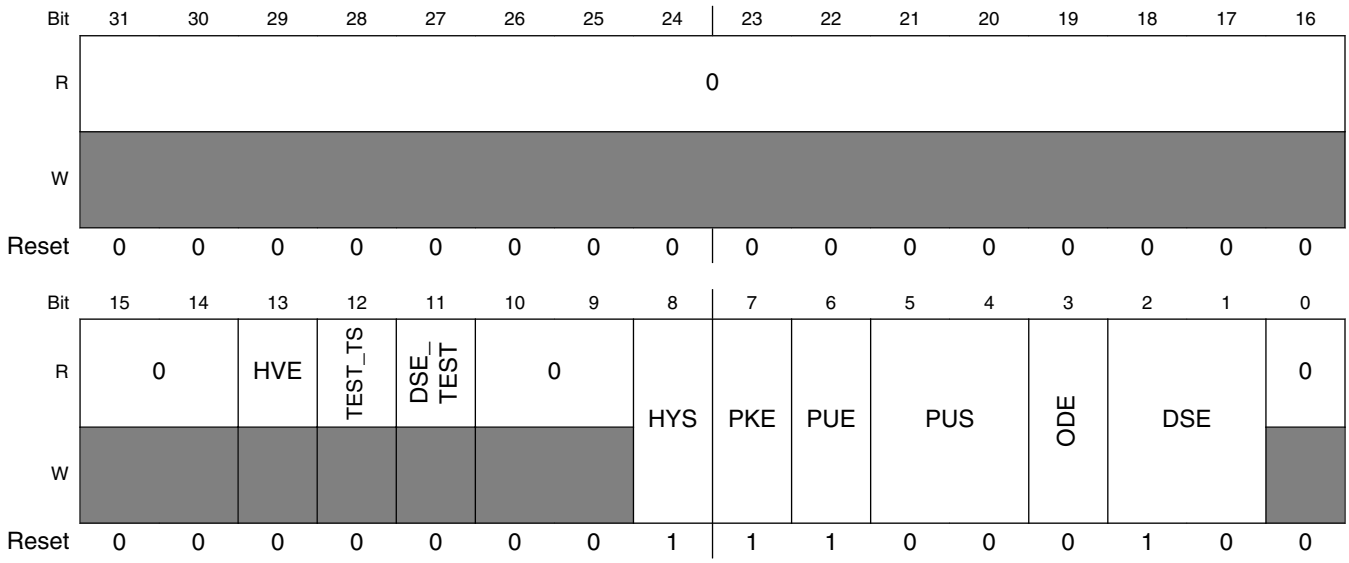
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT13 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_DAT13. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_DAT13. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_DAT13. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_DAT13. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_DAT13. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.266 IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT14 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT14)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT14 is 53FA_8000h base + 424h offset = 53FA_8424h



IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT14 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_DAT14. 0 Hysteresis Disabled 1 Hysteresis Enabled

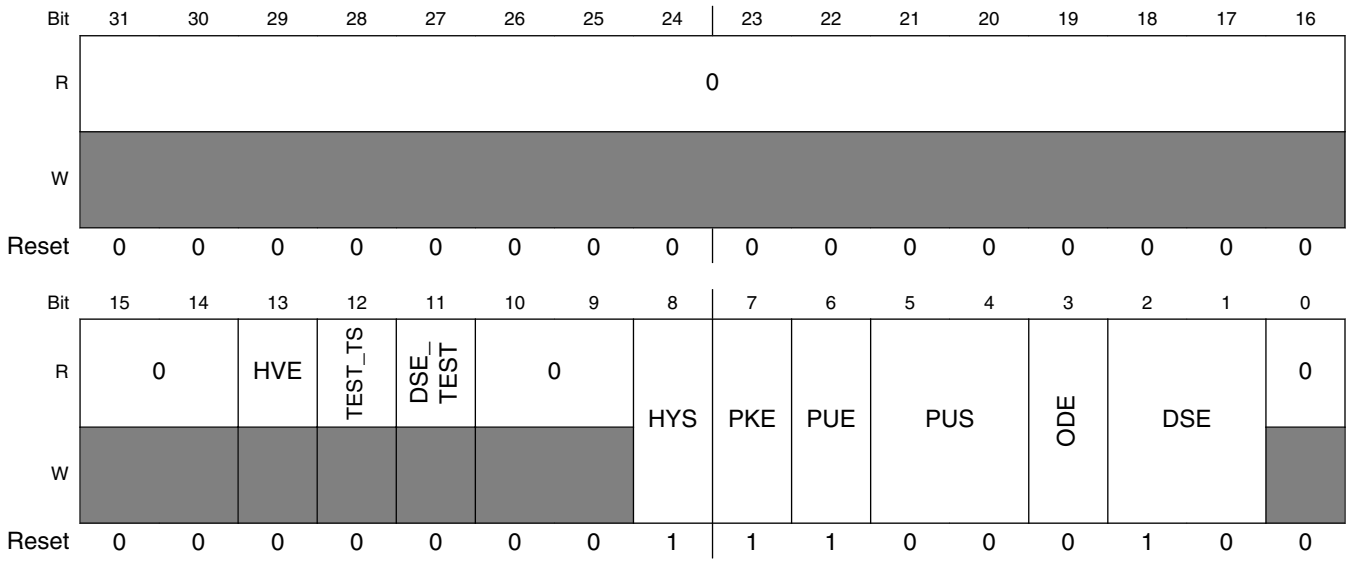
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT14 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_DAT14. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_DAT14. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_DAT14. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_DAT14. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_DAT14. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.267 IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT15 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT15)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT15 is 53FA_8000h base + 428h offset = 53FA_8428h



IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT15 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_DAT15. 0 Hysteresis Disabled 1 Hysteresis Enabled

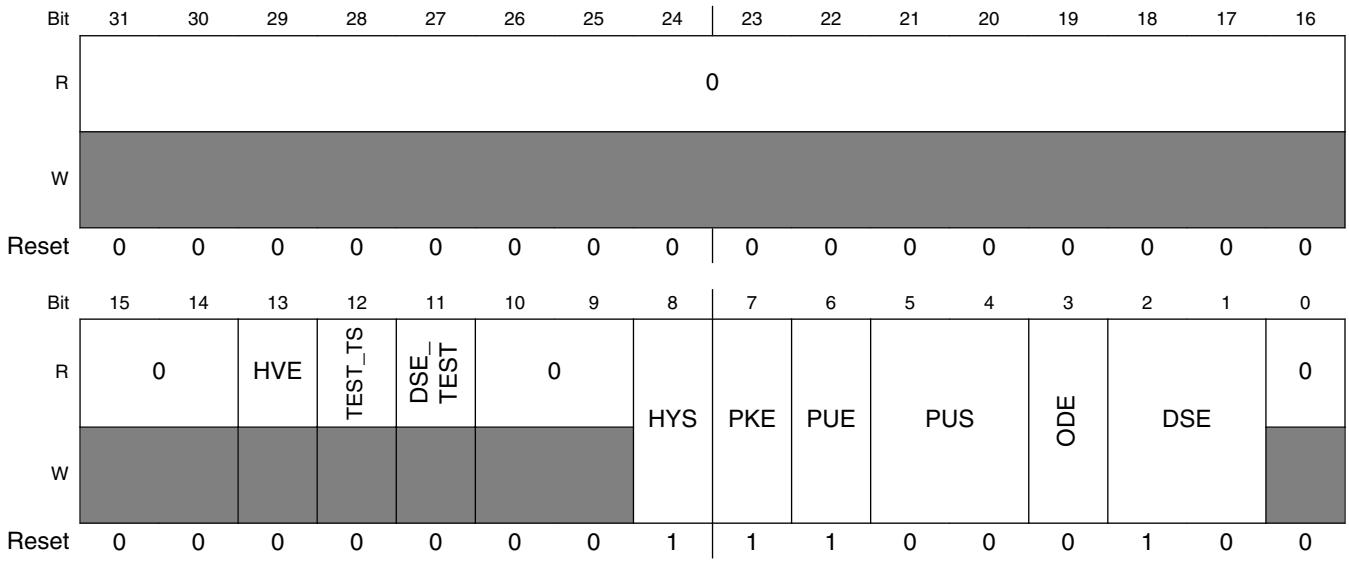
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT15 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_DAT15. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_DAT15. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_DAT15. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_DAT15. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_DAT15. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.268 IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT16 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT16)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT16 is 53FA_8000h base + 42Ch offset = 53FA_842Ch



IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT16 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_DAT16. 0 Hysteresis Disabled 1 Hysteresis Enabled

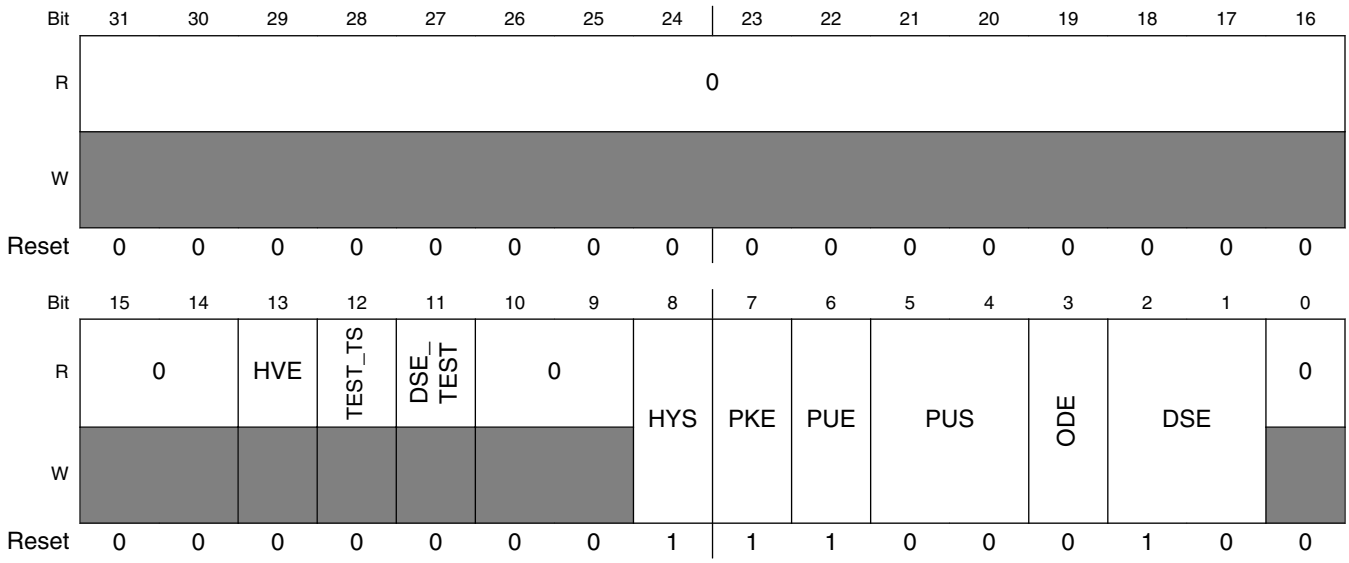
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT16 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_DAT16. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_DAT16. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_DAT16. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_DAT16. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_DAT16. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.269 IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT17 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT17)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT17 is 53FA_8000h base + 430h offset = 53FA_8430h



IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT17 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_DAT17. 0 Hysteresis Disabled 1 Hysteresis Enabled

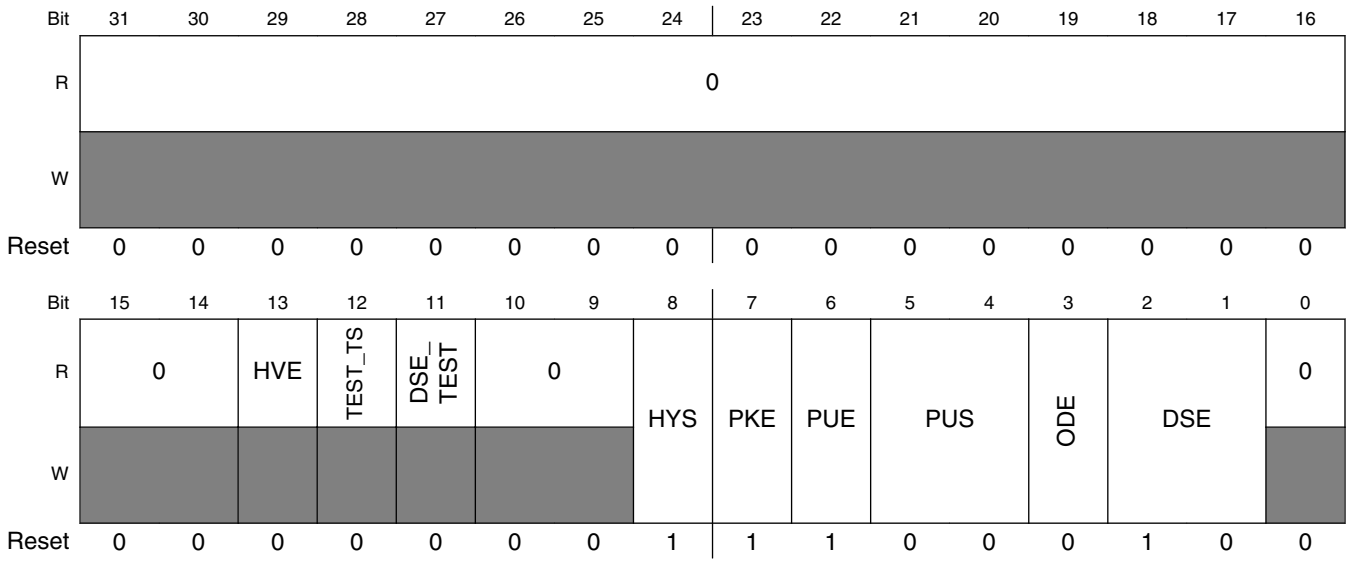
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT17 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_DAT17. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_DAT17. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_DAT17. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_DAT17. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_DAT17. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.270 IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT18 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT18)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT18 is 53FA_8000h base + 434h offset = 53FA_8434h



IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT18 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_DAT18. 0 Hysteresis Disabled 1 Hysteresis Enabled

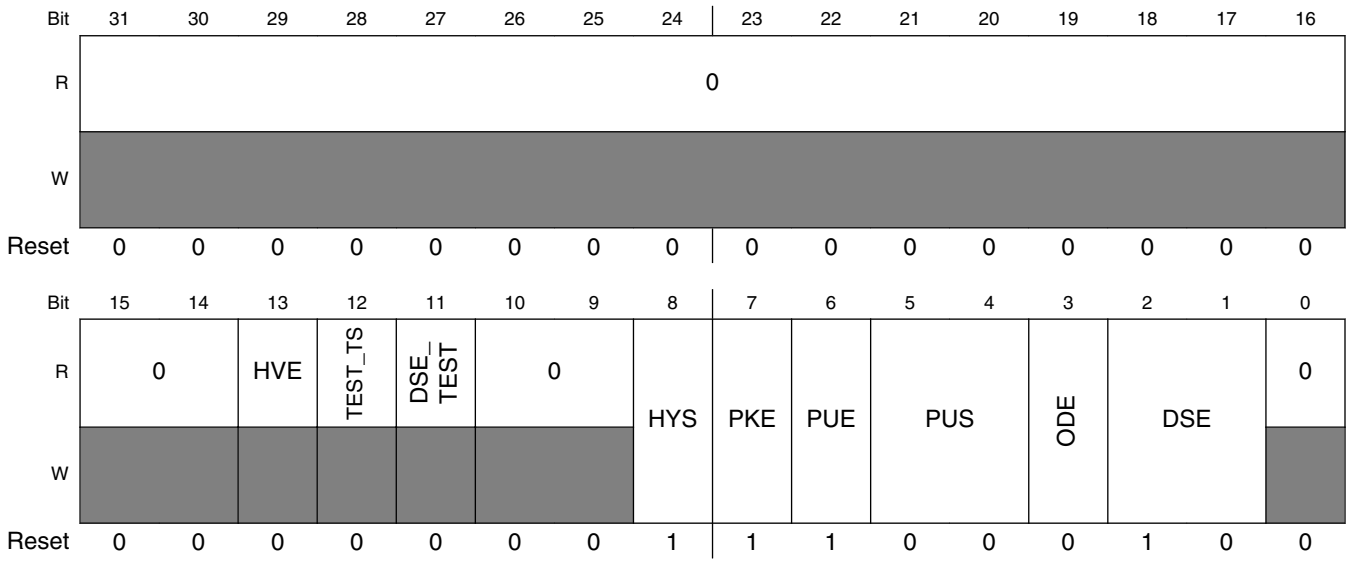
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT18 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_DAT18. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_DAT18. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_DAT18. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_DAT18. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_DAT18. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.271 IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT19 (IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT19)

Address: IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT19 is 53FA_8000h base + 438h offset = 53FA_8438h



IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT19 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: CSI0_DAT19. 0 Hysteresis Disabled 1 Hysteresis Enabled

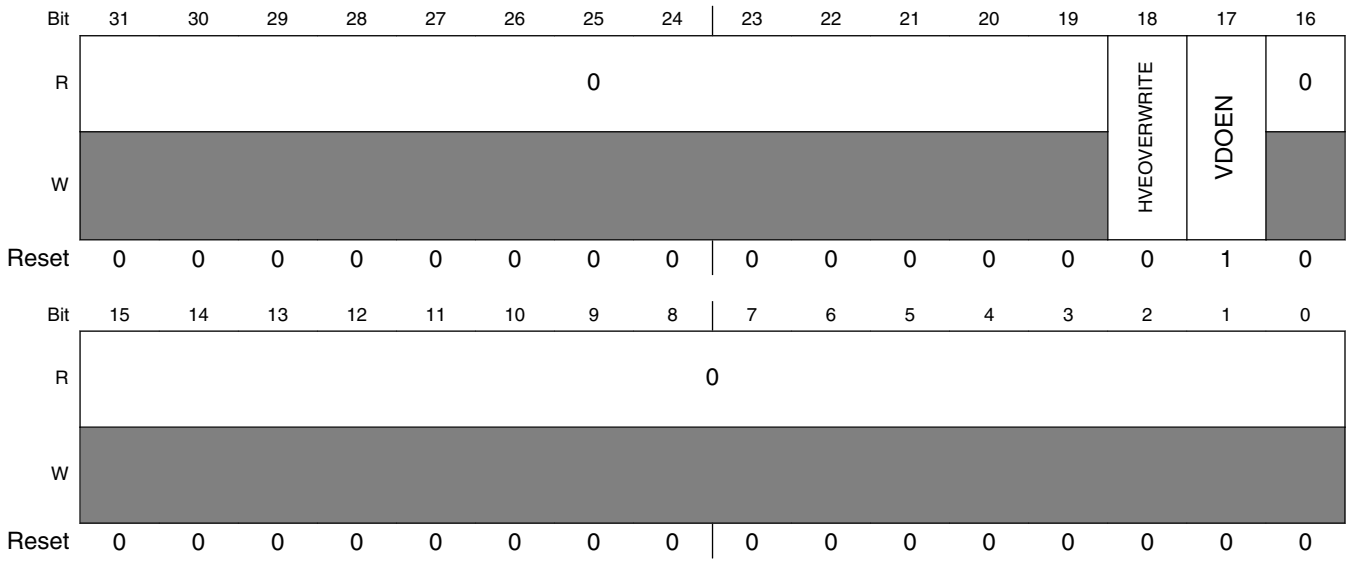
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_CSI0_DAT19 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: CSI0_DAT19. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: CSI0_DAT19. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: CSI0_DAT19. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: CSI0_DAT19. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: CSI0_DAT19. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.272 IOMUXC_SW_PAD_CTL_PAD_NVCC_CSI__0 (IOMUXC_SW_PAD_CTL_PAD_NVCC_CSI__0)

Address: IOMUXC_SW_PAD_CTL_PAD_NVCC_CSI__0 is 53FA_8000h base + 43Ch offset = 53FA_843Ch

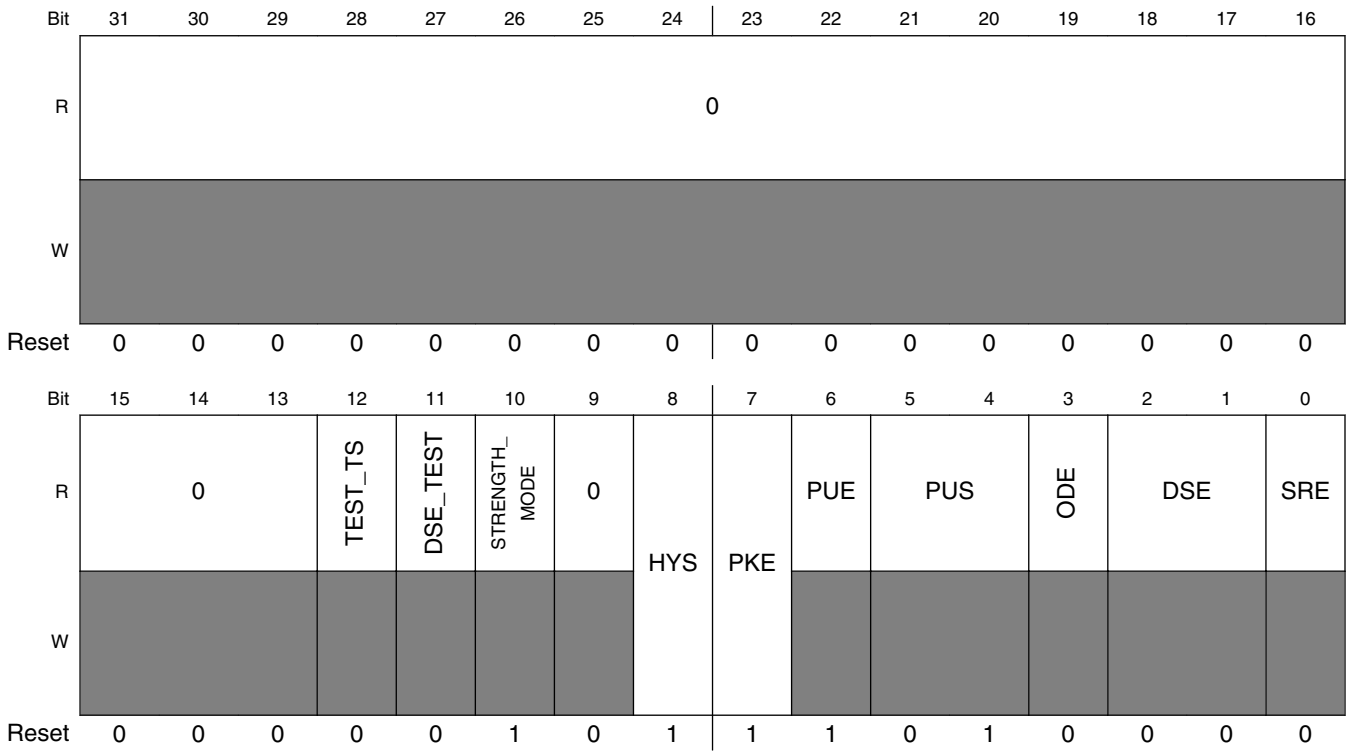


IOMUXC_SW_PAD_CTL_PAD_NVCC_CSI__0 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18 HVEOVERWRITE	HVEOVERWRITE (ipp_owr) Select one out of next values for pad: NVCC_CSI__0. This field affects the voltage levels for the following PADS: CSI0_DAT19, CSI0_DAT18, CSI0_DAT17, CSI0_DAT16, CSI0_DAT15, CSI0_DAT14, CSI0_DAT13, CSI0_DAT12, CSI0_DAT11, CSI0_DAT10, CSI0_DAT9, CSI0_DAT8, CSI0_DAT7, CSI0_DAT6, CSI0_DAT5, CSI0_DAT4, CSI0_VSYNC, CSI0_DATA_EN, CSI0_MCLK, CSI0_PIXCLK 0 PAD support high voltage (3.0V-3.6V). Out of reset value is 0. The HVEOVERWRITE field should be updated before automatic voltage detector is disabled. 1 PAD support low voltage (1.65V-3.1V)
17 VDOEN	VDOEN (ipp_oen) Select one out of next values for pad: NVCC_CSI__0. 0 Voltage detector output disable and HVEOVERWRITE field will be used. It is recommended to disable the automatic voltage detector after boot and update the HVEOVERWRITE field with the actual I/O supply value. 1 Voltage detector output enable. (default 1) **NOTE:**
16–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.273 IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS (IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS)

Address: IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS is 53FA_8000h base + 440h offset = 53FA_8440h



IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: JTAG_TMS. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: JTAG_TMS. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Read Only Field 1 Pull
5–4 PUS	Pull Up / Down Config. Field Read Only Field 01 47 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Read Only Field 0 Open Drain Disabled
2–1 DSE	Drive Strength Field Read Only Field 00 Low Drive Strength
0 SRE	Slew Rate Field Read Only Field 0 Slow Slew Rate

43.3.274 IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD (IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD)

Address: IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD is 53FA_8000h base + 444h offset = 53FA_8444h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	DSE_TEST	STRENGTH_MODE	0	HYS	PKE	PUE	PUS		ODE	DSE		SRE
W																
Reset	0	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: JTAG_MOD. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: JTAG_MOD. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Read Only Field 1 Pull
5–4 PUS	Pull Up / Down Config. Field Read Only Field 10 100 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Read Only Field 0 Open Drain Disabled
2–1 DSE	Drive Strength Field Read Only Field 00 Low Drive Strength
0 SRE	Slew Rate Field Read Only Field 0 Slow Slew Rate

43.3.275 IOMUXC_SW_PAD_CTL_PAD_JTAG_TRSTB (IOMUXC_SW_PAD_CTL_PAD_JTAG_TRSTB)

Address: IOMUXC_SW_PAD_CTL_PAD_JTAG_TRSTB is 53FA_8000h base + 448h offset = 53FA_8448h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	DSE_TEST	STRENGTH_MODE	0	HYS	PKE	PUE	PUS		ODE	DSE		SRE
W																
Reset	0	0	0	0	0	1	0	0	1	1	0	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_JTAG_TRSTB field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_JTAG_TRSTB field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: JTAG_TRSTB. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: JTAG_TRSTB. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Read Only Field 1 Pull
5–4 PUS	Pull Up / Down Config. Field Read Only Field 01 47 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Read Only Field 0 Open Drain Disabled
2–1 DSE	Drive Strength Field Read Only Field 00 Low Drive Strength
0 SRE	Slew Rate Field Read Only Field 0 Slow Slew Rate

43.3.276 IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI)

Address: IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI is 53FA_8000h base + 44Ch offset = 53FA_844Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	DSE_TEST	STRENGTH_MODE	0	HYS	PKE	PUE	PUS		ODE	DSE		SRE
W																
Reset	0	0	0	0	0	1	0	1	1	1	0	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: JTAG_TDI. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: JTAG_TDI. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Read Only Field 1 Pull
5–4 PUS	Pull Up / Down Config. Field Read Only Field 01 47 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Read Only Field 0 Open Drain Disabled
2–1 DSE	Drive Strength Field Read Only Field 00 Low Drive Strength
0 SRE	Slew Rate Field Read Only Field 0 Slow Slew Rate

43.3.277 IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK (IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK)

Address: IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK is 53FA_8000h base + 450h offset = 53FA_8450h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	DSE_TEST	STRENGTH_MODE	0	HYS	PKE	PUE	PUS		ODE	DSE		SRE
W																
Reset	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

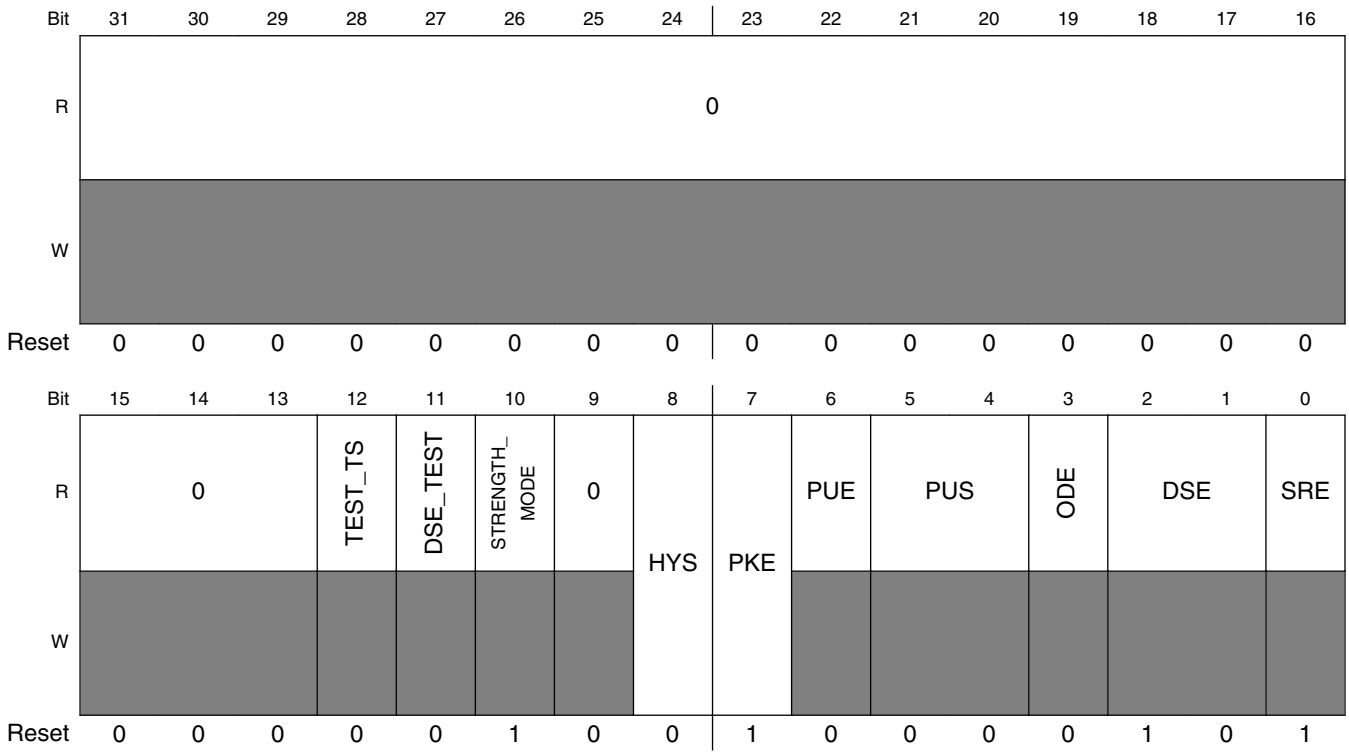
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: JTAG_TCK. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: JTAG_TCK. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Read Only Field 1 Pull
5–4 PUS	Pull Up / Down Config. Field Read Only Field 00 100 [KOhm] Pull Down
3 ODE	Open Drain Enable Field Read Only Field 0 Open Drain Disabled
2–1 DSE	Drive Strength Field Read Only Field 00 Low Drive Strength
0 SRE	Slew Rate Field Read Only Field 0 Slow Slew Rate

43.3.278 IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO)

Address: IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO is 53FA_8000h base + 454h offset = 53FA_8454h



IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: JTAG_TDO. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: JTAG_TDO. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Read Only Field 0 Keeper
5–4 PUS	Pull Up / Down Config. Field Read Only Field 00 100 [KOhm] Pull Down
3 ODE	Open Drain Enable Field Read Only Field 0 Open Drain Disabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: JTAG_TDO. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: JTAG_TDO. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.279 IOMUXC_SW_PAD_CTL_PAD_EIM_A25 (IOMUXC_SW_PAD_CTL_PAD_EIM_A25)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_A25 is 53FA_8000h base + 458h offset = 53FA_8458h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_A25 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_A25. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_A25 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_A25. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_A25. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_A25. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_A25. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_A25. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.280 IOMUXC_SW_PAD_CTL_PAD_EIM_EB2 (IOMUXC_SW_PAD_CTL_PAD_EIM_EB2)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_EB2 is 53FA_8000h base + 45Ch offset = 53FA_845Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_EB2 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_EB2. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_EB2 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_EB2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_EB2. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_EB2. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_EB2. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_EB2. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.281 IOMUXC_SW_PAD_CTL_PAD_EIM_D16 (IOMUXC_SW_PAD_CTL_PAD_EIM_D16)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_D16 is 53FA_8000h base + 460h offset = 53FA_8460h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_D16 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_D16. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_D16 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_D16. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_D16. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_D16. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_D16. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_D16. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.282 IOMUXC_SW_PAD_CTL_PAD_EIM_D17 (IOMUXC_SW_PAD_CTL_PAD_EIM_D17)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_D17 is 53FA_8000h base + 464h offset = 53FA_8464h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_D17 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_D17. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_D17 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_D17. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_D17. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_D17. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_D17. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_D17. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.283 IOMUXC_SW_PAD_CTL_PAD_EIM_D18 (IOMUXC_SW_PAD_CTL_PAD_EIM_D18)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_D18 is 53FA_8000h base + 468h offset = 53FA_8468h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_D18 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_D18. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_D18 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_D18. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_D18. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_D18. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_D18. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_D18. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.284 IOMUXC_SW_PAD_CTL_PAD_EIM_D19 (IOMUXC_SW_PAD_CTL_PAD_EIM_D19)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_D19 is 53FA_8000h base + 46Ch offset = 53FA_846Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_D19 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_D19. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_D19 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_D19. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_D19. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_D19. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_D19. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_D19. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.285 IOMUXC_SW_PAD_CTL_PAD_EIM_D20 (IOMUXC_SW_PAD_CTL_PAD_EIM_D20)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_D20 is 53FA_8000h base + 470h offset = 53FA_8470h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_D20 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_D20. 0 Hysteresis Disabled 1 Hysteresis Enabled

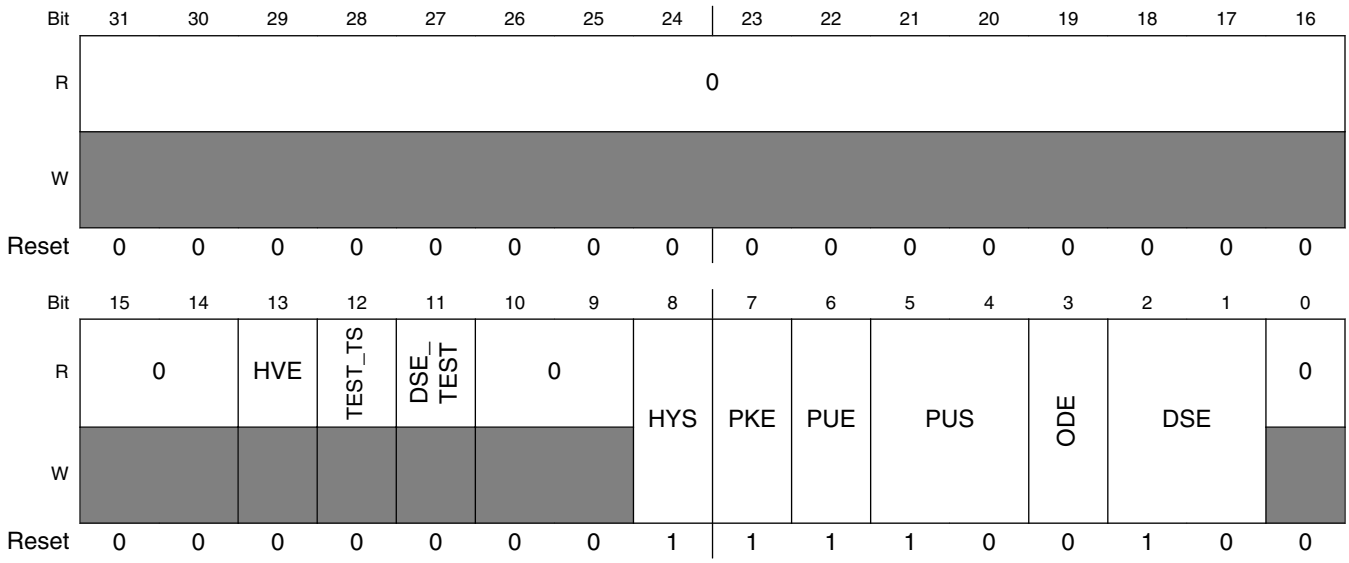
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_D20 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_D20. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_D20. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_D20. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_D20. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_D20. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.286 IOMUXC_SW_PAD_CTL_PAD_EIM_D21 (IOMUXC_SW_PAD_CTL_PAD_EIM_D21)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_D21 is 53FA_8000h base + 474h offset = 53FA_8474h



IOMUXC_SW_PAD_CTL_PAD_EIM_D21 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_D21. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_D21 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_D21. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_D21. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_D21. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_D21. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_D21. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.287 IOMUXC_SW_PAD_CTL_PAD_EIM_D22 (IOMUXC_SW_PAD_CTL_PAD_EIM_D22)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_D22 is 53FA_8000h base + 478h offset = 53FA_8478h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_D22 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_D22. 0 Hysteresis Disabled 1 Hysteresis Enabled

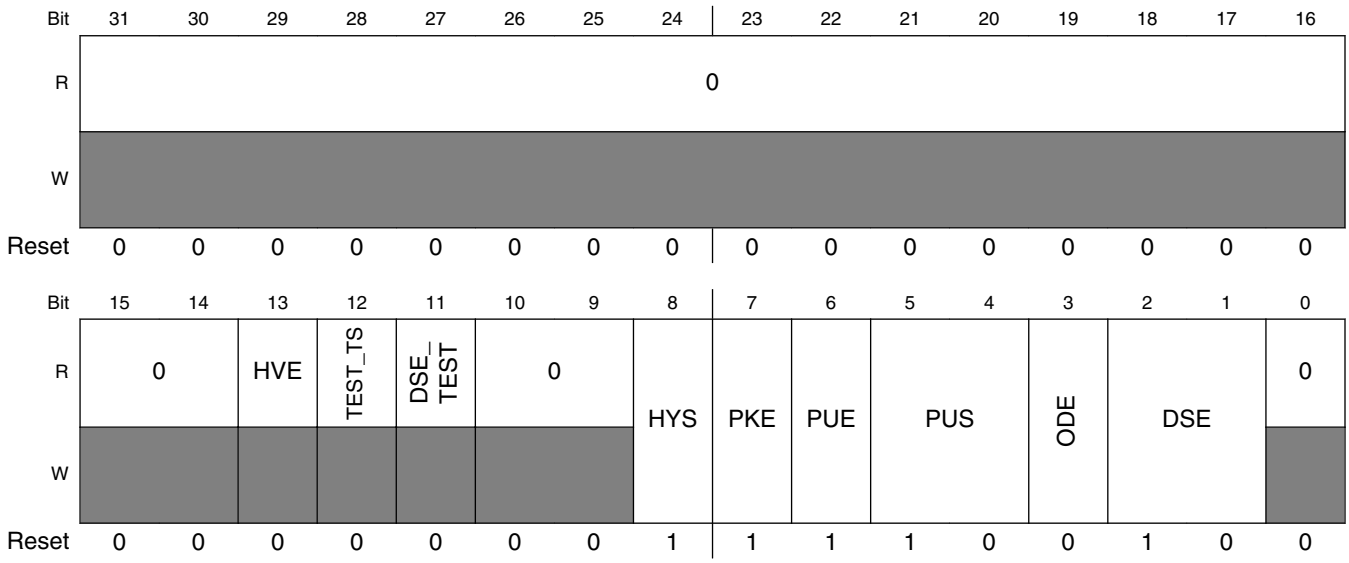
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_D22 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_D22. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_D22. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_D22. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_D22. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_D22. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.288 IOMUXC_SW_PAD_CTL_PAD_EIM_D23 (IOMUXC_SW_PAD_CTL_PAD_EIM_D23)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_D23 is 53FA_8000h base + 47Ch offset = 53FA_847Ch



IOMUXC_SW_PAD_CTL_PAD_EIM_D23 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_D23. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_D23 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_D23. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_D23. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_D23. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_D23. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_D23. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.289 IOMUXC_SW_PAD_CTL_PAD_EIM_EB3 (IOMUXC_SW_PAD_CTL_PAD_EIM_EB3)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_EB3 is 53FA_8000h base + 480h offset = 53FA_8480h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_EB3 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_EB3. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_EB3 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_EB3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_EB3. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_EB3. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_EB3. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_EB3. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.290 IOMUXC_SW_PAD_CTL_PAD_EIM_D24 (IOMUXC_SW_PAD_CTL_PAD_EIM_D24)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_D24 is 53FA_8000h base + 484h offset = 53FA_8484h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_D24 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_D24. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_D24 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_D24. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_D24. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_D24. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_D24. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_D24. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.291 IOMUXC_SW_PAD_CTL_PAD_EIM_D25 (IOMUXC_SW_PAD_CTL_PAD_EIM_D25)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_D25 is 53FA_8000h base + 488h offset = 53FA_8488h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_D25 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_D25. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_D25 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_D25. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_D25. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_D25. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_D25. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_D25. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.292 IOMUXC_SW_PAD_CTL_PAD_EIM_D26 (IOMUXC_SW_PAD_CTL_PAD_EIM_D26)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_D26 is 53FA_8000h base + 48Ch offset = 53FA_848Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_D26 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_D26. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_D26 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_D26. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_D26. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_D26. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_D26. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_D26. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.293 IOMUXC_SW_PAD_CTL_PAD_EIM_D27 (IOMUXC_SW_PAD_CTL_PAD_EIM_D27)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_D27 is 53FA_8000h base + 490h offset = 53FA_8490h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_D27 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_D27. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_D27 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_D27. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_D27. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_D27. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_D27. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_D27. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.294 IOMUXC_SW_PAD_CTL_PAD_EIM_D28 (IOMUXC_SW_PAD_CTL_PAD_EIM_D28)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_D28 is 53FA_8000h base + 494h offset = 53FA_8494h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_D28 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_D28. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_D28 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_D28. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_D28. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_D28. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_D28. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_D28. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.295 IOMUXC_SW_PAD_CTL_PAD_EIM_D29 (IOMUXC_SW_PAD_CTL_PAD_EIM_D29)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_D29 is 53FA_8000h base + 498h offset = 53FA_8498h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_D29 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_D29. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_D29 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_D29. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_D29. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_D29. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_D29. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_D29. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.296 IOMUXC_SW_PAD_CTL_PAD_EIM_D30 (IOMUXC_SW_PAD_CTL_PAD_EIM_D30)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_D30 is 53FA_8000h base + 49Ch offset = 53FA_849Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_D30 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_D30. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_D30 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_D30. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_D30. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_D30. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_D30. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_D30. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.297 IOMUXC_SW_PAD_CTL_PAD_EIM_D31 (IOMUXC_SW_PAD_CTL_PAD_EIM_D31)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_D31 is 53FA_8000h base + 4A0h offset = 53FA_84A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_D31 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_D31. 0 Hysteresis Disabled 1 Hysteresis Enabled

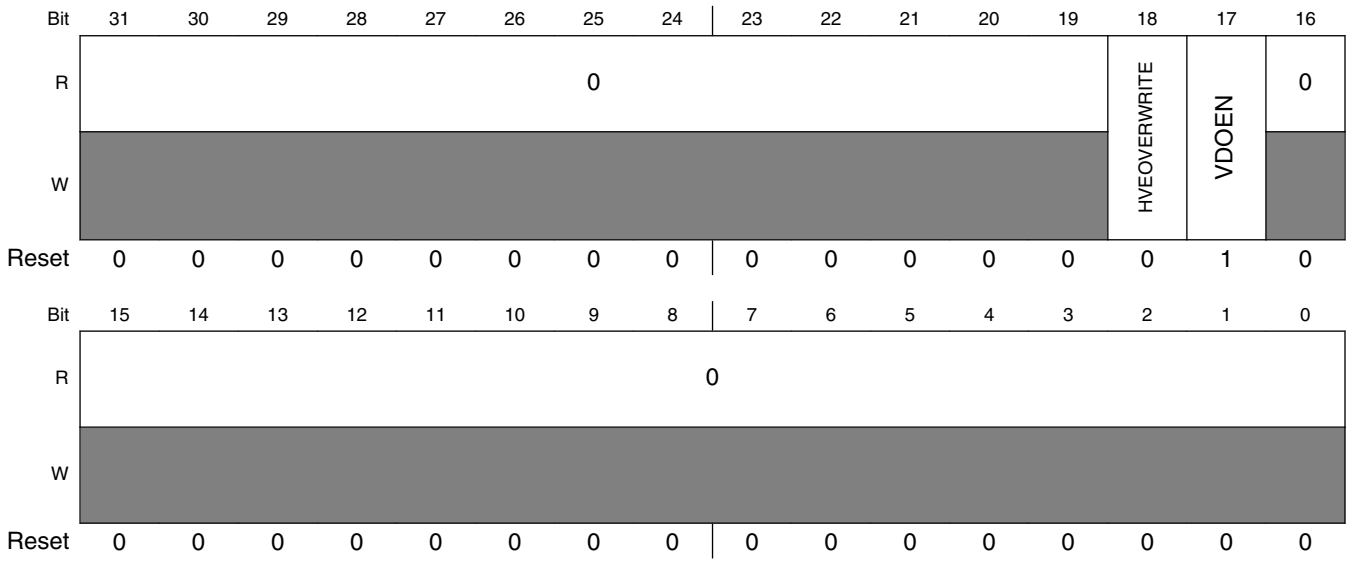
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_D31 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_D31. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_D31. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_D31. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_D31. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_D31. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.298 IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__1 (IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__1)

Address: IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__1 is 53FA_8000h base + 4A4h offset = 53FA_84A4h

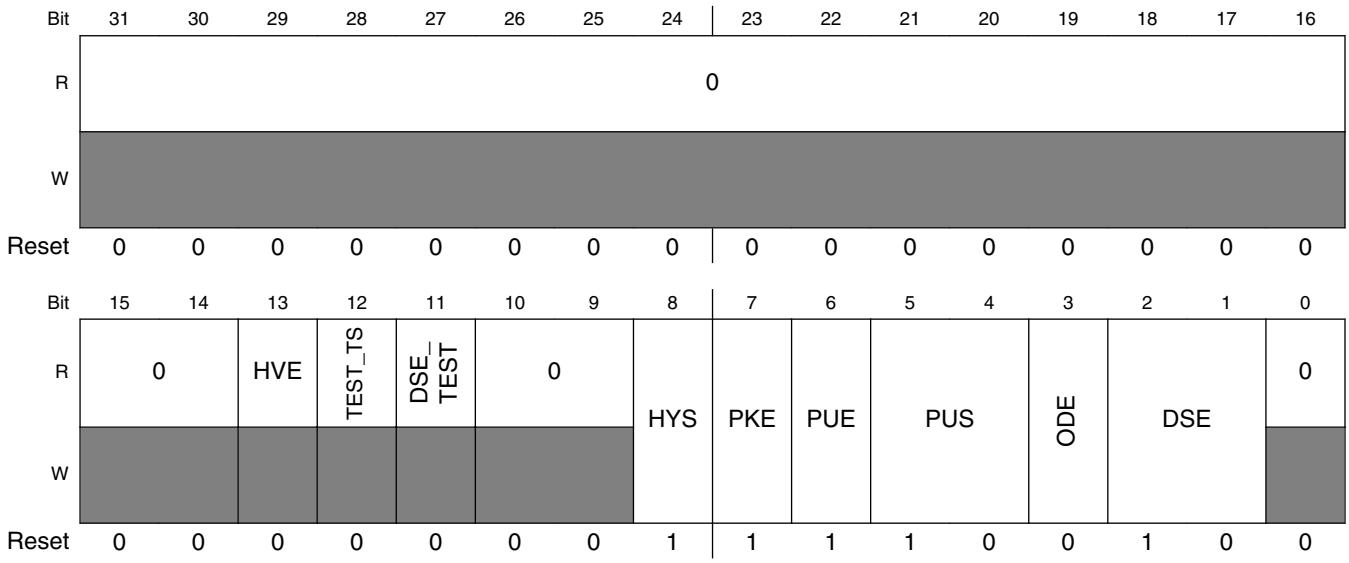


IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__1 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18 HVEOVERWRITE	HVEOVERWRITE (ipp_owr) Select one out of next values for pad: NVCC_EIM__1. This field affects the voltage levels for the following PADS: EIM_D31, EIM_D30, EIM_D29, EIM_D28, EIM_D27, EIM_D26, EIM_D25, EIM_D24, EIM_D23, EIM_D22, EIM_D21, EIM_D20, EIM_D19, EIM_D18, EIM_D17, EIM_D16 0 PAD support high voltage (3.0V-3.6V). Out of reset value is 0. The HVEOVERWRITE field should be updated before automatic voltage detector is disabled. 1 PAD support low voltage (1.65V-3.1V)
17 VDOEN	VDOEN (ipp_oen) Select one out of next values for pad: NVCC_EIM__1. 0 Voltage detector output disable and HVEOVERWRITE field will be used. It is recommended to disable the automatic voltage detector after boot and update the HVEOVERWRITE field with the actual I/O supply value. 1 Voltage detector output enable. (default 1)
16–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.299 IOMUXC_SW_PAD_CTL_PAD_EIM_A24 (IOMUXC_SW_PAD_CTL_PAD_EIM_A24)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_A24 is 53FA_8000h base + 4A8h offset = 53FA_84A8h



IOMUXC_SW_PAD_CTL_PAD_EIM_A24 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_A24. 0 Hysteresis Disabled 1 Hysteresis Enabled

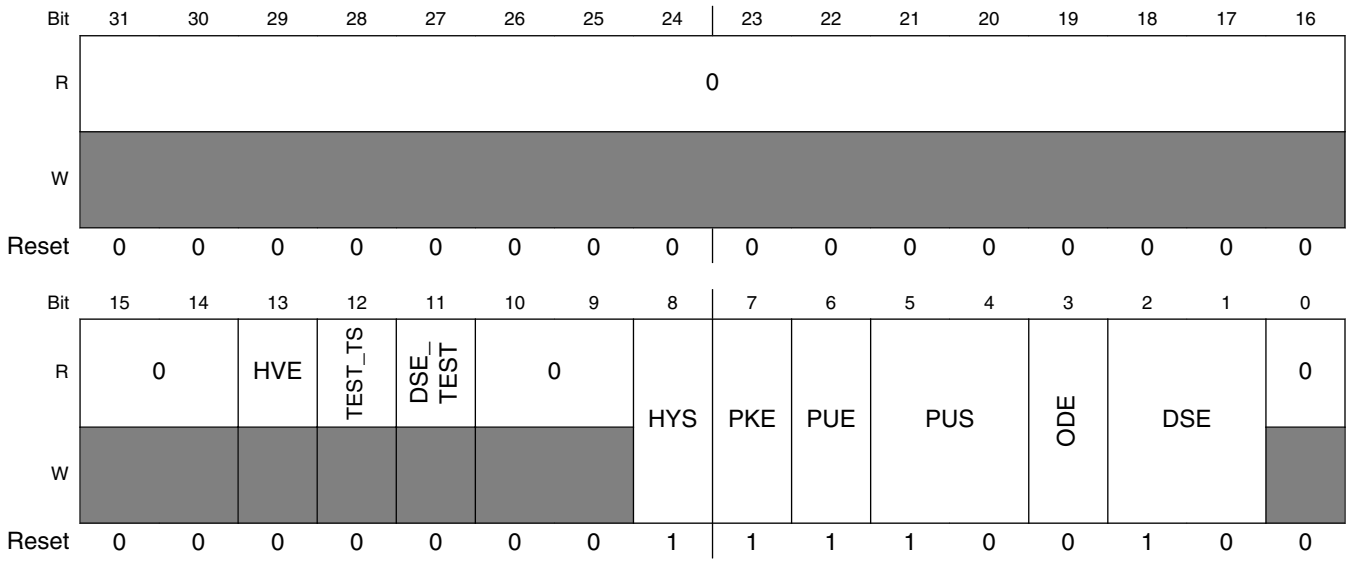
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_A24 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_A24. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_A24. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_A24. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_A24. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_A24. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.300 IOMUXC_SW_PAD_CTL_PAD_EIM_A23 (IOMUXC_SW_PAD_CTL_PAD_EIM_A23)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_A23 is 53FA_8000h base + 4ACh offset = 53FA_84ACh



IOMUXC_SW_PAD_CTL_PAD_EIM_A23 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_A23. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_A23 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_A23. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_A23. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_A23. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_A23. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_A23. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.301 IOMUXC_SW_PAD_CTL_PAD_EIM_A22 (IOMUXC_SW_PAD_CTL_PAD_EIM_A22)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_A22 is 53FA_8000h base + 4B0h offset = 53FA_84B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_A22 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_A22. 0 Hysteresis Disabled 1 Hysteresis Enabled

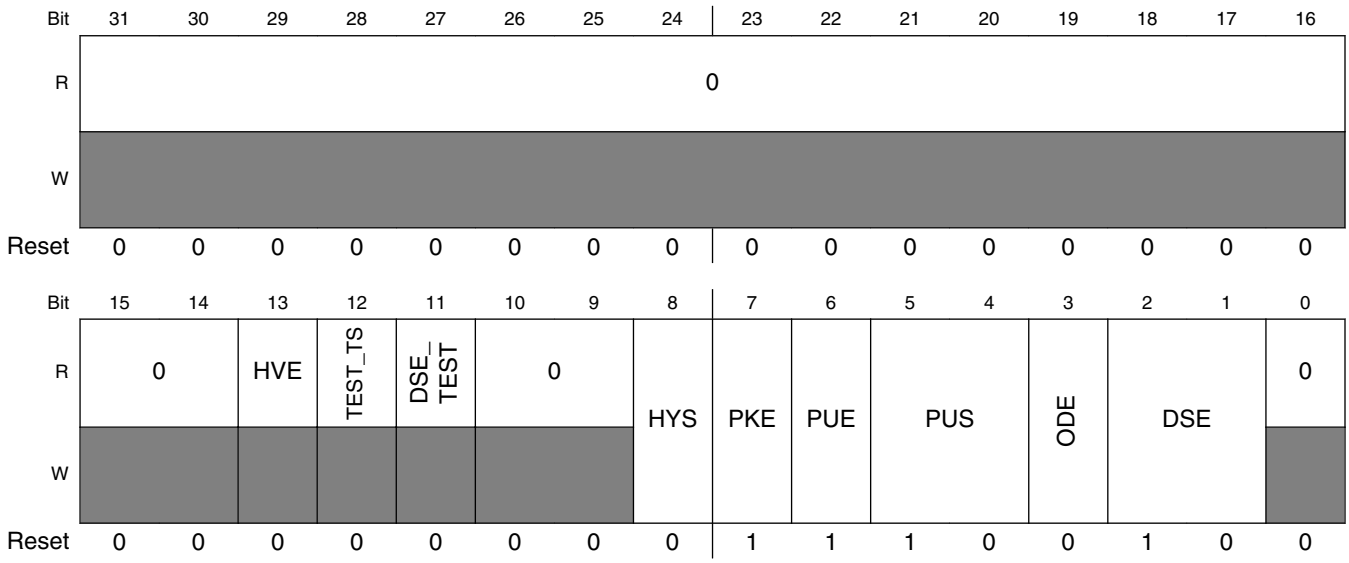
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_A22 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_A22. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_A22. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_A22. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_A22. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_A22. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.302 IOMUXC_SW_PAD_CTL_PAD_EIM_A21 (IOMUXC_SW_PAD_CTL_PAD_EIM_A21)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_A21 is 53FA_8000h base + 4B4h offset = 53FA_84B4h



IOMUXC_SW_PAD_CTL_PAD_EIM_A21 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_A21. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_A21 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_A21. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_A21. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_A21. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_A21. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_A21. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.303 IOMUXC_SW_PAD_CTL_PAD_EIM_A20 (IOMUXC_SW_PAD_CTL_PAD_EIM_A20)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_A20 is 53FA_8000h base + 4B8h offset = 53FA_84B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_A20 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_A20. 0 Hysteresis Disabled 1 Hysteresis Enabled

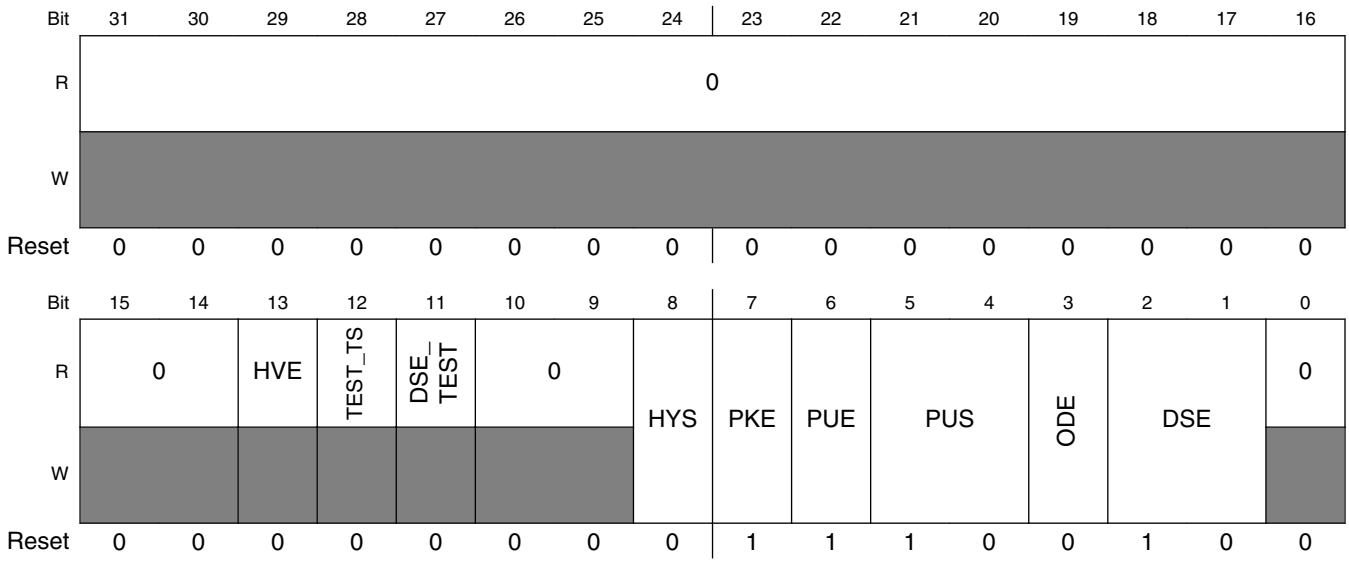
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_A20 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_A20. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_A20. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_A20. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_A20. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_A20. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.304 IOMUXC_SW_PAD_CTL_PAD_EIM_A19 (IOMUXC_SW_PAD_CTL_PAD_EIM_A19)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_A19 is 53FA_8000h base + 4BCh offset = 53FA_84BCh



IOMUXC_SW_PAD_CTL_PAD_EIM_A19 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_A19. 0 Hysteresis Disabled 1 Hysteresis Enabled

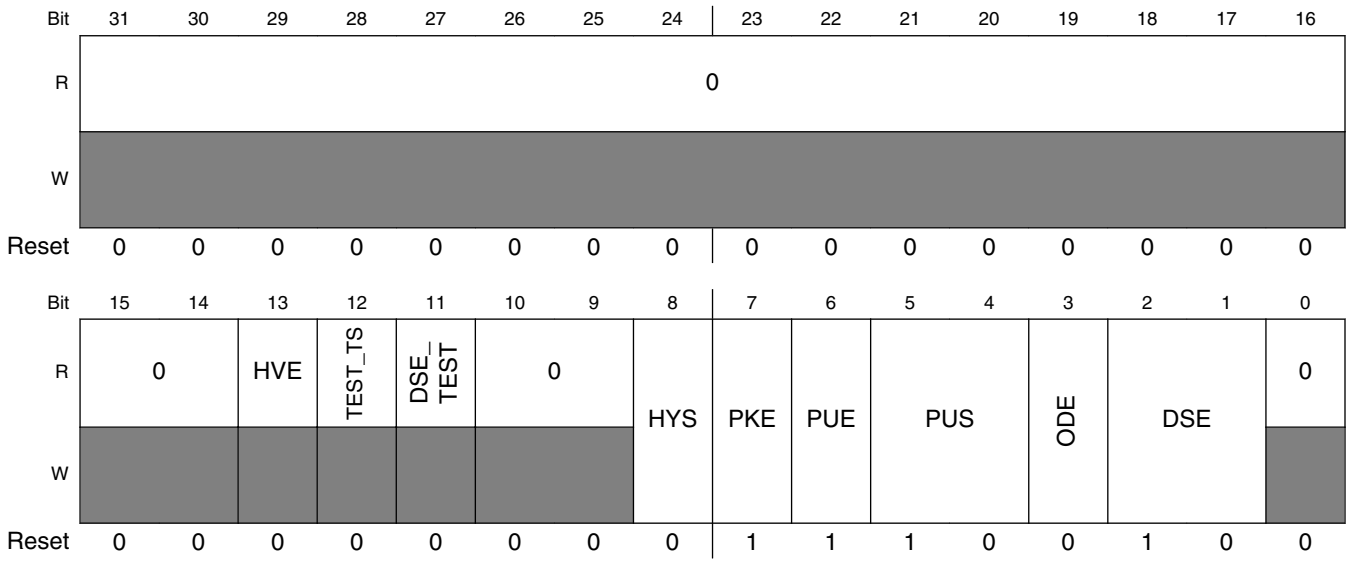
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_A19 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_A19. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_A19. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_A19. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_A19. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_A19. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.305 IOMUXC_SW_PAD_CTL_PAD_EIM_A18 (IOMUXC_SW_PAD_CTL_PAD_EIM_A18)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_A18 is 53FA_8000h base + 4C0h offset = 53FA_84C0h



IOMUXC_SW_PAD_CTL_PAD_EIM_A18 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_A18. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_A18 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_A18. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_A18. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_A18. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_A18. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_A18. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.306 IOMUXC_SW_PAD_CTL_PAD_EIM_A17 (IOMUXC_SW_PAD_CTL_PAD_EIM_A17)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_A17 is 53FA_8000h base + 4C4h offset = 53FA_84C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_A17 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_A17. 0 Hysteresis Disabled 1 Hysteresis Enabled

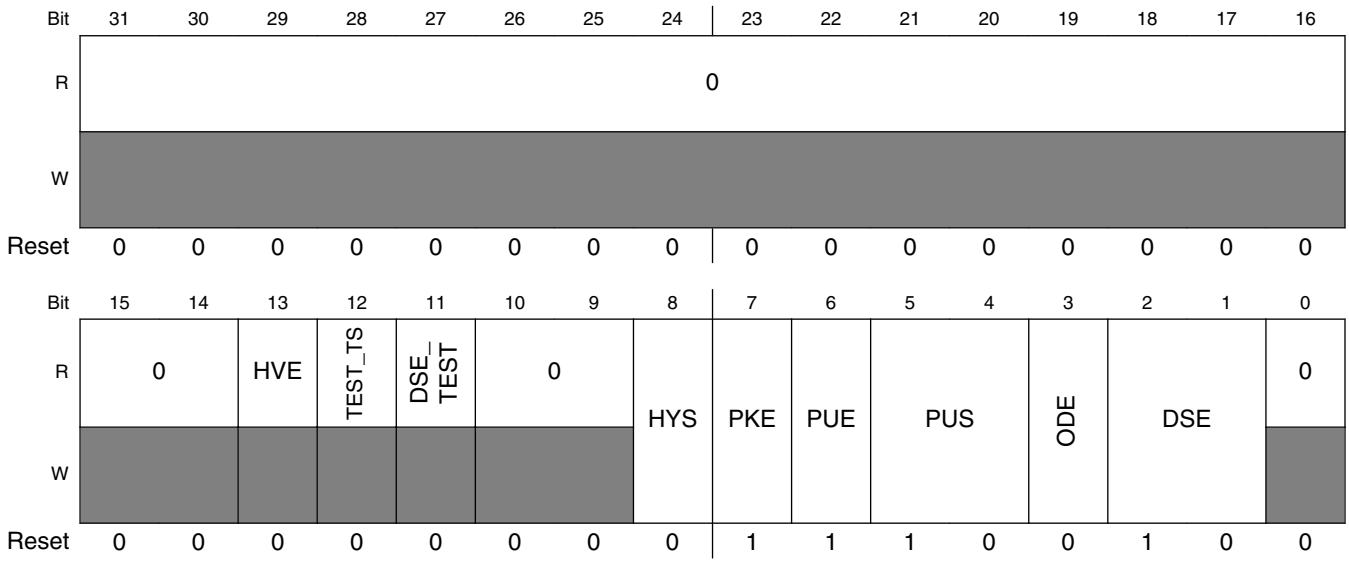
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_A17 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_A17. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_A17. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_A17. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_A17. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_A17. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.307 IOMUXC_SW_PAD_CTL_PAD_EIM_A16 (IOMUXC_SW_PAD_CTL_PAD_EIM_A16)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_A16 is 53FA_8000h base + 4C8h offset = 53FA_84C8h



IOMUXC_SW_PAD_CTL_PAD_EIM_A16 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_A16. 0 Hysteresis Disabled 1 Hysteresis Enabled

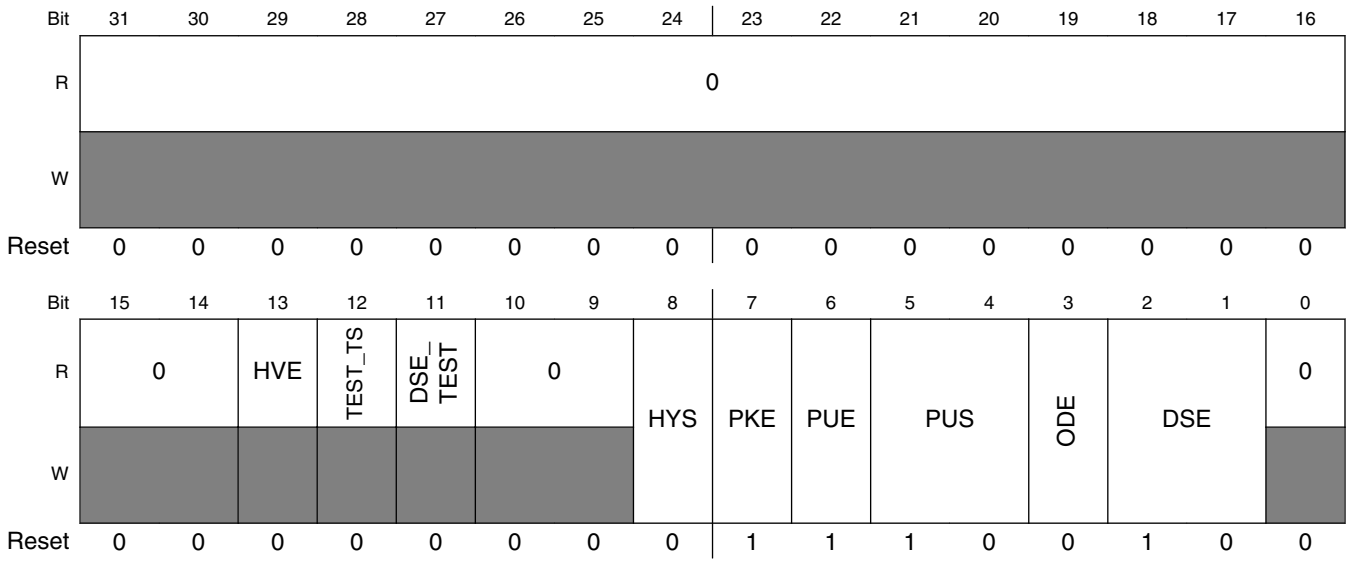
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_A16 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_A16. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_A16. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_A16. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_A16. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_A16. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.308 IOMUXC_SW_PAD_CTL_PAD_EIM_CS0 (IOMUXC_SW_PAD_CTL_PAD_EIM_CS0)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_CS0 is 53FA_8000h base + 4CCh offset = 53FA_84CCh



IOMUXC_SW_PAD_CTL_PAD_EIM_CS0 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_CS0. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_CS0 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_CS0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_CS0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_CS0. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_CS0. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_CS0. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.309 IOMUXC_SW_PAD_CTL_PAD_EIM_CS1 (IOMUXC_SW_PAD_CTL_PAD_EIM_CS1)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_CS1 is 53FA_8000h base + 4D0h offset = 53FA_84D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_CS1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_CS1. 0 Hysteresis Disabled 1 Hysteresis Enabled

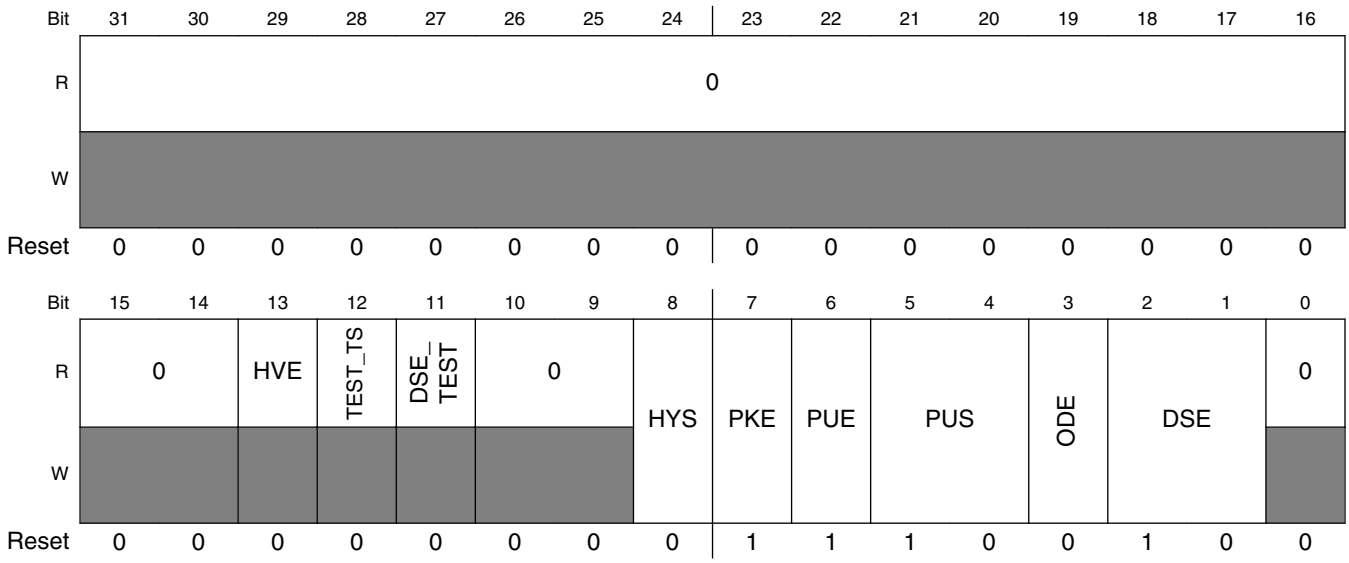
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_CS1 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_CS1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_CS1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_CS1. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_CS1. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_CS1. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.310 IOMUXC_SW_PAD_CTL_PAD_EIM_OE (IOMUXC_SW_PAD_CTL_PAD_EIM_OE)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_OE is 53FA_8000h base + 4D4h offset = 53FA_84D4h



IOMUXC_SW_PAD_CTL_PAD_EIM_OE field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_OE. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_OE field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_OE. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_OE. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_OE. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_OE. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_OE. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.311 IOMUXC_SW_PAD_CTL_PAD_EIM_RW (IOMUXC_SW_PAD_CTL_PAD_EIM_RW)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_RW is 53FA_8000h base + 4D8h offset = 53FA_84D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_RW field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_RW. 0 Hysteresis Disabled 1 Hysteresis Enabled

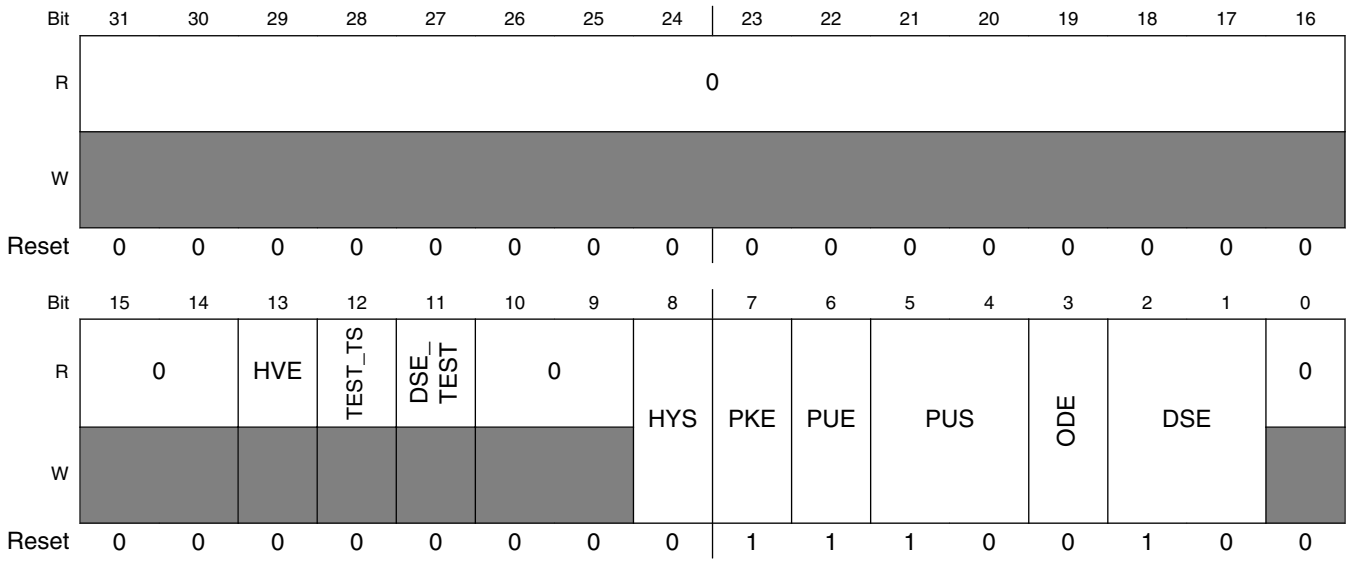
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_RW field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_RW. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_RW. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_RW. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_RW. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_RW. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.312 IOMUXC_SW_PAD_CTL_PAD_EIM_LBA (IOMUXC_SW_PAD_CTL_PAD_EIM_LBA)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_LBA is 53FA_8000h base + 4DCh offset = 53FA_84DCh



IOMUXC_SW_PAD_CTL_PAD_EIM_LBA field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_LBA. 0 Hysteresis Disabled 1 Hysteresis Enabled

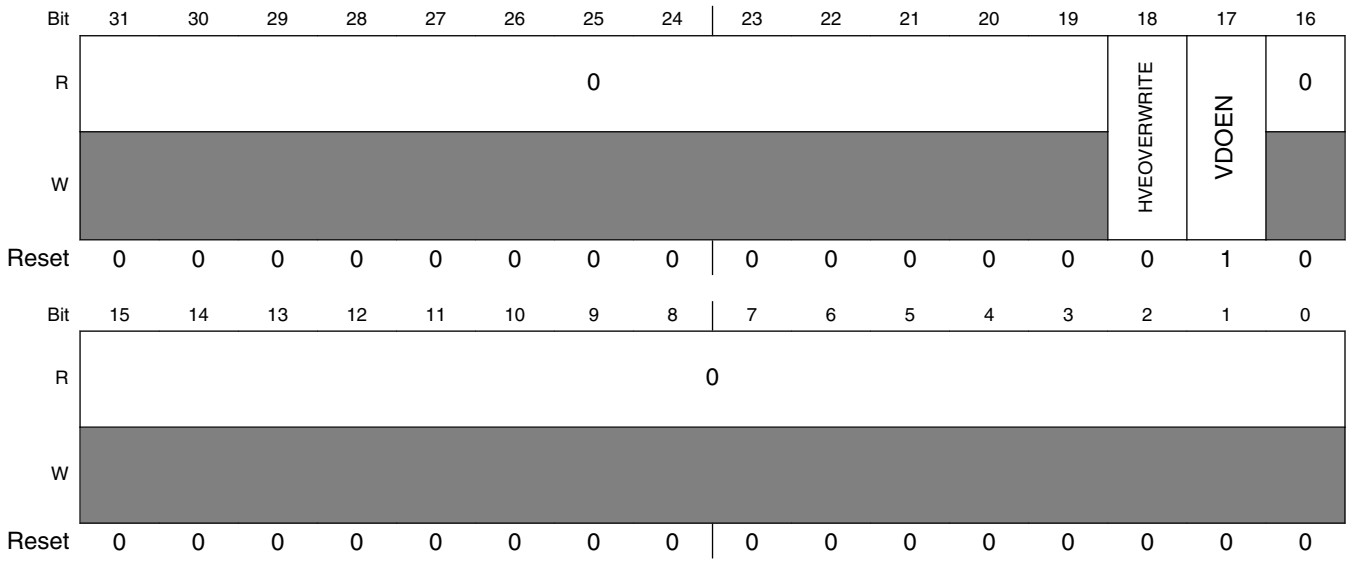
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_LBA field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_LBA. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_LBA. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_LBA. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_LBA. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_LBA. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.313 IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__4 (IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__4)

Address: IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__4 is 53FA_8000h base + 4E0h offset = 53FA_84E0h



IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__4 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18 HVEOVERWRITE	HVEOVERWRITE (ipp_owr) Select one out of next values for pad: NVCC_EIM__4. This field affects the voltage levels for the following PADS: EIM_A25, EIM_EB2, EIM_EB3, EIM_A24, EIM_A23, EIM_A22, EIM_A21, EIM_A20, EIM_A19, EIM_A18, EIM_A17, EIM_A16, EIM_CS0, EIM_CS1, EIM_OE 0 PAD support high voltage (3.0V-3.6V). Out of reset value is 0. The HVEOVERWRITE field should be updated before automatic voltage detector is disabled. 1 PAD support low voltage (1.65V-3.1V)
17 VDOEN	VDOEN (ipp_oen) Select one out of next values for pad: NVCC_EIM__4. 0 Voltage detector output disable and HVEOVERWRITE field will be used. It is recommended to disable the automatic voltage detector after boot and update the HVEOVERWRITE field with the actual I/O supply value. 1 Voltage detector output enable. (default 1)
16–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.314 IOMUXC_SW_PAD_CTL_PAD_EIM_EB0 (IOMUXC_SW_PAD_CTL_PAD_EIM_EB0)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_EB0 is 53FA_8000h base + 4E4h offset = 53FA_84E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_EB0 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_EB0. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_EB0 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_EB0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_EB0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_EB0. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_EB0. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_EB0. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.315 IOMUXC_SW_PAD_CTL_PAD_EIM_EB1 (IOMUXC_SW_PAD_CTL_PAD_EIM_EB1)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_EB1 is 53FA_8000h base + 4E8h offset = 53FA_84E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_EB1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_EB1. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_EB1 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_EB1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_EB1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_EB1. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_EB1. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_EB1. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.316 IOMUXC_SW_PAD_CTL_PAD_EIM_DA0 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA0)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_DA0 is 53FA_8000h base + 4ECh offset = 53FA_84ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DA0 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_DA0. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DA0 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_DA0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_DA0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_DA0. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_DA0. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_DA0. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.317 IOMUXC_SW_PAD_CTL_PAD_EIM_DA1 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA1)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_DA1 is 53FA_8000h base + 4F0h offset = 53FA_84F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DA1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_DA1. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DA1 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_DA1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_DA1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_DA1. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_DA1. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_DA1. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.318 IOMUXC_SW_PAD_CTL_PAD_EIM_DA2 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA2)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_DA2 is 53FA_8000h base + 4F4h offset = 53FA_84F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DA2 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_DA2. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DA2 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_DA2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_DA2. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_DA2. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_DA2. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_DA2. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.319 IOMUXC_SW_PAD_CTL_PAD_EIM_DA3 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA3)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_DA3 is 53FA_8000h base + 4F8h offset = 53FA_84F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DA3 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_DA3. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DA3 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_DA3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_DA3. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_DA3. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_DA3. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_DA3. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.320 IOMUXC_SW_PAD_CTL_PAD_EIM_DA4 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA4)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_DA4 is 53FA_8000h base + 4FCh offset = 53FA_84FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DA4 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_DA4. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DA4 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_DA4. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_DA4. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_DA4. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_DA4. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_DA4. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.321 IOMUXC_SW_PAD_CTL_PAD_EIM_DA5 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA5)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_DA5 is 53FA_8000h base + 500h offset = 53FA_8500h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DA5 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_DA5. 0 Hysteresis Disabled 1 Hysteresis Enabled

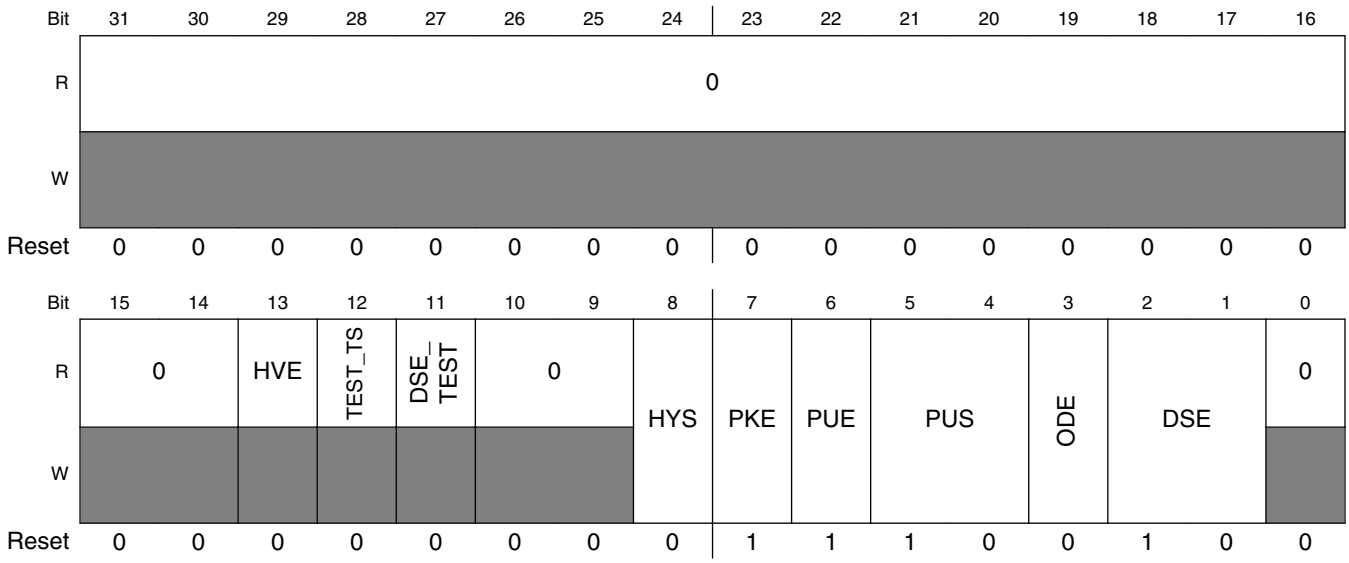
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DA5 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_DA5. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_DA5. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_DA5. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_DA5. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_DA5. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.322 IOMUXC_SW_PAD_CTL_PAD_EIM_DA6 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA6)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_DA6 is 53FA_8000h base + 504h offset = 53FA_8504h



IOMUXC_SW_PAD_CTL_PAD_EIM_DA6 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_DA6. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DA6 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_DA6. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_DA6. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_DA6. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_DA6. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_DA6. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.323 IOMUXC_SW_PAD_CTL_PAD_EIM_DA7 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA7)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_DA7 is 53FA_8000h base + 508h offset = 53FA_8508h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DA7 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_DA7. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DA7 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_DA7. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_DA7. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_DA7. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_DA7. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_DA7. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.324 IOMUXC_SW_PAD_CTL_PAD_EIM_DA8 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA8)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_DA8 is 53FA_8000h base + 50Ch offset = 53FA_850Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DA8 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_DA8. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DA8 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_DA8. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_DA8. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_DA8. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_DA8. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_DA8. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.325 IOMUXC_SW_PAD_CTL_PAD_EIM_DA9 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA9)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_DA9 is 53FA_8000h base + 510h offset = 53FA_8510h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DA9 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_DA9. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DA9 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_DA9. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_DA9. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_DA9. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_DA9. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_DA9. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.326 IOMUXC_SW_PAD_CTL_PAD_EIM_DA10 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA10)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_DA10 is 53FA_8000h base + 514h offset = 53FA_8514h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DA10 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_DA10. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DA10 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_DA10. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_DA10. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_DA10. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_DA10. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_DA10. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.327 IOMUXC_SW_PAD_CTL_PAD_EIM_DA11 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA11)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_DA11 is 53FA_8000h base + 518h offset = 53FA_8518h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DA11 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_DA11. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DA11 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_DA11. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_DA11. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_DA11. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_DA11. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_DA11. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.328 IOMUXC_SW_PAD_CTL_PAD_EIM_DA12 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA12)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_DA12 is 53FA_8000h base + 51Ch offset = 53FA_851Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DA12 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_DA12. 0 Hysteresis Disabled 1 Hysteresis Enabled

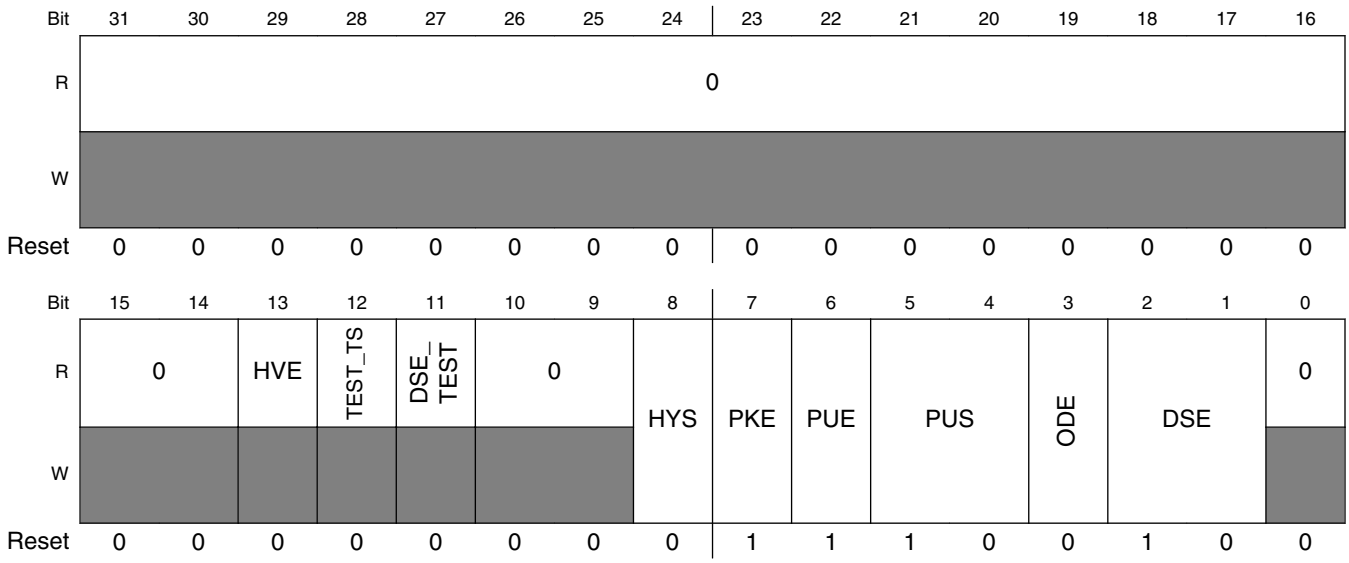
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DA12 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_DA12. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_DA12. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_DA12. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_DA12. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_DA12. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.329 IOMUXC_SW_PAD_CTL_PAD_EIM_DA13 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA13)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_DA13 is 53FA_8000h base + 520h offset = 53FA_8520h



IOMUXC_SW_PAD_CTL_PAD_EIM_DA13 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_DA13. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DA13 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_DA13. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_DA13. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_DA13. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_DA13. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_DA13. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.330 IOMUXC_SW_PAD_CTL_PAD_EIM_DA14 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA14)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_DA14 is 53FA_8000h base + 524h offset = 53FA_8524h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DA14 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_DA14. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DA14 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_DA14. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_DA14. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_DA14. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_DA14. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_DA14. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.331 IOMUXC_SW_PAD_CTL_PAD_EIM_DA15 (IOMUXC_SW_PAD_CTL_PAD_EIM_DA15)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_DA15 is 53FA_8000h base + 528h offset = 53FA_8528h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_DA15 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_DA15. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_DA15 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_DA15. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_DA15. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_DA15. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_DA15. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_DA15. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.332 IOMUXC_SW_PAD_CTL_PAD_NANDF_WE_B (IOMUXC_SW_PAD_CTL_PAD_NANDF_WE_B)

Address: IOMUXC_SW_PAD_CTL_PAD_NANDF_WE_B is 53FA_8000h base + 52Ch offset = 53FA_852Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_NANDF_WE_B field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: NANDF_WE_B. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NANDF_WE_B field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: NANDF_WE_B. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: NANDF_WE_B. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: NANDF_WE_B. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: NANDF_WE_B. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: NANDF_WE_B. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.333 IOMUXC_SW_PAD_CTL_PAD_NANDF_RE_B (IOMUXC_SW_PAD_CTL_PAD_NANDF_RE_B)

Address: IOMUXC_SW_PAD_CTL_PAD_NANDF_RE_B is 53FA_8000h base + 530h offset = 53FA_8530h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_NANDF_RE_B field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: NANDF_RE_B. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NANDF_RE_B field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: NANDF_RE_B. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: NANDF_RE_B. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: NANDF_RE_B. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: NANDF_RE_B. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: NANDF_RE_B. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.334 IOMUXC_SW_PAD_CTL_PAD_EIM_WAIT (IOMUXC_SW_PAD_CTL_PAD_EIM_WAIT)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_WAIT is 53FA_8000h base + 534h offset = 53FA_8534h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_WAIT field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_WAIT. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_WAIT field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_WAIT. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_WAIT. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: EIM_WAIT. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: EIM_WAIT. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_WAIT. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.335 IOMUXC_SW_PAD_CTL_PAD_EIM_BCLK (IOMUXC_SW_PAD_CTL_PAD_EIM_BCLK)

Address: IOMUXC_SW_PAD_CTL_PAD_EIM_BCLK is 53FA_8000h base + 538h offset = 53FA_8538h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_EIM_BCLK field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: EIM_BCLK. 0 Hysteresis Disabled 1 Hysteresis Enabled

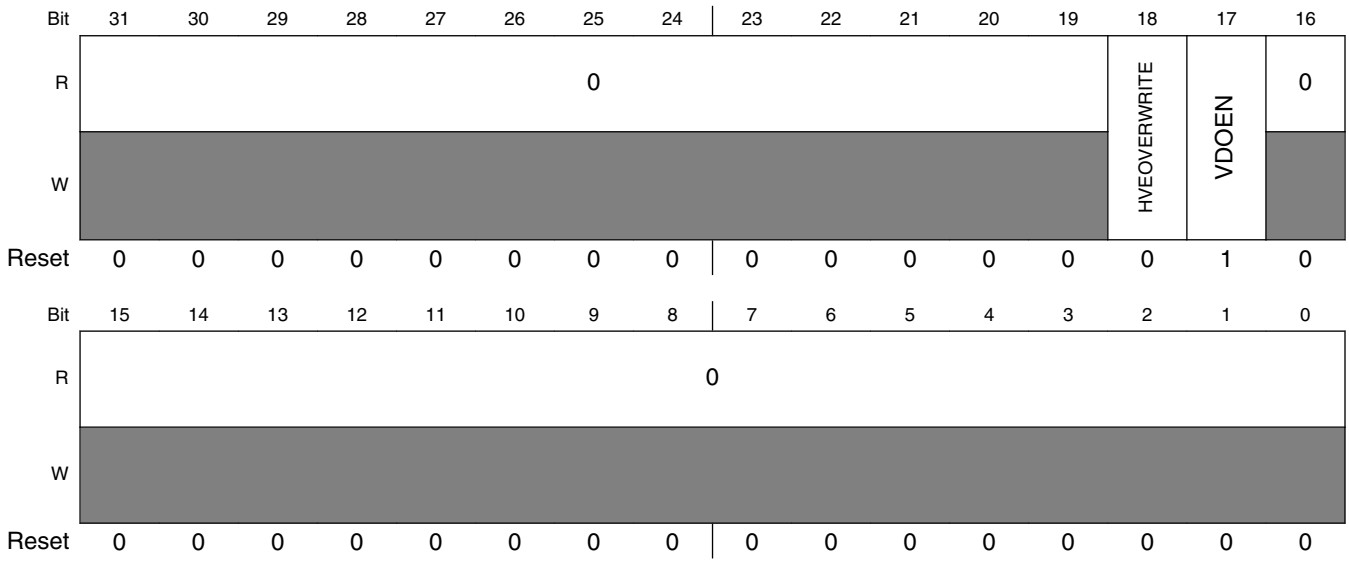
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_EIM_BCLK field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: EIM_BCLK. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: EIM_BCLK. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Read Only Field 10 100 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Read Only Field 0 Open Drain Disabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: EIM_BCLK. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.336 IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__7 (IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__7)

Address: IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__7 is 53FA_8000h base + 53Ch offset = 53FA_853Ch

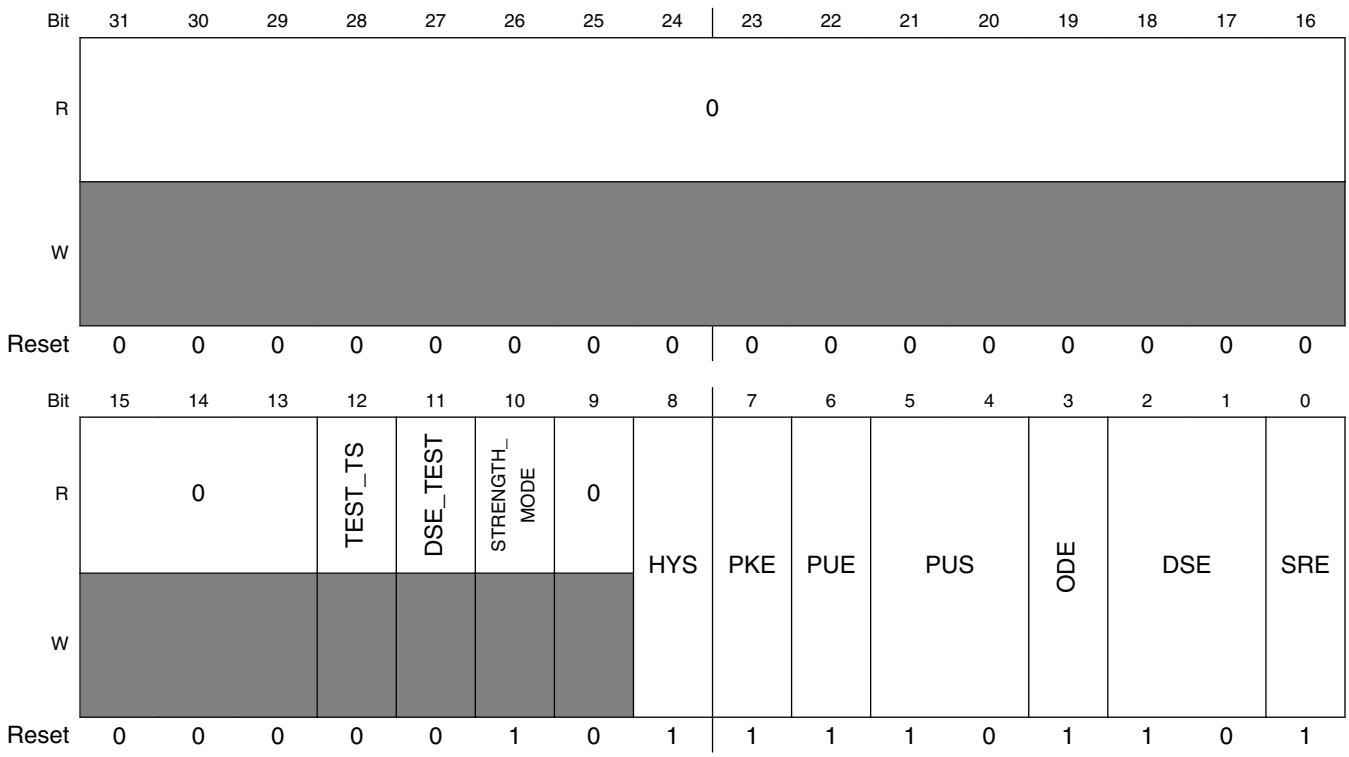


IOMUXC_SW_PAD_CTL_PAD_NVCC_EIM__7 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18 HVEOVERWRITE	HVEOVERWRITE (ipp_owr) Select one out of next values for pad: NVCC_EIM__7. This field affects the voltage levels for the following PADS: EIM_RW, EIM_LBA, EIM_EB0, EIM_EB1, EIM_DA0, EIM_DA1 EIM_DA2, EIM_DA3, EIM_DA4, EIM_DA5, EIM_DA6, EIM_DA7, EIM_DA8, EIM_DA9, EIM_DA10, EIM_DA11, EIM_DA12, EIM_DA13, EIM_DA14, EIM_DA15, NANDF_WE_B, NANDF_RE_B, EIM_WAIT, EIM_BCLK 0 PAD support high voltage (3.0V-3.6V). Out of reset value is 0. The HVEOVERWRITE field should be updated before automatic voltage detector is disabled. 1 PAD support low voltage (1.65V-3.1V)
17 VDOEN	VDOEN (ipp_oen) Select one out of next values for pad: NVCC_EIM__7. 0 Voltage detector output disable and HVEOVERWRITE field will be used. It is recommended to disable the automatic voltage detector after boot and update the HVEOVERWRITE field with the actual I/O supply value. 1 Voltage detector output enable. (default 1)
16–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.337 IOMUXC_SW_PAD_CTL_PAD_GPIO_10 (IOMUXC_SW_PAD_CTL_PAD_GPIO_10)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_10 is 53FA_8000h base + 540h offset = 53FA_8540h



IOMUXC_SW_PAD_CTL_PAD_GPIO_10 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

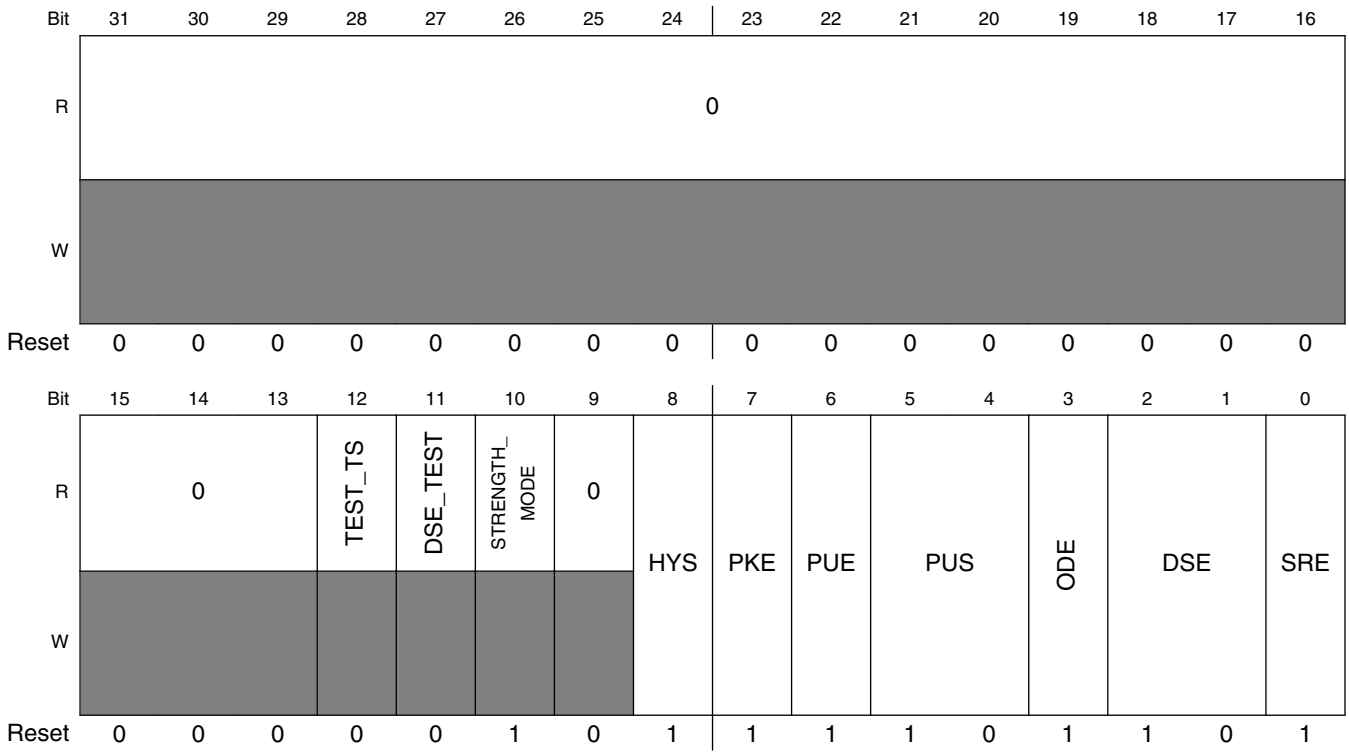
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_10 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_10. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_10. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_10. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_10. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_10. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_10. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_10. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.338 IOMUXC_SW_PAD_CTL_PAD_GPIO_11 (IOMUXC_SW_PAD_CTL_PAD_GPIO_11)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_11 is 53FA_8000h base + 544h offset = 53FA_8544h



IOMUXC_SW_PAD_CTL_PAD_GPIO_11 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

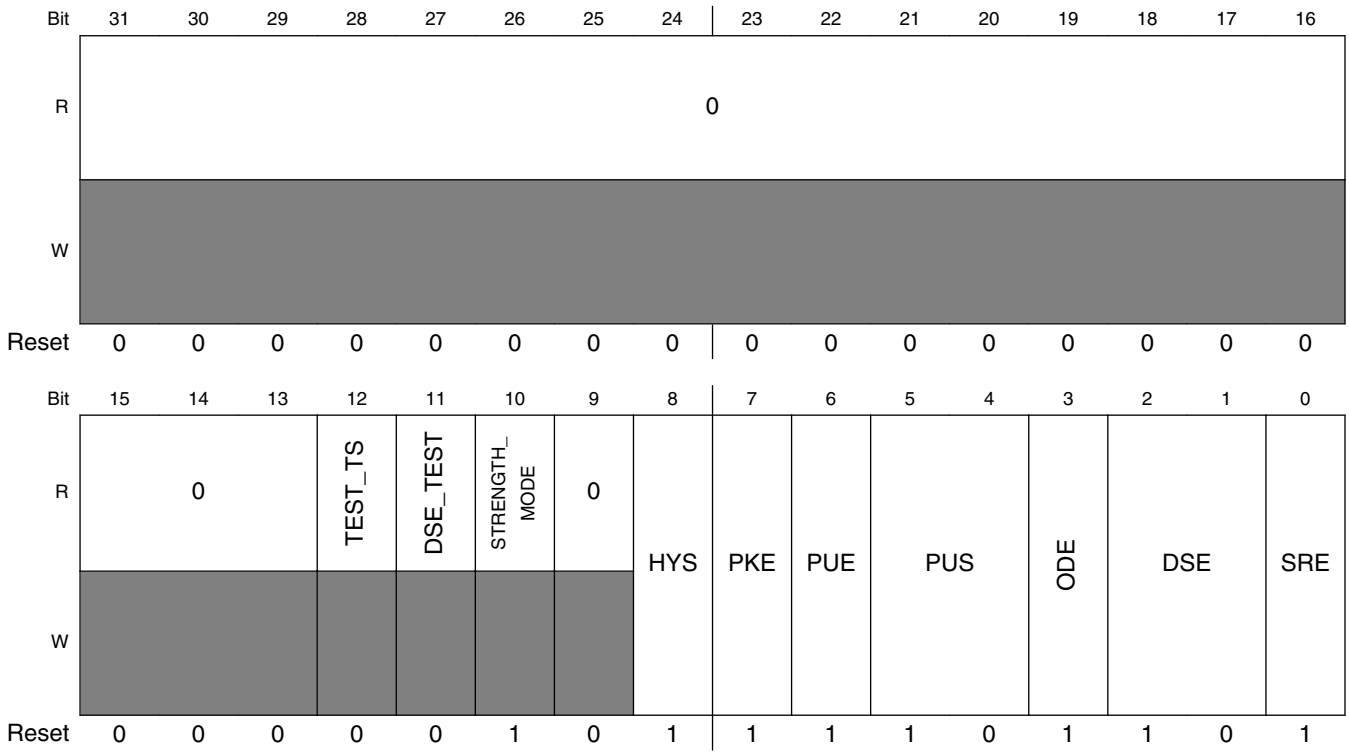
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_11 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_11. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_11. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_11. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_11. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_11. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_11. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_11. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.339 IOMUXC_SW_PAD_CTL_PAD_GPIO_12 (IOMUXC_SW_PAD_CTL_PAD_GPIO_12)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_12 is 53FA_8000h base + 548h offset = 53FA_8548h



IOMUXC_SW_PAD_CTL_PAD_GPIO_12 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_12 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_12. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_12. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_12. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_12. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_12. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_12. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_12. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.340 IOMUXC_SW_PAD_CTL_PAD_GPIO_13 (IOMUXC_SW_PAD_CTL_PAD_GPIO_13)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_13 is 53FA_8000h base + 54Ch offset = 53FA_854Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	DSE_TEST	STRENGTH_MODE	0	HYS	PKE	PUE	PUS		ODE	DSE		SRE
W																
Reset	0	0	0	0	0	1	0	1	1	1	1	0	1	1	0	1

IOMUXC_SW_PAD_CTL_PAD_GPIO_13 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_13 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_13. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_13. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_13. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_13. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_13. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_13. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_13. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.341 IOMUXC_SW_PAD_CTL_PAD_GPIO_14 (IOMUXC_SW_PAD_CTL_PAD_GPIO_14)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_14 is 53FA_8000h base + 550h offset = 53FA_8550h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	DSE_TEST	STRENGTH_MODE	0	HYS	PKE	PUE	PUS		ODE	DSE		SRE
W																
Reset	0	0	0	0	0	1	0	1	1	1	1	0	1	1	0	1

IOMUXC_SW_PAD_CTL_PAD_GPIO_14 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_14 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_14. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_14. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_14. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_14. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_14. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_14. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO_14. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.342 IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM3 (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM3)

Address: IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM3 is 53FA_8000h base + 554h offset = 53FA_8554h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					0		ODT			DSE			0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	0		DDR_INPUT	HYS	PKE	PUE	PUS		0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM3 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 Reserved	This read-only field is reserved and always has the value zero. ddr_sel Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Please note that changes are not reflected once the field is updated using the GCR register. 0 DDR2/DDR3 mode
24–22 ODT	On Die Termination Field Read Only Field 000 DISABLED (HiZ)
21–19 DSE	Refer to Table 43-2.
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM3 field descriptions (continued)

Field	Description
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field 0 CMOS input type
8 HYS	Hyst. Enable Field Read Only Field 0 Hysteresis Disabled
7 PKE	Pull / Keep Enable Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRPKE Please note that changes are not reflected once the field is updated using the GCR register. 0 Pull/Keeper Disabled
6 PUE	Pull / Keep Select Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRPK Please note that changes are not reflected once the field is updated using the GCR register. 0 Keeper
5–4 PUS	Pull Up / Down Config. Field Read Only Field 10 100 [KOhm] Pull Up
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.343 IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS3 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS3)

Address: IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS3 is 53FA_8000h base + 558h offset = 53FA_8558h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					0			0		DSE			0		
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	0		DDR_INPUT	HYS	PKE	PUE	PUS		0			
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS3 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 Reserved	This read-only field is reserved and always has the value zero. ddr_sel Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Please note that changes are not reflected once the field is updated using the GCR register. 0 DDR2/DDR3 mode
24–22 Reserved	This read-only field is reserved and always has the value zero. On Die Termination Field Read-only field. 000 DISABLED (HiZ)
21–19 DSE	Refer to Table 43-2.
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled

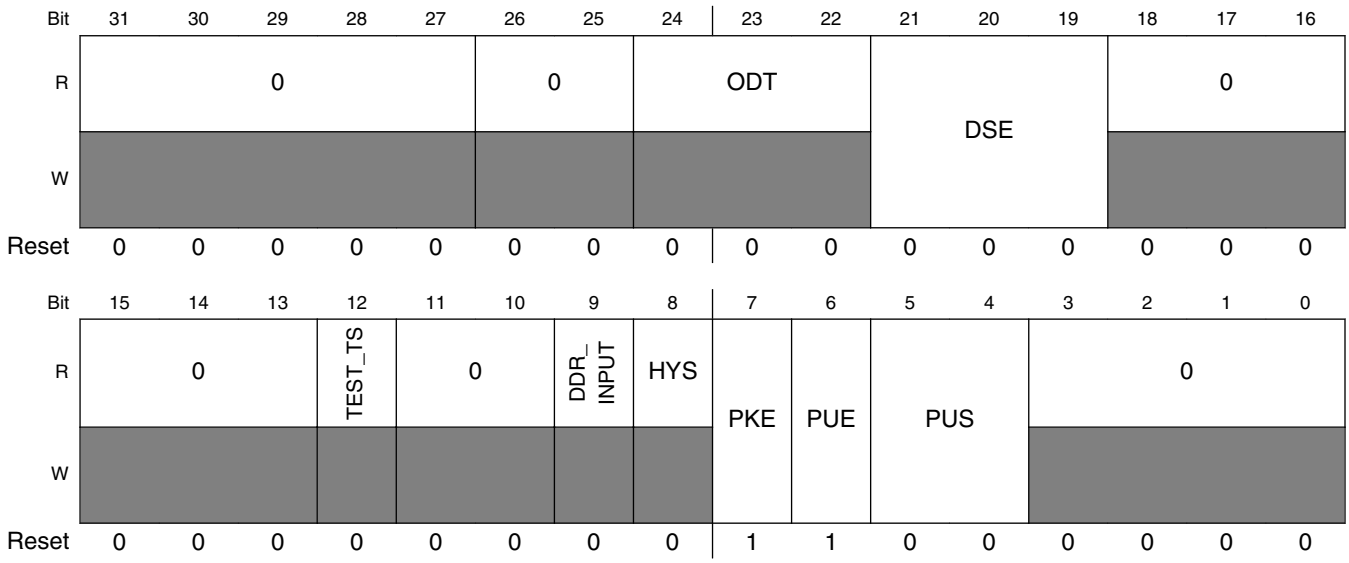
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS3 field descriptions (continued)

Field	Description
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL Please note that changes are not reflected once the field is updated using the GCR register. 0 CMOS input type
8 HYS	Hyst. Enable Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRHYS Please note that changes are not reflected once the field is updated using the GCR register. 0 Hysteresis Disabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DRAM_SDQS3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DRAM_SDQS3. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DRAM_SDQS3. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.344 IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE1 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE1)

Address: IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE1 is 53FA_8000h base + 55Ch offset = 53FA_855Ch



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE1 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 Reserved	This read-only field is reserved and always has the value zero. ddr_sel Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Please note that changes are not reflected once the field is updated using the GCR register. 0 DDR2/DDR3 mode
24–22 ODT	On Die Termination field Read-only field 000 DISABLED (HiZ)
21–19 DSE	Drive Strength Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_CTLDS Please note that changes are not reflected once the field is updated using the GCR register. 0 DISABLED (HiZ)
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE1 field descriptions (continued)

Field	Description
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field 0 CMOS input type
8 HYS	Hyst. Enable Field Read Only Field 0 Hysteresis Disabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DRAM_SDCKE1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DRAM_SDCKE1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DRAM_SDCKE1. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.345 IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM2 (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM2)

Address: IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM2 is 53FA_8000h base + 560h offset = 53FA_8560h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					0		ODT			DSE			0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	0		DDR_INPUT	HYS	PKE	PUE	PUS		0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM2 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 Reserved	This read-only field is reserved and always has the value zero. ddr_sel Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Please note that changes are not reflected once the field is updated using the GCR register. 0 DDR2/DDR3 mode
24–22 ODT	On Die Termination field Read-only field 000 DISABLED (HiZ)
21–19 DSE	Refer to Table 43-2.
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled

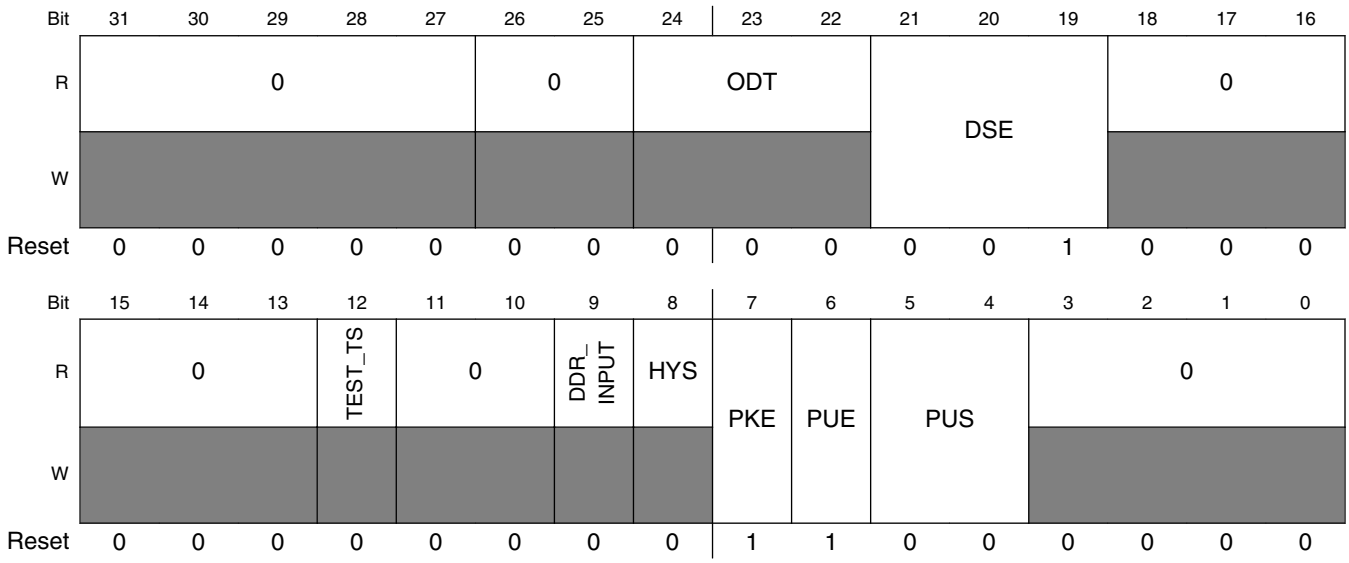
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM2 field descriptions (continued)

Field	Description
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field 0 CMOS input type
8 HYS	Hyst. Enable Field Read Only Field 0 Hysteresis Disabled
7 PKE	Pull / Keep Enable Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRPKE Please note that changes are not reflected once the field is updated using the GCR register. 0 Pull/Keeper Disabled
6 PUE	Pull / Keep Select Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRPK Please note that changes are not reflected once the field is updated using the GCR register. 0 Keeper
5–4 PUS	Pull Up / Down Config. Field Read Only Field 10 100 [KOhm] Pull Up
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.346 IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT1 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT1)

Address: IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT1 is 53FA_8000h base + 564h offset = 53FA_8564h



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT1 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 Reserved	This read-only field is reserved and always has the value zero. ddr_sel Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Please note that changes are not reflected once the field is updated using the GCR register. 0 DDR2/DDR3 mode
24–22 ODT	On Die Termination field Read-only field 000 DISABLED (HiZ)
21–19 DSE	Refer to Table 43-2.
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT1 field descriptions (continued)

Field	Description
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field 0 CMOS input type
8 HYS	Hyst. Enable Field Read Only Field 0 Hysteresis Disabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DRAM_SDODT1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DRAM_SDODT1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DRAM_SDODT1. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.347 IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS2 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS2)

Address: IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS2 is 53FA_8000h base + 568h offset = 53FA_8568h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					0			0		DSE			0		
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	0		DDR_INPUT	HYS	PKE	PUE	PUS		0			
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS2 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 Reserved	This read-only field is reserved and always has the value zero. ddr_sel Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Please note that changes are not reflected once the field is updated using the GCR register. 0 DDR2/DDR3 mode
24–22 Reserved	This read-only field is reserved and always has the value zero. On Die Termination field Read-only field 000 DISABLED (HiZ)
21–19 DSE	Refer to Table 43-2.
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS2 field descriptions (continued)

Field	Description
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL Please note that changes are not reflected once the field is updated using the GCR register. 0 CMOS input type
8 HYS	Hyst. Enable Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRHYS Please note that changes are not reflected once the field is updated using the GCR register. 0 Hysteresis Disabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DRAM_SDQS2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DRAM_SDQS2. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DRAM_SDQS2. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.348 IOMUXC_SW_PAD_CTL_PAD_DRAM_RESET (IOMUXC_SW_PAD_CTL_PAD_DRAM_RESET)

Address: IOMUXC_SW_PAD_CTL_PAD_DRAM_RESET is 53FA_8000h base + 56Ch offset = 53FA_856Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					0			ODT		DSE			0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	0		DDR_INPUT	HYS	PKE	PUE	PUS		0			
W																
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DRAM_RESET field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 Reserved	This read-only field is reserved and always has the value zero. ddr_sel Field Select one out of next values for pad: DRAM_RESET. 00 DDR2/DDR3 mode 01 LPDDR2 mode 10 Reserved 11 Reserved
24–22 ODT	On Die Termination field Read-only field 000 DISABLED (HiZ)
21–19 DSE	Refer to Table 43-2.
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled

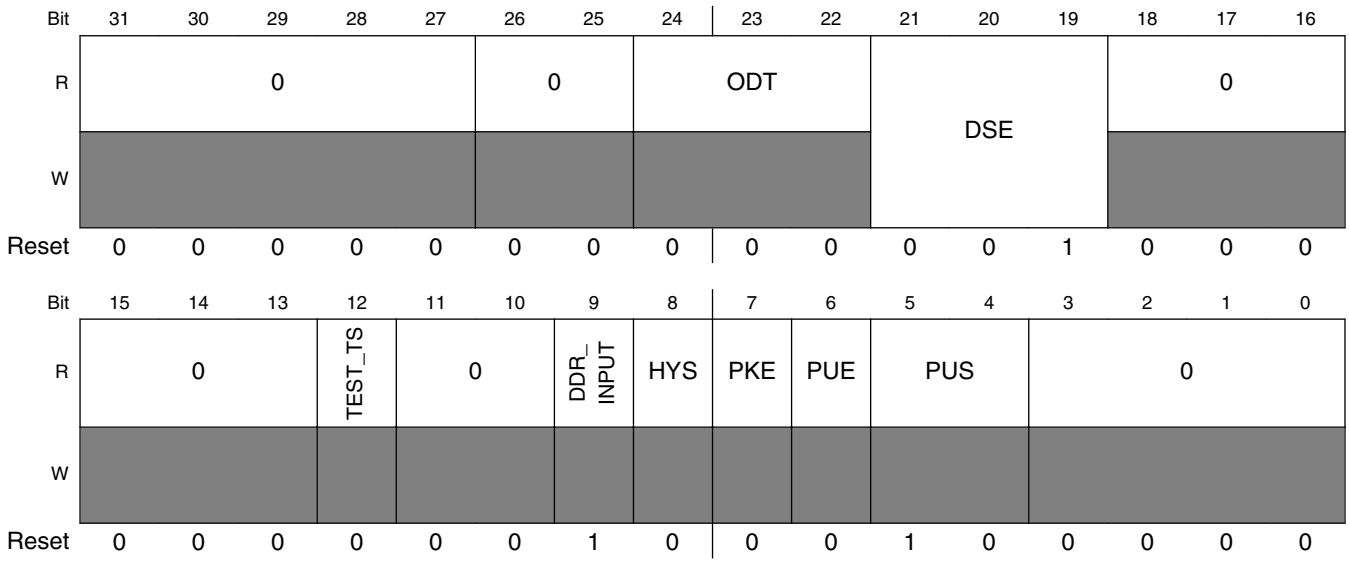
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_RESET field descriptions (continued)

Field	Description
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field 0 CMOS input type
8 HYS	Hyst. Enable Field Read Only Field 0 Hysteresis Disabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DRAM_RESET. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DRAM_RESET. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DRAM_RESET. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.349 IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_1 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_1)

Address: IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_1 is 53FA_8000h base + 570h offset = 53FA_8570h



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_1 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 Reserved	This read-only field is reserved and always has the value zero. ddr_sel Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Please note that changes are not reflected once the field is updated using the GCR register. 0 DDR2/DDR3 mode
24–22 ODT	On Die Termination field Read-only field 000 DISABLED (HiZ)
21–19 DSE	Refer to Table 43-2.
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled

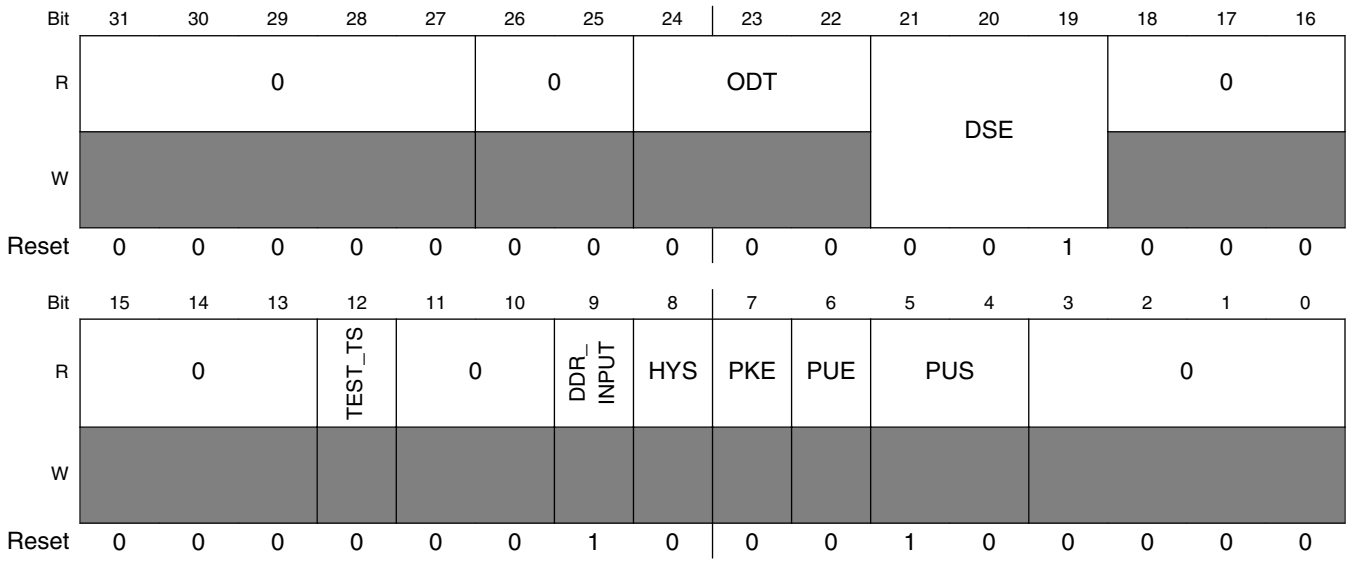
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_1 field descriptions (continued)

Field	Description
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field 1 DDR2 input type
8 HYS	Hyst. Enable Field Read Only Field 0 Hysteresis Disabled
7 PKE	Pull / Keep Enable Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRPKE Please note that changes are not reflected once the field is updated using the GCR register. 0 Pull/Keeper Disabled
6 PUE	Pull / Keep Select Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRPK Please note that changes are not reflected once the field is updated using the GCR register. 0 Keeper
5–4 PUS	Pull Up / Down Config. Field Read Only Field 10 100 [KOhm] Pull Up
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.350 IOMUXC_SW_PAD_CTL_PAD_DRAM_CAS (IOMUXC_SW_PAD_CTL_PAD_DRAM_CAS)

Address: IOMUXC_SW_PAD_CTL_PAD_DRAM_CAS is 53FA_8000h base + 574h offset = 53FA_8574h



IOMUXC_SW_PAD_CTL_PAD_DRAM_CAS field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 Reserved	This read-only field is reserved and always has the value zero. ddr_sel Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Please note that changes are not reflected once the field is updated using the GCR register. 0 DDR2/DDR3 mode
24–22 ODT	On Die Termination field Read-only field 000 DISABLED (HiZ)
21–19 DSE	Refer to Table 43-2.
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled

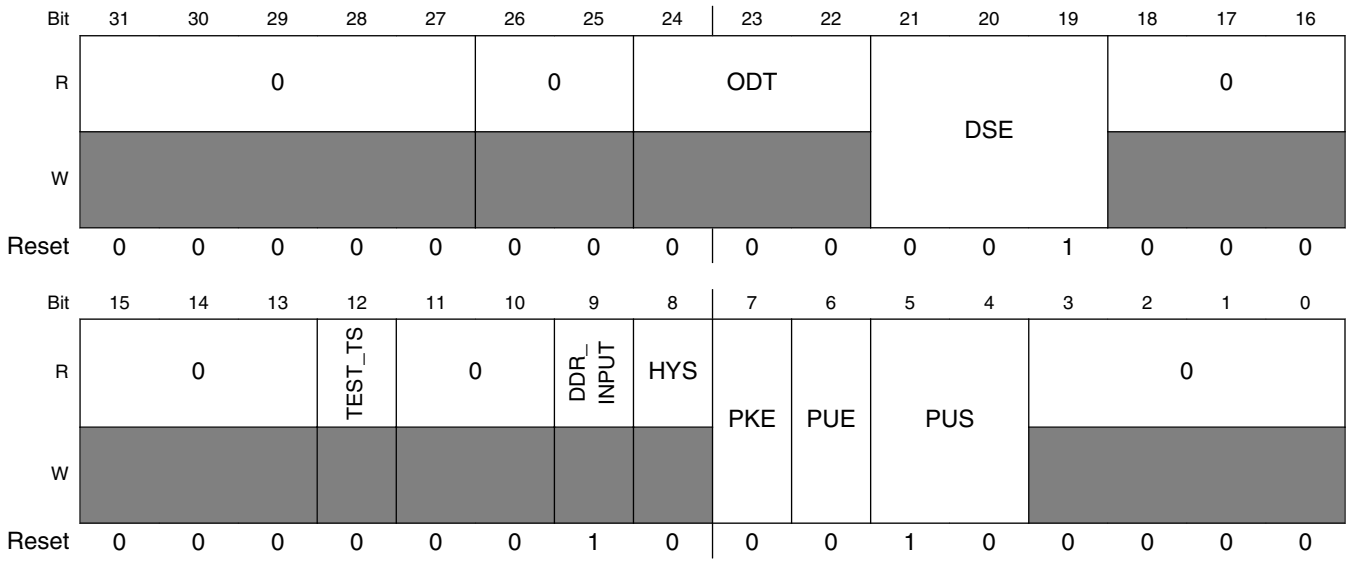
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_CAS field descriptions (continued)

Field	Description
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field 0 CMOS input type
8 HYS	Hyst. Enable Field Read Only Field 0 Hysteresis Disabled
7 PKE	Pull / Keep Enable Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRPKE Please note that changes are not reflected once the field is updated using the GCR register. 0 Pull/Keeper Disabled
6 PUE	Pull / Keep Select Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRPK Please note that changes are not reflected once the field is updated using the GCR register. 0 Keeper
5–4 PUS	Pull Up / Down Config. Field Read Only Field 10 100 [KOhm] Pull Up
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.351 IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_0 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_0)

Address: IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_0 is 53FA_8000h base + 578h offset = 53FA_8578h



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_0 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 Reserved	This read-only field is reserved and always has the value zero. ddr_sel Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Please note that changes are not reflected once the field is updated using the GCR register. 0 DDR2/DDR3 mode
24–22 ODT	On Die Termination field Read-only field 000 DISABLED (HiZ)
21–19 DSE	Refer to Table 43-2.
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled

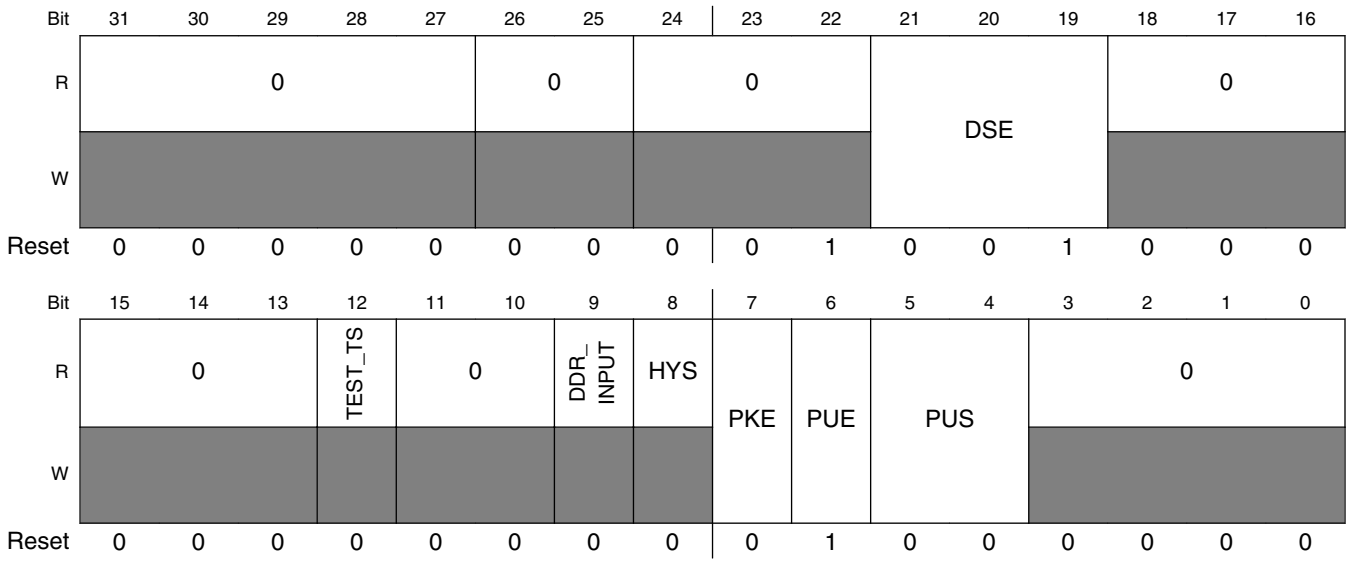
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCLK_0 field descriptions (continued)

Field	Description
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field 1 DDR2 input type
8 HYS	Hyst. Enable Field Read Only Field 0 Hysteresis Disabled
7 PKE	Pull / Keep Enable Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRPKE Please note that changes are not reflected once the field is updated using the GCR register. 0 Pull/Keeper Disabled
6 PUE	Pull / Keep Select Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRPK Please note that changes are not reflected once the field is updated using the GCR register. 0 Keeper
5–4 PUS	Pull Up / Down Config. Field Read Only Field 10 100 [KOhm] Pull Up
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.352 IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS0 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS0)

Address: IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS0 is 53FA_8000h base + 57Ch offset = 53FA_857Ch



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS0 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 Reserved	This read-only field is reserved and always has the value zero. ddr_sel Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Please note that changes are not reflected once the field is updated using the GCR register. 0 DDR2/DDR3 mode
24–22 Reserved	This read-only field is reserved and always has the value zero. On Die Termination field Read-only field 000 DISABLED (HiZ)
21–19 DSE	Refer to Table 43-2.
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled

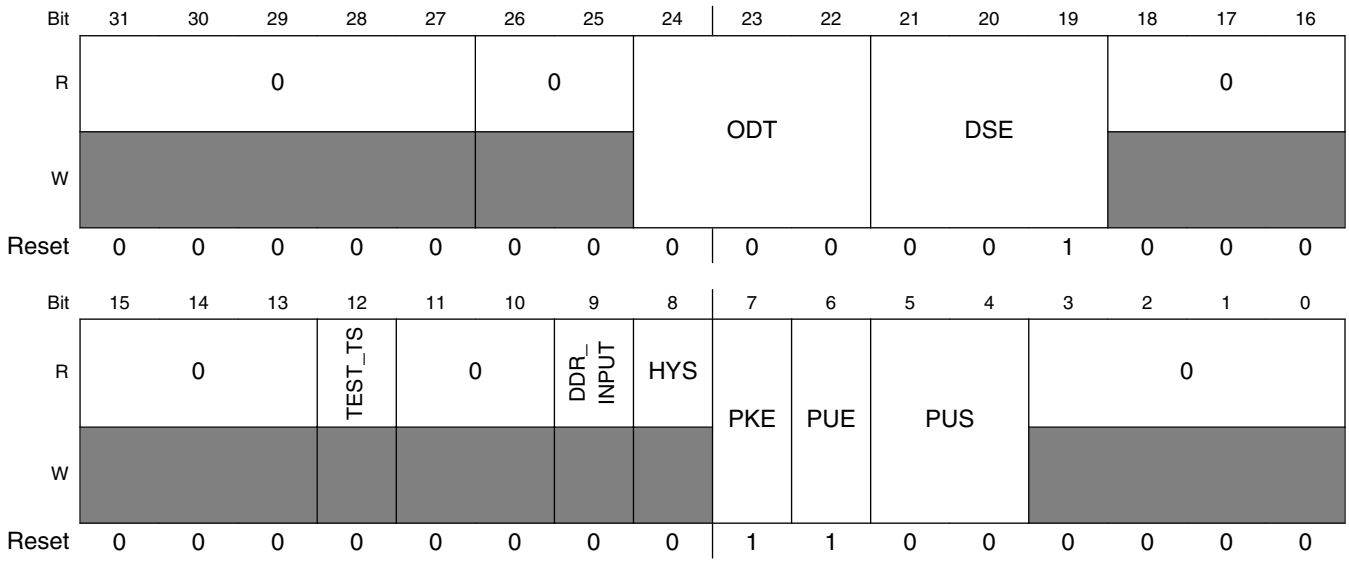
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS0 field descriptions (continued)

Field	Description
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL Please note that changes are not reflected once the field is updated using the GCR register. 0 CMOS input type
8 HYS	Hyst. Enable Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRHYS Please note that changes are not reflected once the field is updated using the GCR register. 0 Hysteresis Disabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DRAM_SDQS0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DRAM_SDQS0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DRAM_SDQS0. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.353 IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT0 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT0)

Address: IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT0 is 53FA_8000h base + 580h offset = 53FA_8580h



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT0 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 Reserved	This read-only field is reserved and always has the value zero. ddr_sel Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Please note that changes are not reflected once the field is updated using the GCR register. 0 DDR2/DDR3 mode
24–22 ODT	On Die Termination Field Read Only Field 000 DISABLED (HiZ)
21–19 DSE	Refer to Table 43-2.
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDODT0 field descriptions (continued)

Field	Description
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field 0 CMOS input type
8 HYS	Hyst. Enable Field Read Only Field 0 Hysteresis Disabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DRAM_SDODT0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DRAM_SDODT0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DRAM_SDODT0. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.354 IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM0 (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM0)

Address: IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM0 is 53FA_8000h base + 584h offset = 53FA_8584h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					0		ODT			DSE			0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	0		DDR_INPUT	HYS	PKE	PUE	PUS		0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM0 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 Reserved	This read-only field is reserved and always has the value zero. ddr_sel Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Please note that changes are not reflected once the field is updated using the GCR register. 0 DDR2/DDR3 mode
24–22 ODT	On Die Termination Field Read Only Field 000 DISABLED (HiZ)
21–19 DSE	Refer to Table 43-2.
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled

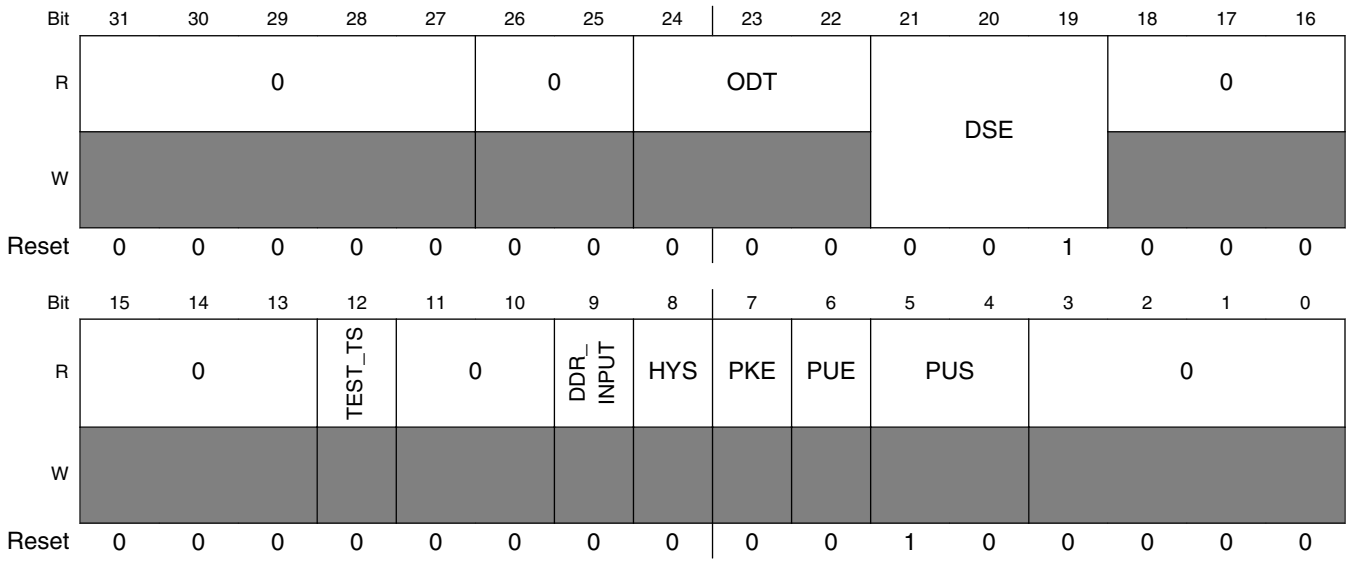
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM0 field descriptions (continued)

Field	Description
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field 0 CMOS input type
8 HYS	Hyst. Enable Field Read Only Field 0 Hysteresis Disabled
7 PKE	Pull / Keep Enable Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRPKE Please note that changes are not reflected once the field is updated using the GCR register. 0 Pull/Keeper Disabled
6 PUE	Pull / Keep Select Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRPK Please note that changes are not reflected once the field is updated using the GCR register. 0 Keeper
5–4 PUS	Pull Up / Down Config. Field Read Only Field 10 100 [KOhm] Pull Up
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.355 IOMUXC_SW_PAD_CTL_PAD_DRAM_RAS (IOMUXC_SW_PAD_CTL_PAD_DRAM_RAS)

Address: IOMUXC_SW_PAD_CTL_PAD_DRAM_RAS is 53FA_8000h base + 588h offset = 53FA_8588h



IOMUXC_SW_PAD_CTL_PAD_DRAM_RAS field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 Reserved	This read-only field is reserved and always has the value zero. ddr_sel Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Please note that changes are not reflected once the field is updated using the GCR register. 0 DDR2/DDR3 mode
24–22 ODT	On Die Termination Field Read Only Field 000 DISABLED (HiZ)
21–19 DSE	Refer to Table 43-2.
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled

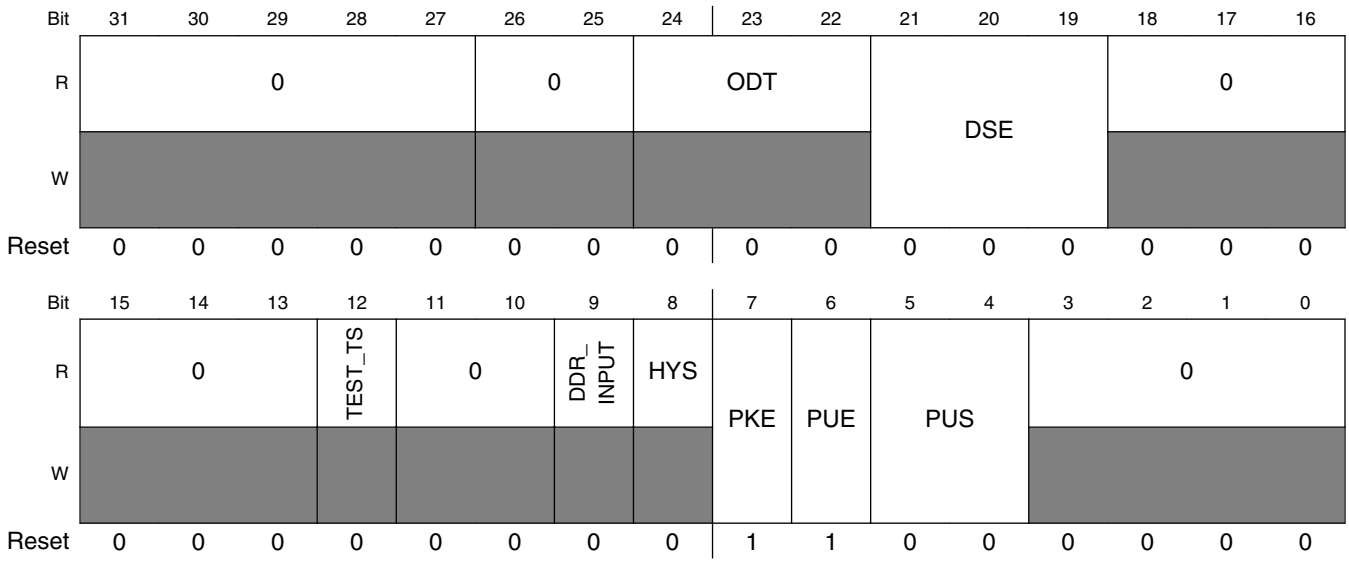
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_RAS field descriptions (continued)

Field	Description
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field 0 CMOS input type
8 HYS	Hyst. Enable Field Read Only Field 0 Hysteresis Disabled
7 PKE	Pull / Keep Enable Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRPKE Please note that changes are not reflected once the field is updated using the GCR register. 0 Pull/Keeper Disabled
6 PUE	Pull / Keep Select Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRPK Please note that changes are not reflected once the field is updated using the GCR register. 0 Keeper
5–4 PUS	Pull Up / Down Config. Field Read Only Field 10 100 [KOhm] Pull Up
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.356 IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE0 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE0)

Address: IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE0 is 53FA_8000h base + 58Ch offset = 53FA_858Ch



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE0 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 Reserved	This read-only field is reserved and always has the value zero. ddr_sel Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Please note that changes are not reflected once the field is updated using the GCR register. 0 DDR2/DDR3 mode
24–22 ODT	On Die Termination Field Read Only Field 000 DISABLED (HiZ)
21–19 DSE	Drive Strength Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_CTLDS Please note that changes are not reflected once the field is updated using the GCR register. 0 DISABLED (HiZ)
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved

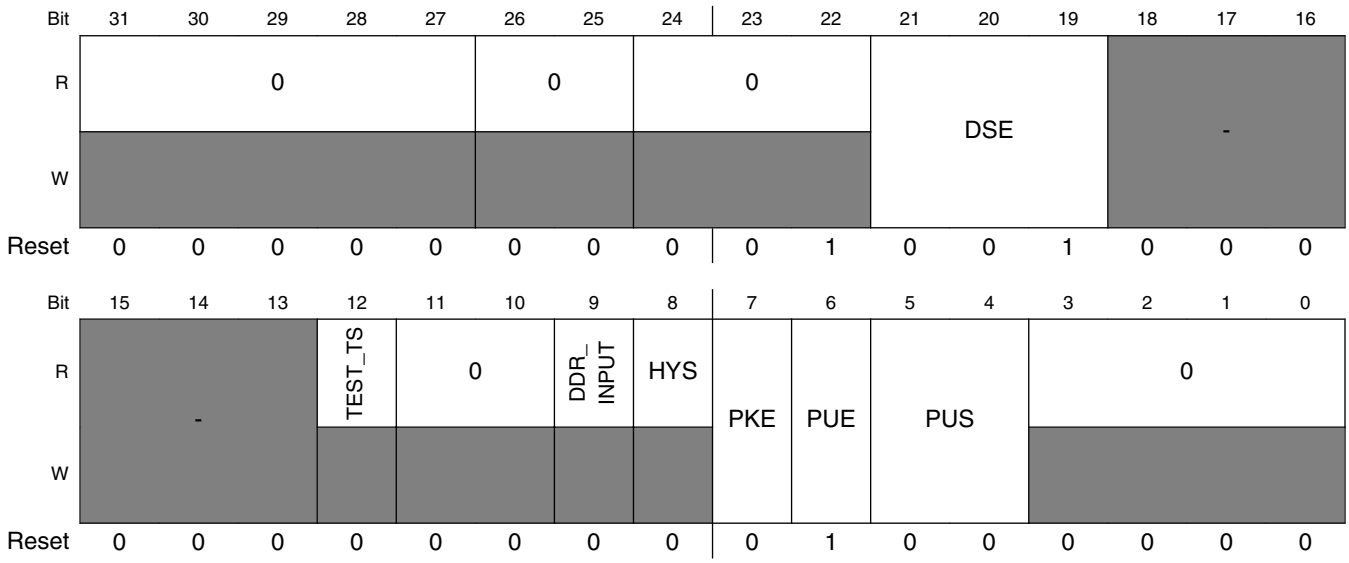
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDCKE0 field descriptions (continued)

Field	Description
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field 0 CMOS input type
8 HYS	Hyst. Enable Field Read Only Field 0 Hysteresis Disabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DRAM_SDCKE0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DRAM_SDCKE0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DRAM_SDCKE0. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.357 IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS1 (IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS1)

Address: IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS1 is 53FA_8000h base + 590h offset = 53FA_8590h



IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS1 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 Reserved	This read-only field is reserved and always has the value zero. ddr_sel Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Please note that changes are not reflected once the field is updated using the GCR register. 0 DDR2/DDR3 mode
24–22 Reserved	This read-only field is reserved and always has the value zero. On Die Termination field Read-only field 000 DISABLED (HiZ)
21–19 DSE	Refer to Table 43-2.
18–13 -	Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_SDQS1 field descriptions (continued)

Field	Description
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL Please note that changes are not reflected once the field is updated using the GCR register. 0 CMOS input type
8 HYS	Hyst. Enable Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRHYS Please note that changes are not reflected once the field is updated using the GCR register. 0 Hysteresis Disabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: DRAM_SDQS1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: DRAM_SDQS1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: DRAM_SDQS1. 00 100 [KOhm] Pull Down 01 47 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.358 IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM1 (IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM1)

Address: IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM1 is 53FA_8000h base + 594h offset = 53FA_8594h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					0		ODT			DSE			0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	0		DDR_INPUT	HYS	PKE	PUE	PUS		0			
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM1 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 Reserved	This read-only field is reserved and always has the value zero. ddr_sel Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE Please note that changes are not reflected once the field is updated using the GCR register. 0 DDR2/DDR3 mode
24–22 ODT	On Die Termination Field Read Only Field 000 DISABLED (HiZ)
21–19 DSE	Refer to Table 43-2.
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled

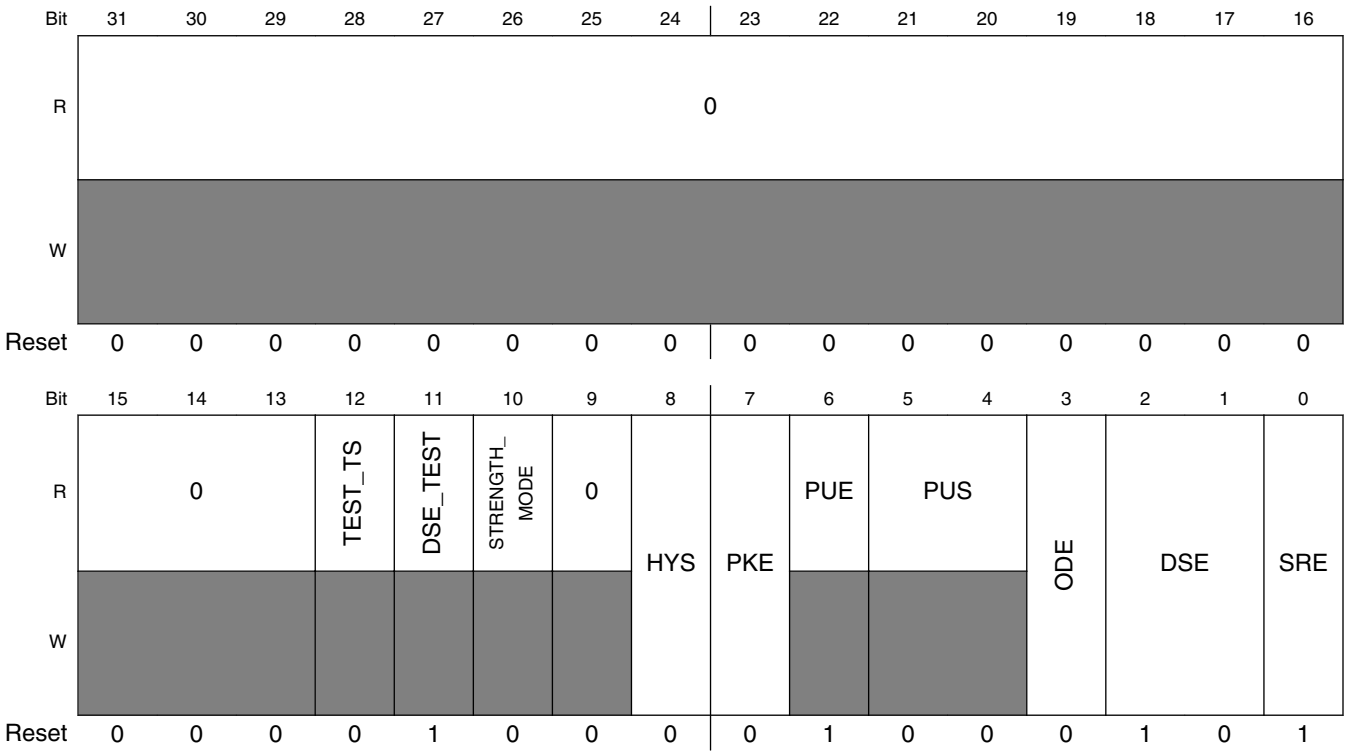
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_DRAM_DQM1 field descriptions (continued)

Field	Description
11–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Read Only Field 0 CMOS input type
8 HYS	Hyst. Enable Field Read Only Field 0 Hysteresis Disabled
7 PKE	Pull / Keep Enable Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRPKE Please note that changes are not reflected once the field is updated using the GCR register. 0 Pull/Keeper Disabled
6 PUE	Pull / Keep Select Field Read Only Field Can be configured using Group Control Register (GCR): IOMUXC_SW_PAD_CTL_GRP_DDRPK Please note that changes are not reflected once the field is updated using the GCR register. 0 Keeper
5–4 PUS	Pull Up / Down Config. Field Read Only Field 10 100 [KOhm] Pull Up
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.359 IOMUXC_SW_PAD_CTL_PAD_PMIC_ON_REQ (IOMUXC_SW_PAD_CTL_PAD_PMIC_ON_REQ)

Address: IOMUXC_SW_PAD_CTL_PAD_PMIC_ON_REQ is 53FA_8000h base + 598h offset = 53FA_8598h



IOMUXC_SW_PAD_CTL_PAD_PMIC_ON_REQ field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

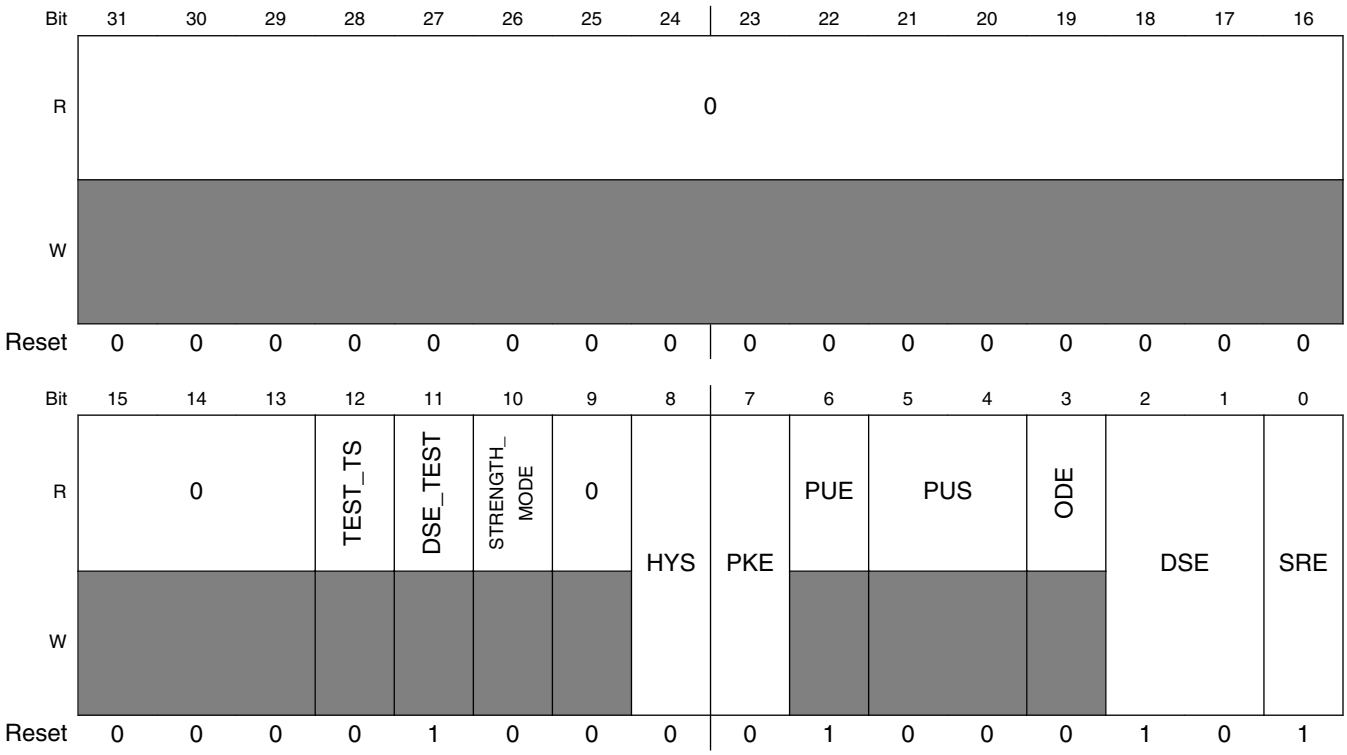
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PMIC_ON_REQ field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: PMIC_ON_REQ. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PMIC_ON_REQ. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Read Only Field 0 Keeper
5–4 PUS	Pull Up / Down Config. Field Read Only Field 00 100 [KOhm] Pull Down
3 ODE	Open Drain Enable Field Select one out of next values for pad: PMIC_ON_REQ. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PMIC_ON_REQ. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: PMIC_ON_REQ. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.360 IOMUXC_SW_PAD_CTL_PAD_PMIC_STBY_REQ (IOMUXC_SW_PAD_CTL_PAD_PMIC_STBY_REQ)

Address: IOMUXC_SW_PAD_CTL_PAD_PMIC_STBY_REQ is 53FA_8000h base + 59Ch offset = 53FA_859Ch



IOMUXC_SW_PAD_CTL_PAD_PMIC_STBY_REQ field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

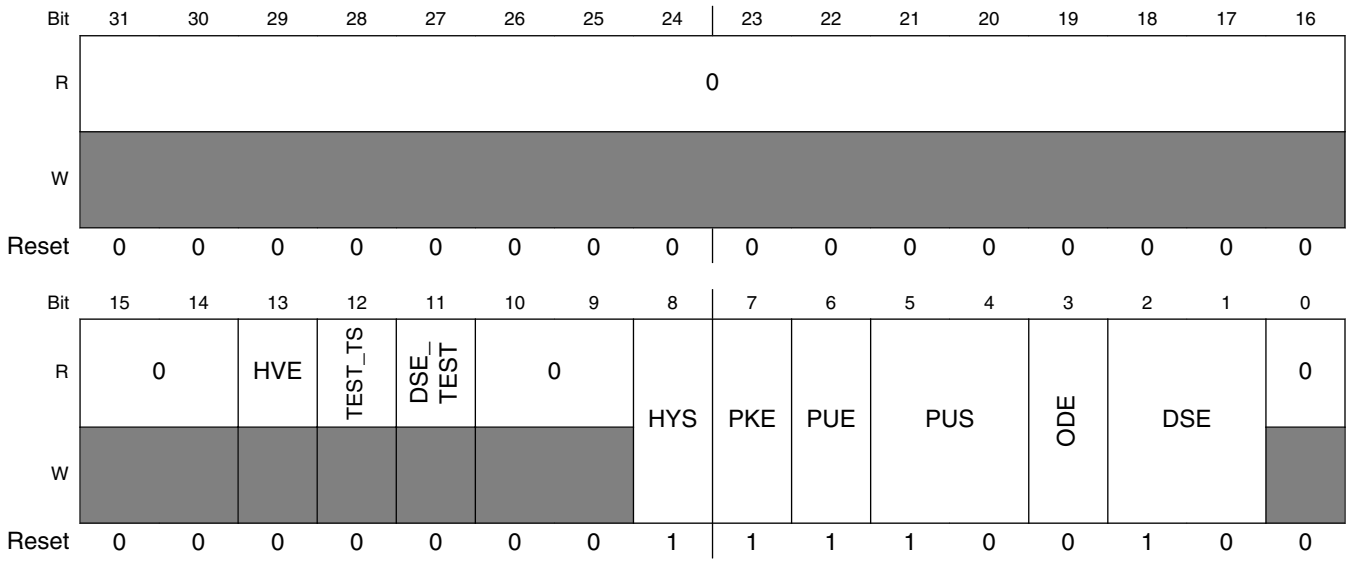
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PMIC_STBY_REQ field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: PMIC_STBY_REQ. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PMIC_STBY_REQ. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Read Only Field 0 Keeper
5–4 PUS	Pull Up / Down Config. Field Read Only Field 00 100 [KOhm] Pull Down
3 ODE	Open Drain Enable Field Read Only Field 0 Open Drain Disabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PMIC_STBY_REQ. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 SRE	Slew Rate Field Select one out of next values for pad: PMIC_STBY_REQ. 0 Slow Slew Rate 1 Fast Slew Rate

43.3.361 IOMUXC_SW_PAD_CTL_PAD_NANDF_CLE (IOMUXC_SW_PAD_CTL_PAD_NANDF_CLE)

Address: IOMUXC_SW_PAD_CTL_PAD_NANDF_CLE is 53FA_8000h base + 5A0h offset = 53FA_85A0h



IOMUXC_SW_PAD_CTL_PAD_NANDF_CLE field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: NANDF_CLE. 0 Hysteresis Disabled 1 Hysteresis Enabled

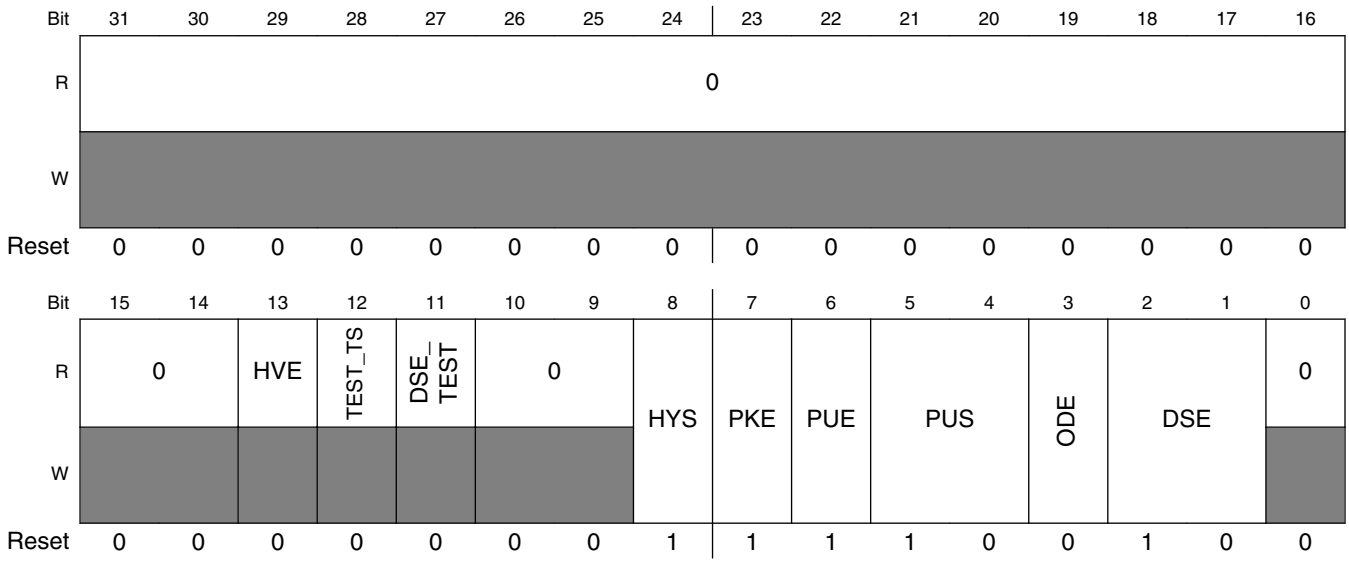
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NANDF_CLE field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: NANDF_CLE. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: NANDF_CLE. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: NANDF_CLE. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: NANDF_CLE. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: NANDF_CLE. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.362 IOMUXC_SW_PAD_CTL_PAD_NANDF_ALE (IOMUXC_SW_PAD_CTL_PAD_NANDF_ALE)

Address: IOMUXC_SW_PAD_CTL_PAD_NANDF_ALE is 53FA_8000h base + 5A4h offset = 53FA_85A4h



IOMUXC_SW_PAD_CTL_PAD_NANDF_ALE field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: NANDF_ALE. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NANDF_ALE field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: NANDF_ALE. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: NANDF_ALE. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: NANDF_ALE. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: NANDF_ALE. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: NANDF_ALE. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.363 IOMUXC_SW_PAD_CTL_PAD_NANDF_WP_B (IOMUXC_SW_PAD_CTL_PAD_NANDF_WP_B)

Address: IOMUXC_SW_PAD_CTL_PAD_NANDF_WP_B is 53FA_8000h base + 5A8h offset = 53FA_85A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_NANDF_WP_B field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: NANDF_WP_B. 0 Hysteresis Disabled 1 Hysteresis Enabled

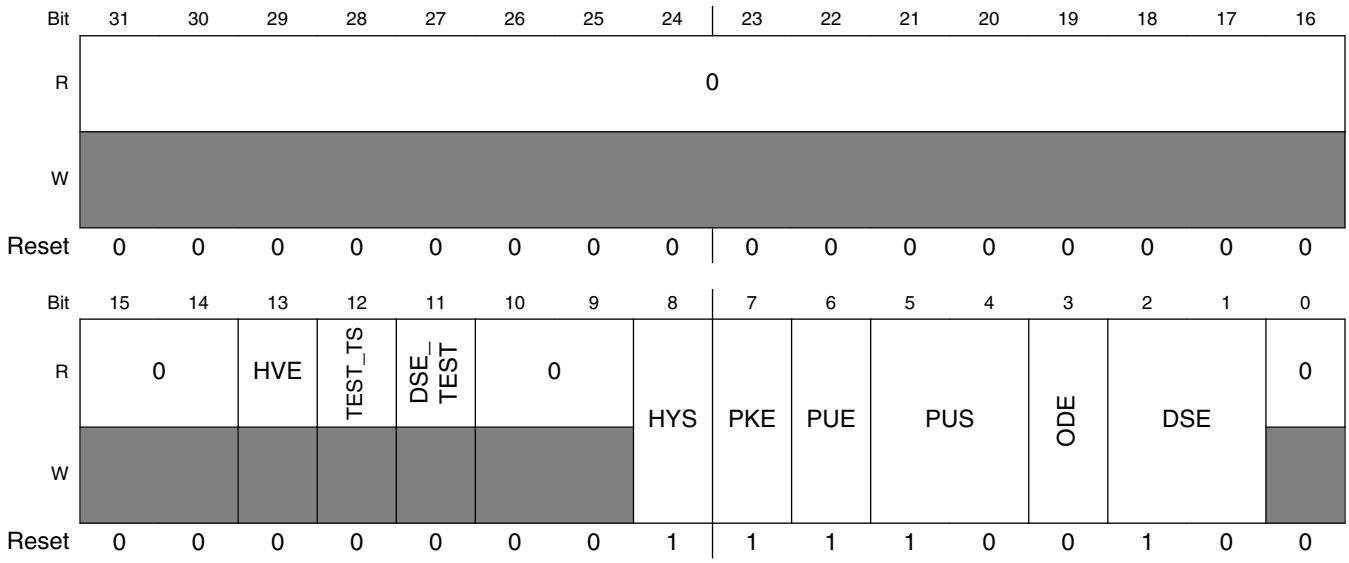
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NANDF_WP_B field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: NANDF_WP_B. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: NANDF_WP_B. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: NANDF_WP_B. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: NANDF_WP_B. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: NANDF_WP_B. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.364 IOMUXC_SW_PAD_CTL_PAD_NANDF_RB0 (IOMUXC_SW_PAD_CTL_PAD_NANDF_RB0)

Address: IOMUXC_SW_PAD_CTL_PAD_NANDF_RB0 is 53FA_8000h base + 5ACh offset = 53FA_85ACh



IOMUXC_SW_PAD_CTL_PAD_NANDF_RB0 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: NANDF_RB0. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NANDF_RB0 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: NANDF_RB0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: NANDF_RB0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: NANDF_RB0. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: NANDF_RB0. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: NANDF_RB0. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.365 IOMUXC_SW_PAD_CTL_PAD_NANDF_CS0 (IOMUXC_SW_PAD_CTL_PAD_NANDF_CS0)

Address: IOMUXC_SW_PAD_CTL_PAD_NANDF_CS0 is 53FA_8000h base + 5B0h offset = 53FA_85B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_NANDF_CS0 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: NANDF_CS0. 0 Hysteresis Disabled 1 Hysteresis Enabled

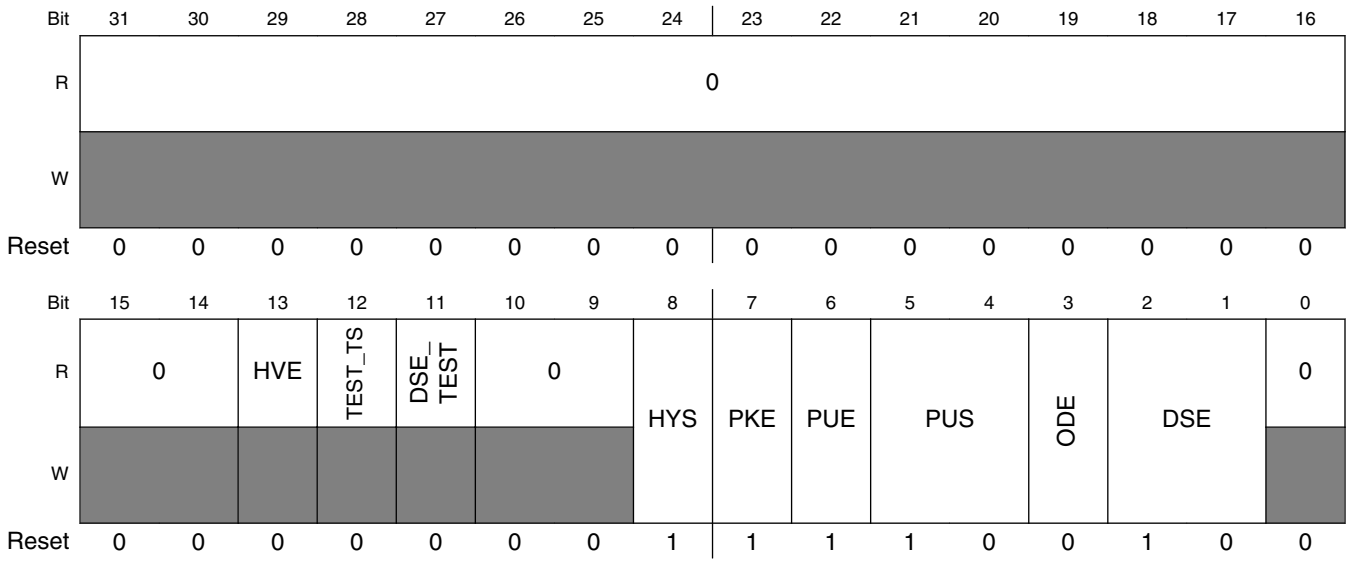
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NANDF_CS0 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: NANDF_CS0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: NANDF_CS0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: NANDF_CS0. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: NANDF_CS0. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: NANDF_CS0. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.366 IOMUXC_SW_PAD_CTL_PAD_NANDF_CS1 (IOMUXC_SW_PAD_CTL_PAD_NANDF_CS1)

Address: IOMUXC_SW_PAD_CTL_PAD_NANDF_CS1 is 53FA_8000h base + 5B4h offset = 53FA_85B4h



IOMUXC_SW_PAD_CTL_PAD_NANDF_CS1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: NANDF_CS1. 0 Hysteresis Disabled 1 Hysteresis Enabled

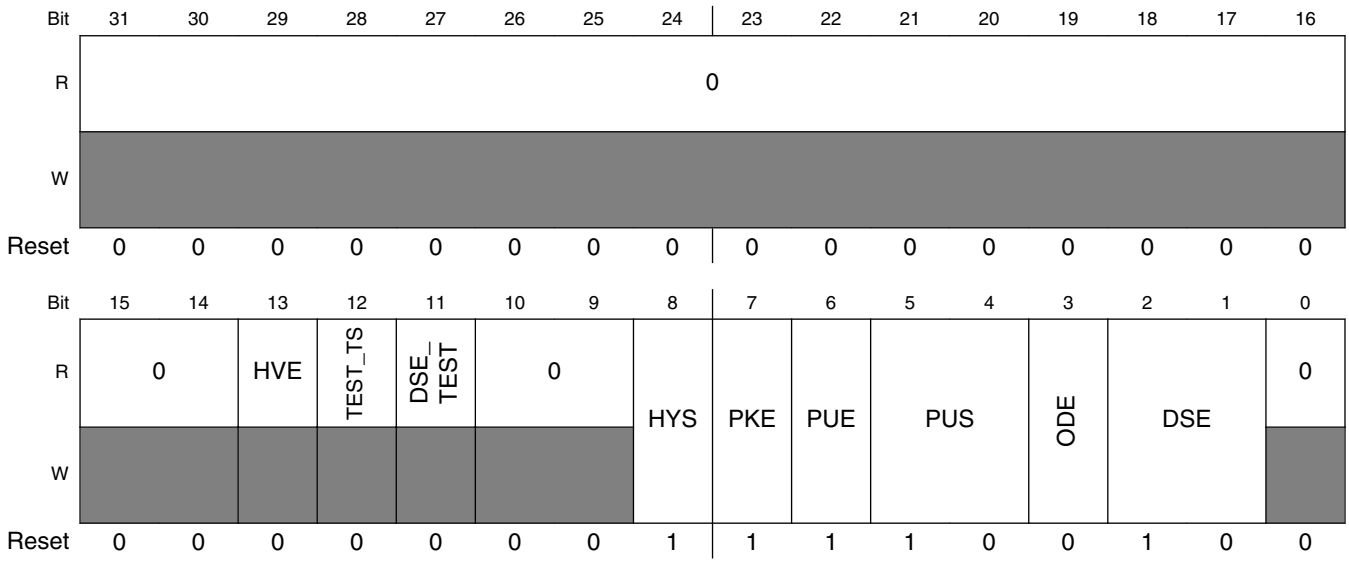
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NANDF_CS1 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: NANDF_CS1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: NANDF_CS1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: NANDF_CS1. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: NANDF_CS1. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: NANDF_CS1. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.367 IOMUXC_SW_PAD_CTL_PAD_NANDF_CS2 (IOMUXC_SW_PAD_CTL_PAD_NANDF_CS2)

Address: IOMUXC_SW_PAD_CTL_PAD_NANDF_CS2 is 53FA_8000h base + 5B8h offset = 53FA_85B8h



IOMUXC_SW_PAD_CTL_PAD_NANDF_CS2 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: NANDF_CS2. 0 Hysteresis Disabled 1 Hysteresis Enabled

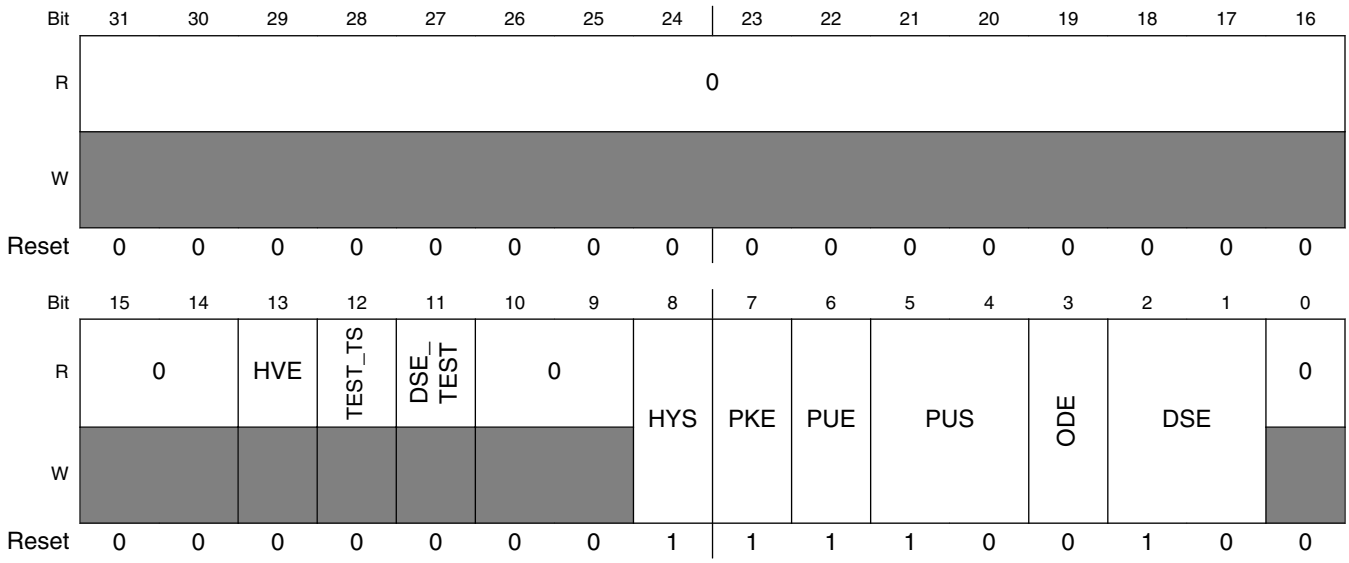
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NANDF_CS2 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: NANDF_CS2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: NANDF_CS2. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: NANDF_CS2. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: NANDF_CS2. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: NANDF_CS2. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.368 IOMUXC_SW_PAD_CTL_PAD_NANDF_CS3 (IOMUXC_SW_PAD_CTL_PAD_NANDF_CS3)

Address: IOMUXC_SW_PAD_CTL_PAD_NANDF_CS3 is 53FA_8000h base + 5BCh offset = 53FA_85BCh



IOMUXC_SW_PAD_CTL_PAD_NANDF_CS3 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: NANDF_CS3. 0 Hysteresis Disabled 1 Hysteresis Enabled

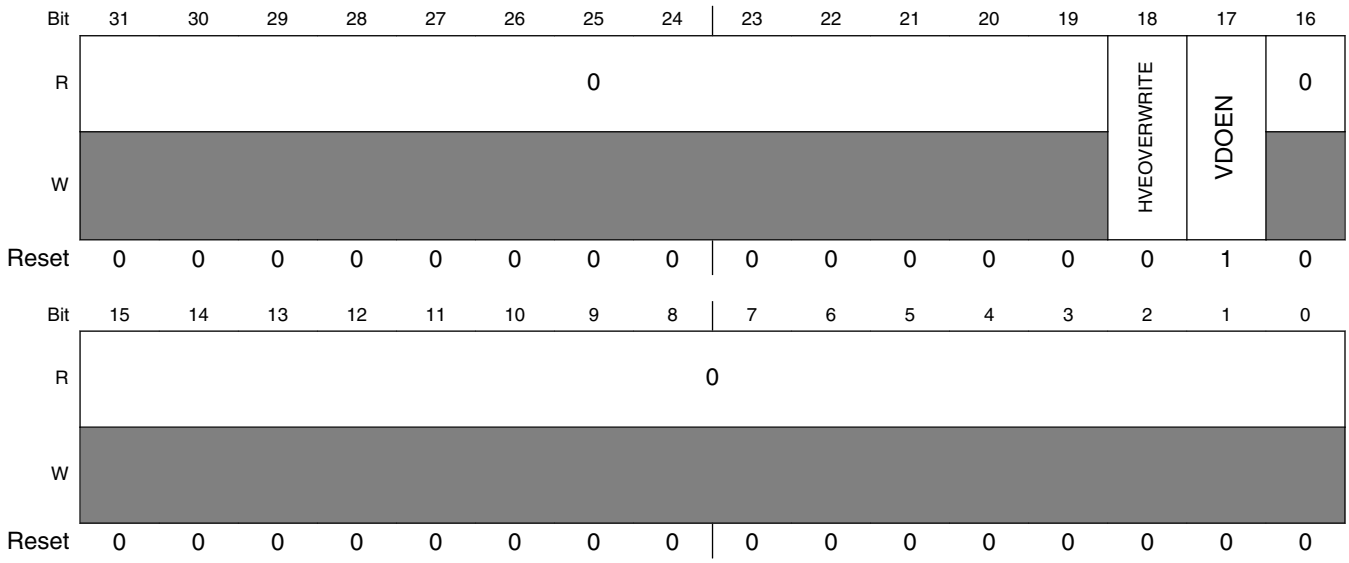
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NANDF_CS3 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: NANDF_CS3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: NANDF_CS3. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: NANDF_CS3. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: NANDF_CS3. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: NANDF_CS3. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.369 IOMUXC_SW_PAD_CTL_PAD_NVCC_NANDF (IOMUXC_SW_PAD_CTL_PAD_NVCC_NANDF)

Address: IOMUXC_SW_PAD_CTL_PAD_NVCC_NANDF is 53FA_8000h base + 5C0h offset = 53FA_85C0h



IOMUXC_SW_PAD_CTL_PAD_NVCC_NANDF field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18 HVEOVERWRITE	HVEOVERWRITE (ipp_owr) Select one out of next values for pad: NVCC_NANDF. This field affects the voltage levels for the following PADS: NANDF_CLE, NANDF_ALE, NANDF_WP_B, NANDF_RB0, NANDF_CS0, NANDF_CS1, NANDF_CS2, NANDF_CS3 0 PAD support high voltage (3.0V-3.6V). Out of reset value is 0. The HVEOVERWRITE field should be updated before automatic voltage detector is disabled. 1 PAD support low voltage (1.65V-3.1V)
17 VDOEN	VDOEN (ipp_oen) Select one out of next values for pad: NVCC_NANDF. 0 Voltage detector output disable and HVEOVERWRITE field will be used. It is recommended to disable the automatic voltage detector after boot and update the HVEOVERWRITE field with the actual I/O supply value. 1 Voltage detector output enable. (default 1)
16–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.370 IOMUXC_SW_PAD_CTL_PAD_FEC_MDIO (IOMUXC_SW_PAD_CTL_PAD_FEC_MDIO)

Address: IOMUXC_SW_PAD_CTL_PAD_FEC_MDIO is 53FA_8000h base + 5C4h offset = 53FA_85C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_FEC_MDIO field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: FEC_MDIO. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_FEC_MDIO field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: FEC_MDIO. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: FEC_MDIO. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: FEC_MDIO. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: FEC_MDIO. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: FEC_MDIO. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.371 IOMUXC_SW_PAD_CTL_PAD_FEC_REF_CLK (IOMUXC_SW_PAD_CTL_PAD_FEC_REF_CLK)

Address: IOMUXC_SW_PAD_CTL_PAD_FEC_REF_CLK is 53FA_8000h base + 5C8h offset = 53FA_85C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_FEC_REF_CLK field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: FEC_REF_CLK. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_FEC_REF_CLK field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: FEC_REF_CLK. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: FEC_REF_CLK. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: FEC_REF_CLK. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: FEC_REF_CLK. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: FEC_REF_CLK. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.372 IOMUXC_SW_PAD_CTL_PAD_FEC_RX_ER (IOMUXC_SW_PAD_CTL_PAD_FEC_RX_ER)

Address: IOMUXC_SW_PAD_CTL_PAD_FEC_RX_ER is 53FA_8000h base + 5CCh offset = 53FA_85CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_FEC_RX_ER field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: FEC_RX_ER. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_FEC_RX_ER field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: FEC_RX_ER. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: FEC_RX_ER. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: FEC_RX_ER. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: FEC_RX_ER. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: FEC_RX_ER. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.373 IOMUXC_SW_PAD_CTL_PAD_FEC_CRS_DV (IOMUXC_SW_PAD_CTL_PAD_FEC_CRS_DV)

Address: IOMUXC_SW_PAD_CTL_PAD_FEC_CRS_DV is 53FA_8000h base + 5D0h offset = 53FA_85D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_FEC_CRS_DV field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: FEC_CRS_DV. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_FEC_CRS_DV field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: FEC_CRS_DV. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: FEC_CRS_DV. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: FEC_CRS_DV. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: FEC_CRS_DV. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: FEC_CRS_DV. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.374 IOMUXC_SW_PAD_CTL_PAD_FEC_RXD1 (IOMUXC_SW_PAD_CTL_PAD_FEC_RXD1)

Address: IOMUXC_SW_PAD_CTL_PAD_FEC_RXD1 is 53FA_8000h base + 5D4h offset = 53FA_85D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_FEC_RXD1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: FEC_RXD1. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_FEC_RXD1 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: FEC_RXD1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: FEC_RXD1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: FEC_RXD1. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: FEC_RXD1. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: FEC_RXD1. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.375 IOMUXC_SW_PAD_CTL_PAD_FEC_RXD0 (IOMUXC_SW_PAD_CTL_PAD_FEC_RXD0)

Address: IOMUXC_SW_PAD_CTL_PAD_FEC_RXD0 is 53FA_8000h base + 5D8h offset = 53FA_85D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_FEC_RXD0 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: FEC_RXD0. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_FEC_RXD0 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: FEC_RXD0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: FEC_RXD0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: FEC_RXD0. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: FEC_RXD0. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: FEC_RXD0. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.376 IOMUXC_SW_PAD_CTL_PAD_FEC_TX_EN (IOMUXC_SW_PAD_CTL_PAD_FEC_TX_EN)

Address: IOMUXC_SW_PAD_CTL_PAD_FEC_TX_EN is 53FA_8000h base + 5DCh offset = 53FA_85DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_FEC_TX_EN field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: FEC_TX_EN. 0 Hysteresis Disabled 1 Hysteresis Enabled

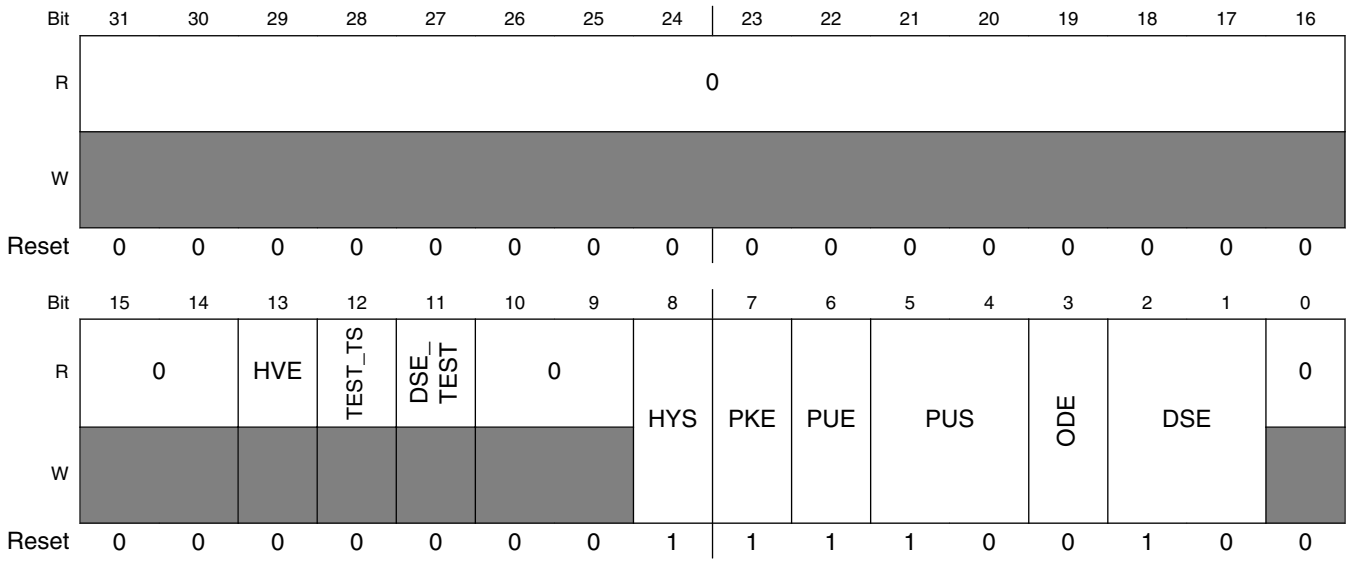
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_FEC_TX_EN field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: FEC_TX_EN. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: FEC_TX_EN. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: FEC_TX_EN. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: FEC_TX_EN. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: FEC_TX_EN. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.377 IOMUXC_SW_PAD_CTL_PAD_FEC_TXD1 (IOMUXC_SW_PAD_CTL_PAD_FEC_TXD1)

Address: IOMUXC_SW_PAD_CTL_PAD_FEC_TXD1 is 53FA_8000h base + 5E0h offset = 53FA_85E0h



IOMUXC_SW_PAD_CTL_PAD_FEC_TXD1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: FEC_TXD1. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_FEC_TXD1 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: FEC_TXD1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: FEC_TXD1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: FEC_TXD1. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: FEC_TXD1. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: FEC_TXD1. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.378 IOMUXC_SW_PAD_CTL_PAD_FEC_TXD0 (IOMUXC_SW_PAD_CTL_PAD_FEC_TXD0)

Address: IOMUXC_SW_PAD_CTL_PAD_FEC_TXD0 is 53FA_8000h base + 5E4h offset = 53FA_85E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_FEC_TXD0 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: FEC_TXD0. 0 Hysteresis Disabled 1 Hysteresis Enabled

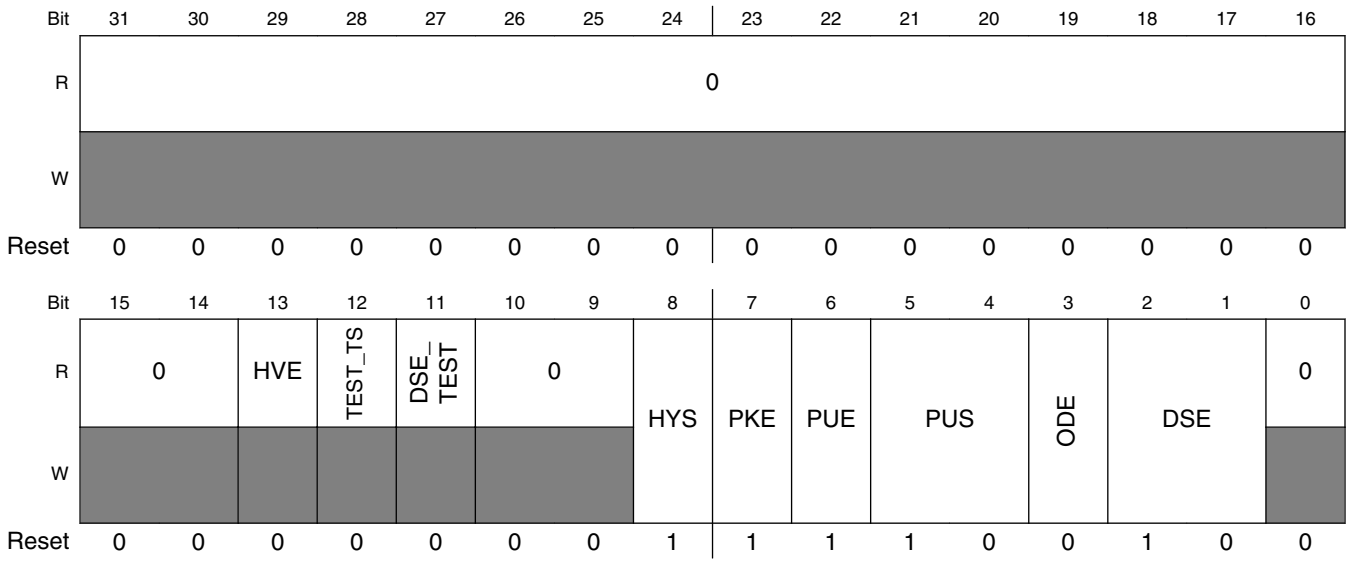
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_FEC_TXD0 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: FEC_TXD0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: FEC_TXD0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: FEC_TXD0. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: FEC_TXD0. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: FEC_TXD0. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.379 IOMUXC_SW_PAD_CTL_PAD_FEC_MDC (IOMUXC_SW_PAD_CTL_PAD_FEC_MDC)

Address: IOMUXC_SW_PAD_CTL_PAD_FEC_MDC is 53FA_8000h base + 5E8h offset = 53FA_85E8h



IOMUXC_SW_PAD_CTL_PAD_FEC_MDC field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: FEC_MDC. 0 Hysteresis Disabled 1 Hysteresis Enabled

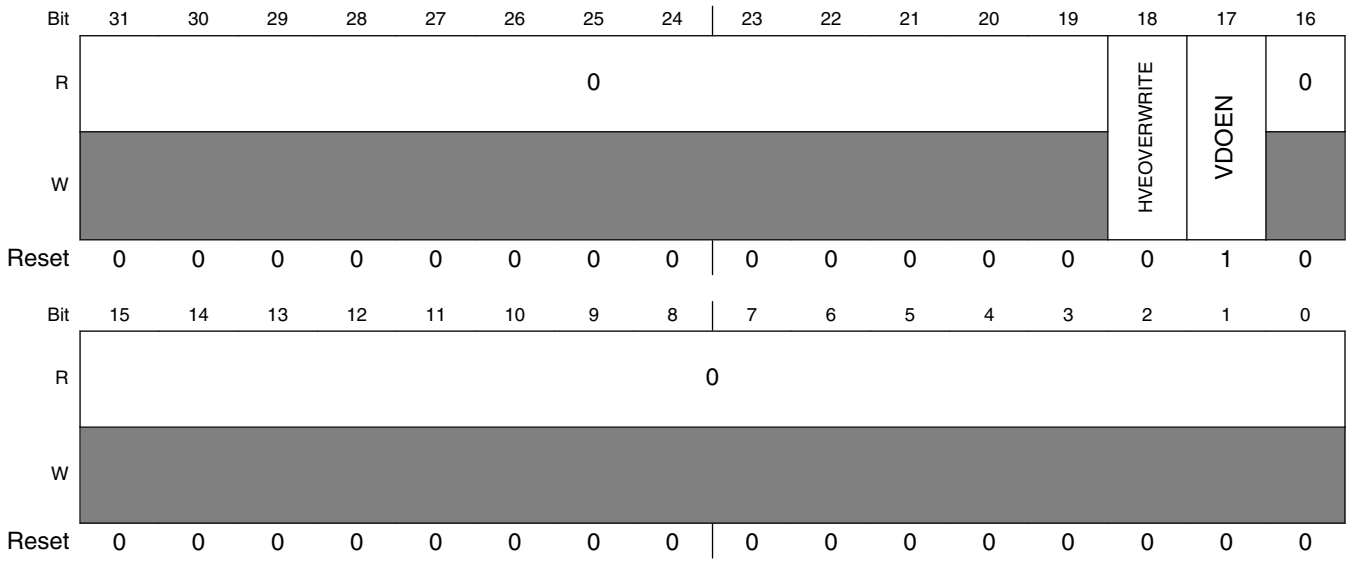
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_FEC_MDC field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: FEC_MDC. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: FEC_MDC. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: FEC_MDC. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: FEC_MDC. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: FEC_MDC. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.380 IOMUXC_SW_PAD_CTL_PAD_NVCC_FEC (IOMUXC_SW_PAD_CTL_PAD_NVCC_FEC)

Address: IOMUXC_SW_PAD_CTL_PAD_NVCC_FEC is 53FA_8000h base + 5ECh offset = 53FA_85ECh

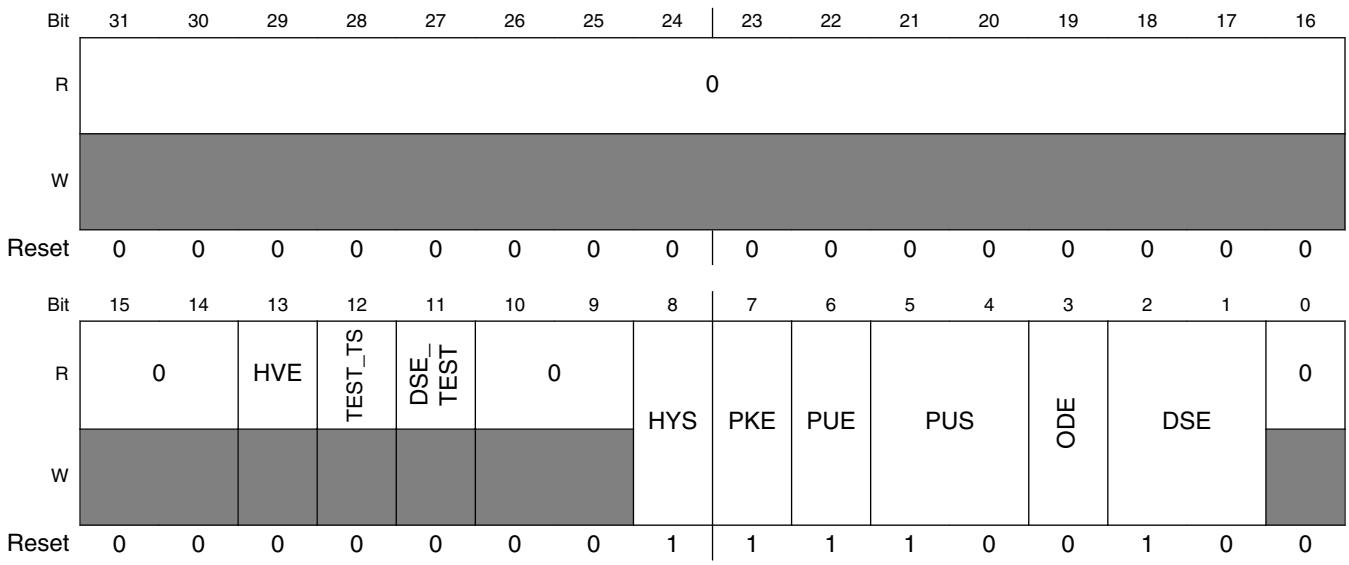


IOMUXC_SW_PAD_CTL_PAD_NVCC_FEC field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18 HVEOVERWRITE	HVEOVERWRITE (ipp_owr) Select one out of next values for pad: NVCC_FEC. This field affects the voltage levels for the following PADS: FEC_MDC, FEC_TXD0, FEC_TXD1, FEC_TX_EN, FEC_RXD0, FEC_RXD1, FEC_CRS_DV, FEC_RX_ER, FEC_REF_CLK, FEC_MDIO 0 PAD support high voltage (3.0V-3.6V). Out of reset value is 0. The HVEOVERWRITE field should be updated before automatic voltage detector is disabled. 1 PAD support low voltage (1.65V-3.1V)
17 VDOEN	VDOEN (ipp_oen) Select one out of next values for pad: NVCC_FEC. 0 Voltage detector output disable and HVEOVERWRITE field will be used. It is recommended to disable the automatic voltage detector after boot and update the HVEOVERWRITE field with the actual I/O supply value. 1 Voltage detector output enable. (default 1)
16–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.381 IOMUXC_SW_PAD_CTL_PAD_PATA_DIOW (IOMUXC_SW_PAD_CTL_PAD_PATA_DIOW)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DIOW is 53FA_8000h base + 5F0h offset = 53FA_85F0h



IOMUXC_SW_PAD_CTL_PAD_PATA_DIOW field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DIOW. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DIOW field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DIOW. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DIOW. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DIOW. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DIOW. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DIOW. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.382 IOMUXC_SW_PAD_CTL_PAD_PATA_DMACK (IOMUXC_SW_PAD_CTL_PAD_PATA_DMACK)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DMACK is 53FA_8000h base + 5F4h offset = 53FA_85F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_PATA_DMACK field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DMACK. 0 Hysteresis Disabled 1 Hysteresis Enabled

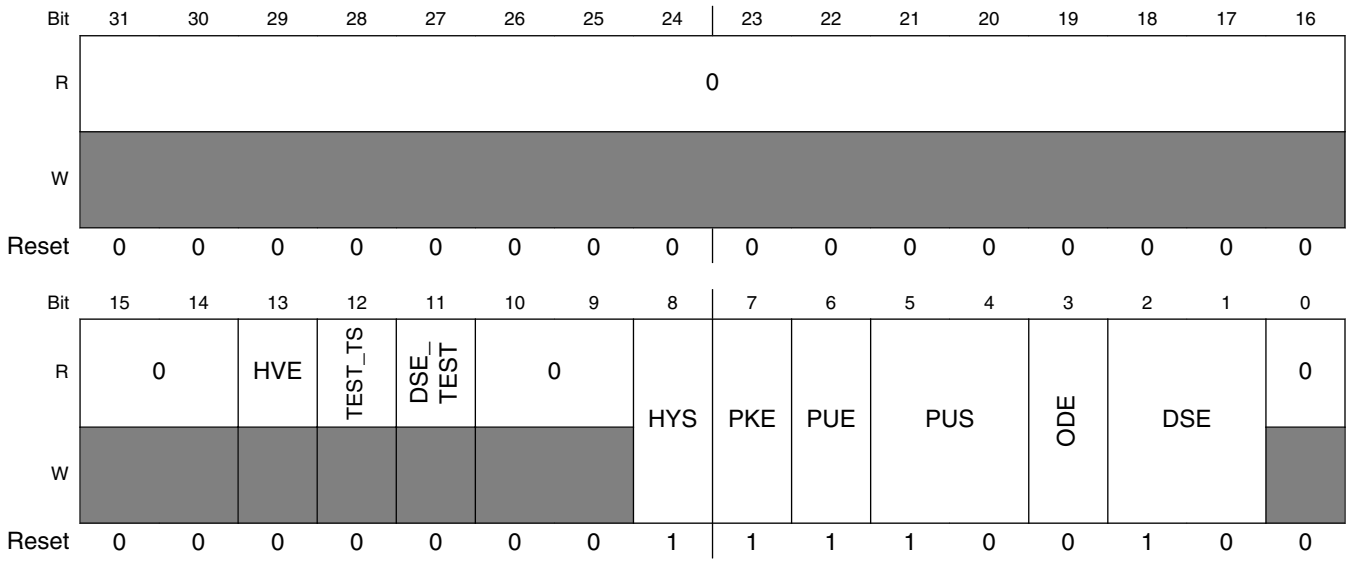
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DMACK field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DMACK. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DMACK. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DMACK. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DMACK. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DMACK. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.383 IOMUXC_SW_PAD_CTL_PAD_PATA_DMARQ (IOMUXC_SW_PAD_CTL_PAD_PATA_DMARQ)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DMARQ is 53FA_8000h base + 5F8h offset = 53FA_85F8h



IOMUXC_SW_PAD_CTL_PAD_PATA_DMARQ field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DMARQ. 0 Hysteresis Disabled 1 Hysteresis Enabled

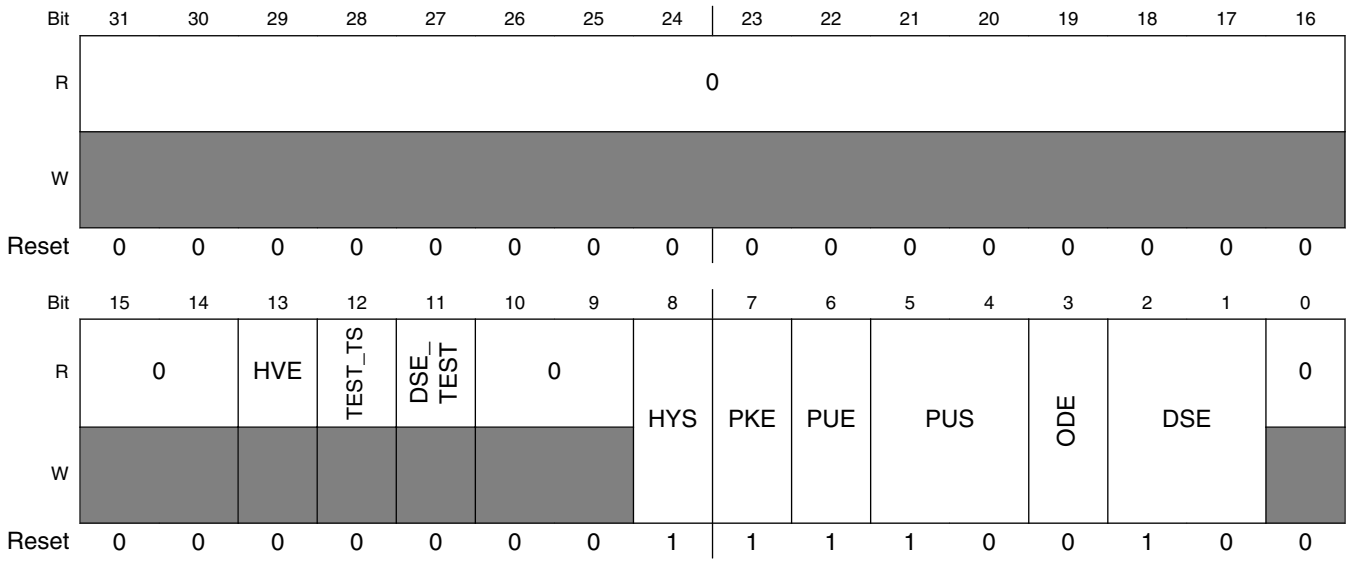
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DMARQ field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DMARQ. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DMARQ. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DMARQ. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DMARQ. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DMARQ. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.384 IOMUXC_SW_PAD_CTL_PAD_PATA_BUFFER_EN (IOMUXC_SW_PAD_CTL_PAD_PATA_BUFFER_EN)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_BUFFER_EN is 53FA_8000h base + 5FCh offset = 53FA_85FCh



IOMUXC_SW_PAD_CTL_PAD_PATA_BUFFER_EN field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_BUFFER_EN. 0 Hysteresis Disabled 1 Hysteresis Enabled

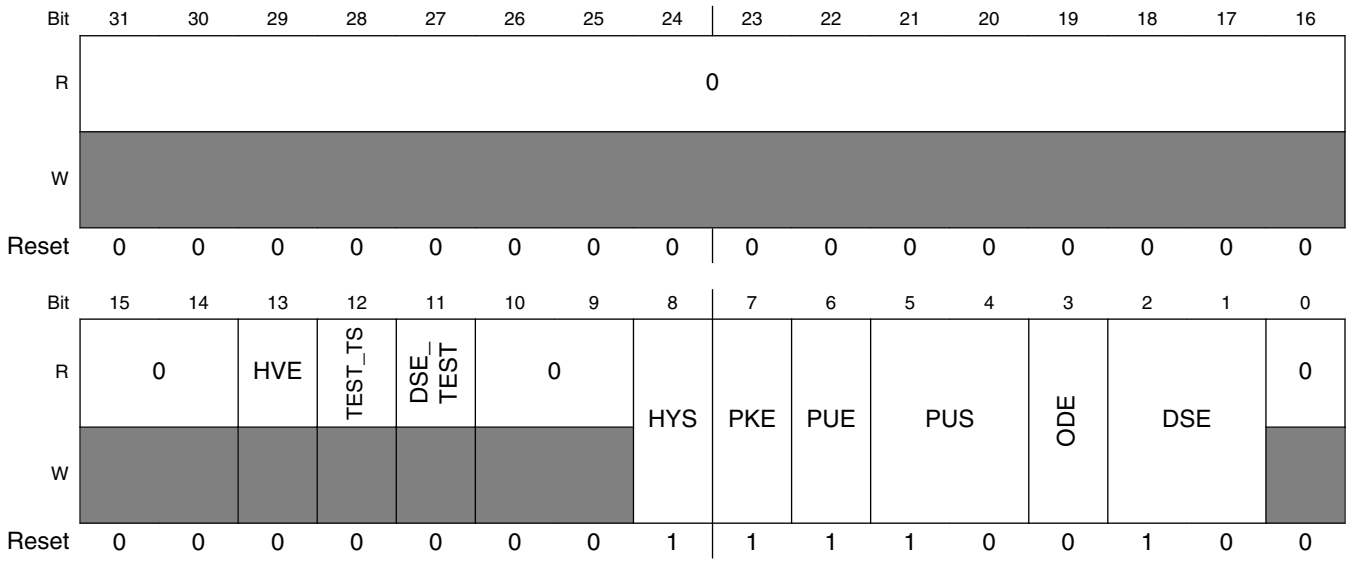
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_BUFFER_EN field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_BUFFER_EN. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_BUFFER_EN. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_BUFFER_EN. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_BUFFER_EN. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_BUFFER_EN. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.385 IOMUXC_SW_PAD_CTL_PAD_PATA_INTRQ (IOMUXC_SW_PAD_CTL_PAD_PATA_INTRQ)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_INTRQ is 53FA_8000h base + 600h offset = 53FA_8600h



IOMUXC_SW_PAD_CTL_PAD_PATA_INTRQ field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_INTRQ. 0 Hysteresis Disabled 1 Hysteresis Enabled

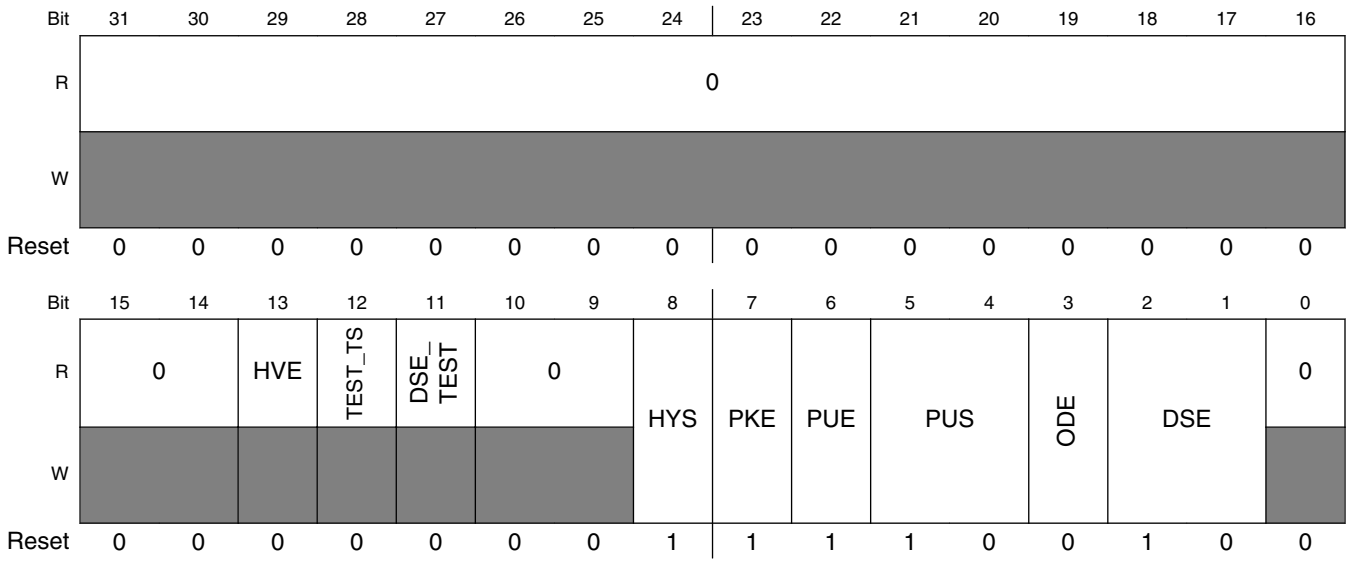
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_INTRQ field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_INTRQ. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_INTRQ. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_INTRQ. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_INTRQ. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_INTRQ. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.386 IOMUXC_SW_PAD_CTL_PAD_PATA_DIOR (IOMUXC_SW_PAD_CTL_PAD_PATA_DIOR)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DIOR is 53FA_8000h base + 604h offset = 53FA_8604h



IOMUXC_SW_PAD_CTL_PAD_PATA_DIOR field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DIOR. 0 Hysteresis Disabled 1 Hysteresis Enabled

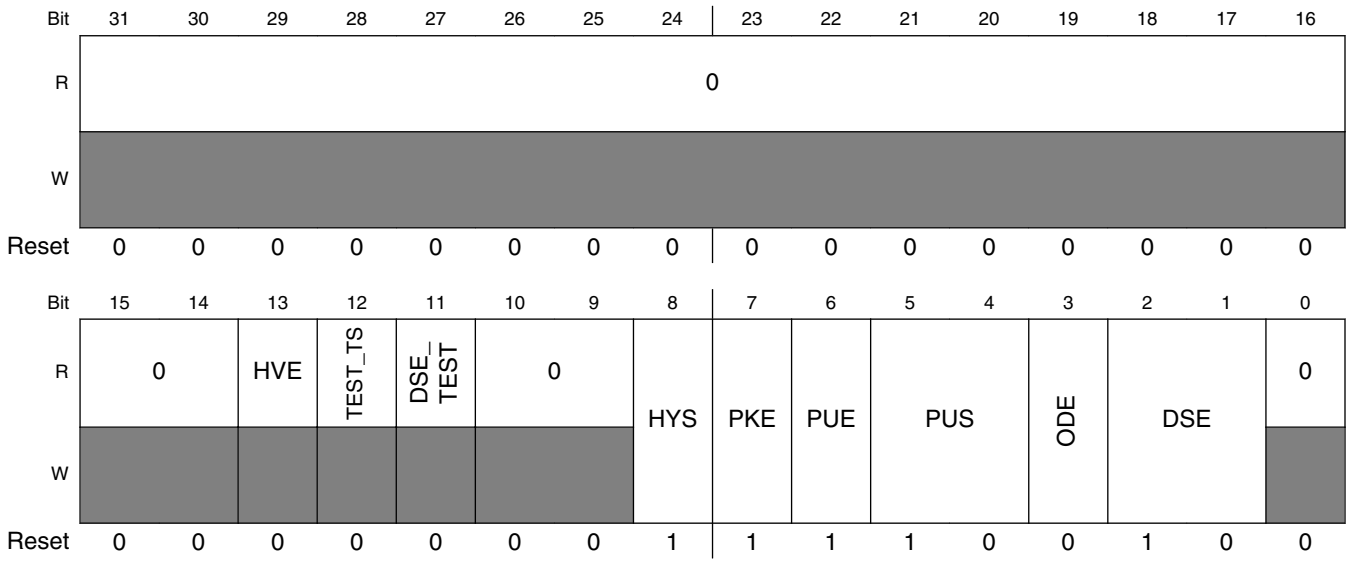
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DIOR field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DIOR. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DIOR. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DIOR. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DIOR. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DIOR. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.387 IOMUXC_SW_PAD_CTL_PAD_PATA_RESET_B (IOMUXC_SW_PAD_CTL_PAD_PATA_RESET_B)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_RESET_B is 53FA_8000h base + 608h offset = 53FA_8608h



IOMUXC_SW_PAD_CTL_PAD_PATA_RESET_B field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_RESET_B. 0 Hysteresis Disabled 1 Hysteresis Enabled

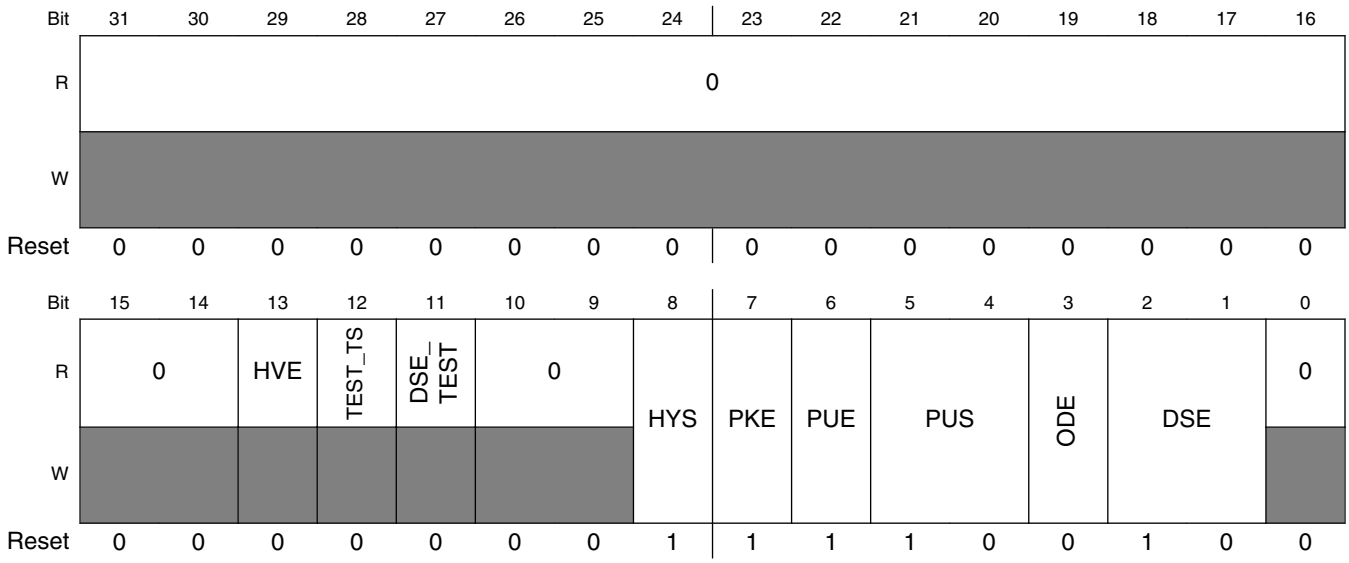
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_RESET_B field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_RESET_B. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_RESET_B. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_RESET_B. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_RESET_B. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_RESET_B. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.388 IOMUXC_SW_PAD_CTL_PAD_PATA_IORDY (IOMUXC_SW_PAD_CTL_PAD_PATA_IORDY)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_IORDY is 53FA_8000h base + 60Ch offset = 53FA_860Ch



IOMUXC_SW_PAD_CTL_PAD_PATA_IORDY field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_IORDY. 0 Hysteresis Disabled 1 Hysteresis Enabled

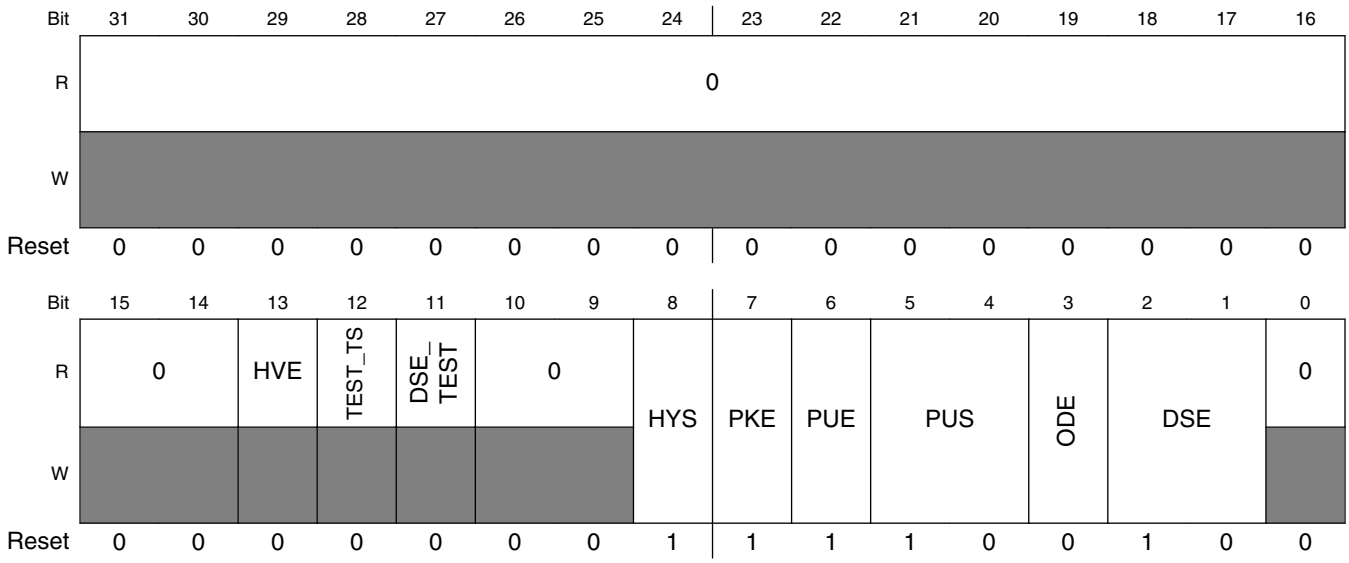
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_IORDY field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_IORDY. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_IORDY. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_IORDY. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_IORDY. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_IORDY. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.389 IOMUXC_SW_PAD_CTL_PAD_PATA_DA_0 (IOMUXC_SW_PAD_CTL_PAD_PATA_DA_0)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DA_0 is 53FA_8000h base + 610h offset = 53FA_8610h



IOMUXC_SW_PAD_CTL_PAD_PATA_DA_0 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DA_0. 0 Hysteresis Disabled 1 Hysteresis Enabled

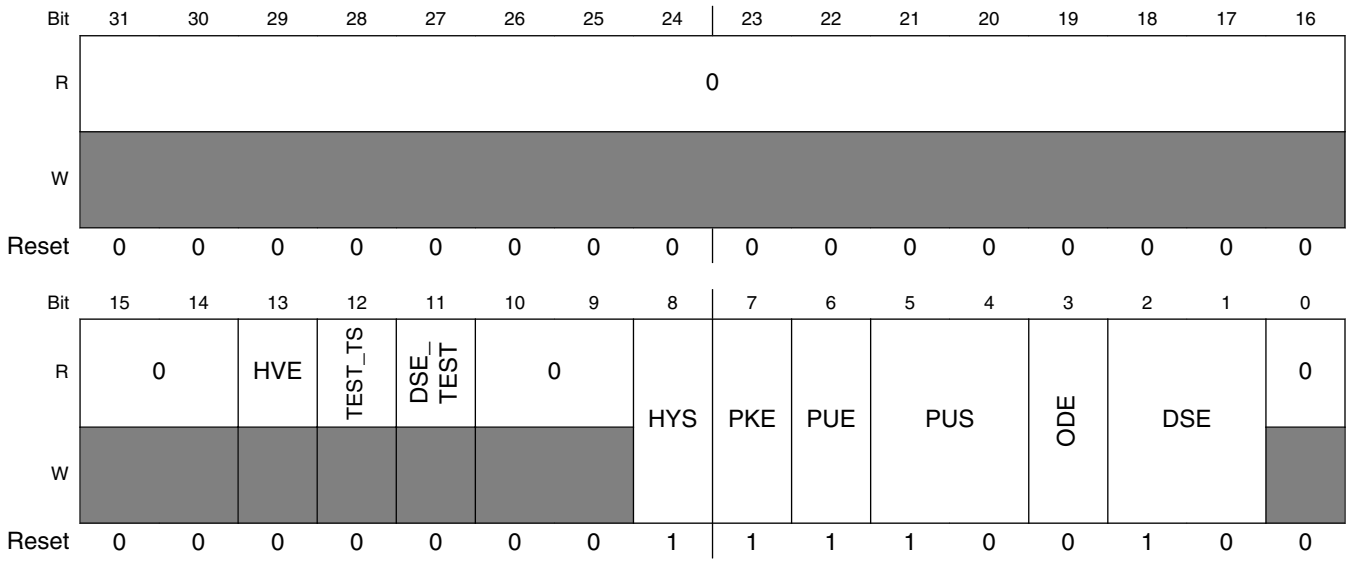
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DA_0 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DA_0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DA_0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DA_0. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DA_0. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DA_0. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.390 IOMUXC_SW_PAD_CTL_PAD_PATA_DA_1 (IOMUXC_SW_PAD_CTL_PAD_PATA_DA_1)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DA_1 is 53FA_8000h base + 614h offset = 53FA_8614h



IOMUXC_SW_PAD_CTL_PAD_PATA_DA_1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DA_1. 0 Hysteresis Disabled 1 Hysteresis Enabled

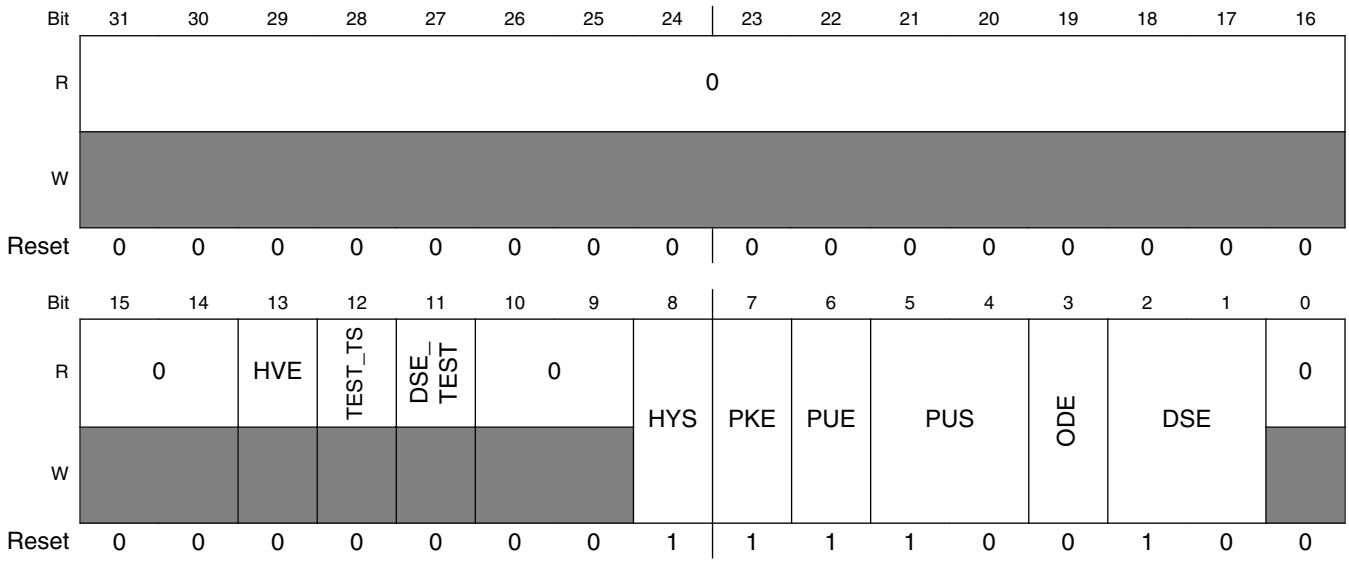
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DA_1 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DA_1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DA_1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DA_1. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DA_1. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DA_1. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.391 IOMUXC_SW_PAD_CTL_PAD_PATA_DA_2 (IOMUXC_SW_PAD_CTL_PAD_PATA_DA_2)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DA_2 is 53FA_8000h base + 618h offset = 53FA_8618h



IOMUXC_SW_PAD_CTL_PAD_PATA_DA_2 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DA_2. 0 Hysteresis Disabled 1 Hysteresis Enabled

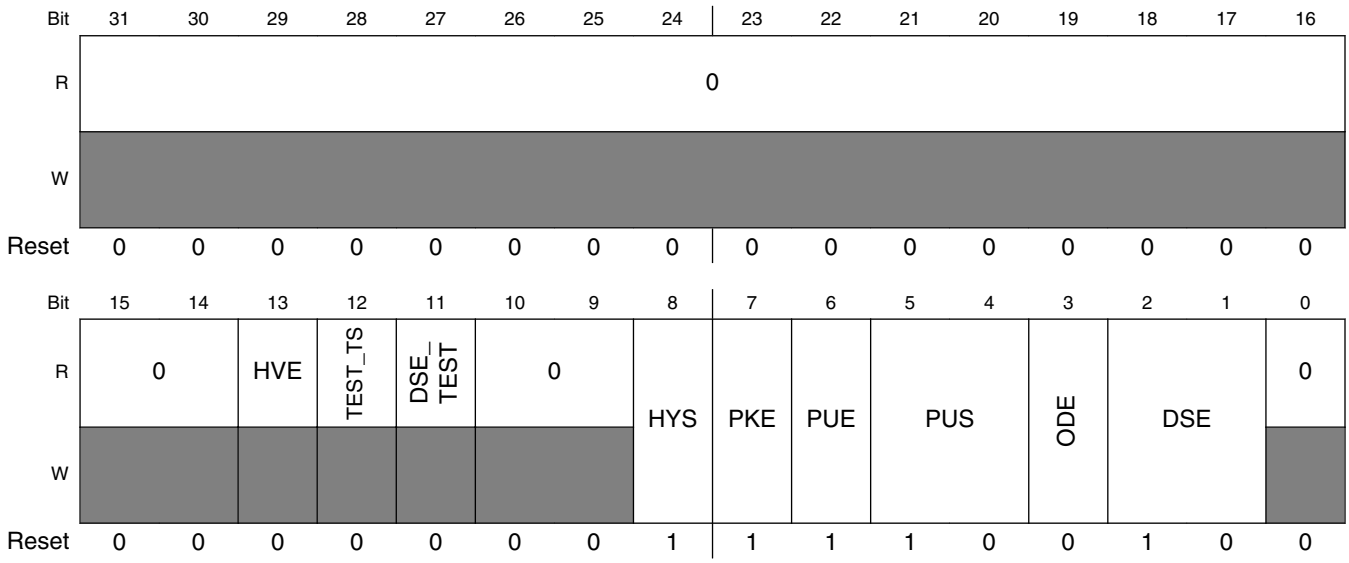
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DA_2 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DA_2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DA_2. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DA_2. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DA_2. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DA_2. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.392 IOMUXC_SW_PAD_CTL_PAD_PATA_CS_0 (IOMUXC_SW_PAD_CTL_PAD_PATA_CS_0)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_CS_0 is 53FA_8000h base + 61Ch offset = 53FA_861Ch



IOMUXC_SW_PAD_CTL_PAD_PATA_CS_0 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_CS_0. 0 Hysteresis Disabled 1 Hysteresis Enabled

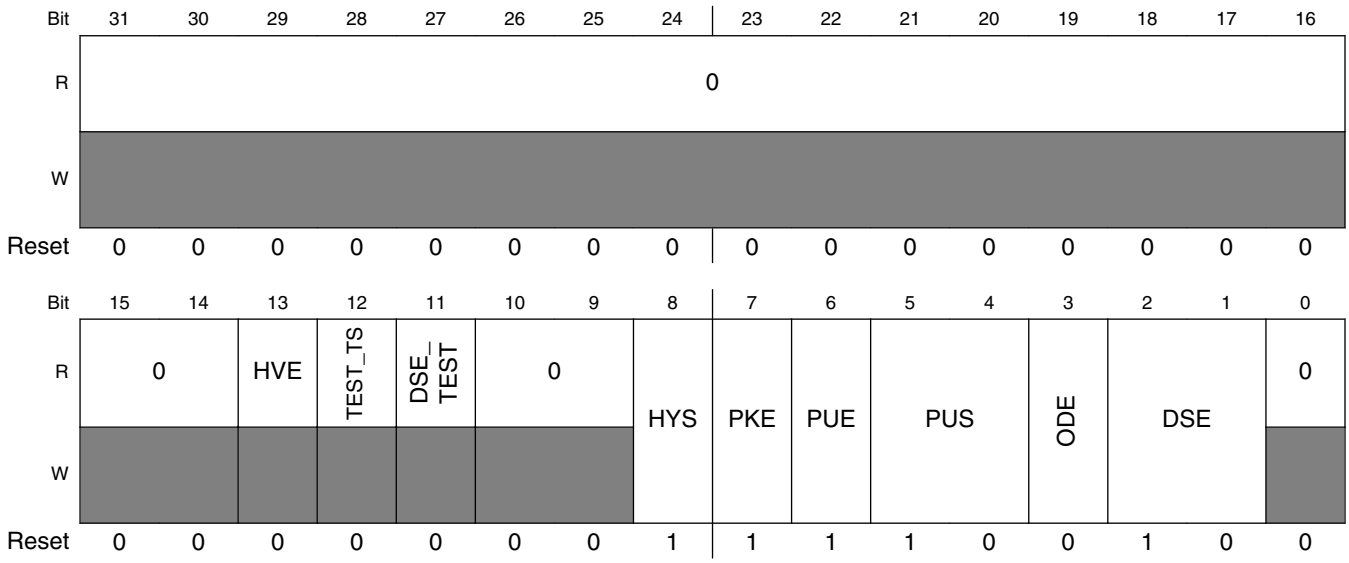
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_CS_0 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_CS_0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_CS_0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_CS_0. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_CS_0. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_CS_0. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.393 IOMUXC_SW_PAD_CTL_PAD_PATA_CS_1 (IOMUXC_SW_PAD_CTL_PAD_PATA_CS_1)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_CS_1 is 53FA_8000h base + 620h offset = 53FA_8620h



IOMUXC_SW_PAD_CTL_PAD_PATA_CS_1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_CS_1. 0 Hysteresis Disabled 1 Hysteresis Enabled

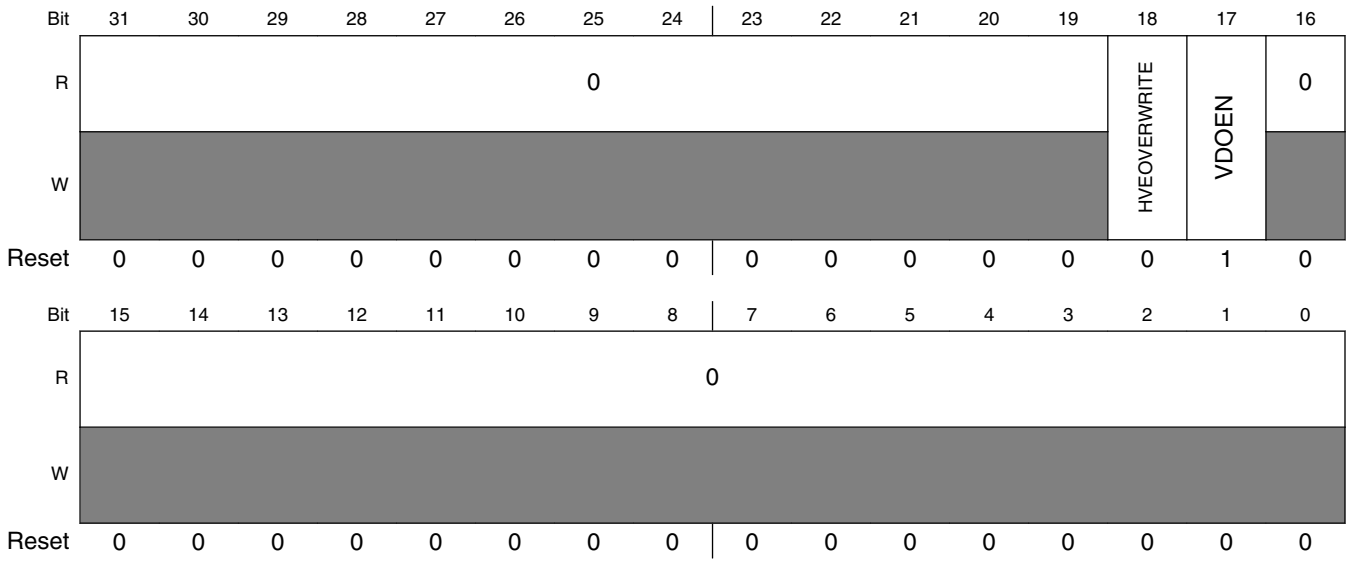
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_CS_1 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_CS_1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_CS_1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_CS_1. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_CS_1. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_CS_1. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.394 IOMUXC_SW_PAD_CTL_PAD_NVCC_PATA__2 (IOMUXC_SW_PAD_CTL_PAD_NVCC_PATA__2)

Address: IOMUXC_SW_PAD_CTL_PAD_NVCC_PATA__2 is 53FA_8000h base + 624h offset = 53FA_8624h

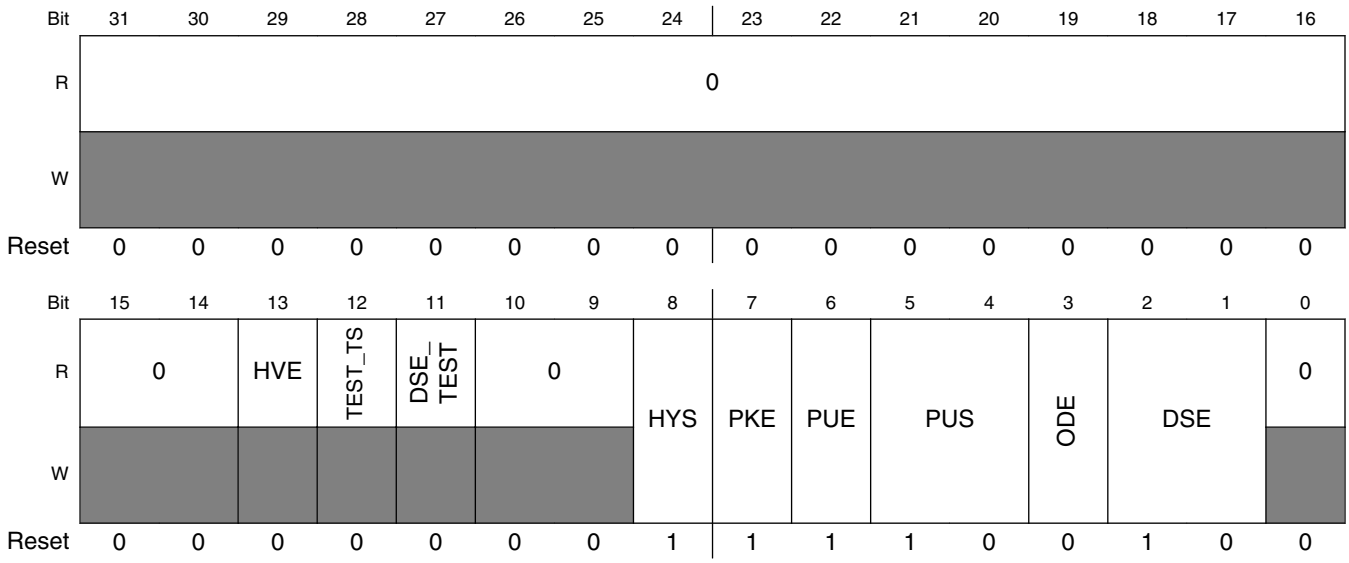


IOMUXC_SW_PAD_CTL_PAD_NVCC_PATA__2 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18 HVEOVERWRITE	HVEOVERWRITE (ipp_owr) Select one out of next values for pad: NVCC_PATA__2. This field affects the voltage levels for the following PADS: PATA_CS_1, PATA_CS_0, PATA_DA_2, PATA_DA_1, PATA_DA_0, PATA_IORDY, PATA_RESET_B, PATA_DIOR, PATA_INTRQ, PATA_BUFFER_EN, PATA_DMARQ, PATA_DMACK, PATA_DIOW 0 PAD support high voltage (3.0V-3.6V). Out of reset value is 0. The HVEOVERWRITE field should be updated before automatic voltage detector is disabled. 1 PAD support low voltage (1.65V-3.1V)
17 VDOEN	VDOEN (ipp_oen) Select one out of next values for pad: NVCC_PATA__2. 0 Voltage detector output disable and HVEOVERWRITE field will be used. It is recommended to disable the automatic voltage detector after boot and update the HVEOVERWRITE field with the actual I/O supply value. 1 Voltage detector output enable. (default 1)
16–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.395 IOMUXC_SW_PAD_CTL_PAD_PATA_DATA0 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA0)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DATA0 is 53FA_8000h base + 628h offset = 53FA_8628h



IOMUXC_SW_PAD_CTL_PAD_PATA_DATA0 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DATA0. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA0 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DATA0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DATA0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DATA0. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DATA0. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DATA0. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.396 IOMUXC_SW_PAD_CTL_PAD_PATA_DATA1 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA1)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DATA1 is 53FA_8000h base + 62Ch offset = 53FA_862Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DATA1. 0 Hysteresis Disabled 1 Hysteresis Enabled

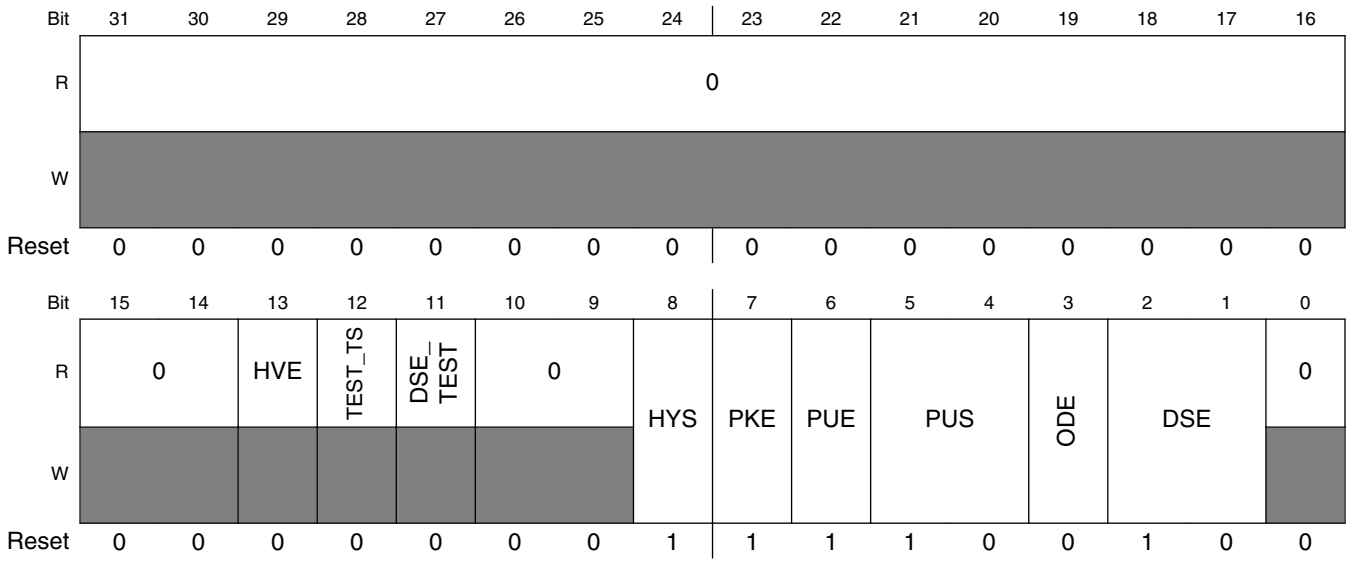
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA1 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DATA1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DATA1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DATA1. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DATA1. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DATA1. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.397 IOMUXC_SW_PAD_CTL_PAD_PATA_DATA2 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA2)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DATA2 is 53FA_8000h base + 630h offset = 53FA_8630h



IOMUXC_SW_PAD_CTL_PAD_PATA_DATA2 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DATA2. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA2 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DATA2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DATA2. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DATA2. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DATA2. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DATA2. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.398 IOMUXC_SW_PAD_CTL_PAD_PATA_DATA3 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA3)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DATA3 is 53FA_8000h base + 634h offset = 53FA_8634h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA3 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DATA3. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA3 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DATA3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DATA3. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DATA3. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DATA3. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DATA3. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.399 IOMUXC_SW_PAD_CTL_PAD_PATA_DATA4 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA4)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DATA4 is 53FA_8000h base + 638h offset = 53FA_8638h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA4 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DATA4. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA4 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DATA4. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DATA4. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DATA4. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DATA4. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DATA4. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.400 IOMUXC_SW_PAD_CTL_PAD_PATA_DATA5 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA5)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DATA5 is 53FA_8000h base + 63Ch offset = 53FA_863Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA5 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DATA5. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA5 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DATA5. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DATA5. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DATA5. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DATA5. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DATA5. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.401 IOMUXC_SW_PAD_CTL_PAD_PATA_DATA6 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA6)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DATA6 is 53FA_8000h base + 640h offset = 53FA_8640h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA6 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DATA6. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA6 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DATA6. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DATA6. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DATA6. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DATA6. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DATA6. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.402 IOMUXC_SW_PAD_CTL_PAD_PATA_DATA7 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA7)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DATA7 is 53FA_8000h base + 644h offset = 53FA_8644h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA7 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DATA7. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA7 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DATA7. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DATA7. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DATA7. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DATA7. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DATA7. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.403 IOMUXC_SW_PAD_CTL_PAD_PATA_DATA8 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA8)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DATA8 is 53FA_8000h base + 648h offset = 53FA_8648h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA8 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DATA8. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA8 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DATA8. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DATA8. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DATA8. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DATA8. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DATA8. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.404 IOMUXC_SW_PAD_CTL_PAD_PATA_DATA9 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA9)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DATA9 is 53FA_8000h base + 64Ch offset = 53FA_864Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA9 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DATA9. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA9 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DATA9. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DATA9. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DATA9. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DATA9. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DATA9. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.405 IOMUXC_SW_PAD_CTL_PAD_PATA_DATA10 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA10)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DATA10 is 53FA_8000h base + 650h offset = 53FA_8650h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA10 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DATA10. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA10 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DATA10. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DATA10. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DATA10. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DATA10. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DATA10. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.406 IOMUXC_SW_PAD_CTL_PAD_PATA_DATA11 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA11)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DATA11 is 53FA_8000h base + 654h offset = 53FA_8654h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA11 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DATA11. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA11 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DATA11. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DATA11. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DATA11. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DATA11. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DATA11. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.407 IOMUXC_SW_PAD_CTL_PAD_PATA_DATA12 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA12)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DATA12 is 53FA_8000h base + 658h offset = 53FA_8658h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA12 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DATA12. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA12 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DATA12. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DATA12. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DATA12. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DATA12. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DATA12. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.408 IOMUXC_SW_PAD_CTL_PAD_PATA_DATA13 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA13)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DATA13 is 53FA_8000h base + 65Ch offset = 53FA_865Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA13 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DATA13. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA13 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DATA13. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DATA13. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DATA13. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DATA13. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DATA13. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.409 IOMUXC_SW_PAD_CTL_PAD_PATA_DATA14 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA14)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DATA14 is 53FA_8000h base + 660h offset = 53FA_8660h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA14 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DATA14. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA14 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DATA14. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DATA14. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DATA14. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DATA14. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DATA14. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.410 IOMUXC_SW_PAD_CTL_PAD_PATA_DATA15 (IOMUXC_SW_PAD_CTL_PAD_PATA_DATA15)

Address: IOMUXC_SW_PAD_CTL_PAD_PATA_DATA15 is 53FA_8000h base + 664h offset = 53FA_8664h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA15 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: PATA_DATA15. 0 Hysteresis Disabled 1 Hysteresis Enabled

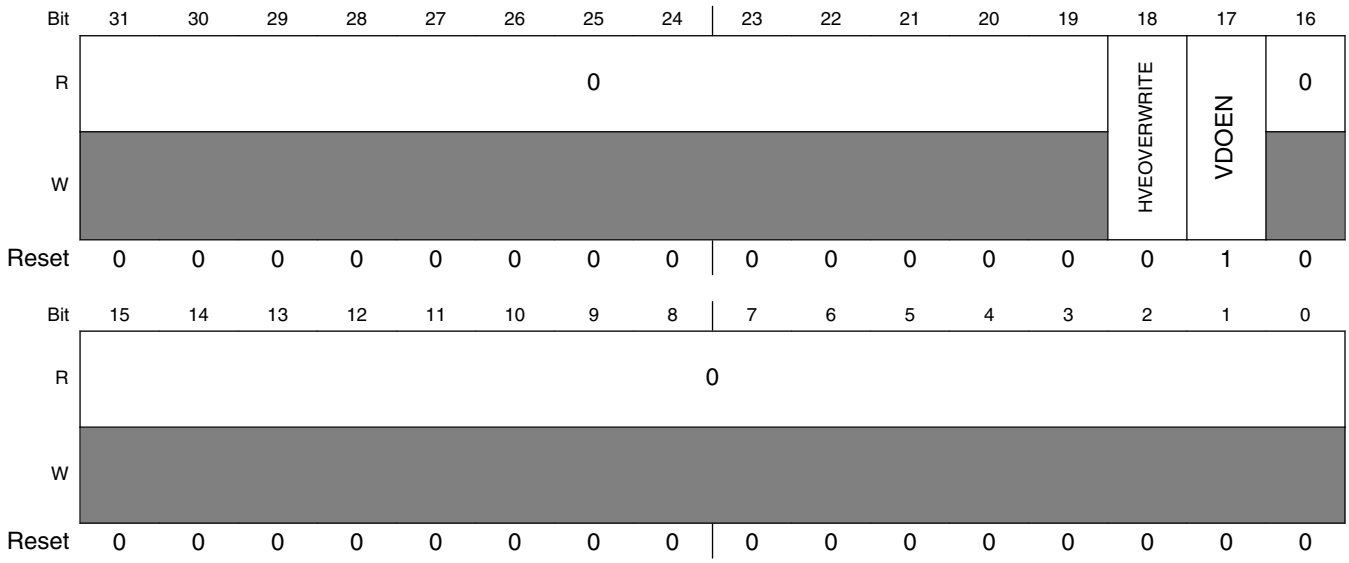
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PATA_DATA15 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: PATA_DATA15. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: PATA_DATA15. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: PATA_DATA15. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: PATA_DATA15. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: PATA_DATA15. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.411 IOMUXC_SW_PAD_CTL_PAD_NVCC_PATA__0 (IOMUXC_SW_PAD_CTL_PAD_NVCC_PATA__0)

Address: IOMUXC_SW_PAD_CTL_PAD_NVCC_PATA__0 is 53FA_8000h base + 668h offset = 53FA_8668h

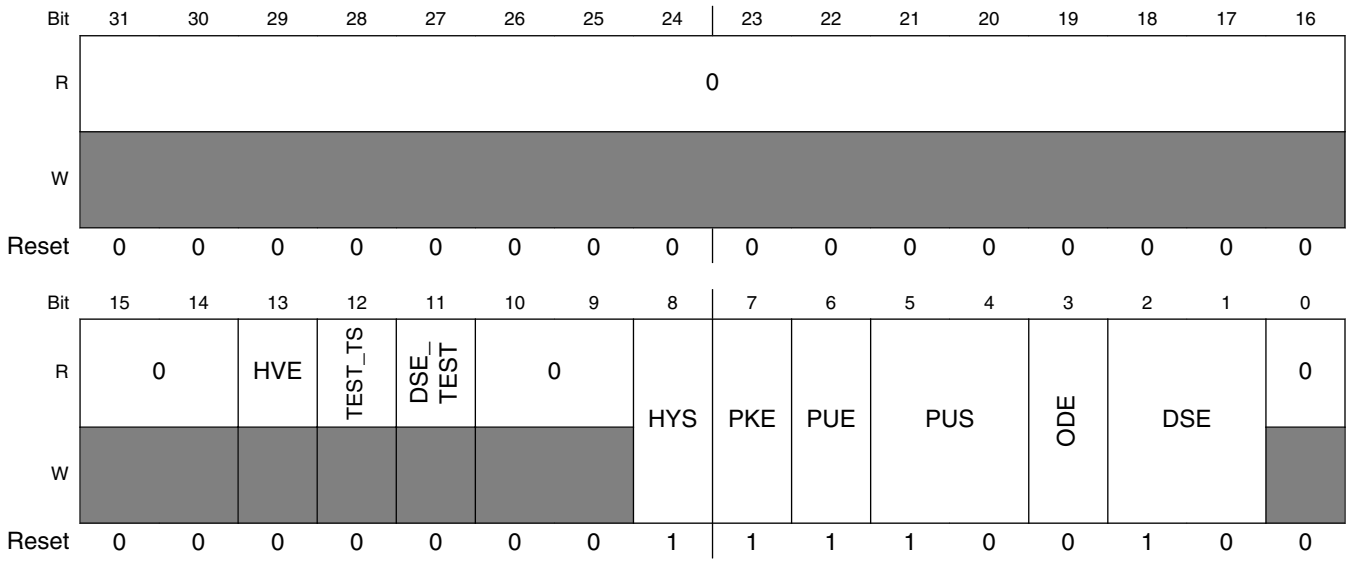


IOMUXC_SW_PAD_CTL_PAD_NVCC_PATA__0 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18 HVEOVERWRITE	HVEOVERWRITE (ipp_owr) Select one out of next values for pad: NVCC_PATA__0. This field affects the voltage levels for the following PADS: PATA_DATA15, PATA_DATA14, PATA_DATA13, PATA_DATA12, PATA_DATA11, PATA_DATA10, PATA_DATA9, PATA_DATA8, PATA_DATA7, PATA_DATA6, PATA_DATA5, PATA_DATA4, PATA_DATA3, PATA_DATA2, PATA_DATA1, PATA_DATA0 0 PAD support high voltage (3.0V-3.6V). Out of reset value is 0. The HVEOVERWRITE field should be updated before automatic voltage detector is disabled. 1 PAD support low voltage (1.65V-3.1V)
17 VDOEN	VDOEN (ipp_oen) Select one out of next values for pad: NVCC_PATA__0. 0 Voltage detector output disable and HVEOVERWRITE field will be used. It is recommended to disable the automatic voltage detector after boot and update the HVEOVERWRITE field with the actual I/O supply value. 1 Voltage detector output enable. (default 1)
16–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.412 IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0 (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0)

Address: IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0 is 53FA_8000h base + 66Ch offset = 53FA_866Ch



IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: SD1_DATA0. 0 Hysteresis Disabled 1 Hysteresis Enabled

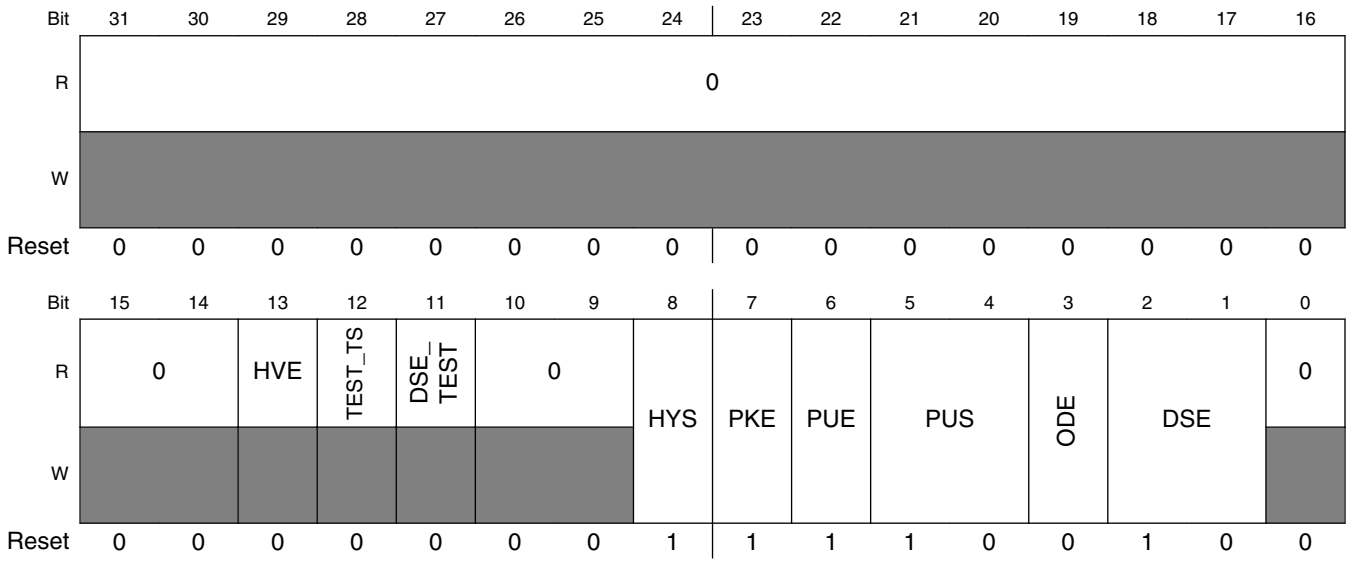
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: SD1_DATA0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: SD1_DATA0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: SD1_DATA0. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_DATA0. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA0. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.413 IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1 (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1)

Address: IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1 is 53FA_8000h base + 670h offset = 53FA_8670h



IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: SD1_DATA1. 0 Hysteresis Disabled 1 Hysteresis Enabled

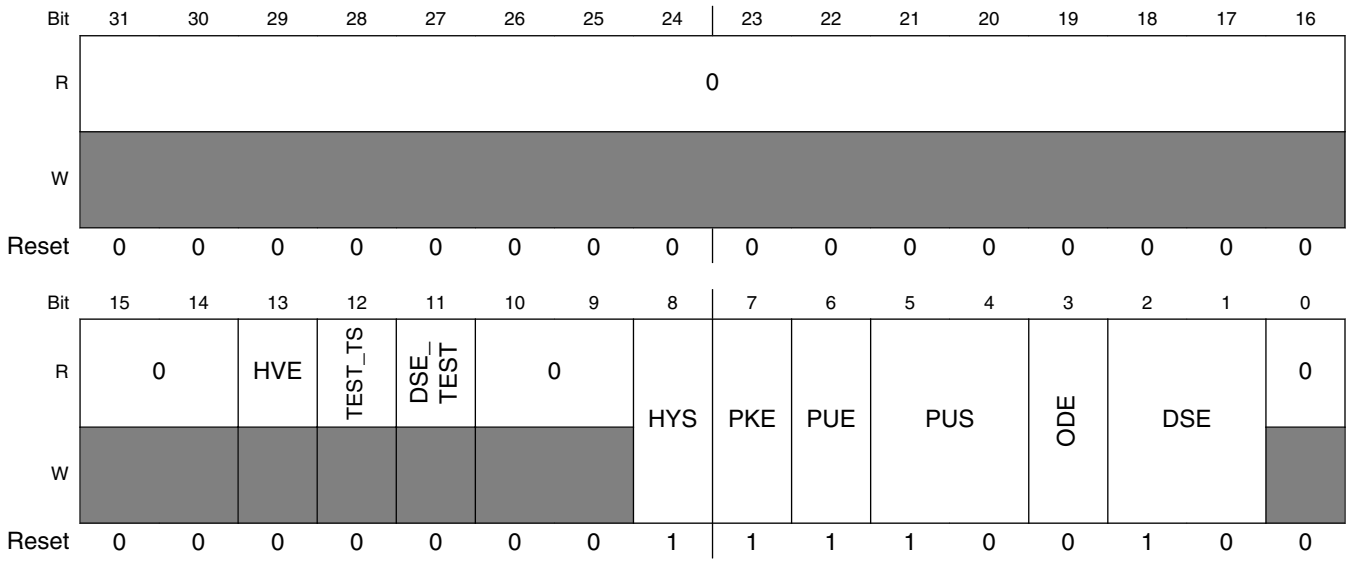
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: SD1_DATA1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: SD1_DATA1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: SD1_DATA1. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_DATA1. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA1. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.414 IOMUXC_SW_PAD_CTL_PAD_SD1_CMD (IOMUXC_SW_PAD_CTL_PAD_SD1_CMD)

Address: IOMUXC_SW_PAD_CTL_PAD_SD1_CMD is 53FA_8000h base + 674h offset = 53FA_8674h



IOMUXC_SW_PAD_CTL_PAD_SD1_CMD field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: SD1_CMD. 0 Hysteresis Disabled 1 Hysteresis Enabled

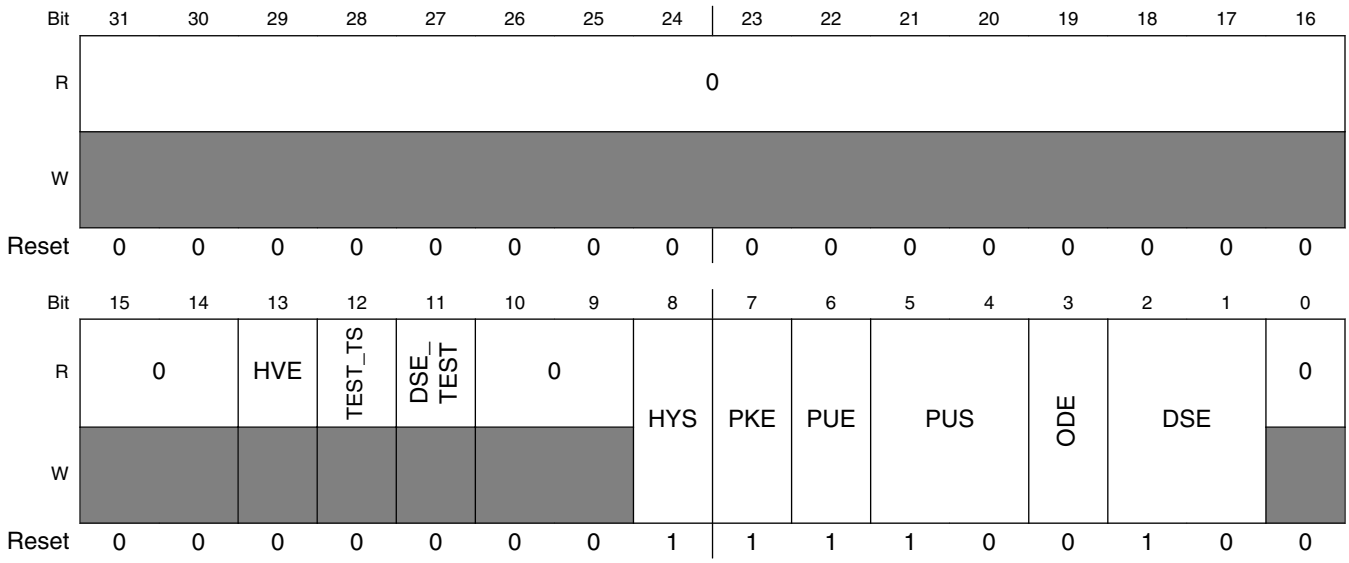
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_CMD field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: SD1_CMD. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: SD1_CMD. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: SD1_CMD. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_CMD. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_CMD. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.415 IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2 (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2)

Address: IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2 is 53FA_8000h base + 678h offset = 53FA_8678h



IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: SD1_DATA2. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: SD1_DATA2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: SD1_DATA2. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: SD1_DATA2. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_DATA2. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA2. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.416 IOMUXC_SW_PAD_CTL_PAD_SD1_CLK (IOMUXC_SW_PAD_CTL_PAD_SD1_CLK)

Address: IOMUXC_SW_PAD_CTL_PAD_SD1_CLK is 53FA_8000h base + 67Ch offset = 53FA_867Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_SD1_CLK field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: SD1_CLK. 0 Hysteresis Disabled 1 Hysteresis Enabled

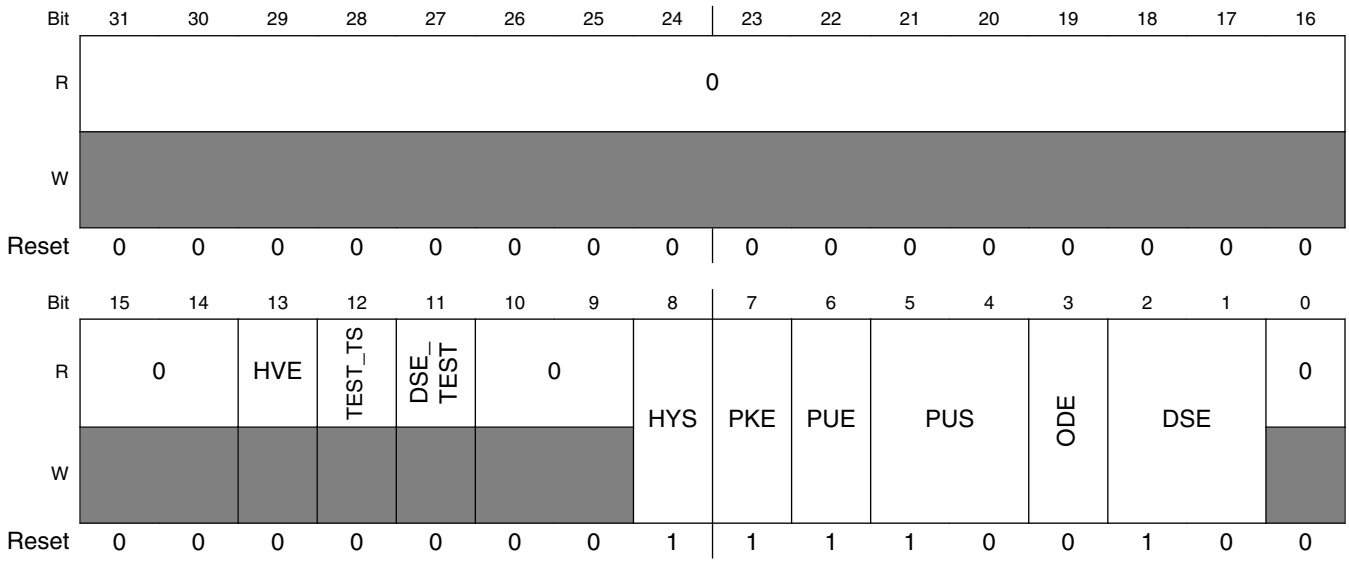
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_CLK field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: SD1_CLK. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: SD1_CLK. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: SD1_CLK. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_CLK. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_CLK. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.417 IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3 (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3)

Address: IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3 is 53FA_8000h base + 680h offset = 53FA_8680h



IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: SD1_DATA3. 0 Hysteresis Disabled 1 Hysteresis Enabled

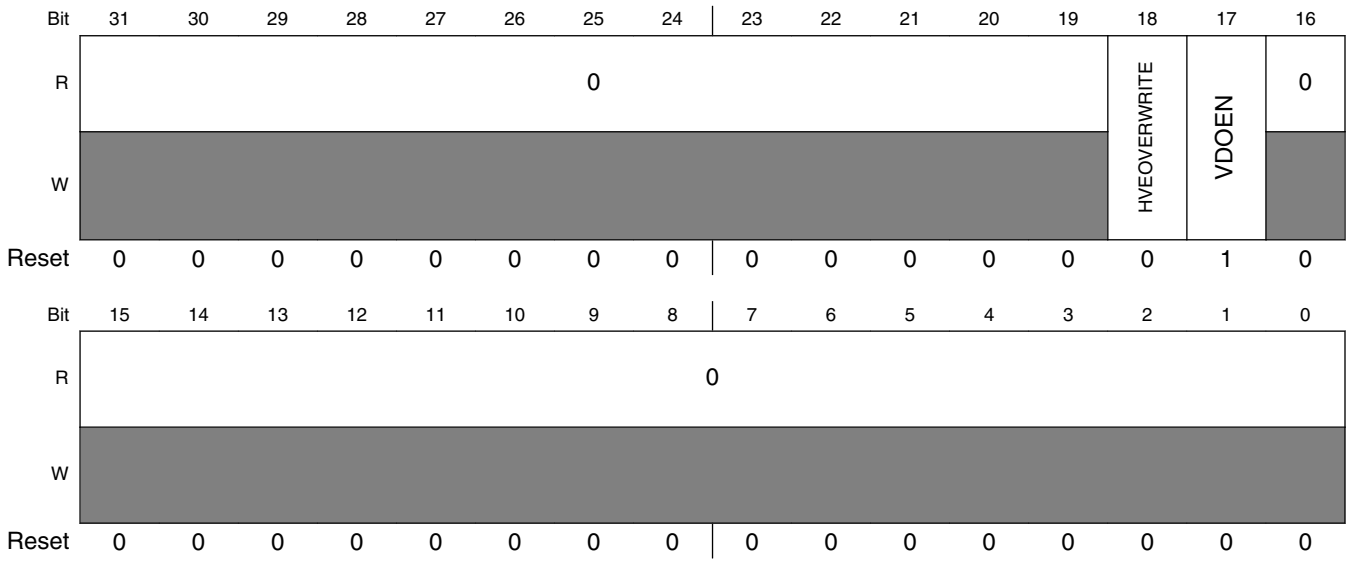
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: SD1_DATA3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: SD1_DATA3. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: SD1_DATA3. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_DATA3. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA3. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.418 IOMUXC_SW_PAD_CTL_PAD_NVCC_SD1 (IOMUXC_SW_PAD_CTL_PAD_NVCC_SD1)

Address: IOMUXC_SW_PAD_CTL_PAD_NVCC_SD1 is 53FA_8000h base + 684h offset = 53FA_8684h

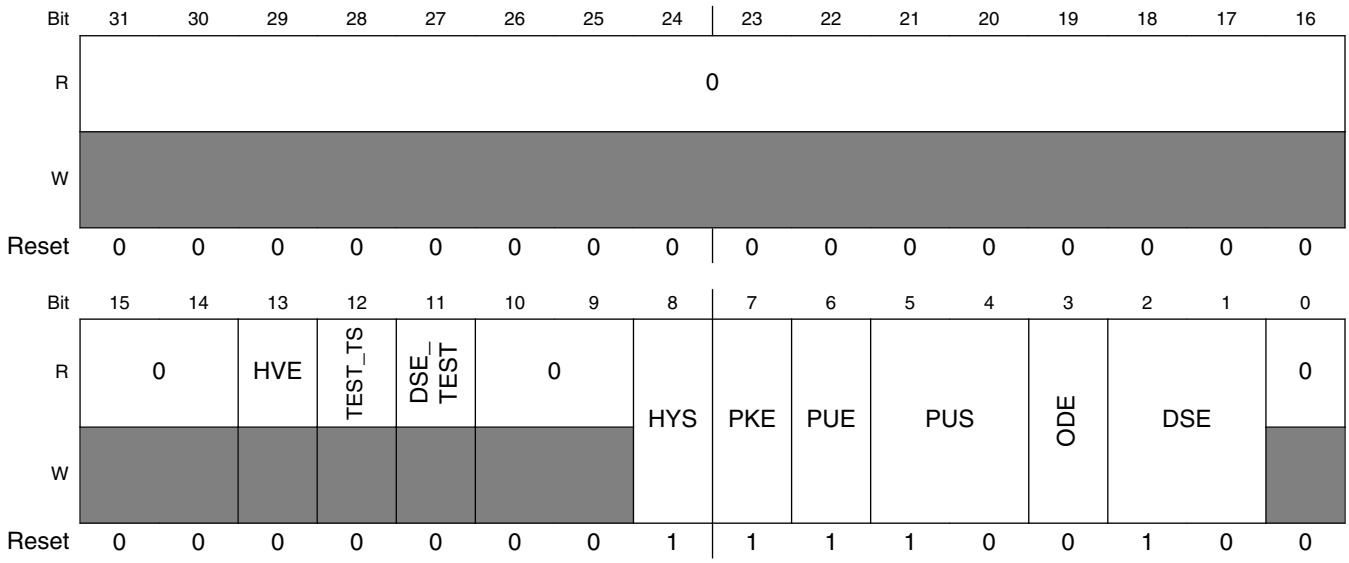


IOMUXC_SW_PAD_CTL_PAD_NVCC_SD1 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18 HVEOVERWRITE	HVEOVERWRITE (ipp_owr) Select one out of next values for pad: NVCC_SD1. This field affects the voltage levels for the following PADS: SD1_DATA3, SD1_CLK, SD1_DATA2, SD1_CMD, SD1_DATA1, SD1_DATA0 0 PAD support high voltage (3.0V-3.6V). Out of reset value is 0. The HVEOVERWRITE field should be updated before automatic voltage detector is disabled. 1 PAD support low voltage (1.65V-3.1V)
17 VDOEN	VDOEN (ipp_oen) Select one out of next values for pad: NVCC_SD1. 0 Voltage detector output disable and HVEOVERWRITE field will be used. It is recommended to disable the automatic voltage detector after boot and update the HVEOVERWRITE field with the actual I/O supply value. 1 Voltage detector output enable. (default 1)
16–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.419 IOMUXC_SW_PAD_CTL_PAD_SD2_CLK (IOMUXC_SW_PAD_CTL_PAD_SD2_CLK)

Address: IOMUXC_SW_PAD_CTL_PAD_SD2_CLK is 53FA_8000h base + 688h offset = 53FA_8688h



IOMUXC_SW_PAD_CTL_PAD_SD2_CLK field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: SD2_CLK. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_CLK field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: SD2_CLK. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: SD2_CLK. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: SD2_CLK. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: SD2_CLK. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD2_CLK. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.420 IOMUXC_SW_PAD_CTL_PAD_SD2_CMD (IOMUXC_SW_PAD_CTL_PAD_SD2_CMD)

Address: IOMUXC_SW_PAD_CTL_PAD_SD2_CMD is 53FA_8000h base + 68Ch offset = 53FA_868Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_SD2_CMD field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: SD2_CMD. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_CMD field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: SD2_CMD. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: SD2_CMD. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: SD2_CMD. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: SD2_CMD. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD2_CMD. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.421 IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3 (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3)

Address: IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3 is 53FA_8000h base + 690h offset = 53FA_8690h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: SD2_DATA3. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: SD2_DATA3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: SD2_DATA3. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: SD2_DATA3. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: SD2_DATA3. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD2_DATA3. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.422 IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2 (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2)

Address: IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2 is 53FA_8000h base + 694h offset = 53FA_8694h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: SD2_DATA2. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: SD2_DATA2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: SD2_DATA2. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: SD2_DATA2. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: SD2_DATA2. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD2_DATA2. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.423 IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1 (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1)

Address: IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1 is 53FA_8000h base + 698h offset = 53FA_8698h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: SD2_DATA1. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: SD2_DATA1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: SD2_DATA1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: SD2_DATA1. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: SD2_DATA1. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD2_DATA1. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.424 IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0 (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0)

Address: IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0 is 53FA_8000h base + 69Ch offset = 53FA_869Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		HVE	TEST_TS	DSE_TEST	0		HYS	PKE	PUE	PUS		ODE	DSE		0
W																
Reset	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: SD2_DATA0. 0 Hysteresis Disabled 1 Hysteresis Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: SD2_DATA0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: SD2_DATA0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: SD2_DATA0. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: SD2_DATA0. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: SD2_DATA0. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.425 IOMUXC_SW_PAD_CTL_PAD_NVCC_SD2 (IOMUXC_SW_PAD_CTL_PAD_NVCC_SD2)

Address: IOMUXC_SW_PAD_CTL_PAD_NVCC_SD2 is 53FA_8000h base + 6A0h offset = 53FA_86A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													HVEOVERWRITE	VDOEN	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

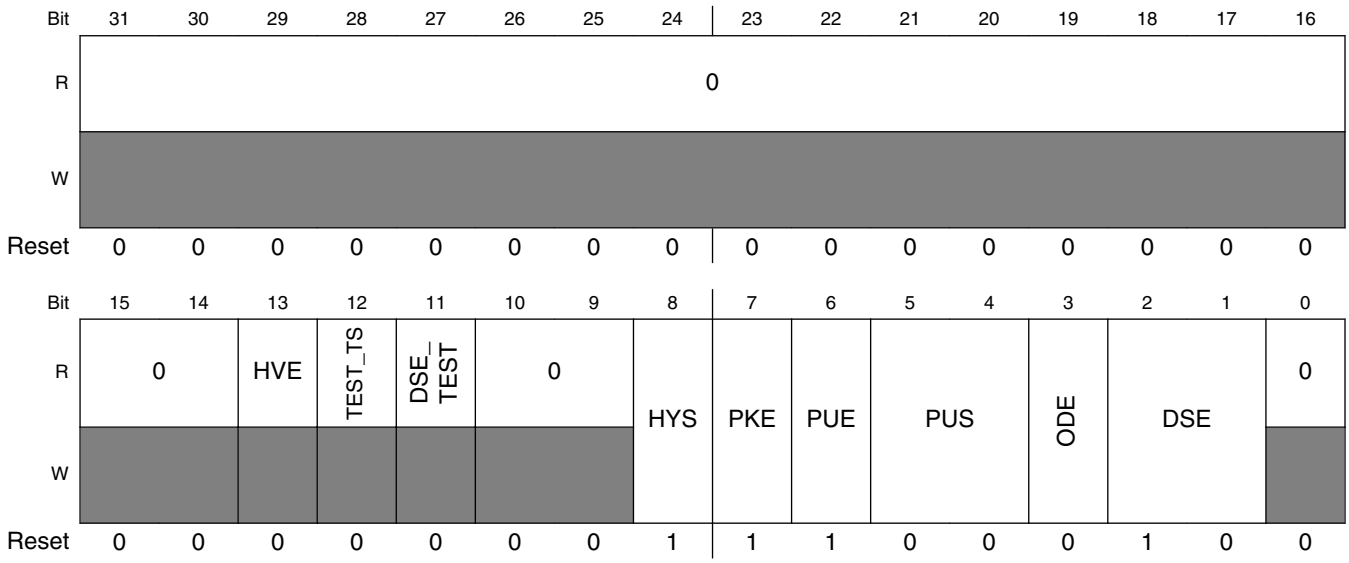
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_NVCC_SD2 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18 HVEOVERWRITE	HVEOVERWRITE (ipp_owr) Select one out of next values for pad: NVCC_SD2. This field affects the voltage levels for the following PADS: SD2_DATA0,SD2_DATA1, SD2_DATA2, SD2_DATA3, SD2_CMD, SD2_CLK 0 PAD support high voltage (3.0V-3.6V). Out of reset value is 0. The HVEOVERWRITE field should be updated before automatic voltage detector is disabled. 1 PAD support low voltage (1.65V-3.1V)
17 VDOEN	VDOEN (ipp_oen) Select one out of next values for pad: NVCC_SD2. 0 Voltage detector output disable and HVEOVERWRITE field will be used. It is recommended to disable the automatic voltage detector after boot and update the HVEOVERWRITE field with the actual I/O supply value. 1 Voltage detector output enable. (default 1)
16–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.426 IOMUXC_SW_PAD_CTL_PAD_GPIO_0 (IOMUXC_SW_PAD_CTL_PAD_GPIO_0)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_0 is 53FA_8000h base + 6A4h offset = 53FA_86A4h



IOMUXC_SW_PAD_CTL_PAD_GPIO_0 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_0. 0 Hysteresis Disabled 1 Hysteresis Enabled

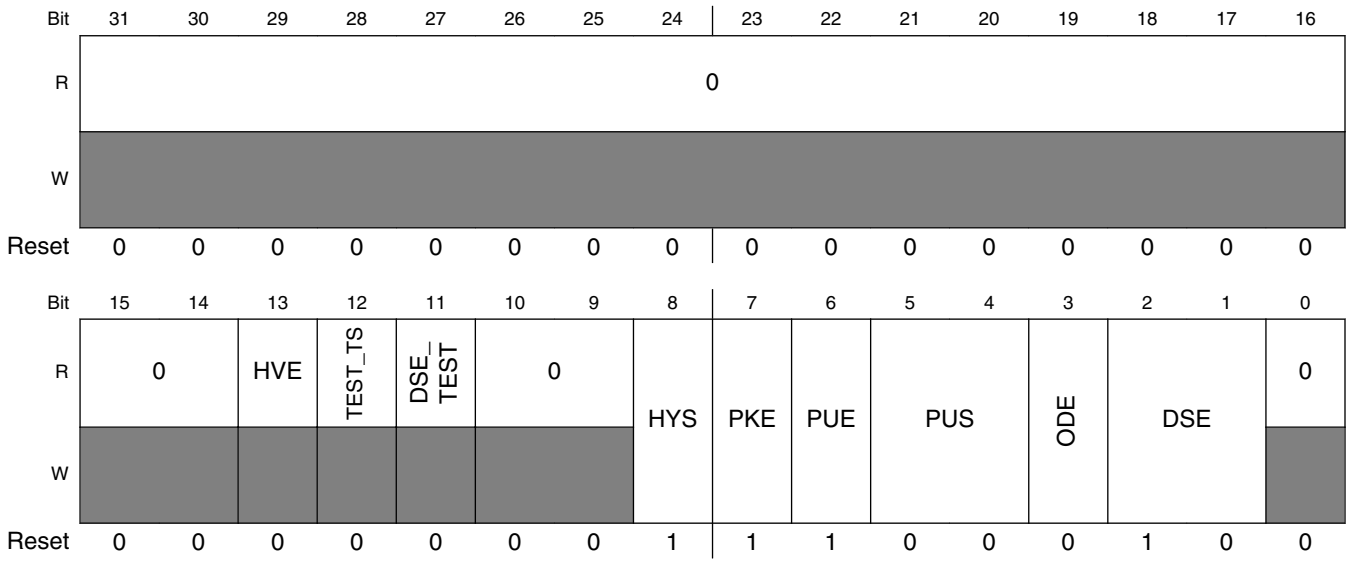
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_0 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_0. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_0. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_0. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_0. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.427 IOMUXC_SW_PAD_CTL_PAD_GPIO_1 (IOMUXC_SW_PAD_CTL_PAD_GPIO_1)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_1 is 53FA_8000h base + 6A8h offset = 53FA_86A8h



IOMUXC_SW_PAD_CTL_PAD_GPIO_1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_1. 0 Hysteresis Disabled 1 Hysteresis Enabled

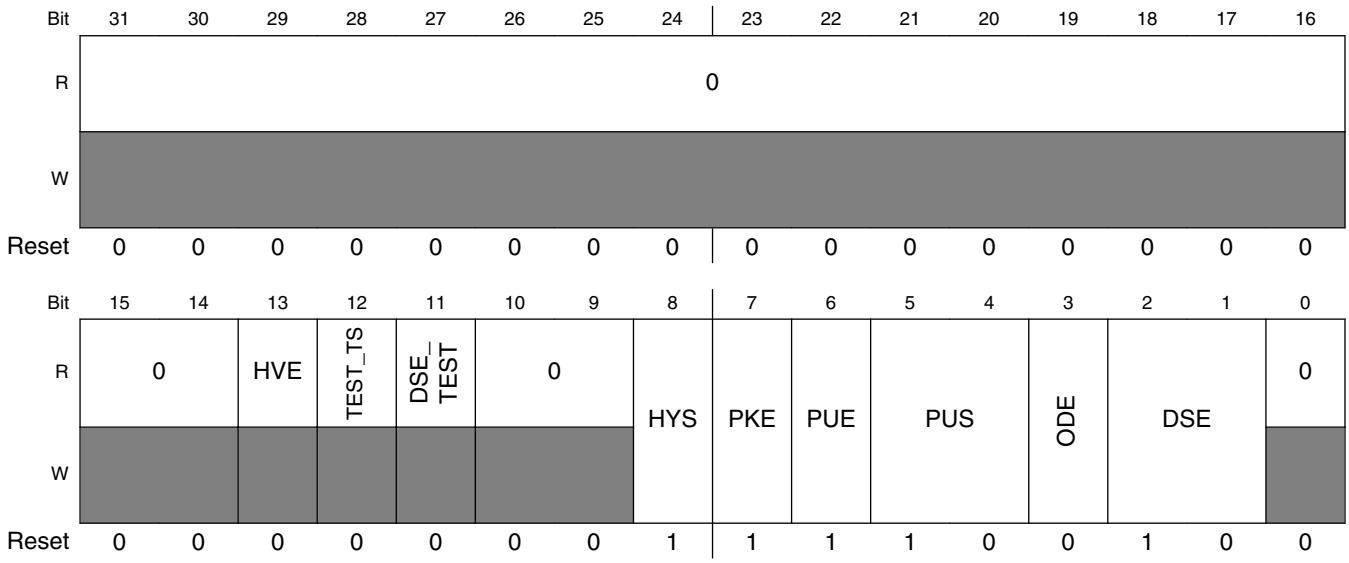
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_1 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_1. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_1. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_1. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_1. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.428 IOMUXC_SW_PAD_CTL_PAD_GPIO_9 (IOMUXC_SW_PAD_CTL_PAD_GPIO_9)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_9 is 53FA_8000h base + 6ACh offset = 53FA_86ACh



IOMUXC_SW_PAD_CTL_PAD_GPIO_9 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_9. 0 Hysteresis Disabled 1 Hysteresis Enabled

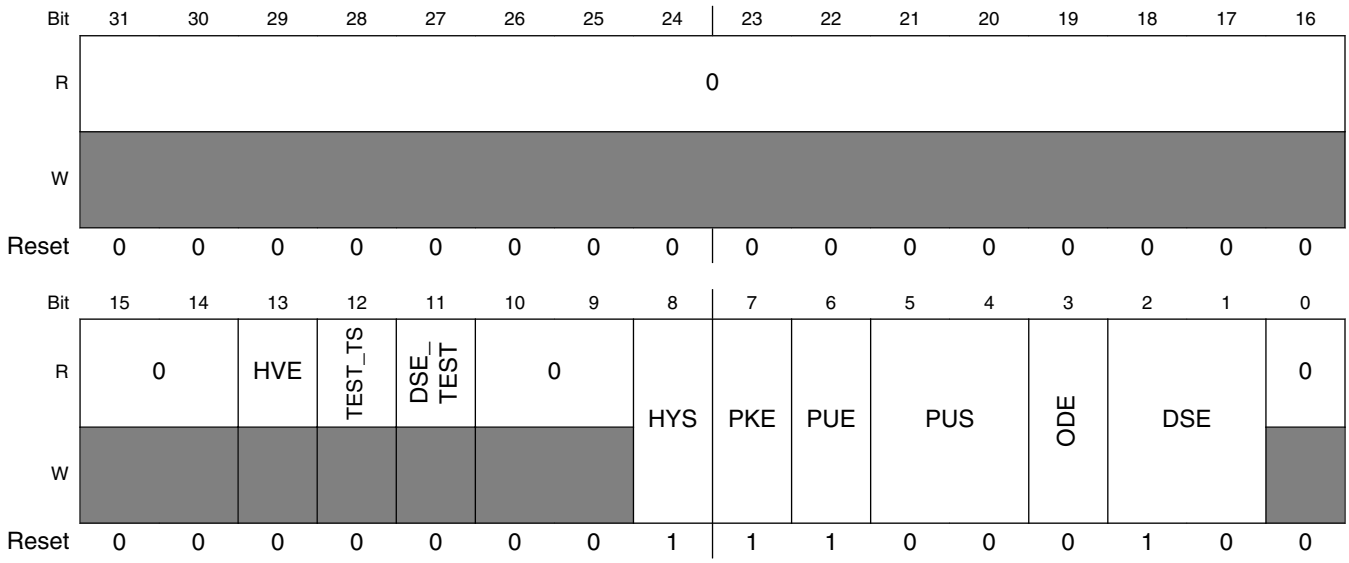
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_9 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_9. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_9. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_9. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_9. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_9. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.429 IOMUXC_SW_PAD_CTL_PAD_GPIO_3 (IOMUXC_SW_PAD_CTL_PAD_GPIO_3)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_3 is 53FA_8000h base + 6B0h offset = 53FA_86B0h



IOMUXC_SW_PAD_CTL_PAD_GPIO_3 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_3. 0 Hysteresis Disabled 1 Hysteresis Enabled

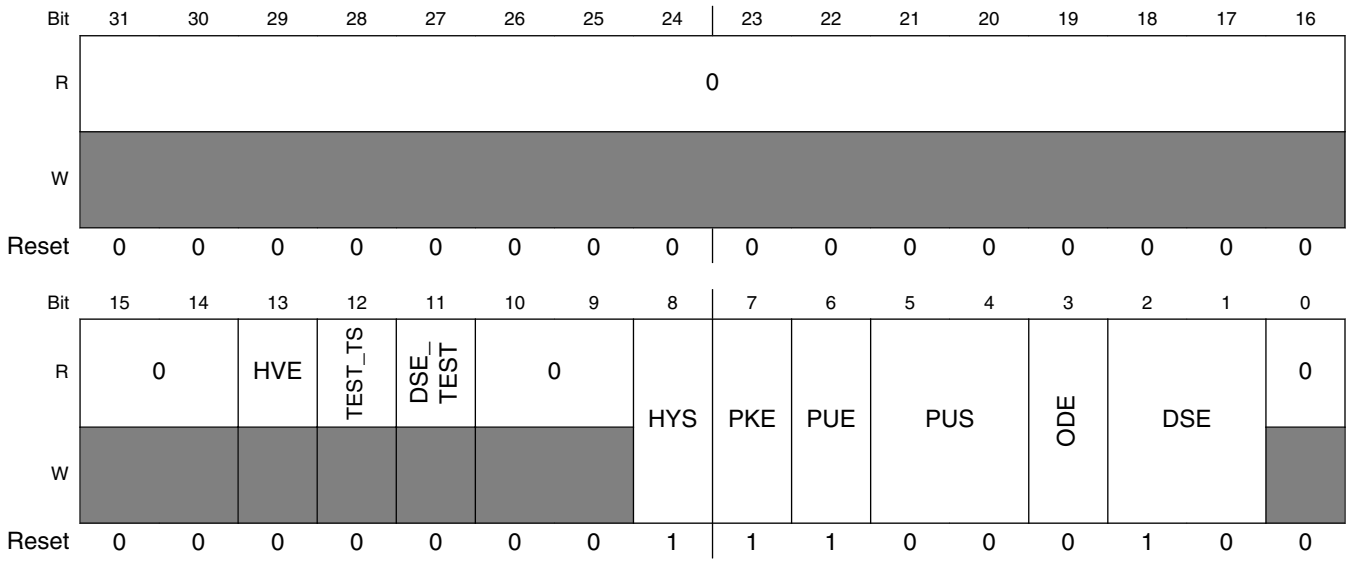
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_3 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_3. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_3. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_3. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_3. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.430 IOMUXC_SW_PAD_CTL_PAD_GPIO_6 (IOMUXC_SW_PAD_CTL_PAD_GPIO_6)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_6 is 53FA_8000h base + 6B4h offset = 53FA_86B4h



IOMUXC_SW_PAD_CTL_PAD_GPIO_6 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_6. 0 Hysteresis Disabled 1 Hysteresis Enabled

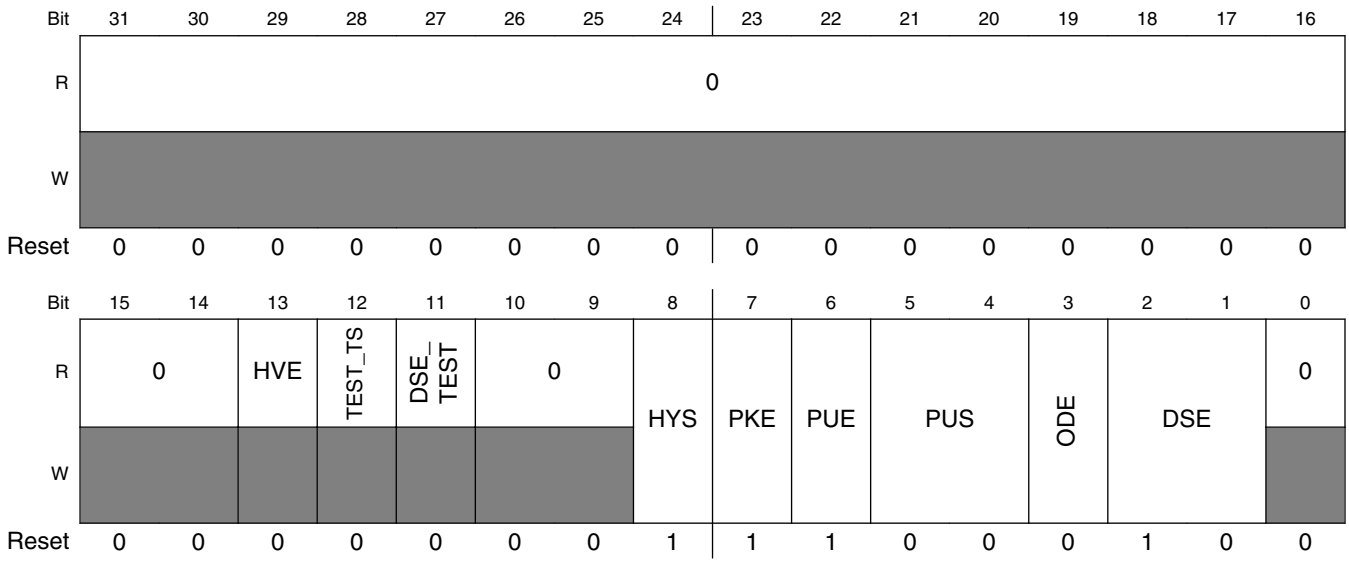
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_6 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_6. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_6. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_6. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_6. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_6. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.431 IOMUXC_SW_PAD_CTL_PAD_GPIO_2 (IOMUXC_SW_PAD_CTL_PAD_GPIO_2)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_2 is 53FA_8000h base + 6B8h offset = 53FA_86B8h



IOMUXC_SW_PAD_CTL_PAD_GPIO_2 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_2. 0 Hysteresis Disabled 1 Hysteresis Enabled

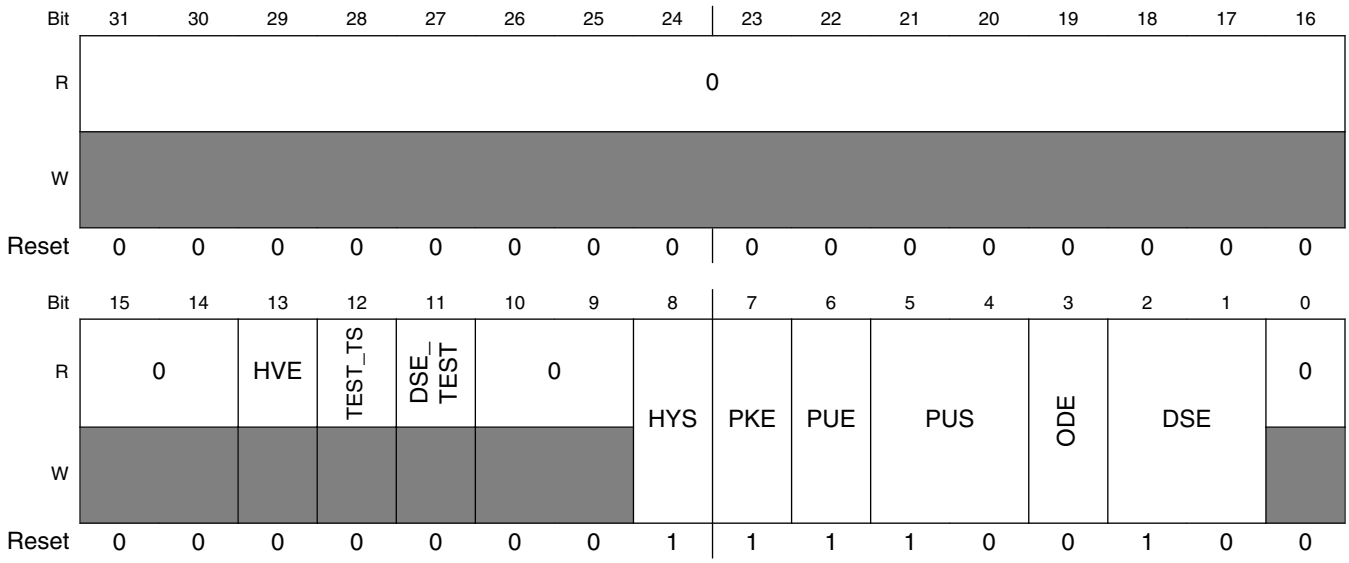
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_2 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_2. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_2. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_2. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_2. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.432 IOMUXC_SW_PAD_CTL_PAD_GPIO_4 (IOMUXC_SW_PAD_CTL_PAD_GPIO_4)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_4 is 53FA_8000h base + 6BCh offset = 53FA_86BCh



IOMUXC_SW_PAD_CTL_PAD_GPIO_4 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_4. 0 Hysteresis Disabled 1 Hysteresis Enabled

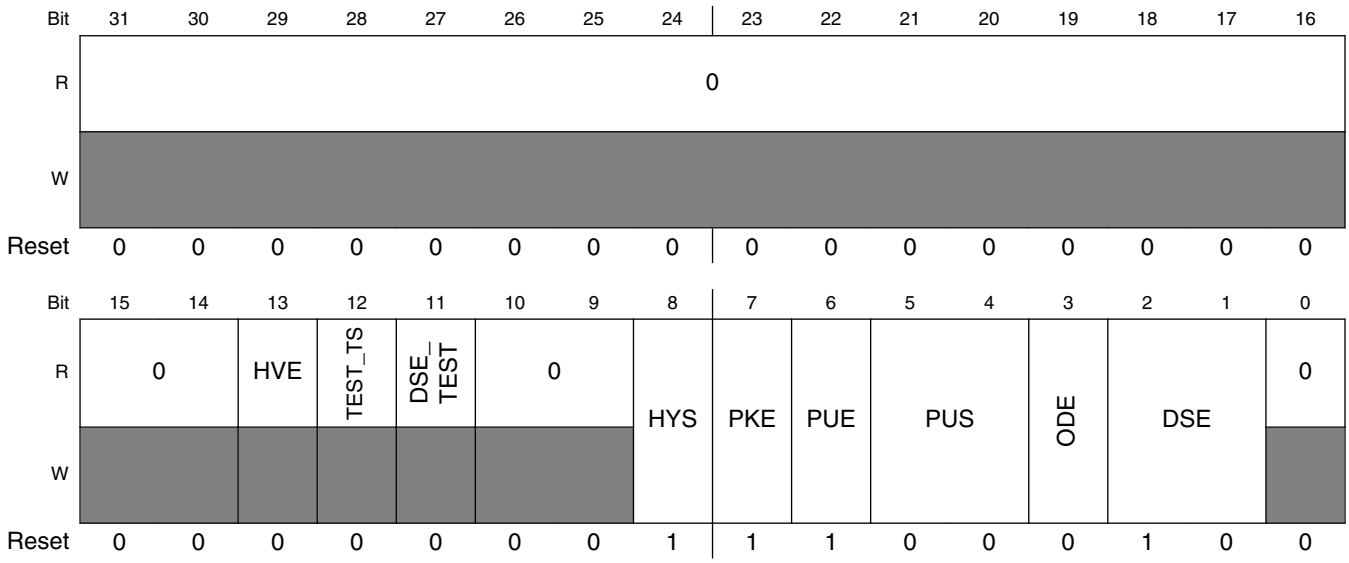
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_4 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_4. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_4. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_4. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_4. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_4. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.433 IOMUXC_SW_PAD_CTL_PAD_GPIO_5 (IOMUXC_SW_PAD_CTL_PAD_GPIO_5)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_5 is 53FA_8000h base + 6C0h offset = 53FA_86C0h



IOMUXC_SW_PAD_CTL_PAD_GPIO_5 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_5. 0 Hysteresis Disabled 1 Hysteresis Enabled

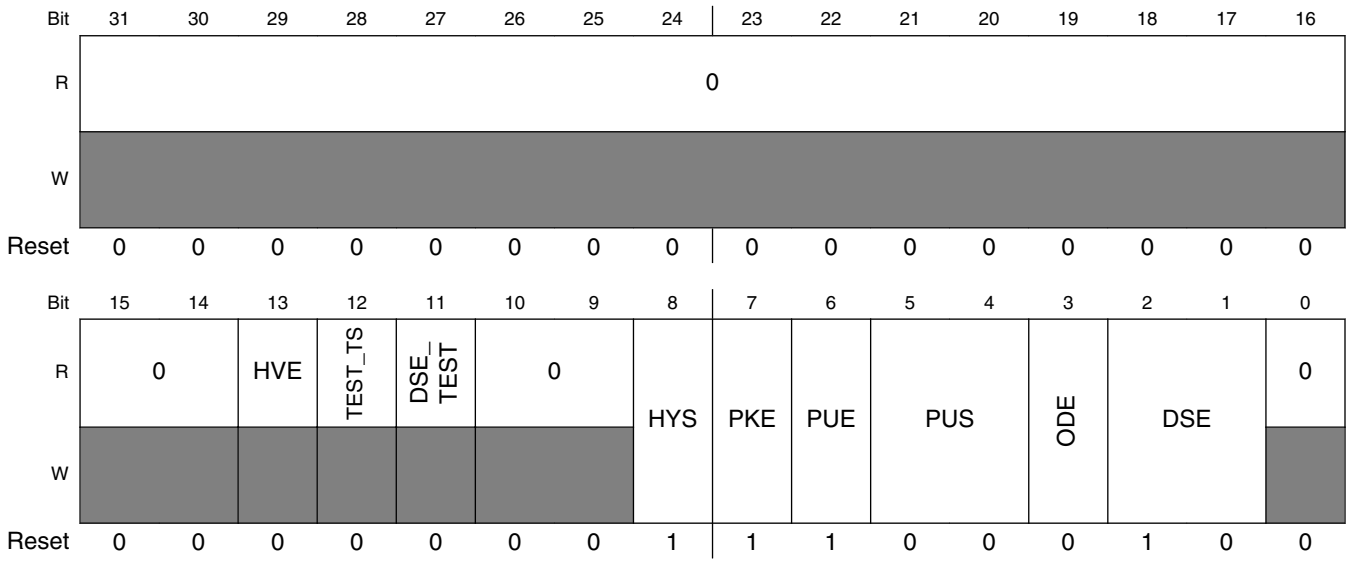
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_5 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_5. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_5. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_5. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_5. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_5. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.434 IOMUXC_SW_PAD_CTL_PAD_GPIO_7 (IOMUXC_SW_PAD_CTL_PAD_GPIO_7)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_7 is 53FA_8000h base + 6C4h offset = 53FA_86C4h



IOMUXC_SW_PAD_CTL_PAD_GPIO_7 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_7. 0 Hysteresis Disabled 1 Hysteresis Enabled

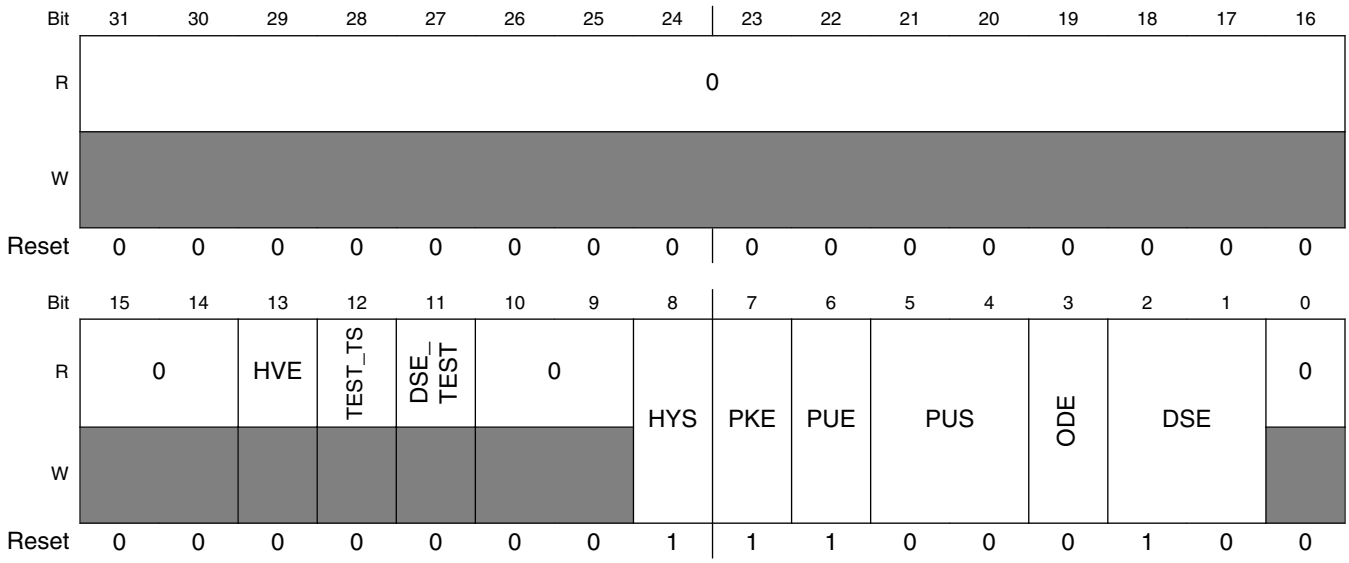
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_7 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_7. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_7. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_7. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_7. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_7. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.435 IOMUXC_SW_PAD_CTL_PAD_GPIO_8 (IOMUXC_SW_PAD_CTL_PAD_GPIO_8)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_8 is 53FA_8000h base + 6C8h offset = 53FA_86C8h



IOMUXC_SW_PAD_CTL_PAD_GPIO_8 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_8. 0 Hysteresis Disabled 1 Hysteresis Enabled

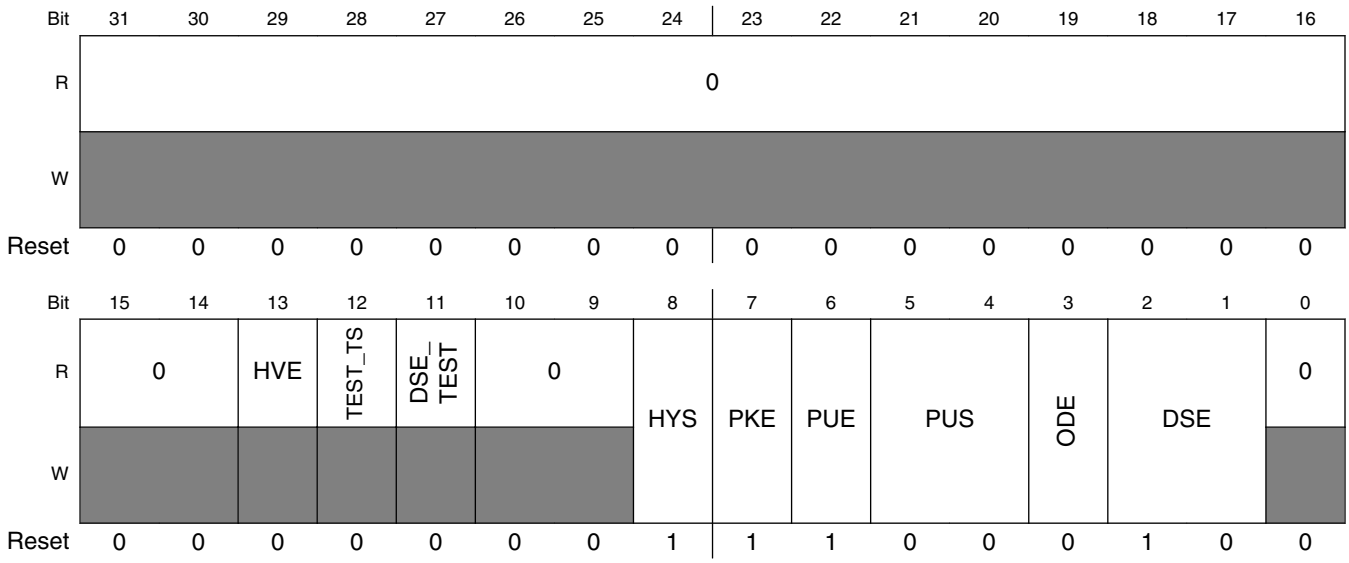
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_8 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_8. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_8. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_8. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_8. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_8. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.436 IOMUXC_SW_PAD_CTL_PAD_GPIO_16 (IOMUXC_SW_PAD_CTL_PAD_GPIO_16)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_16 is 53FA_8000h base + 6CCh offset = 53FA_86CCh



IOMUXC_SW_PAD_CTL_PAD_GPIO_16 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_16. 0 Hysteresis Disabled 1 Hysteresis Enabled

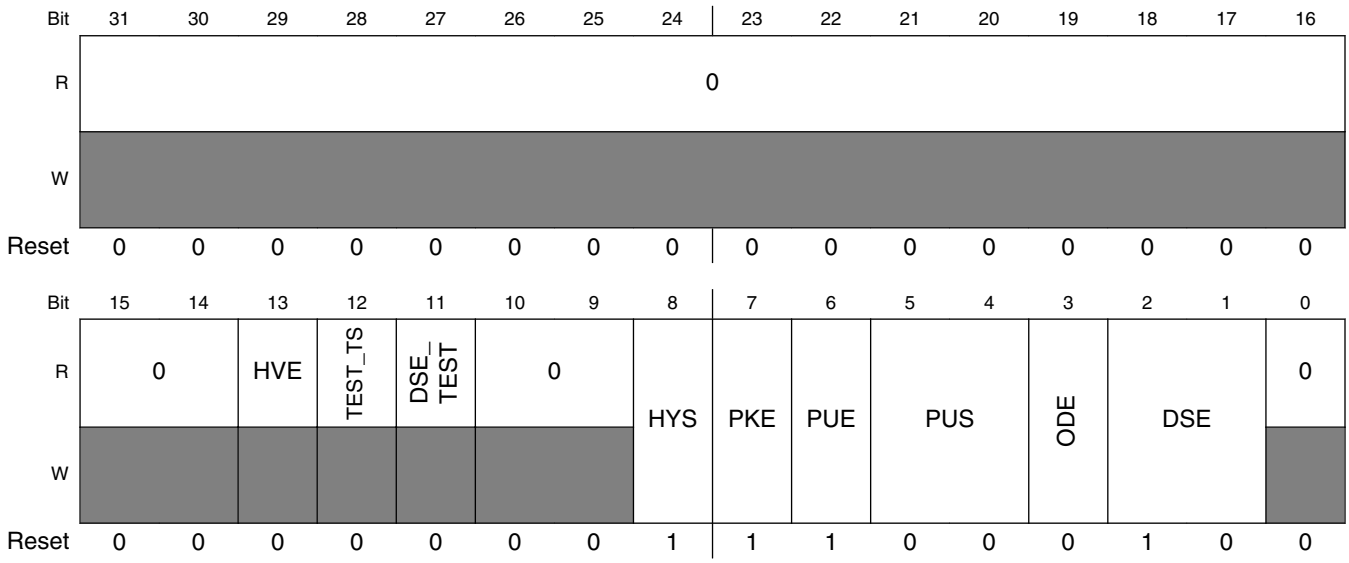
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_16 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_16. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_16. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_16. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_16. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_16. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.437 IOMUXC_SW_PAD_CTL_PAD_GPIO_17 (IOMUXC_SW_PAD_CTL_PAD_GPIO_17)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_17 is 53FA_8000h base + 6D0h offset = 53FA_86D0h



IOMUXC_SW_PAD_CTL_PAD_GPIO_17 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_17. 0 Hysteresis Disabled 1 Hysteresis Enabled

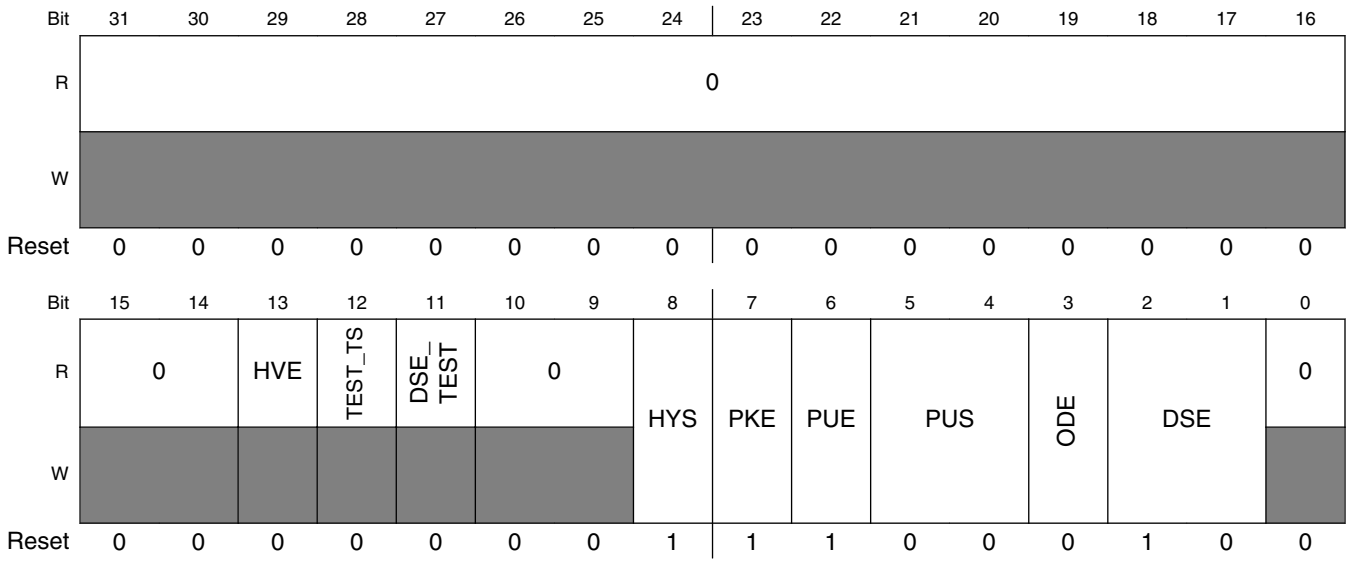
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_17 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_17. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_17. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_17. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_17. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_17. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.438 IOMUXC_SW_PAD_CTL_PAD_GPIO_18 (IOMUXC_SW_PAD_CTL_PAD_GPIO_18)

Address: IOMUXC_SW_PAD_CTL_PAD_GPIO_18 is 53FA_8000h base + 6D4h offset = 53FA_86D4h



IOMUXC_SW_PAD_CTL_PAD_GPIO_18 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 HVE	low/high output voltage Field Read Only Field 0 Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO_18. 0 Hysteresis Disabled 1 Hysteresis Enabled

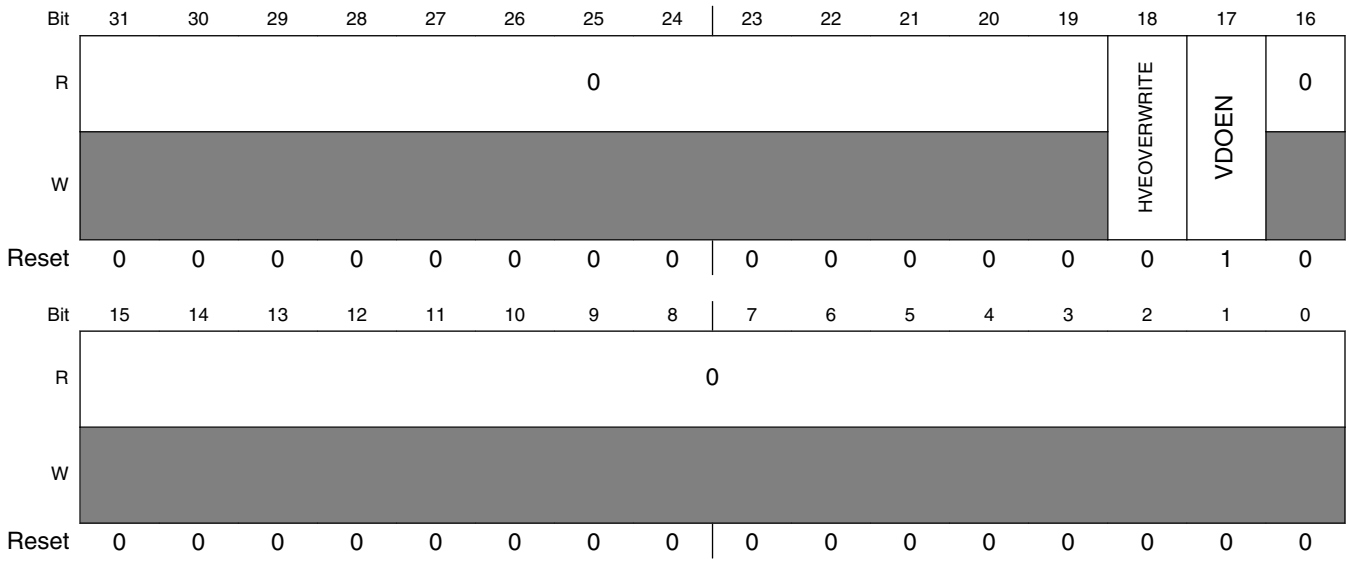
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO_18 field descriptions (continued)

Field	Description
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO_18. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO_18. 0 Keeper 1 Pull
5–4 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO_18. 00 360 [KOhm] Pull Down 01 75 [KOhm] Pull Up 10 100 [KOhm] Pull Up 11 22 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO_18. 0 Open Drain Disabled 1 Open Drain Enabled
2–1 DSE	Drive Strength Field Select one out of next values for pad: GPIO_18. 00 Low Drive Strength 01 Medium Drive Strength 10 High Drive Strength 11 Max Drive Strength
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.439 IOMUXC_SW_PAD_CTL_PAD_NVCC_GPIO (IOMUXC_SW_PAD_CTL_PAD_NVCC_GPIO)

Address: IOMUXC_SW_PAD_CTL_PAD_NVCC_GPIO is 53FA_8000h base + 6D8h offset = 53FA_86D8h



IOMUXC_SW_PAD_CTL_PAD_NVCC_GPIO field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18 HVEOVERWRITE	HVEOVERWRITE (ipp_owr) Select one out of next values for pad: NVCC_GPIO. This field affects the voltage levels for the following PADS: GPIO_18, GPIO_17, GPIO_16, GPIO_8, GPIO_7, GPIO_5, GPIO_4, GPIO_2, GPIO_6, GPIO_3, GPIO_9, GPIO_1, GPIO_0 0 PAD support high voltage (3.0V-3.6V). Out of reset value is 0. The HVEOVERWRITE field should be updated before automatic voltage detector is disabled. 1 PAD support low voltage (1.65V-3.1V)
17 VDOEN	VDOEN (ipp_oen) Select one out of next values for pad: NVCC_GPIO. 0 Voltage detector output disable and HVEOVERWRITE field will be used. It is recommended to disable the automatic voltage detector after boot and update the HVEOVERWRITE field with the actual I/O supply value. 1 Voltage detector output enable. (default 1)
16–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.440 IOMUXC_SW_PAD_CTL_PAD_POR_B (IOMUXC_SW_PAD_CTL_PAD_POR_B)

Address: IOMUXC_SW_PAD_CTL_PAD_POR_B is 53FA_8000h base + 6DCh offset = 53FA_86DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	DSE_TEST	STRENGTH_MODE	0	HYS	PKE	PUE	PUS		ODE	DSE		SRE
W																
Reset	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_POR_B field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_POR_B field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: POR_B. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: POR_B. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Read Only Field 1 Pull
5–4 PUS	Pull Up / Down Config. Field Read Only Field 10 100 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Read Only Field 0 Open Drain Disabled
2–1 DSE	Drive Strength Field Read Only Field 00 Low Drive Strength
0 SRE	Slew Rate Field Read Only Field 0 Slow Slew Rate

43.3.441 IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE1 (IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE1)

Address: IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE1 is 53FA_8000h base + 6E0h offset = 53FA_86E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	DSE_TEST	STRENGTH_MODE	0	HYS	PKE	PUE	PUS		ODE	DSE		SRE
W																
Reset	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE1 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE1 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: BOOT_MODE1. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: BOOT_MODE1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Read Only Field 1 Pull
5–4 PUS	Pull Up / Down Config. Field Read Only Field 00 100 [KOhm] Pull Down
3 ODE	Open Drain Enable Field Read Only Field 0 Open Drain Disabled
2–1 DSE	Drive Strength Field Read Only Field 00 Low Drive Strength
0 SRE	Slew Rate Field Read Only Field 0 Slow Slew Rate

43.3.442 IOMUXC_SW_PAD_CTL_PAD_RESET_IN_B (IOMUXC_SW_PAD_CTL_PAD_RESET_IN_B)

Address: IOMUXC_SW_PAD_CTL_PAD_RESET_IN_B is 53FA_8000h base + 6E4h offset = 53FA_86E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	DSE_TEST	STRENGTH_MODE	0	HYS	PKE	PUE	PUS		ODE	DSE		SRE
W																
Reset	0	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_RESET_IN_B field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

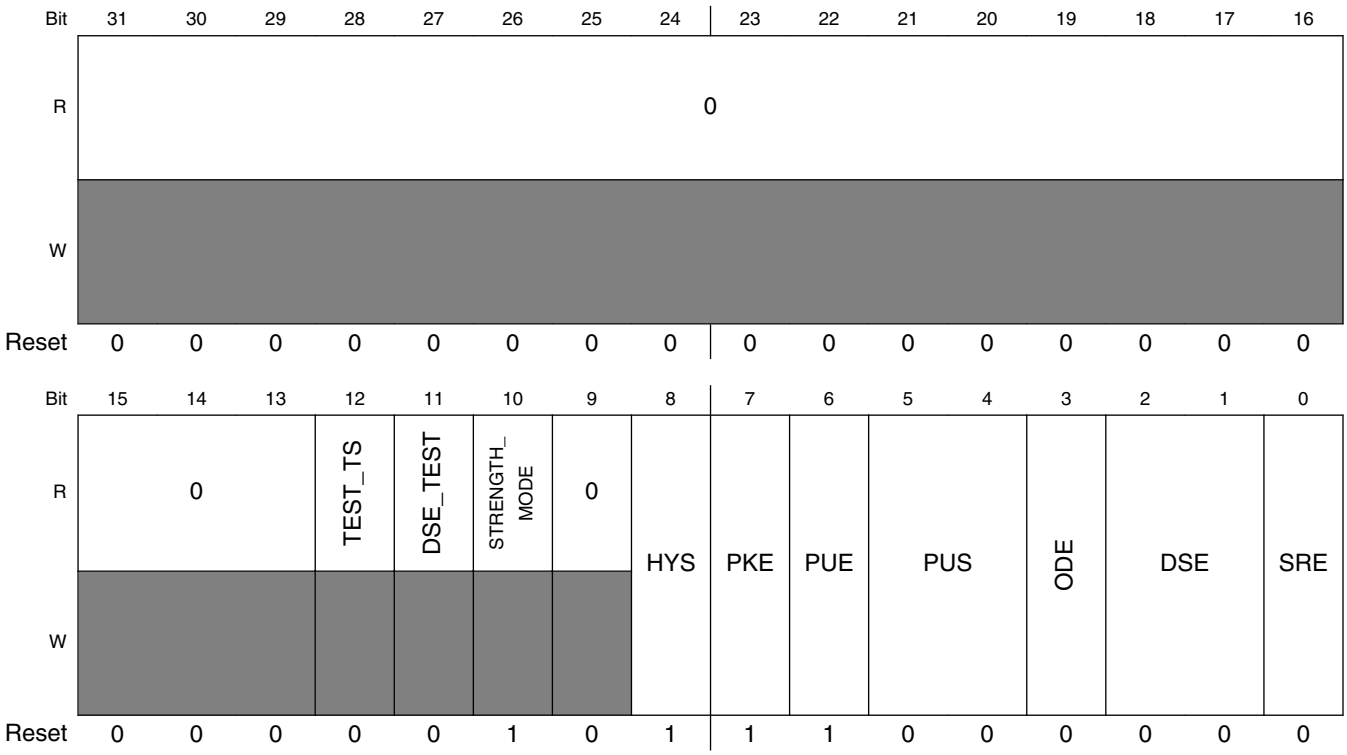
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_RESET_IN_B field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: RESET_IN_B. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Read Only Field 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Read Only Field 1 Pull
5–4 PUS	Pull Up / Down Config. Field Read Only Field 10 100 [KOhm] Pull Up
3 ODE	Open Drain Enable Field Read Only Field 0 Open Drain Disabled
2–1 DSE	Drive Strength Field Read Only Field 00 Low Drive Strength
0 SRE	Slew Rate Field Read Only Field 0 Slow Slew Rate

43.3.443 IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE0 (IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE0)

Address: IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE0 is 53FA_8000h base + 6E8h offset = 53FA_86E8h



IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE0 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE0 field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: BOOT_MODE0. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: BOOT_MODE0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Read Only Field 1 Pull
5–4 PUS	Pull Up / Down Config. Field Read Only Field 00 100 [KOhm] Pull Down
3 ODE	Open Drain Enable Field Read Only Field 0 Open Drain Disabled
2–1 DSE	Drive Strength Field Read Only Field 00 Low Drive Strength
0 SRE	Slew Rate Field Read Only Field 0 Slow Slew Rate

43.3.444 IOMUXC_SW_PAD_CTL_PAD_TEST_MODE (IOMUXC_SW_PAD_CTL_PAD_TEST_MODE)

Address: IOMUXC_SW_PAD_CTL_PAD_TEST_MODE is 53FA_8000h base + 6ECh offset = 53FA_86ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TEST_TS	DSE_TEST	STRENGTH_MODE	0	HYS	PKE	PUE	PUS		ODE	DSE		SRE
W																
Reset	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_TEST_MODE field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 TEST_TS	test_ts Field Read Only Field 0 Test ts disabled
11 DSE_TEST	dse test Field Read Only Field 0 Regular
10 STRENGTH_MODE	strength mode Field Read Only Field 1 4 level mode
9 Reserved	This read-only field is reserved and always has the value zero. Reserved

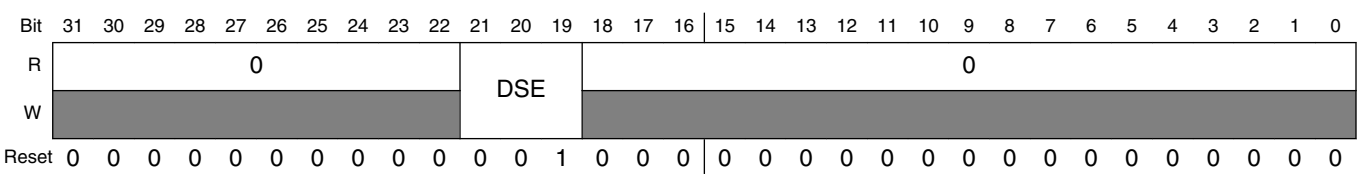
Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_TEST_MODE field descriptions (continued)

Field	Description
8 HYS	Hyst. Enable Field Select one out of next values for pad: TEST_MODE. 0 Hysteresis Disabled 1 Hysteresis Enabled
7 PKE	Pull / Keep Enable Field Select one out of next values for pad: TEST_MODE. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6 PUE	Pull / Keep Select Field Read Only Field 1 Pull
5–4 PUS	Pull Up / Down Config. Field Read Only Field 00 100 [KOhm] Pull Down
3 ODE	Open Drain Enable Field Read Only Field 0 Open Drain Disabled
2–1 DSE	Drive Strength Field Read Only Field 00 Low Drive Strength
0 SRE	Slew Rate Field Read Only Field 0 Slow Slew Rate

43.3.445 IOMUXC_SW_PAD_CTL_GRP_ADDDS (IOMUXC_SW_PAD_CTL_GRP_ADDDS)

Address: IOMUXC_SW_PAD_CTL_GRP_ADDDS is 53FA_8000h base + 6F0h offset = 53FA_86F0h

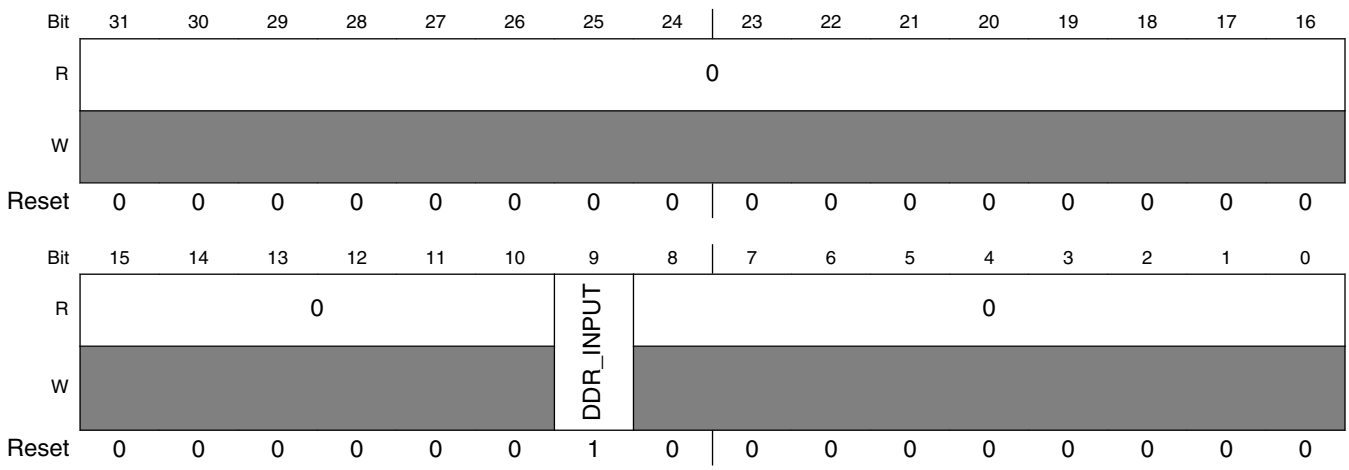


IOMUXC_SW_PAD_CTL_GRP_ADDDS field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value zero. Reserved
21–19 DSE	Refer to Table 43-2.
18–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.446 IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL (IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL)

Address: IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL is 53FA_8000h base + 6F4h offset = 53FA_86F4h

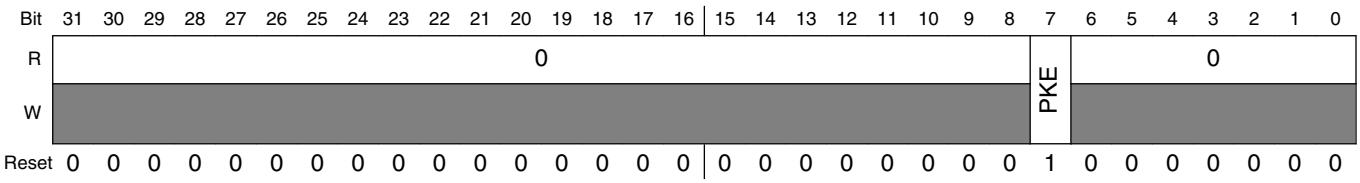


IOMUXC_SW_PAD_CTL_GRP_DDRMODE_CTL field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Select one out of next values for group: DDRMODE_CTL (Pads: DRAM_SDQS0 DRAM_SDQS1 DRAM_SDQS2 DRAM_SDQS3). 0 CMOS input type 1 DDR2 input type
8–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.447 IOMUXC_SW_PAD_CTL_GRP_DDRPKE (IOMUXC_SW_PAD_CTL_GRP_DDRPKE)

Address: IOMUXC_SW_PAD_CTL_GRP_DDRPKE is 53FA_8000h base + 6FCh offset = 53FA_86FCh

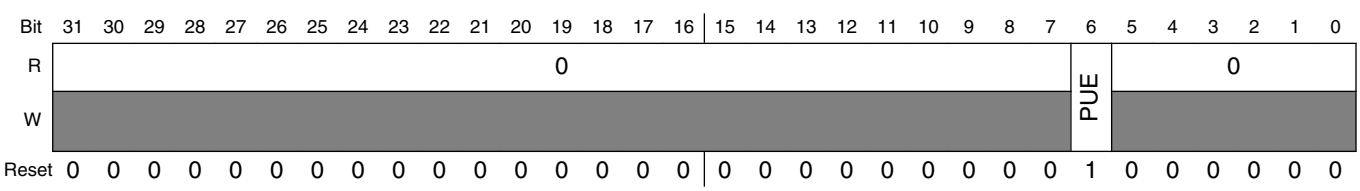


IOMUXC_SW_PAD_CTL_GRP_DDRPKE field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7 PKE	Pull / Keep Enable Field Select one out of next values for group: DDRPKE (Pads: DRAM_A0 DRAM_A1 DRAM_A10 DRAM_A11 DRAM_A12 DRAM_A13 DRAM_A14 DRAM_A15 DRAM_A2 DRAM_A3 DRAM_A4 DRAM_A5 DRAM_A6 DRAM_A7 DRAM_A8 DRAM_A9 DRAM_CAS DRAM_CS0 DRAM_CS1 DRAM_D0 DRAM_D1 DRAM_D10 DRAM_D11 DRAM_D12 DRAM_D13 DRAM_D14 DRAM_D15 DRAM_D16 DRAM_D17 DRAM_D18 DRAM_D19 DRAM_D2 DRAM_D20 DRAM_D21 DRAM_D22 DRAM_D23 DRAM_D24 DRAM_D25 DRAM_D26 DRAM_D27 DRAM_D28 DRAM_D29 DRAM_D3 DRAM_D30 DRAM_D31 DRAM_D4 DRAM_D5 DRAM_D6 DRAM_D7 DRAM_D8 DRAM_D9 DRAM_DQM0 DRAM_DQM1 DRAM_DQM2 DRAM_DQM3 DRAM_RAS DRAM_SDBA0 DRAM_SDBA1 DRAM_SDBA2 DRAM_SDCLK_0 DRAM_SDCLK_1 DRAM_SDWE). 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
6–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.448 IOMUXC_SW_PAD_CTL_GRP_DDRPK (IOMUXC_SW_PAD_CTL_GRP_DDRPK)

Address: IOMUXC_SW_PAD_CTL_GRP_DDRPK is 53FA_8000h base + 708h offset = 53FA_8708h

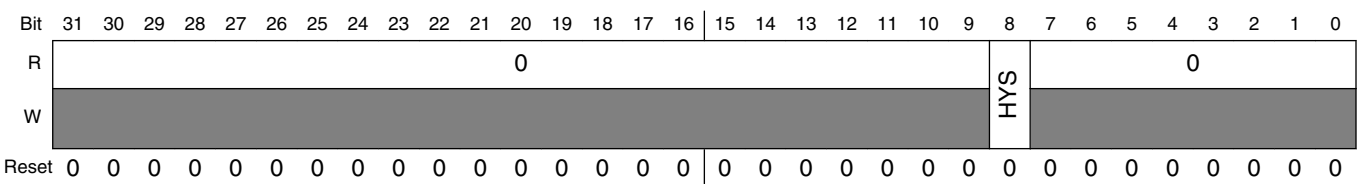


IOMUXC_SW_PAD_CTL_GRP_DDRPK field descriptions

Field	Description
31–7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6 PUE	Pull / Keep Select Field Select one out of next values for group: DDRPK (Pads: DRAM_A0 DRAM_A1 DRAM_A10 DRAM_A11 DRAM_A12 DRAM_A13 DRAM_A14 DRAM_A15 DRAM_A2 DRAM_A3 DRAM_A4 DRAM_A5 DRAM_A6 DRAM_A7 DRAM_A8 DRAM_A9 DRAM_CAS DRAM_CS0 DRAM_CS1 DRAM_D0 DRAM_D1 DRAM_D10 DRAM_D11 DRAM_D12 DRAM_D13 DRAM_D14 DRAM_D15 DRAM_D16 DRAM_D17 DRAM_D18 DRAM_D19 DRAM_D2 DRAM_D20 DRAM_D21 DRAM_D22 DRAM_D23 DRAM_D24 DRAM_D25 DRAM_D26 DRAM_D27 DRAM_D28 DRAM_D29 DRAM_D3 DRAM_D30 DRAM_D31 DRAM_D4 DRAM_D5 DRAM_D6 DRAM_D7 DRAM_D8 DRAM_D9 DRAM_DQM0 DRAM_DQM1 DRAM_DQM2 DRAM_DQM3 DRAM_RAS DRAM_SDBA0 DRAM_SDBA1 DRAM_SDBA2 DRAM_SDCLK_0 DRAM_SDCLK_1 DRAM_SDWE). 0 Keeper 1 Pull
5–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.449 IOMUXC_SW_PAD_CTL_GRP_DDRHYS (IOMUXC_SW_PAD_CTL_GRP_DDRHYS)

Address: IOMUXC_SW_PAD_CTL_GRP_DDRHYS is 53FA_8000h base + 710h offset = 53FA_8710h



IOMUXC_SW_PAD_CTL_GRP_DDRHYS field descriptions

Field	Description
31–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8 HYS	Hyst. Enable Field Select one out of next values for group: DDRHYS (Pads: DRAM_D0 DRAM_D1 DRAM_D10 DRAM_D11 DRAM_D12 DRAM_D13 DRAM_D14 DRAM_D15 DRAM_D16 DRAM_D17 DRAM_D18 DRAM_D19 DRAM_D2 DRAM_D20 DRAM_D21 DRAM_D22 DRAM_D23 DRAM_D24 DRAM_D25 DRAM_D26 DRAM_D27 DRAM_D28 DRAM_D29 DRAM_D3 DRAM_D30 DRAM_D31 DRAM_D4 DRAM_D5 DRAM_D6 DRAM_D7 DRAM_D8 DRAM_D9 DRAM_SDQS0 DRAM_SDQS1 DRAM_SDQS2 DRAM_SDQS3). 0 Hysteresis Disabled 1 Hysteresis Enabled
7–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.450 IOMUXC_SW_PAD_CTL_GRP_DDRMODE (IOMUXC_SW_PAD_CTL_GRP_DDRMODE)

Address: IOMUXC_SW_PAD_CTL_GRP_DDRMODE is 53FA_8000h base + 714h offset = 53FA_8714h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						DDR_INPUT		0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_PAD_CTL_GRP_DDRMODE field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DDR_INPUT	DDR / CMOS Input Mode Field Select one out of next values for group: DDRMODE (Pads: DRAM_D0 DRAM_D1 DRAM_D10 DRAM_D11 DRAM_D12 DRAM_D13 DRAM_D14 DRAM_D15 DRAM_D16 DRAM_D17 DRAM_D18 DRAM_D19 DRAM_D2 DRAM_D20 DRAM_D21 DRAM_D22 DRAM_D23 DRAM_D24 DRAM_D25 DRAM_D26 DRAM_D27 DRAM_D28 DRAM_D29 DRAM_D3 DRAM_D30 DRAM_D31 DRAM_D4 DRAM_D5 DRAM_D6 DRAM_D7 DRAM_D8 DRAM_D9). 0 CMOS input type 1 DDR2 input type
8–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.451 IOMUXC_SW_PAD_CTL_GRP_B0DS (IOMUXC_SW_PAD_CTL_GRP_B0DS)

Address: IOMUXC_SW_PAD_CTL_GRP_B0DS is 53FA_8000h base + 718h offset = 53FA_8718h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										DSE						0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_PAD_CTL_GRP_B0DS field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value zero. Reserved
21–19 DSE	Refer to Table 43-2.
18–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.452 IOMUXC_SW_PAD_CTL_GRP_B1DS (IOMUXC_SW_PAD_CTL_GRP_B1DS)

Address: IOMUXC_SW_PAD_CTL_GRP_B1DS is 53FA_8000h base + 71Ch offset = 53FA_871Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										DSE			0																			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_PAD_CTL_GRP_B1DS field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value zero. Reserved
21–19 DSE	Refer to Table 43-2.
18–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.453 IOMUXC_SW_PAD_CTL_GRP_CTLDS (IOMUXC_SW_PAD_CTL_GRP_CTLDS)

Address: IOMUXC_SW_PAD_CTL_GRP_CTLDS is 53FA_8000h base + 720h offset = 53FA_8720h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										DSE			0																			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_PAD_CTL_GRP_CTLDS field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value zero. Reserved
21–19 DSE	Refer to Table 43-2.
18–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.454 IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE (IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE)

Address: IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE is 53FA_8000h base + 724h offset = 53FA_8724h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					DDR_SEL		0																								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–25 DDR_SEL	ddr_sel Field Select one out of next values for group: DDR_TYPE (Pads: DRAM_A0 DRAM_A1 DRAM_A10 DRAM_A11 DRAM_A12 DRAM_A13 DRAM_A14 DRAM_A15 DRAM_A2 DRAM_A3 DRAM_A4 DRAM_A5 DRAM_A6 DRAM_A7 DRAM_A8 DRAM_A9 DRAM_CAS DRAM_CS0 DRAM_CS1 DRAM_D0 DRAM_D1 DRAM_D10 DRAM_D11 DRAM_D12 DRAM_D13 DRAM_D14 DRAM_D15 DRAM_D16 DRAM_D17 DRAM_D18 DRAM_D19 DRAM_D2 DRAM_D20 DRAM_D21 DRAM_D22 DRAM_D23 DRAM_D24 DRAM_D25 DRAM_D26 DRAM_D27 DRAM_D28 DRAM_D29 DRAM_D3 DRAM_D30 DRAM_D31 DRAM_D4 DRAM_D5 DRAM_D6 DRAM_D7 DRAM_D8 DRAM_D9 DRAM_DQM0 DRAM_DQM1 DRAM_DQM2 DRAM_DQM3 DRAM_RAS DRAM_SDBA0 DRAM_SDBA1 DRAM_SDBA2 DRAM_SDCKE0 DRAM_SDCKE1 DRAM_SDCLK_0 DRAM_SDCLK_1 DRAM_SDODT0 DRAM_SDODT1 DRAM_SDQS0 DRAM_SDQS1 DRAM_SDQS2 DRAM_SDQS3 DRAM_SDWE). 2-bit control signal to select ZQ resistor for driver calibration: NVCC_EMI_DRAM=1.8+/-0.1V (DDR2), ddr_sel='00': 300 Ohm NVCC_EMI_DRAM=1.8+/-0.1V (DDR2), ddr_sel='01': 180 Ohm NVCC_EMI_DRAM=1.8+/-0.1V (DDR2), ddr_sel='10': 200 Ohm NVCC_EMI_DRAM=1.8+/-0.1V (DDR2), ddr_sel='11': 140 Ohm NVCC_EMI_DRAM=1.5V+/-5% (DDR3), ddr_sel='00': 240 Ohm NVCC_EMI_DRAM=1.5V+/-5% (DDR3), ddr_sel="01: 160 Ohm NVCC_EMI_DRAM=1.2V+10/-5% (LPDDR2), ddr_sel='01': 160 Ohm NVCC_EMI_DRAM=1.2V+10/-5% (LPDDR2), ddr_sel='10': 240 Ohm

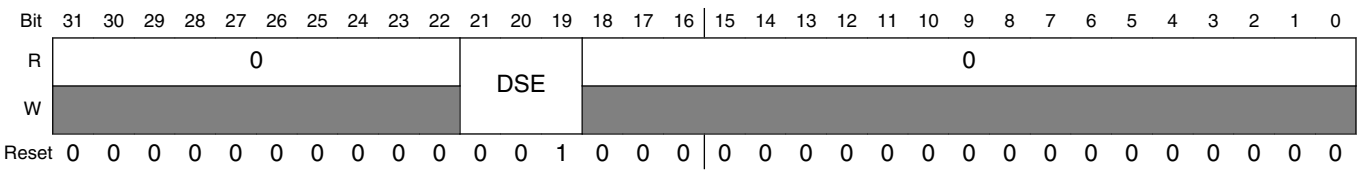
Table continues on the next page...

IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE field descriptions (continued)

Field	Description
	NVCC_EMI_DRAM=1.2V+10/-5% (LPDDR2), ddr_sel='11': 160 Ohm
24–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.455 IOMUXC_SW_PAD_CTL_GRP_B2DS (IOMUXC_SW_PAD_CTL_GRP_B2DS)

Address: IOMUXC_SW_PAD_CTL_GRP_B2DS is 53FA_8000h base + 728h offset = 53FA_8728h

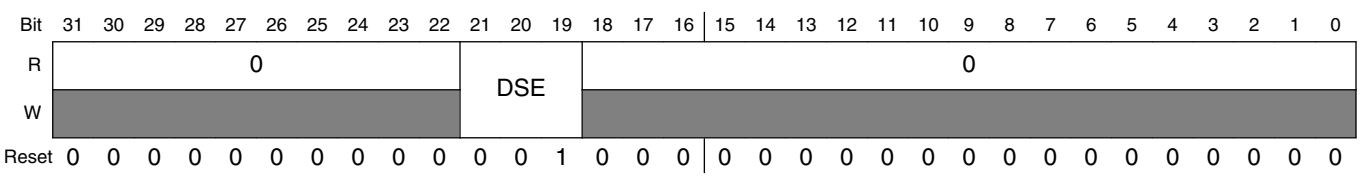


IOMUXC_SW_PAD_CTL_GRP_B2DS field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value zero. Reserved
21–19 DSE	Refer to Table 43-2.
18–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.456 IOMUXC_SW_PAD_CTL_GRP_B3DS (IOMUXC_SW_PAD_CTL_GRP_B3DS)

Address: IOMUXC_SW_PAD_CTL_GRP_B3DS is 53FA_8000h base + 72Ch offset = 53FA_872Ch



IOMUXC_SW_PAD_CTL_GRP_B3DS field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_SW_PAD_CTL_GRP_B3DS field descriptions (continued)

Field	Description
21–19 DSE	Refer to Table 43-2.
18–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

43.3.457 IOMUXC_AUDMUX_P4_INPUT_DA_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P4_INPUT_DA_AMX_SELECT_INPUT)

Address: IOMUXC_AUDMUX_P4_INPUT_DA_AMX_SELECT_INPUT is 53FA_8000h base + 730h offset = 53FA_8730h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_AUDMUX_P4_INPUT_DA_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audmux, In Pin: p4_input_da_amx 0 Selecting Pad: DISP0_DAT23 for Mode: ALT3. 1 Selecting Pad: SD2_DATA0 for Mode: ALT3.

43.3.458 IOMUXC_AUDMUX_P4_INPUT_DB_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P4_INPUT_DB_AMX_SELECT_INPUT)

Address: IOMUXC_AUDMUX_P4_INPUT_DB_AMX_SELECT_INPUT is 53FA_8000h base + 734h offset = 53FA_8734h

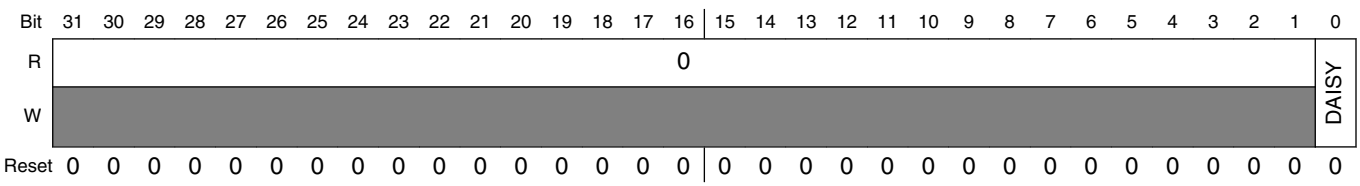
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_AUDMUX_P4_INPUT_DB_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audmux, In Pin: p4_input_db_amx 0 Selecting Pad: DISP0_DAT21 for Mode: ALT3. 1 Selecting Pad: SD2_DATA2 for Mode: ALT3.

43.3.459 IOMUXC_AUDMUX_P4_INPUT_RXCLK_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P4_INPUT_RXCLK_AMX_SELECT_INPUT)

Address: IOMUXC_AUDMUX_P4_INPUT_RXCLK_AMX_SELECT_INPUT is 53FA_8000h base + 738h offset = 53FA_8738h

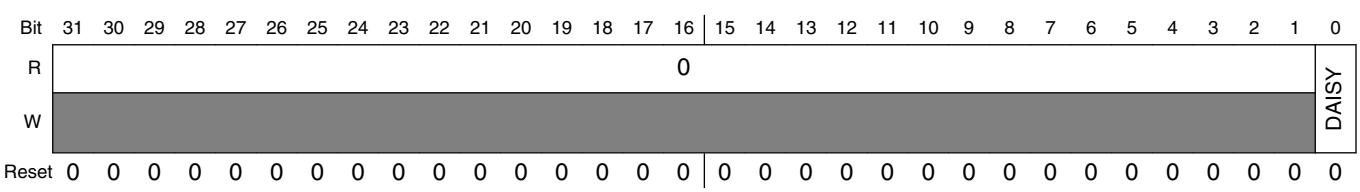


IOMUXC_AUDMUX_P4_INPUT_RXCLK_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audmux, In Pin: p4_input_rxclk_amx 0 Selecting Pad: DISP0_DAT19 for Mode: ALT4. 1 Selecting Pad: SD2_CMD for Mode: ALT3.

43.3.460 IOMUXC_AUDMUX_P4_INPUT_RXFS_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P4_INPUT_RXFS_AMX_SELECT_INPUT)

Address: IOMUXC_AUDMUX_P4_INPUT_RXFS_AMX_SELECT_INPUT is 53FA_8000h base + 73Ch offset = 53FA_873Ch



IOMUXC_AUDMUX_P4_INPUT_RXFS_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audmux, In Pin: p4_input_rxfs_amx 0 Selecting Pad: DISP0_DAT18 for Mode: ALT4. 1 Selecting Pad: SD2_CLK for Mode: ALT3.

43.3.461 IOMUXC_AUDMUX_P4_INPUT_TXCLK_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P4_INPUT_TXCLK_AMX_SELECT_INPUT)

Address: IOMUXC_AUDMUX_P4_INPUT_TXCLK_AMX_SELECT_INPUT is 53FA_8000h base + 740h offset = 53FA_8740h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_AUDMUX_P4_INPUT_TXCLK_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audmux, In Pin: p4_input_txclk_amx 0 Selecting Pad: DISP0_DAT20 for Mode: ALT3. 1 Selecting Pad: SD2_DATA3 for Mode: ALT3.

43.3.462 IOMUXC_AUDMUX_P4_INPUT_TXFS_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P4_INPUT_TXFS_AMX_SELECT_INPUT)

Address: IOMUXC_AUDMUX_P4_INPUT_TXFS_AMX_SELECT_INPUT is 53FA_8000h base + 744h offset = 53FA_8744h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_AUDMUX_P4_INPUT_TXFS_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audmux, In Pin: p4_input_txfs_amx 0 Selecting Pad: DISP0_DAT22 for Mode: ALT3. 1 Selecting Pad: SD2_DATA1 for Mode: ALT3.

43.3.463 IOMUXC_AUDMUX_P5_INPUT_DA_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P5_INPUT_DA_AMX_SELECT_INPUT)

Address: IOMUXC_AUDMUX_P5_INPUT_DA_AMX_SELECT_INPUT is 53FA_8000h base + 748h offset = 53FA_8748h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_AUDMUX_P5_INPUT_DA_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audmux, In Pin: p5_input_da_amx 0 Selecting Pad: KEY_ROW1 for Mode: ALT2. 1 Selecting Pad: DISP0_DAT19 for Mode: ALT3.

43.3.464 IOMUXC_AUDMUX_P5_INPUT_DB_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P5_INPUT_DB_AMX_SELECT_INPUT)

Address: IOMUXC_AUDMUX_P5_INPUT_DB_AMX_SELECT_INPUT is 53FA_8000h base + 74Ch offset = 53FA_874Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_AUDMUX_P5_INPUT_DB_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audmux, In Pin: p5_input_db_amx 0 Selecting Pad: KEY_ROW0 for Mode: ALT2. 1 Selecting Pad: DISP0_DAT17 for Mode: ALT3.

43.3.465 IOMUXC_AUDMUX_P5_INPUT_RXCLK_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P5_INPUT_RXCLK_AMX_SELECT_INPUT)

Address: IOMUXC_AUDMUX_P5_INPUT_RXCLK_AMX_SELECT_INPUT is 53FA_8000h base + 750h offset = 53FA_8750h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_AUDMUX_P5_INPUT_RXCLK_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audmux, In Pin: p5_input_rxclk_amx 0 Selecting Pad: DISP0_DAT14 for Mode: ALT3. 1 Selecting Pad: EIM_D25 for Mode: ALT5.

43.3.466 IOMUXC_AUDMUX_P5_INPUT_RXFS_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P5_INPUT_RXFS_AMX_SELECT_INPUT)

Address: IOMUXC_AUDMUX_P5_INPUT_RXFS_AMX_SELECT_INPUT is 53FA_8000h base + 754h offset = 53FA_8754h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_AUDMUX_P5_INPUT_RXFS_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audmux, In Pin: p5_input_rxfs_amx 0 Selecting Pad: DISP0_DAT13 for Mode: ALT3. 1 Selecting Pad: EIM_D24 for Mode: ALT5.

43.3.467 IOMUXC_AUDMUX_P5_INPUT_TXCLK_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P5_INPUT_TXCLK_AMX_SELECT_INPUT)

Address: IOMUXC_AUDMUX_P5_INPUT_TXCLK_AMX_SELECT_INPUT is 53FA_8000h base + 758h offset = 53FA_8758h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_AUDMUX_P5_INPUT_TXCLK_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audmux, In Pin: p5_input_txclk_amx 0 Selecting Pad: KEY_COL0 for Mode: ALT2. 1 Selecting Pad: DISP0_DAT16 for Mode: ALT3.

43.3.468 IOMUXC_AUDMUX_P5_INPUT_TXFS_AMX_SELECT_INPUT (IOMUXC_AUDMUX_P5_INPUT_TXFS_AMX_SELECT_INPUT)

Address: IOMUXC_AUDMUX_P5_INPUT_TXFS_AMX_SELECT_INPUT is 53FA_8000h base + 75Ch offset = 53FA_875Ch

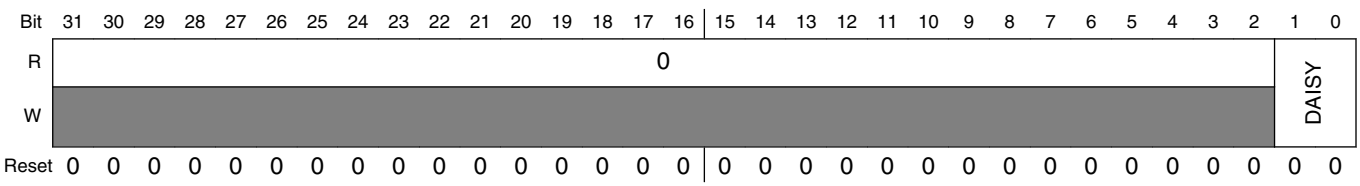
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_AUDMUX_P5_INPUT_TXFS_AMX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: audmux, In Pin: p5_input_txfs_amx 0 Selecting Pad: KEY_COL1 for Mode: ALT2. 1 Selecting Pad: DISP0_DAT18 for Mode: ALT3.

43.3.469 IOMUXC_CAN1_IPP_IND_CANRX_SELECT_INPUT (IOMUXC_CAN1_IPP_IND_CANRX_SELECT_INPUT)

Address: IOMUXC_CAN1_IPP_IND_CANRX_SELECT_INPUT is 53FA_8000h base + 760h offset = 53FA_8760h

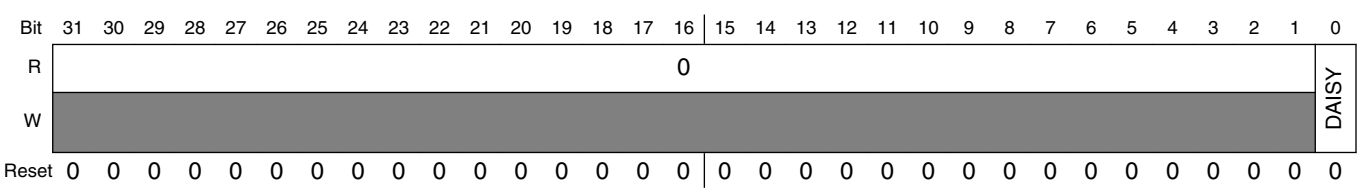


IOMUXC_CAN1_IPP_IND_CANRX_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: can1, In Pin: ipp_ind_canrx 00 Selecting Pad: KEY_ROW2 for Mode: ALT2. 01 Selecting Pad: PATA_DIOR for Mode: ALT4. 10 Selecting Pad: GPIO_8 for Mode: ALT3.

43.3.470 IOMUXC_CAN2_IPP_IND_CANRX_SELECT_INPUT (IOMUXC_CAN2_IPP_IND_CANRX_SELECT_INPUT)

Address: IOMUXC_CAN2_IPP_IND_CANRX_SELECT_INPUT is 53FA_8000h base + 764h offset = 53FA_8764h



IOMUXC_CAN2_IPP_IND_CANRX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: can2, In Pin: ipp_ind_canrx 0 Selecting Pad: KEY_ROW4 for Mode: ALT2. 1 Selecting Pad: PATA_IORDY for Mode: ALT4.

43.3.471 IOMUXC_CCM_IPP_ASRC_EXT_SELECT_INPUT (IOMUXC_CCM_IPP_ASRC_EXT_SELECT_INPUT)

Address: IOMUXC_CCM_IPP_ASRC_EXT_SELECT_INPUT is 53FA_8000h base + 768h offset = 53FA_8768h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_CCM_IPP_ASRC_EXT_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ccm, In Pin: ipp_asrc_ext 0 Selecting Pad: KEY_ROW3 for Mode: ALT3. 1 Selecting Pad: GPIO_18 for Mode: ALT5.

43.3.472 IOMUXC_CCM_IPP_DI1_CLK_SELECT_INPUT (IOMUXC_CCM_IPP_DI1_CLK_SELECT_INPUT)

Address: IOMUXC_CCM_IPP_DI1_CLK_SELECT_INPUT is 53FA_8000h base + 76Ch offset = 53FA_876Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_CCM_IPP_DI1_CLK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ccm, In Pin: ipp_di1_clk 0 Selecting Pad: EIM_EB2 for Mode: ALT2. 1 Selecting Pad: EIM_DA13 for Mode: ALT4.

43.3.473 IOMUXC_CCM_PLL1_BYPASS_CLK_SELECT_INPUT (IOMUXC_CCM_PLL1_BYPASS_CLK_SELECT_INPUT)

Address: IOMUXC_CCM_PLL1_BYPASS_CLK_SELECT_INPUT is 53FA_8000h base + 770h offset = 53FA_8770h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_CCM_PLL1_BYPASS_CLK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ccm, In Pin: pll1_bypass_clk 0 Selecting Pad: SD1_CMD for Mode: ALT7. 1 Selecting Pad: GPIO_5 for Mode: ALT7.

43.3.474 IOMUXC_CCM_PLL2_BYPASS_CLK_SELECT_INPUT (IOMUXC_CCM_PLL2_BYPASS_CLK_SELECT_INPUT)

Address: IOMUXC_CCM_PLL2_BYPASS_CLK_SELECT_INPUT is 53FA_8000h base + 774h offset = 53FA_8774h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_CCM_PLL2_BYPASS_CLK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ccm, In Pin: pll2_bypass_clk 0 Selecting Pad: SD1_DATA2 for Mode: ALT7. 1 Selecting Pad: GPIO_7 for Mode: ALT7.

43.3.475 IOMUXC_CCM_PLL3_BYPASS_CLK_SELECT_INPUT (IOMUXC_CCM_PLL3_BYPASS_CLK_SELECT_INPUT)

Address: IOMUXC_CCM_PLL3_BYPASS_CLK_SELECT_INPUT is 53FA_8000h base + 778h offset = 53FA_8778h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_CCM_PLL3_BYPASS_CLK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ccm, In Pin: pll3_bypass_clk 0 Selecting Pad: SD1_DATA0 for Mode: ALT7. 1 Selecting Pad: GPIO_8 for Mode: ALT7.

43.3.476 IOMUXC_CCM_PLL4_BYPASS_CLK_SELECT_INPUT (IOMUXC_CCM_PLL4_BYPASS_CLK_SELECT_INPUT)

Address: IOMUXC_CCM_PLL4_BYPASS_CLK_SELECT_INPUT is 53FA_8000h base + 77Ch offset = 53FA_877Ch

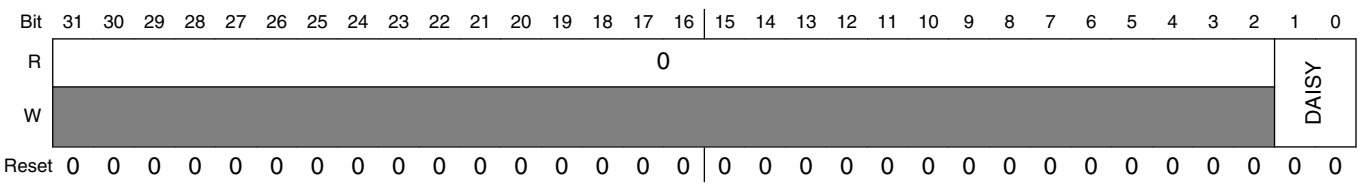
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_CCM_PLL4_BYPASS_CLK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ccm, In Pin: pll4_bypass_clk 0 Selecting Pad: KEY_ROW3 for Mode: ALT6. 1 Selecting Pad: SD1_DATA1 for Mode: ALT7.

43.3.477 IOMUXC_CSPI_IPP_CSPI_CLK_IN_SELECT_INPUT (IOMUXC_CSPI_IPP_CSPI_CLK_IN_SELECT_INPUT)

Address: IOMUXC_CSPI_IPP_CSPI_CLK_IN_SELECT_INPUT is 53FA_8000h base + 780h offset = 53FA_8780h

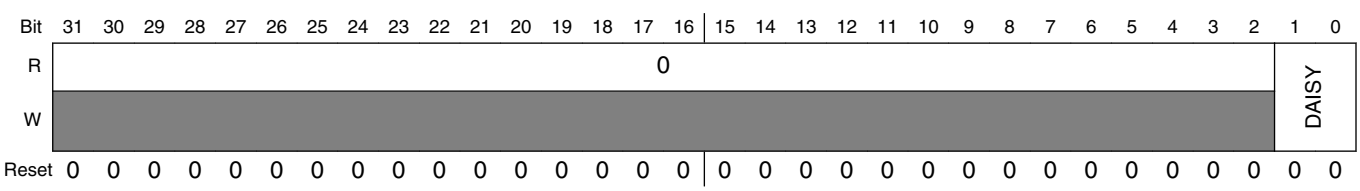


IOMUXC_CSPI_IPP_CSPI_CLK_IN_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: cspi, In Pin: ipp_cspi_clk_in 00 Selecting Pad: DISP0_DAT0 for Mode: ALT2. 01 Selecting Pad: EIM_D21 for Mode: ALT4. 10 Selecting Pad: SD1_CLK for Mode: ALT5. 11 Selecting Pad: SD2_CLK for Mode: ALT5.

43.3.478 IOMUXC_CSPI_IPP_IND_MISO_SELECT_INPUT (IOMUXC_CSPI_IPP_IND_MISO_SELECT_INPUT)

Address: IOMUXC_CSPI_IPP_IND_MISO_SELECT_INPUT is 53FA_8000h base + 784h offset = 53FA_8784h



IOMUXC_CSPI_IPP_IND_MISO_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: cspi, In Pin: ipp_ind_miso 00 Selecting Pad: DISP0_DAT2 for Mode: ALT2. 01 Selecting Pad: EIM_D22 for Mode: ALT4. 10 Selecting Pad: SD1_DATA0 for Mode: ALT5. 11 Selecting Pad: SD2_DATA0 for Mode: ALT5.

43.3.479 IOMUXC_CSPI_IPP_IND_MOSI_SELECT_INPUT (IOMUXC_CSPI_IPP_IND_MOSI_SELECT_INPUT)

Address: IOMUXC_CSPI_IPP_IND_MOSI_SELECT_INPUT is 53FA_8000h base + 788h offset = 53FA_8788h

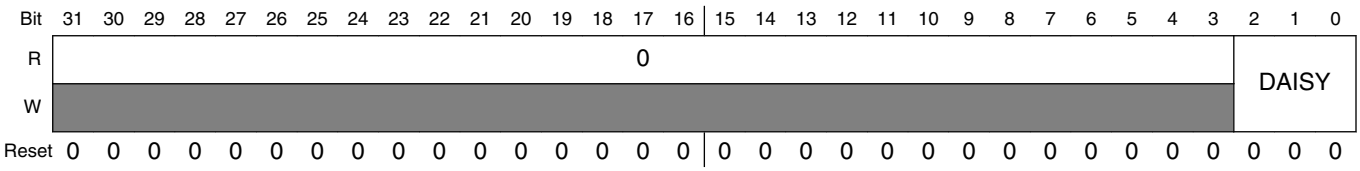
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																														DAISY	
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_CSPI_IPP_IND_MOSI_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: cspi, In Pin: ipp_ind_mosi 00 Selecting Pad: DISP0_DAT1 for Mode: ALT2. 01 Selecting Pad: EIM_D28 for Mode: ALT4. 10 Selecting Pad: SD1_CMD for Mode: ALT5. 11 Selecting Pad: SD2_CMD for Mode: ALT5.

43.3.480 IOMUXC_CSPI_IPP_IND_SS0_B_SELECT_INPUT (IOMUXC_CSPI_IPP_IND_SS0_B_SELECT_INPUT)

Address: IOMUXC_CSPI_IPP_IND_SS0_B_SELECT_INPUT is 53FA_8000h base + 78Ch offset = 53FA_878Ch

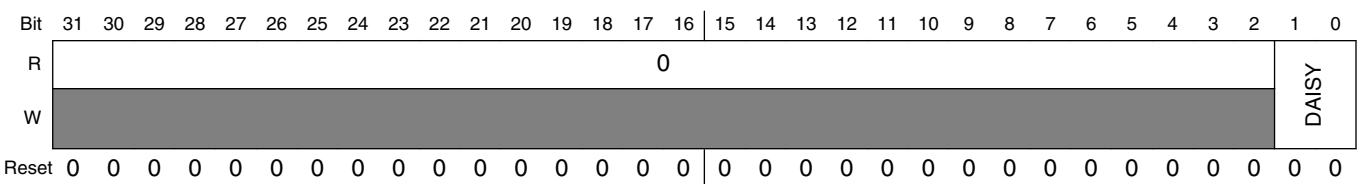


IOMUXC_CSPI_IPP_IND_SS0_B_SELECT_INPUT field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: cspi, In Pin: ipp_ind_ss0_b 000 Selecting Pad: DISP0_DAT3 for Mode: ALT2. 001 Selecting Pad: EIM_D20 for Mode: ALT4. 010 Selecting Pad: EIM_D29 for Mode: ALT4. 011 Selecting Pad: SD1_DATA1 for Mode: ALT5. 100 Selecting Pad: SD2_DATA1 for Mode: ALT5.

43.3.481 IOMUXC_CSPI_IPP_IND_SS1_B_SELECT_INPUT (IOMUXC_CSPI_IPP_IND_SS1_B_SELECT_INPUT)

Address: IOMUXC_CSPI_IPP_IND_SS1_B_SELECT_INPUT is 53FA_8000h base + 790h offset = 53FA_8790h



IOMUXC_CSPI_IPP_IND_SS1_B_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: cspi, In Pin: ipp_ind_ss1_b 00 Selecting Pad: DISP0_DAT4 for Mode: ALT2. 01 Selecting Pad: EIM_A25 for Mode: ALT4.

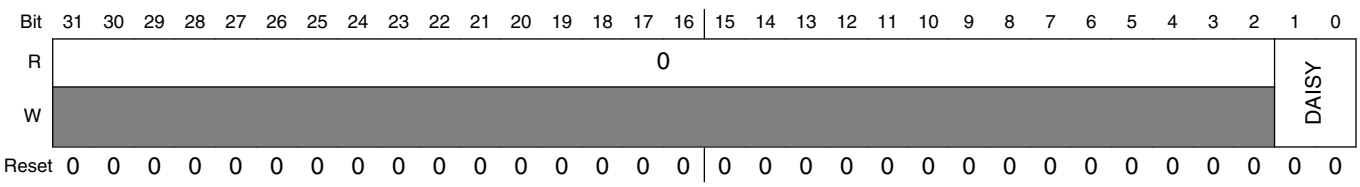
Table continues on the next page...

IOMUXC_CSPI_IPP_IND_SS1_B_SELECT_INPUT field descriptions (continued)

Field	Description
	10 Selecting Pad: SD1_DATA2 for Mode: ALT5. 11 Selecting Pad: SD2_DATA2 for Mode: ALT5.

43.3.482 IOMUXC_CSPI_IPP_IND_SS2_B_SELECT_INPUT (IOMUXC_CSPI_IPP_IND_SS2_B_SELECT_INPUT)

Address: IOMUXC_CSPI_IPP_IND_SS2_B_SELECT_INPUT is 53FA_8000h base + 794h offset = 53FA_8794h

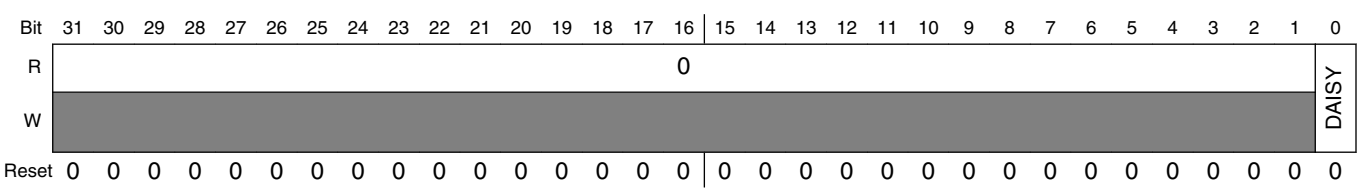


IOMUXC_CSPI_IPP_IND_SS2_B_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: cspi, In Pin: ipp_ind_ss2_b 00 Selecting Pad: DISP0_DAT5 for Mode: ALT2. 01 Selecting Pad: EIM_D24 for Mode: ALT4. 10 Selecting Pad: SD1_DATA3 for Mode: ALT5. 11 Selecting Pad: SD2_DATA3 for Mode: ALT5.

43.3.483 IOMUXC_CSPI_IPP_IND_SS3_B_SELECT_INPUT (IOMUXC_CSPI_IPP_IND_SS3_B_SELECT_INPUT)

Address: IOMUXC_CSPI_IPP_IND_SS3_B_SELECT_INPUT is 53FA_8000h base + 798h offset = 53FA_8798h

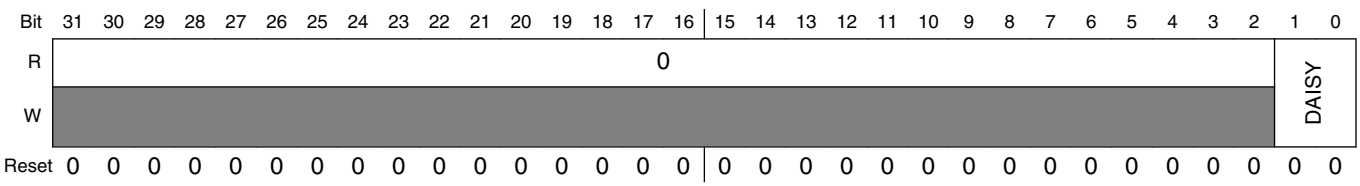


IOMUXC_CSPI_IPP_IND_SS3_B_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: cspi, In Pin: ipp_ind_ss3_b 0 Selecting Pad: DISP0_DAT6 for Mode: ALT2. 1 Selecting Pad: EIM_D25 for Mode: ALT4.

43.3.484 IOMUXC_ECSPI1_IPP_CSPI_CLK_IN_SELECT_INPUT (IOMUXC_ECSPI1_IPP_CSPI_CLK_IN_SELECT_INPUT)

Address: IOMUXC_ECSPI1_IPP_CSPI_CLK_IN_SELECT_INPUT is 53FA_8000h base + 79Ch offset = 53FA_879Ch

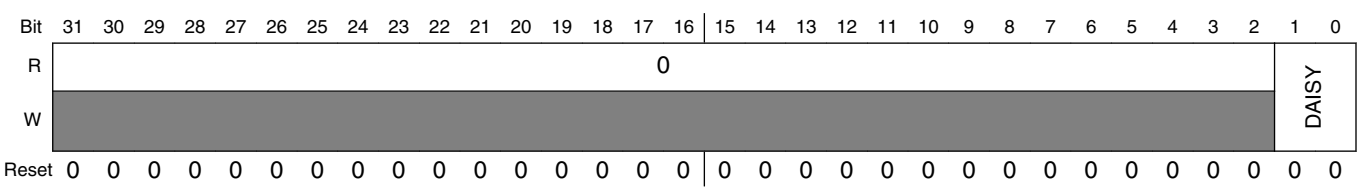


IOMUXC_ECSPI1_IPP_CSPI_CLK_IN_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi1, In Pin: ipp_cspi_clk_in 00 Selecting Pad: KEY_COL0 for Mode: ALT5. 01 Selecting Pad: DISP0_DAT20 for Mode: ALT2. 10 Selecting Pad: CSI0_DAT4 for Mode: ALT3. 11 Selecting Pad: EIM_D16 for Mode: ALT4.

43.3.485 IOMUXC_ECSPI1_IPP_IND_MISO_SELECT_INPUT (IOMUXC_ECSPI1_IPP_IND_MISO_SELECT_INPUT)

Address: IOMUXC_ECSPI1_IPP_IND_MISO_SELECT_INPUT is 53FA_8000h base + 7A0h offset = 53FA_87A0h



IOMUXC_ECSPI1_IPP_IND_MISO_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi1, In Pin: ipp_ind_miso 00 Selecting Pad: KEY_COL1 for Mode: ALT5. 01 Selecting Pad: DISP0_DAT22 for Mode: ALT2. 10 Selecting Pad: CSI0_DAT6 for Mode: ALT3. 11 Selecting Pad: EIM_D17 for Mode: ALT4.

43.3.486 IOMUXC_ECSPI1_IPP_IND_MOSI_SELECT_INPUT (IOMUXC_ECSPI1_IPP_IND_MOSI_SELECT_INPUT)

Address: IOMUXC_ECSPI1_IPP_IND_MOSI_SELECT_INPUT is 53FA_8000h base + 7A4h offset = 53FA_87A4h

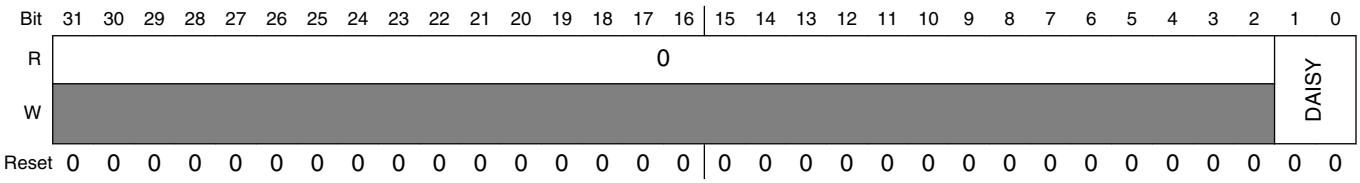
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																														DAISY	
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ECSPI1_IPP_IND_MOSI_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi1, In Pin: ipp_ind_mosi 00 Selecting Pad: KEY_ROW0 for Mode: ALT5. 01 Selecting Pad: DISP0_DAT21 for Mode: ALT2. 10 Selecting Pad: CSI0_DAT5 for Mode: ALT3. 11 Selecting Pad: EIM_D18 for Mode: ALT4.

43.3.487 IOMUXC_ECSPI1_IPP_IND_SS_B_0_SELECT_INPUT (IOMUXC_ECSPI1_IPP_IND_SS_B_0_SELECT_INPUT)

Address: IOMUXC_ECSPI1_IPP_IND_SS_B_0_SELECT_INPUT is 53FA_8000h base + 7A8h offset = 53FA_87A8h

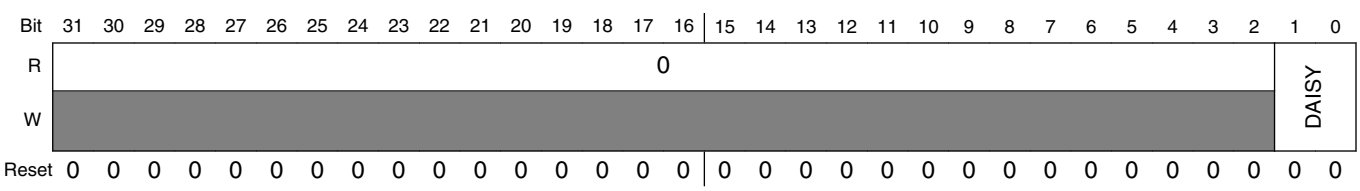


IOMUXC_ECSPI1_IPP_IND_SS_B_0_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi1, In Pin: ipp_ind_ss_b[0] 00 Selecting Pad: KEY_ROW1 for Mode: ALT5. 01 Selecting Pad: DISP0_DAT23 for Mode: ALT2. 10 Selecting Pad: CSI0_DAT7 for Mode: ALT3. 11 Selecting Pad: EIM_EB2 for Mode: ALT4.

43.3.488 IOMUXC_ECSPI1_IPP_IND_SS_B_1_SELECT_INPUT (IOMUXC_ECSPI1_IPP_IND_SS_B_1_SELECT_INPUT)

Address: IOMUXC_ECSPI1_IPP_IND_SS_B_1_SELECT_INPUT is 53FA_8000h base + 7ACh offset = 53FA_87ACh



IOMUXC_ECSPI1_IPP_IND_SS_B_1_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi1, In Pin: ipp_ind_ss_b[1] 00 Selecting Pad: KEY_COL2 for Mode: ALT5. 01 Selecting Pad: DISP0_DAT15 for Mode: ALT2. 10 Selecting Pad: EIM_D19 for Mode: ALT4.

43.3.489 IOMUXC_ECSPI1_IPP_IND_SS_B_2_SELECT_INPUT (IOMUXC_ECSPI1_IPP_IND_SS_B_2_SELECT_INPUT)

Address: IOMUXC_ECSPI1_IPP_IND_SS_B_2_SELECT_INPUT is 53FA_8000h base + 7B0h offset = 53FA_87B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ECSPI1_IPP_IND_SS_B_2_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi1, In Pin: ipp_ind_ss_b[2] 0 Selecting Pad: KEY_ROW2 for Mode: ALT5. 1 Selecting Pad: EIM_D24 for Mode: ALT3.

43.3.490 IOMUXC_ECSPI1_IPP_IND_SS_B_3_SELECT_INPUT (IOMUXC_ECSPI1_IPP_IND_SS_B_3_SELECT_INPUT)

Address: IOMUXC_ECSPI1_IPP_IND_SS_B_3_SELECT_INPUT is 53FA_8000h base + 7B4h offset = 53FA_87B4h

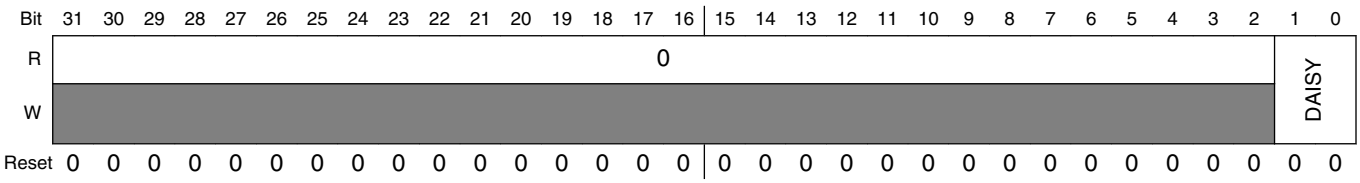
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ECSPI1_IPP_IND_SS_B_3_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi1, In Pin: ipp_ind_ss_b[3] 0 Selecting Pad: KEY_COL3 for Mode: ALT5. 1 Selecting Pad: EIM_D25 for Mode: ALT3.

43.3.491 IOMUXC_ECSPI2_IPP_CSPI_CLK_IN_SELECT_INPUT (IOMUXC_ECSPI2_IPP_CSPI_CLK_IN_SELECT_INPUT)

Address: IOMUXC_ECSPI2_IPP_CSPI_CLK_IN_SELECT_INPUT is 53FA_8000h base + 7B8h offset = 53FA_87B8h

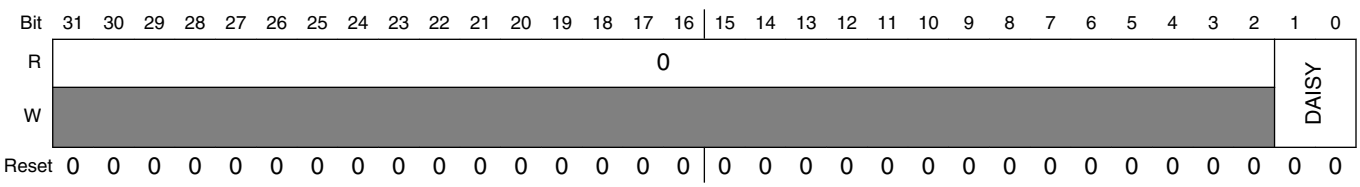


IOMUXC_ECSPI2_IPP_CSPI_CLK_IN_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi2, In Pin: ipp_cspi_clk_in 00 Selecting Pad: DISP0_DAT19 for Mode: ALT2. 01 Selecting Pad: CSI0_DAT8 for Mode: ALT3. 10 Selecting Pad: EIM_CS0 for Mode: ALT2.

43.3.492 IOMUXC_ECSPI2_IPP_IND_MISO_SELECT_INPUT (IOMUXC_ECSPI2_IPP_IND_MISO_SELECT_INPUT)

Address: IOMUXC_ECSPI2_IPP_IND_MISO_SELECT_INPUT is 53FA_8000h base + 7BCh offset = 53FA_87BCh

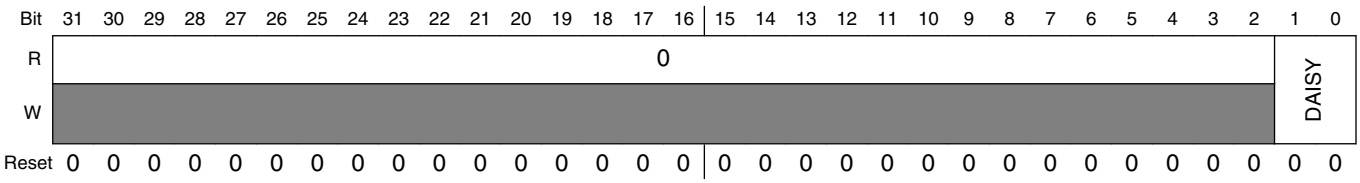


IOMUXC_ECSPI2_IPP_IND_MISO_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi2, In Pin: ipp_ind_miso 00 Selecting Pad: DISP0_DAT17 for Mode: ALT2. 01 Selecting Pad: CSI0_DAT10 for Mode: ALT3. 10 Selecting Pad: EIM_OE for Mode: ALT2.

43.3.493 IOMUXC_ECSPI2_IPP_IND_MOSI_SELECT_INPUT (IOMUXC_ECSPI2_IPP_IND_MOSI_SELECT_INPUT)

Address: IOMUXC_ECSPI2_IPP_IND_MOSI_SELECT_INPUT is 53FA_8000h base + 7C0h offset = 53FA_87C0h

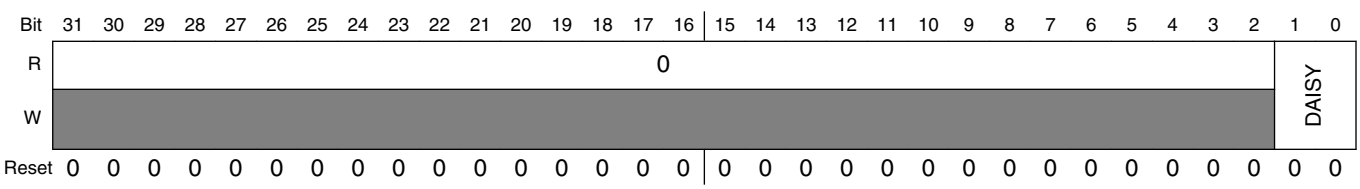


IOMUXC_ECSPI2_IPP_IND_MOSI_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi2, In Pin: ipp_ind_mosi 00 Selecting Pad: DISP0_DAT16 for Mode: ALT2. 01 Selecting Pad: CSI0_DAT9 for Mode: ALT3. 10 Selecting Pad: EIM_CS1 for Mode: ALT2.

43.3.494 IOMUXC_ECSPI2_IPP_IND_SS_B_0_SELECT_INPUT (IOMUXC_ECSPI2_IPP_IND_SS_B_0_SELECT_INPUT)

Address: IOMUXC_ECSPI2_IPP_IND_SS_B_0_SELECT_INPUT is 53FA_8000h base + 7C4h offset = 53FA_87C4h

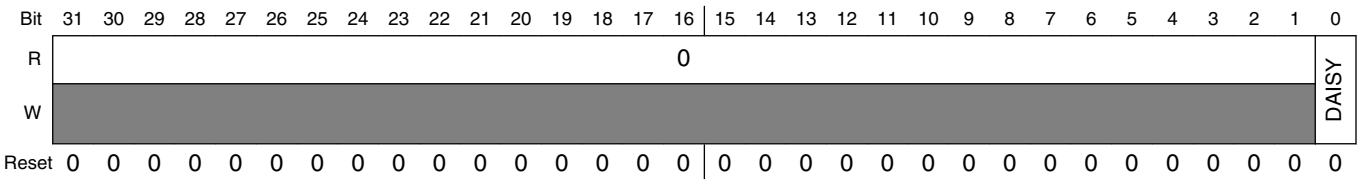


IOMUXC_ECSPI2_IPP_IND_SS_B_0_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi2, In Pin: ipp_ind_ss_b[0] 00 Selecting Pad: DISP0_DAT18 for Mode: ALT2. 01 Selecting Pad: CSI0_DAT11 for Mode: ALT3. 10 Selecting Pad: EIM_RW for Mode: ALT2.

43.3.495 IOMUXC_ECSPI2_IPP_IND_SS_B_1_SELECT_INPUT (IOMUXC_ECSPI2_IPP_IND_SS_B_1_SELECT_INPUT)

Address: IOMUXC_ECSPI2_IPP_IND_SS_B_1_SELECT_INPUT is 53FA_8000h base + 7C8h offset = 53FA_87C8h

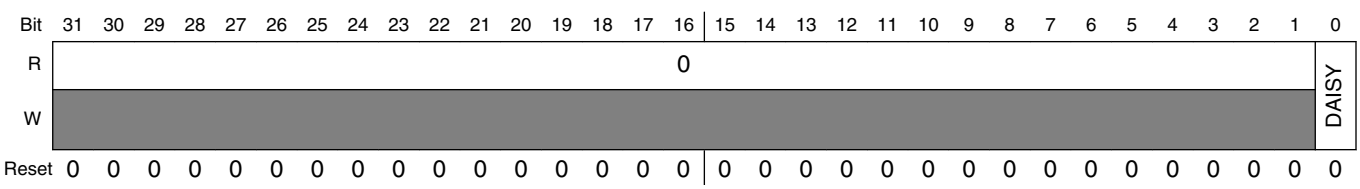


IOMUXC_ECSPI2_IPP_IND_SS_B_1_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ecspi2, In Pin: ipp_ind_ss_b[1] 0 Selecting Pad: DISP0_DAT15 for Mode: ALT3. 1 Selecting Pad: EIM_LBA for Mode: ALT2.

43.3.496 IOMUXC_ESAI1_IPP_IND_FSR_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_FSR_SELECT_INPUT)

Address: IOMUXC_ESAI1_IPP_IND_FSR_SELECT_INPUT is 53FA_8000h base + 7CCh offset = 53FA_87CCh

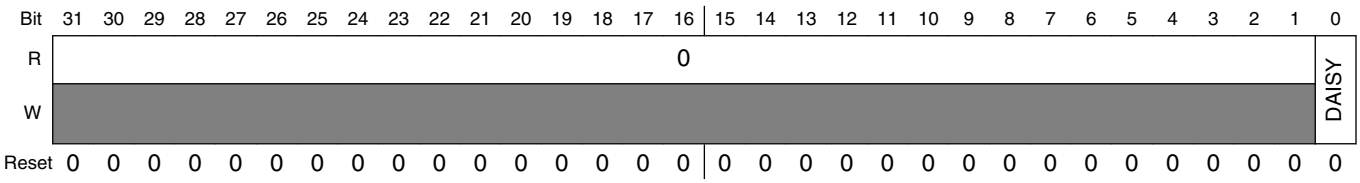


IOMUXC_ESAI1_IPP_IND_FSR_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai1, In Pin: ipp_ind_fsr 0 Selecting Pad: FEC_REF_CLK for Mode: ALT2. 1 Selecting Pad: GPIO_9 for Mode: ALT0.

43.3.497 IOMUXC_ESAI1_IPP_IND_FST_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_FST_SELECT_INPUT)

Address: IOMUXC_ESAI1_IPP_IND_FST_SELECT_INPUT is 53FA_8000h base + 7D0h offset = 53FA_87D0h

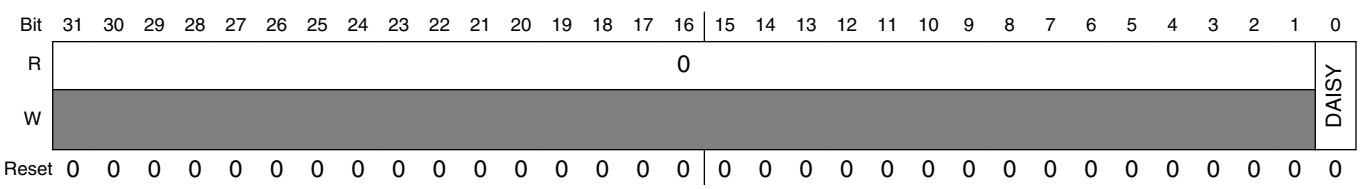


IOMUXC_ESAI1_IPP_IND_FST_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai1, In Pin: ipp_ind_fst 0 Selecting Pad: FEC_RXD1 for Mode: ALT2. 1 Selecting Pad: GPIO_2 for Mode: ALT0.

43.3.498 IOMUXC_ESAI1_IPP_IND_HCKR_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_HCKR_SELECT_INPUT)

Address: IOMUXC_ESAI1_IPP_IND_HCKR_SELECT_INPUT is 53FA_8000h base + 7D4h offset = 53FA_87D4h

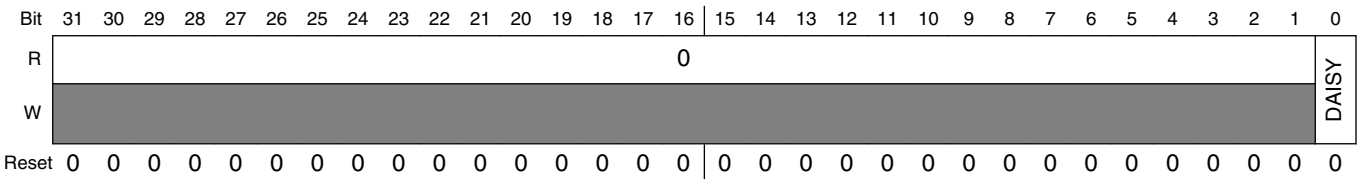


IOMUXC_ESAI1_IPP_IND_HCKR_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai1, In Pin: ipp_ind_hckr 0 Selecting Pad: FEC_RX_ER for Mode: ALT2. 1 Selecting Pad: GPIO_3 for Mode: ALT0.

43.3.499 IOMUXC_ESAI1_IPP_IND_HCKT_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_HCKT_SELECT_INPUT)

Address: IOMUXC_ESAI1_IPP_IND_HCKT_SELECT_INPUT is 53FA_8000h base + 7D8h offset = 53FA_87D8h

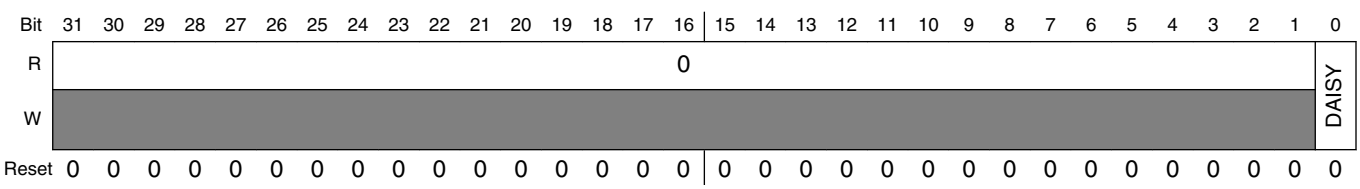


IOMUXC_ESAI1_IPP_IND_HCKT_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai1, In Pin: ipp_ind_hckt 0 Selecting Pad: FEC_RXD0 for Mode: ALT2. 1 Selecting Pad: GPIO_4 for Mode: ALT0.

43.3.500 IOMUXC_ESAI1_IPP_IND_SCKR_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_SCKR_SELECT_INPUT)

Address: IOMUXC_ESAI1_IPP_IND_SCKR_SELECT_INPUT is 53FA_8000h base + 7DCh offset = 53FA_87DCh



IOMUXC_ESAI1_IPP_IND_SCKR_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai1, In Pin: ipp_ind_sckr 0 Selecting Pad: FEC_MDIO for Mode: ALT2. 1 Selecting Pad: GPIO_1 for Mode: ALT0.

43.3.501 IOMUXC_ESAI1_IPP_IND_SCKT_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_SCKT_SELECT_INPUT)

Address: IOMUXC_ESAI1_IPP_IND_SCKT_SELECT_INPUT is 53FA_8000h base + 7E0h offset = 53FA_87E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ESAI1_IPP_IND_SCKT_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai1, In Pin: ipp_ind_sckt 0 Selecting Pad: FEC_CRS_DV for Mode: ALT2. 1 Selecting Pad: GPIO_6 for Mode: ALT0.

43.3.502 IOMUXC_ESAI1_IPP_IND_SDO0_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_SDO0_SELECT_INPUT)

Address: IOMUXC_ESAI1_IPP_IND_SDO0_SELECT_INPUT is 53FA_8000h base + 7E4h offset = 53FA_87E4h

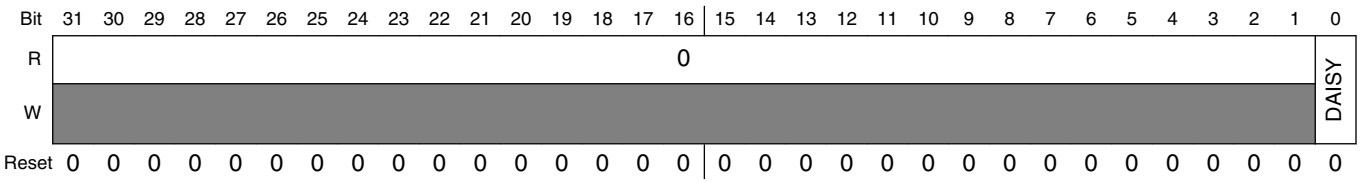
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ESAI1_IPP_IND_SDO0_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai1, In Pin: ipp_ind_sdo0 0 Selecting Pad: NANDF_CS2 for Mode: ALT3. 1 Selecting Pad: GPIO_17 for Mode: ALT0.

43.3.503 IOMUXC_ESAI1_IPP_IND_SDO1_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_SDO1_SELECT_INPUT)

Address: IOMUXC_ESAI1_IPP_IND_SDO1_SELECT_INPUT is 53FA_8000h base + 7E8h offset = 53FA_87E8h

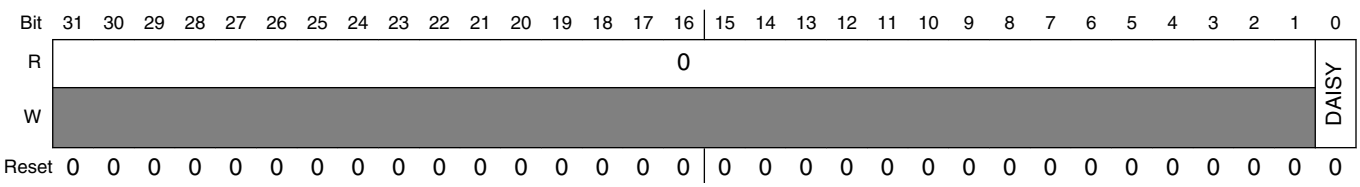


IOMUXC_ESAI1_IPP_IND_SDO1_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai1, In Pin: ipp_ind_sdo1 0 Selecting Pad: NANDF_CS3 for Mode: ALT3. 1 Selecting Pad: GPIO_18 for Mode: ALT0.

43.3.504 IOMUXC_ESAI1_IPP_IND_SDO2_SDI3_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_SDO2_SDI3_SELECT_INPUT)

Address: IOMUXC_ESAI1_IPP_IND_SDO2_SDI3_SELECT_INPUT is 53FA_8000h base + 7ECh offset = 53FA_87ECh

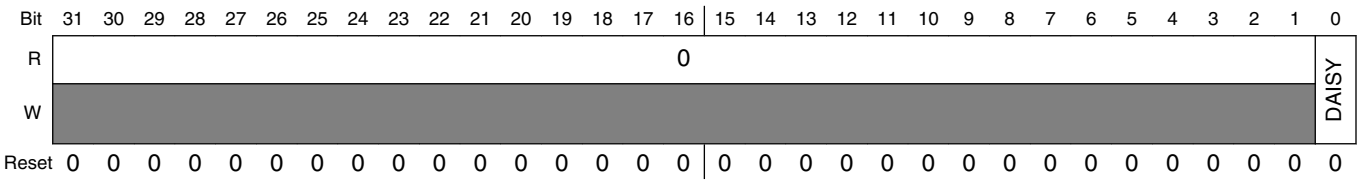


IOMUXC_ESAI1_IPP_IND_SDO2_SDI3_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai1, In Pin: ipp_ind_sdo2_sdi3 0 Selecting Pad: FEC_TXD1 for Mode: ALT2. 1 Selecting Pad: GPIO_5 for Mode: ALT0.

43.3.505 IOMUXC_ESAI1_IPP_IND_SDO3_SDI2_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_SDO3_SDI2_SELECT_INPUT)

Address: IOMUXC_ESAI1_IPP_IND_SDO3_SDI2_SELECT_INPUT is 53FA_8000h base + 7F0h offset = 53FA_87F0h

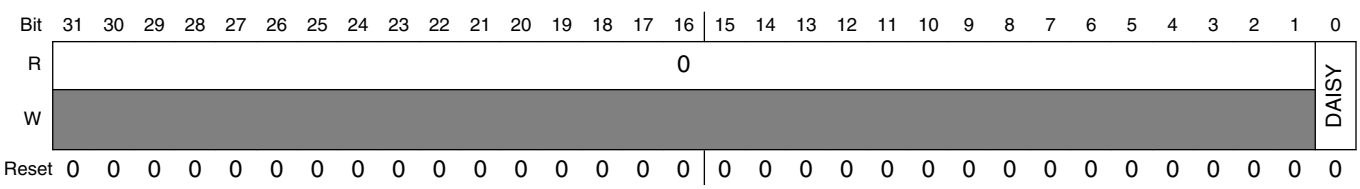


IOMUXC_ESAI1_IPP_IND_SDO3_SDI2_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai1, In Pin: ipp_ind_sdo3_sdi2 0 Selecting Pad: FEC_TX_EN for Mode: ALT2. 1 Selecting Pad: GPIO_16 for Mode: ALT0.

43.3.506 IOMUXC_ESAI1_IPP_IND_SDO4_SDI1_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_SDO4_SDI1_SELECT_INPUT)

Address: IOMUXC_ESAI1_IPP_IND_SDO4_SDI1_SELECT_INPUT is 53FA_8000h base + 7F4h offset = 53FA_87F4h

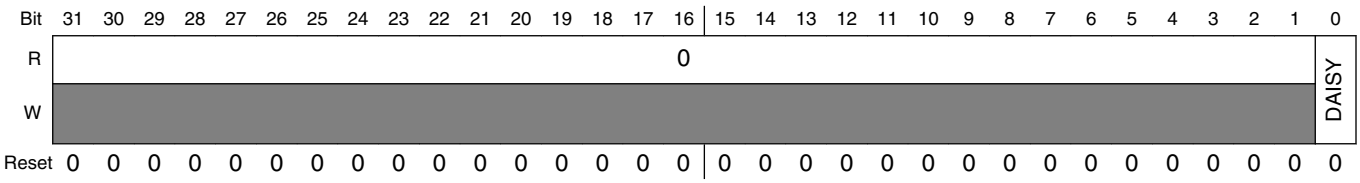


IOMUXC_ESAI1_IPP_IND_SDO4_SDI1_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai1, In Pin: ipp_ind_sdo4_sdi1 0 Selecting Pad: FEC_TXD0 for Mode: ALT2. 1 Selecting Pad: GPIO_7 for Mode: ALT0.

43.3.507 IOMUXC_ESAI1_IPP_IND_SDO5_SDI0_SELECT_INPUT (IOMUXC_ESAI1_IPP_IND_SDO5_SDI0_SELECT_INPUT)

Address: IOMUXC_ESAI1_IPP_IND_SDO5_SDI0_SELECT_INPUT is 53FA_8000h base + 7F8h offset = 53FA_87F8h

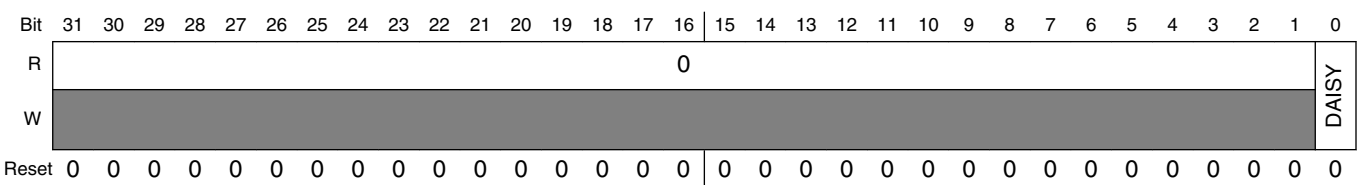


IOMUXC_ESAI1_IPP_IND_SDO5_SDI0_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai1, In Pin: ipp_ind_sdo5_sdi0 0 Selecting Pad: FEC_MDC for Mode: ALT2. 1 Selecting Pad: GPIO_8 for Mode: ALT0.

43.3.508 IOMUXC_ESDHC1_IPP_WP_ON_SELECT_INPUT (IOMUXC_ESDHC1_IPP_WP_ON_SELECT_INPUT)

Address: IOMUXC_ESDHC1_IPP_WP_ON_SELECT_INPUT is 53FA_8000h base + 7FCh offset = 53FA_87FCh



IOMUXC_ESDHC1_IPP_WP_ON_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esdhc1, In Pin: ipp_wp_on 0 Selecting Pad: DI0_PIN4 for Mode: ALT3. 1 Selecting Pad: GPIO_9 for Mode: ALT6.

43.3.509 IOMUXC_FEC_FEC_COL_SELECT_INPUT (IOMUXC_FEC_FEC_COL_SELECT_INPUT)

Address: IOMUXC_FEC_FEC_COL_SELECT_INPUT is 53FA_8000h base + 800h offset = 53FA_8800h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_FEC_FEC_COL_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: fec, In Pin: fec_col 0 Selecting Pad: KEY_ROW1 for Mode: ALT6. 1 Selecting Pad: FEC_MDIO for Mode: ALT3.

43.3.510 IOMUXC_FEC_FEC_MDI_SELECT_INPUT (IOMUXC_FEC_FEC_MDI_SELECT_INPUT)

Address: IOMUXC_FEC_FEC_MDI_SELECT_INPUT is 53FA_8000h base + 804h offset = 53FA_8804h

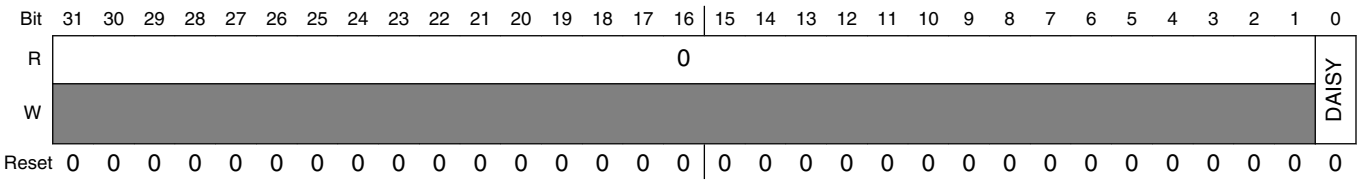
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_FEC_FEC_MDI_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: fec, In Pin: fec_mdi 0 Selecting Pad: KEY_COL2 for Mode: ALT4. 1 Selecting Pad: FEC_MDIO for Mode: ALT0.

43.3.511 IOMUXC_FEC_FEC_RX_CLK_SELECT_INPUT (IOMUXC_FEC_FEC_RX_CLK_SELECT_INPUT)

Address: IOMUXC_FEC_FEC_RX_CLK_SELECT_INPUT is 53FA_8000h base + 808h offset = 53FA_8808h

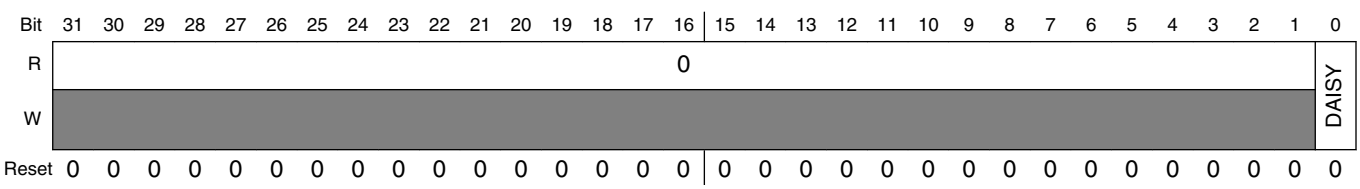


IOMUXC_FEC_FEC_RX_CLK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: fec, In Pin: fec_rx_clk 0 Selecting Pad: KEY_COL1 for Mode: ALT6. 1 Selecting Pad: FEC_RX_ER for Mode: ALT3.

43.3.512 IOMUXC_FIRI_IPP_IND_RXD_SELECT_INPUT (IOMUXC_FIRI_IPP_IND_RXD_SELECT_INPUT)

Address: IOMUXC_FIRI_IPP_IND_RXD_SELECT_INPUT is 53FA_8000h base + 80Ch offset = 53FA_880Ch

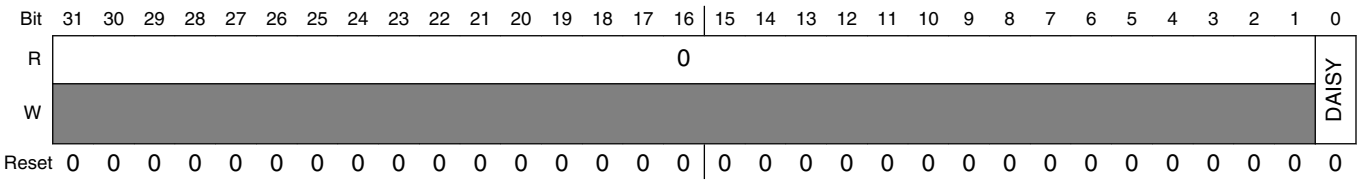


IOMUXC_FIRI_IPP_IND_RXD_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: firi, In Pin: ipp_ind_rxd 0 Selecting Pad: EIM_D26 for Mode: ALT3. 1 Selecting Pad: GPIO_7 for Mode: ALT5.

43.3.513 IOMUXC_GPC_PMIC_RDY_SELECT_INPUT (IOMUXC_GPC_PMIC_RDY_SELECT_INPUT)

Address: IOMUXC_GPC_PMIC_RDY_SELECT_INPUT is 53FA_8000h base + 810h offset = 53FA_8810h

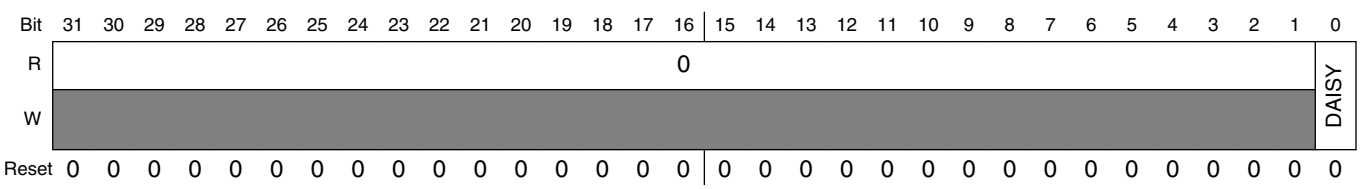


IOMUXC_GPC_PMIC_RDY_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: gpc, In Pin: pmic_rdy 0 Selecting Pad: EIM_EB0 for Mode: ALT5. 1 Selecting Pad: GPIO_17 for Mode: ALT3.

43.3.514 IOMUXC_I2C1_IPP_SCL_IN_SELECT_INPUT (IOMUXC_I2C1_IPP_SCL_IN_SELECT_INPUT)

Address: IOMUXC_I2C1_IPP_SCL_IN_SELECT_INPUT is 53FA_8000h base + 814h offset = 53FA_8814h

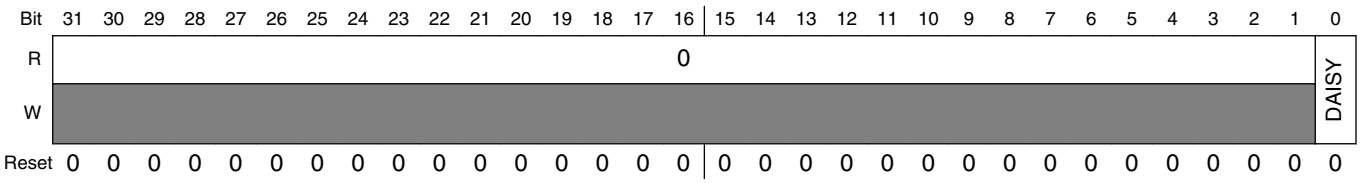


IOMUXC_I2C1_IPP_SCL_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c1, In Pin: ipp_scl_in 0 Selecting Pad: CSI0_DAT9 for Mode: ALT5. 1 Selecting Pad: EIM_D21 for Mode: ALT5.

43.3.515 IOMUXC_I2C1_IPP_SDA_IN_SELECT_INPUT (IOMUXC_I2C1_IPP_SDA_IN_SELECT_INPUT)

Address: IOMUXC_I2C1_IPP_SDA_IN_SELECT_INPUT is 53FA_8000h base + 818h offset = 53FA_8818h

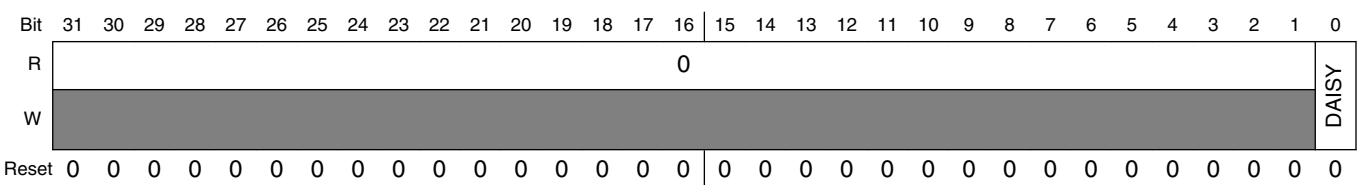


IOMUXC_I2C1_IPP_SDA_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c1, In Pin: ipp_sda_in 0 Selecting Pad: CSI0_DAT8 for Mode: ALT5. 1 Selecting Pad: EIM_D28 for Mode: ALT5.

43.3.516 IOMUXC_I2C2_IPP_SCL_IN_SELECT_INPUT (IOMUXC_I2C2_IPP_SCL_IN_SELECT_INPUT)

Address: IOMUXC_I2C2_IPP_SCL_IN_SELECT_INPUT is 53FA_8000h base + 81Ch offset = 53FA_881Ch

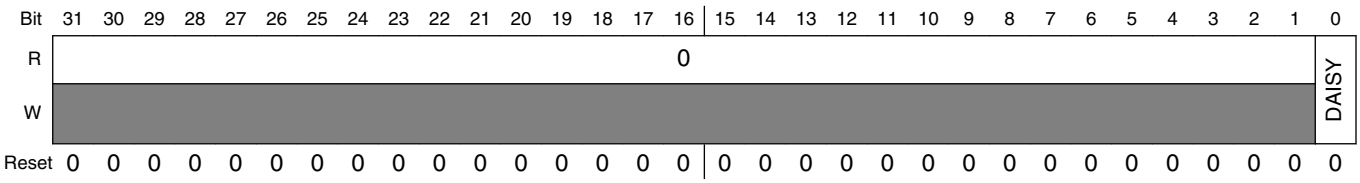


IOMUXC_I2C2_IPP_SCL_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c2, In Pin: ipp_scl_in 0 Selecting Pad: KEY_COL3 for Mode: ALT4. 1 Selecting Pad: EIM_EB2 for Mode: ALT5.

43.3.517 IOMUXC_I2C2_IPP_SDA_IN_SELECT_INPUT (IOMUXC_I2C2_IPP_SDA_IN_SELECT_INPUT)

Address: IOMUXC_I2C2_IPP_SDA_IN_SELECT_INPUT is 53FA_8000h base + 820h offset = 53FA_8820h

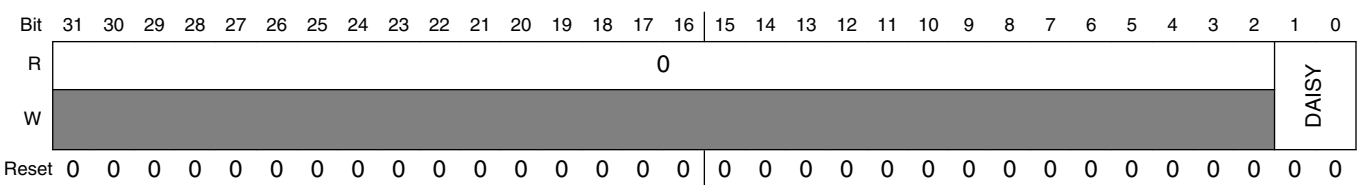


IOMUXC_I2C2_IPP_SDA_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c2, In Pin: ipp_sda_in 0 Selecting Pad: KEY_ROW3 for Mode: ALT4. 1 Selecting Pad: EIM_D16 for Mode: ALT5.

43.3.518 IOMUXC_I2C3_IPP_SCL_IN_SELECT_INPUT (IOMUXC_I2C3_IPP_SCL_IN_SELECT_INPUT)

Address: IOMUXC_I2C3_IPP_SCL_IN_SELECT_INPUT is 53FA_8000h base + 824h offset = 53FA_8824h

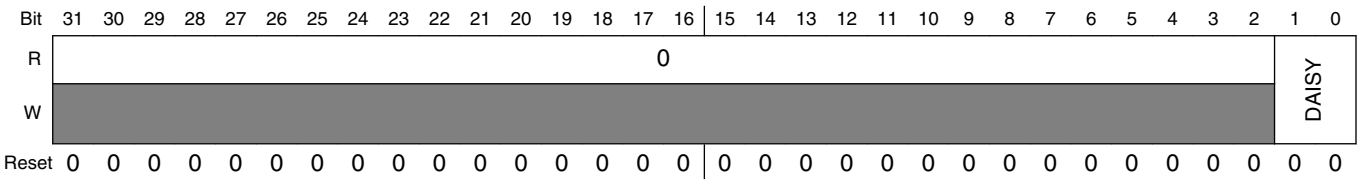


IOMUXC_I2C3_IPP_SCL_IN_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c3, In Pin: ipp_scl_in 00 Selecting Pad: EIM_D17 for Mode: ALT5. 01 Selecting Pad: GPIO_3 for Mode: ALT2. 10 Selecting Pad: GPIO_5 for Mode: ALT6.

43.3.519 IOMUXC_I2C3_IPP_SDA_IN_SELECT_INPUT (IOMUXC_I2C3_IPP_SDA_IN_SELECT_INPUT)

Address: IOMUXC_I2C3_IPP_SDA_IN_SELECT_INPUT is 53FA_8000h base + 828h offset = 53FA_8828h

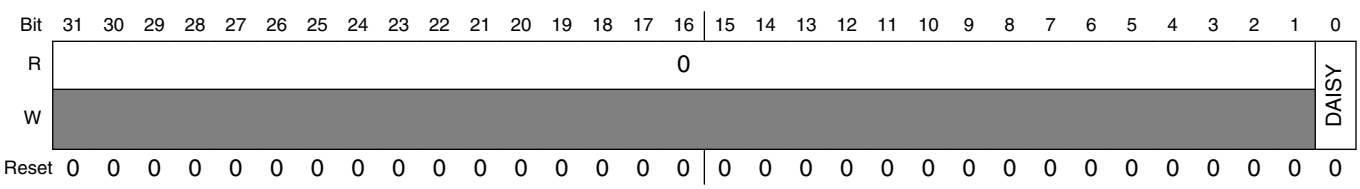


IOMUXC_I2C3_IPP_SDA_IN_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c3, In Pin: ipp_sda_in 00 Selecting Pad: EIM_D18 for Mode: ALT5. 01 Selecting Pad: GPIO_6 for Mode: ALT2. 10 Selecting Pad: GPIO_16 for Mode: ALT6.

43.3.520 IOMUXC_IPU_IPP_DI_0_IND_DISPB_SD_D_SELECT_INPUT (IOMUXC_IPU_IPP_DI_0_IND_DISPB_SD_D_SELECT_INPUT)

Address: IOMUXC_IPU_IPP_DI_0_IND_DISPB_SD_D_SELECT_INPUT is 53FA_8000h base + 82Ch offset = 53FA_882Ch

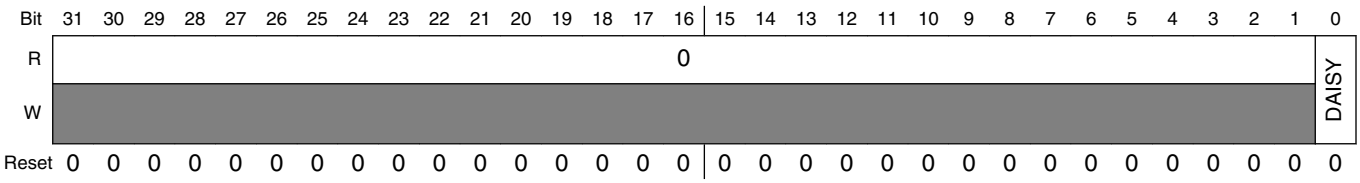


IOMUXC_IPU_IPP_DI_0_IND_DISPB_SD_D_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ipu, In Pin: ipp_di_0_ind_dispb_sd_d 0 Selecting Pad: EIM_D22 for Mode: ALT3. 1 Selecting Pad: EIM_D28 for Mode: ALT3.

43.3.521 IOMUXC_IPU_IPP_DI_1_IND_DISPB_SD_D_SELECT_INPUT (IOMUXC_IPU_IPP_DI_1_IND_DISPB_SD_D_SELECT_INPUT)

Address: IOMUXC_IPU_IPP_DI_1_IND_DISPB_SD_D_SELECT_INPUT is 53FA_8000h base + 830h offset = 53FA_8830h

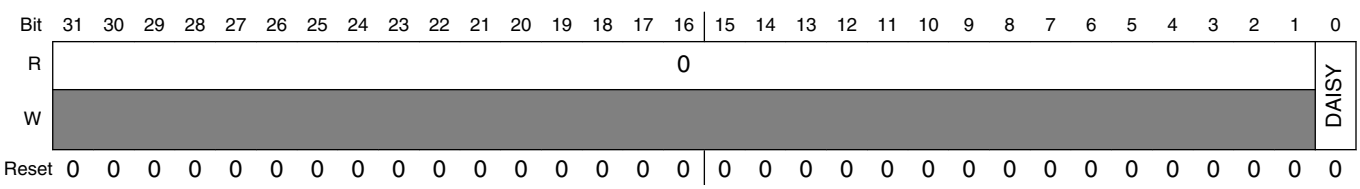


IOMUXC_IPU_IPP_DI_1_IND_DISPB_SD_D_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ipu, In Pin: ipp_di_1_ind_dispb_sd_d 0 Selecting Pad: EIM_D17 for Mode: ALT3. 1 Selecting Pad: EIM_D18 for Mode: ALT3.

43.3.522 IOMUXC_IPU_IPP_IND_SENS1_DATA_EN_SELECT_INPUT (IOMUXC_IPU_IPP_IND_SENS1_DATA_EN_SELECT_INPUT)

Address: IOMUXC_IPU_IPP_IND_SENS1_DATA_EN_SELECT_INPUT is 53FA_8000h base + 834h offset = 53FA_8834h



IOMUXC_IPU_IPP_IND_SENS1_DATA_EN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ipu, In Pin: ipp_ind_sens1_data_en 0 Selecting Pad: EIM_D23 for Mode: ALT6. 1 Selecting Pad: EIM_DA10 for Mode: ALT4.

43.3.523 IOMUXC_IPU_IPP_IND_SENS1_HSYNC_SELECT_INPUT (IOMUXC_IPU_IPP_IND_SENS1_HSYNC_SELECT_INPUT)

Address: IOMUXC_IPU_IPP_IND_SENS1_HSYNC_SELECT_INPUT is 53FA_8000h base + 838h offset = 53FA_8838h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_IPU_IPP_IND_SENS1_HSYNC_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ipu, In Pin: ipp_ind_sens1_hsync 0 Selecting Pad: EIM_EB3 for Mode: ALT6. 1 Selecting Pad: EIM_DA11 for Mode: ALT4.

43.3.524 IOMUXC_IPU_IPP_IND_SENS1_VSYNC_SELECT_INPUT (IOMUXC_IPU_IPP_IND_SENS1_VSYNC_SELECT_INPUT)

Address: IOMUXC_IPU_IPP_IND_SENS1_VSYNC_SELECT_INPUT is 53FA_8000h base + 83Ch offset = 53FA_883Ch

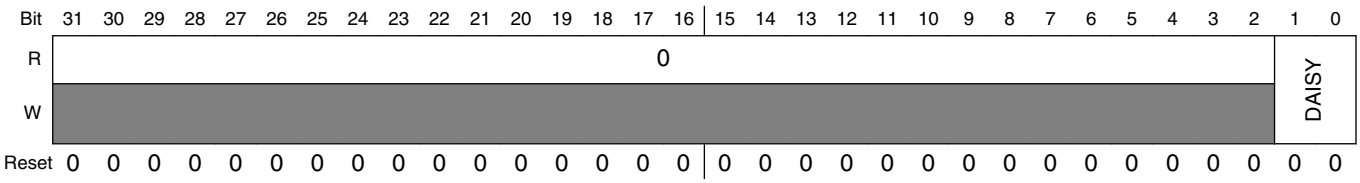
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_IPU_IPP_IND_SENS1_VSYNC_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ipu, In Pin: ipp_ind_sens1_vsync 0 Selecting Pad: EIM_D29 for Mode: ALT6. 1 Selecting Pad: EIM_DA12 for Mode: ALT4.

43.3.525 IOMUXC_KPP_IPP_IND_COL_5_SELECT_INPUT (IOMUXC_KPP_IPP_IND_COL_5_SELECT_INPUT)

Address: IOMUXC_KPP_IPP_IND_COL_5_SELECT_INPUT is 53FA_8000h base + 840h offset = 53FA_8840h

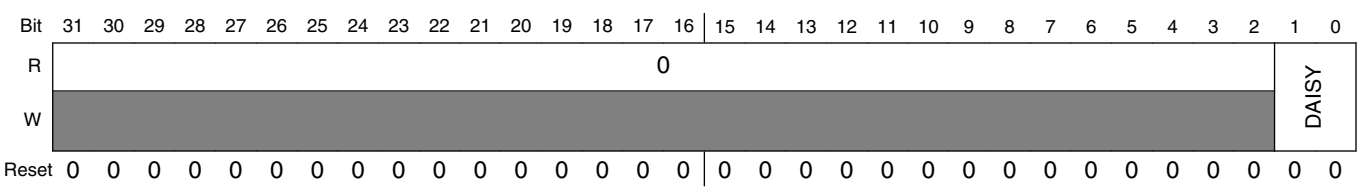


IOMUXC_KPP_IPP_IND_COL_5_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: kpp, In Pin: ipp_ind_col[5] 00 Selecting Pad: GPIO_19 for Mode: ALT0. 01 Selecting Pad: CSI0_DAT4 for Mode: ALT2. 10 Selecting Pad: SD2_CLK for Mode: ALT2. 11 Selecting Pad: GPIO_0 for Mode: ALT2.

43.3.526 IOMUXC_KPP_IPP_IND_COL_6_SELECT_INPUT (IOMUXC_KPP_IPP_IND_COL_6_SELECT_INPUT)

Address: IOMUXC_KPP_IPP_IND_COL_6_SELECT_INPUT is 53FA_8000h base + 844h offset = 53FA_8844h

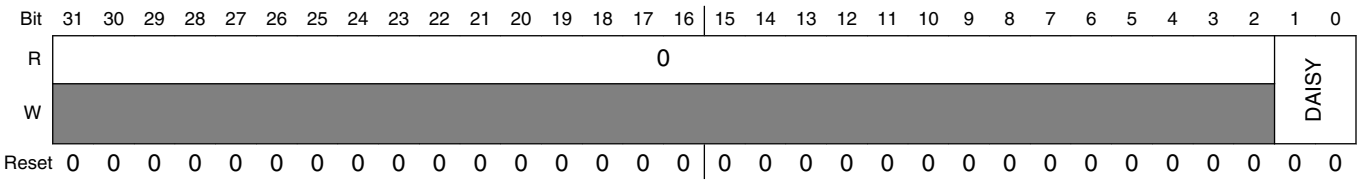


IOMUXC_KPP_IPP_IND_COL_6_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: kpp, In Pin: ipp_ind_col[6] 00 Selecting Pad: CSI0_DAT6 for Mode: ALT2. 01 Selecting Pad: SD2_DATA3 for Mode: ALT2. 10 Selecting Pad: GPIO_9 for Mode: ALT2.

43.3.527 IOMUXC_KPP_IPP_IND_COL_7_SELECT_INPUT (IOMUXC_KPP_IPP_IND_COL_7_SELECT_INPUT)

Address: IOMUXC_KPP_IPP_IND_COL_7_SELECT_INPUT is 53FA_8000h base + 848h offset = 53FA_8848h

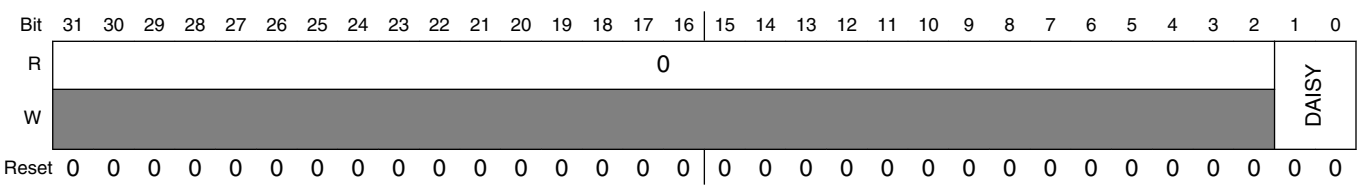


IOMUXC_KPP_IPP_IND_COL_7_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: kpp, In Pin: ipp_ind_col[7] 00 Selecting Pad: CSI0_DAT8 for Mode: ALT2. 01 Selecting Pad: SD2_DATA1 for Mode: ALT2. 10 Selecting Pad: GPIO_4 for Mode: ALT2.

43.3.528 IOMUXC_KPP_IPP_IND_ROW_5_SELECT_INPUT (IOMUXC_KPP_IPP_IND_ROW_5_SELECT_INPUT)

Address: IOMUXC_KPP_IPP_IND_ROW_5_SELECT_INPUT is 53FA_8000h base + 84Ch offset = 53FA_884Ch

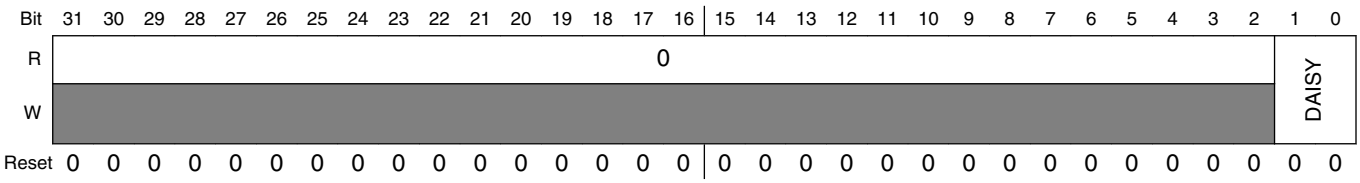


IOMUXC_KPP_IPP_IND_ROW_5_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: kpp, In Pin: ipp_ind_row[5] 00 Selecting Pad: CSI0_DAT5 for Mode: ALT2. 01 Selecting Pad: SD2_CMD for Mode: ALT2. 10 Selecting Pad: GPIO_1 for Mode: ALT2.

43.3.529 IOMUXC_KPP_IPP_IND_ROW_6_SELECT_INPUT (IOMUXC_KPP_IPP_IND_ROW_6_SELECT_INPUT)

Address: IOMUXC_KPP_IPP_IND_ROW_6_SELECT_INPUT is 53FA_8000h base + 850h offset = 53FA_8850h

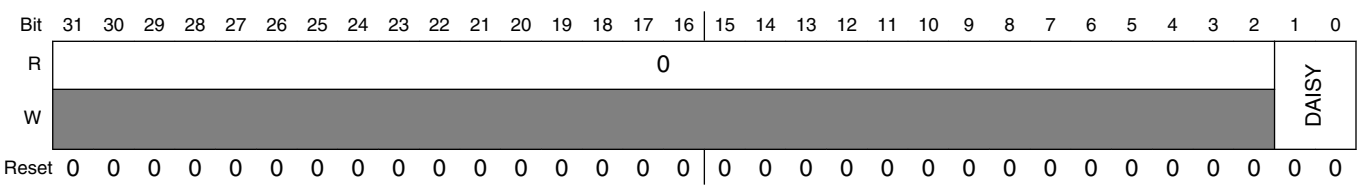


IOMUXC_KPP_IPP_IND_ROW_6_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: kpp, In Pin: ipp_ind_row[6] 00 Selecting Pad: CSI0_DAT7 for Mode: ALT2. 01 Selecting Pad: SD2_DATA2 for Mode: ALT2. 10 Selecting Pad: GPIO_2 for Mode: ALT2.

43.3.530 IOMUXC_KPP_IPP_IND_ROW_7_SELECT_INPUT (IOMUXC_KPP_IPP_IND_ROW_7_SELECT_INPUT)

Address: IOMUXC_KPP_IPP_IND_ROW_7_SELECT_INPUT is 53FA_8000h base + 854h offset = 53FA_8854h

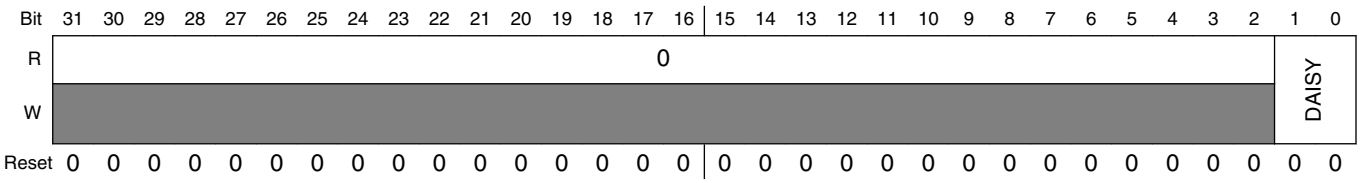


IOMUXC_KPP_IPP_IND_ROW_7_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: kpp, In Pin: ipp_ind_row[7] 00 Selecting Pad: CSI0_DAT9 for Mode: ALT2. 01 Selecting Pad: SD2_DATA0 for Mode: ALT2. 10 Selecting Pad: GPIO_5 for Mode: ALT2.

43.3.531 IOMUXC_MLB_MLBCLK_IN_SELECT_INPUT (IOMUXC_MLB_MLBCLK_IN_SELECT_INPUT)

Address: IOMUXC_MLB_MLBCLK_IN_SELECT_INPUT is 53FA_8000h base + 858h offset = 53FA_8858h

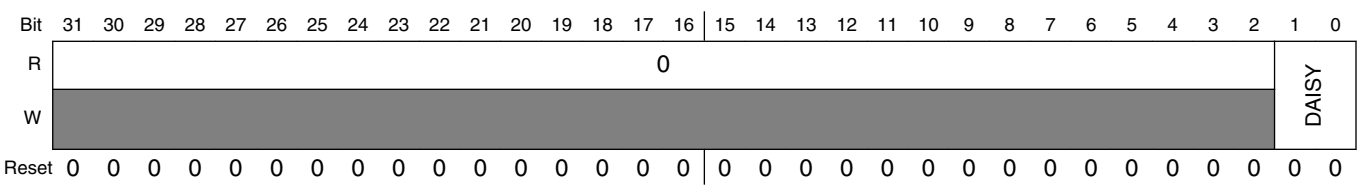


IOMUXC_MLB_MLBCLK_IN_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: mlb, In Pin: mlbclk_in 00 Selecting Pad: NANDF_CS1 for Mode: ALT6. 01 Selecting Pad: FEC_TXD1 for Mode: ALT3. 10 Selecting Pad: GPIO_3 for Mode: ALT7.

43.3.532 IOMUXC_MLB_MLBDAT_IN_SELECT_INPUT (IOMUXC_MLB_MLBDAT_IN_SELECT_INPUT)

Address: IOMUXC_MLB_MLBDAT_IN_SELECT_INPUT is 53FA_8000h base + 85Ch offset = 53FA_885Ch

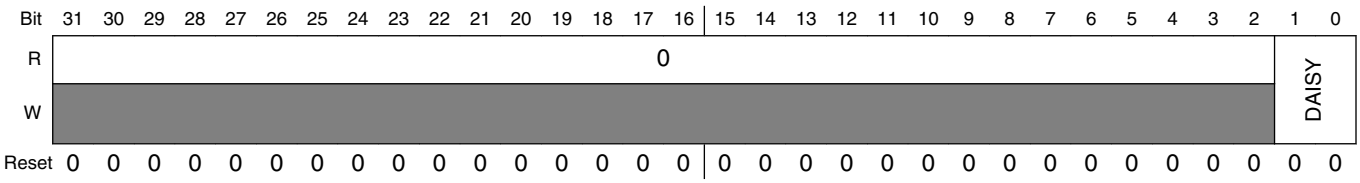


IOMUXC_MLB_MLBDAT_IN_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: mlb, In Pin: mlbdat_in 00 Selecting Pad: NANDF_CS3 for Mode: ALT6. 01 Selecting Pad: FEC_MDC for Mode: ALT3. 10 Selecting Pad: GPIO_2 for Mode: ALT7.

43.3.533 IOMUXC_MLB_MLBSIG_IN_SELECT_INPUT (IOMUXC_MLB_MLBSIG_IN_SELECT_INPUT)

Address: IOMUXC_MLB_MLBSIG_IN_SELECT_INPUT is 53FA_8000h base + 860h offset = 53FA_8860h

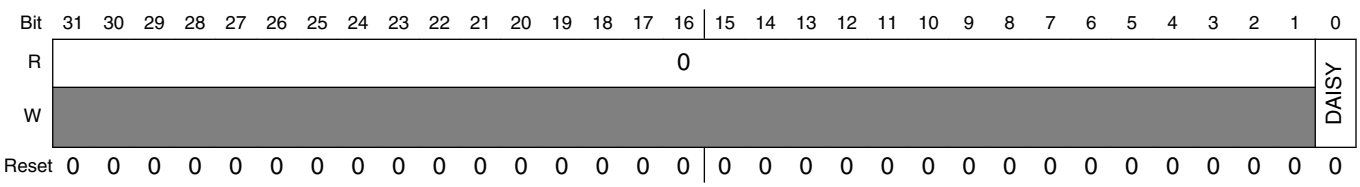


IOMUXC_MLB_MLBSIG_IN_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: mlb, In Pin: mlbsig_in 00 Selecting Pad: NANDF_CS2 for Mode: ALT6. 01 Selecting Pad: FEC_RXD1 for Mode: ALT3. 10 Selecting Pad: GPIO_6 for Mode: ALT7.

43.3.534 IOMUXC_OWIRE_BATTERY_LINE_IN_SELECT_INPUT (IOMUXC_OWIRE_BATTERY_LINE_IN_SELECT_INPUT)

Address: IOMUXC_OWIRE_BATTERY_LINE_IN_SELECT_INPUT is 53FA_8000h base + 864h offset = 53FA_8864h

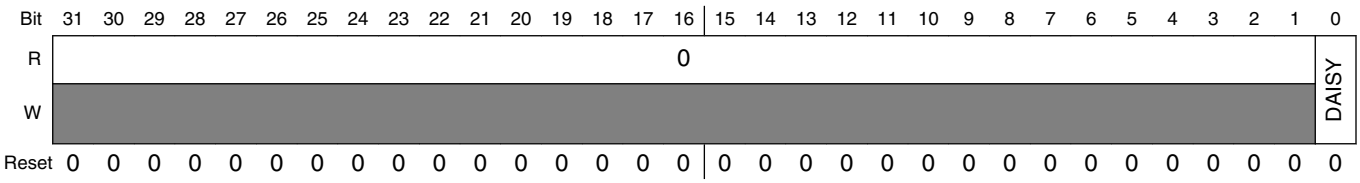


IOMUXC_OWIRE_BATTERY_LINE_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: owire, In Pin: battery_line_in 0 Selecting Pad: PATA_DA_0 for Mode: ALT4. 1 Selecting Pad: GPIO_18 for Mode: ALT3.

43.3.535 IOMUXC_SDMA_EVENTS_14_SELECT_INPUT (IOMUXC_SDMA_EVENTS_14_SELECT_INPUT)

Address: IOMUXC_SDMA_EVENTS_14_SELECT_INPUT is 53FA_8000h base + 868h offset = 53FA_8868h

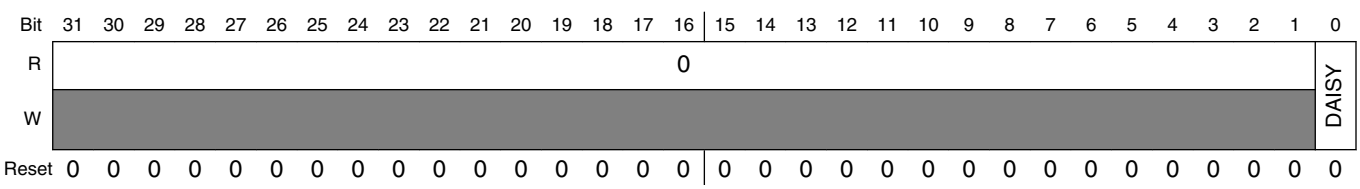


IOMUXC_SDMA_EVENTS_14_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sdma, In Pin: events[14] 0 Selecting Pad: DISP0_DAT16 for Mode: ALT4. 1 Selecting Pad: GPIO_17 for Mode: ALT2.

43.3.536 IOMUXC_SDMA_EVENTS_15_SELECT_INPUT (IOMUXC_SDMA_EVENTS_15_SELECT_INPUT)

Address: IOMUXC_SDMA_EVENTS_15_SELECT_INPUT is 53FA_8000h base + 86Ch offset = 53FA_886Ch

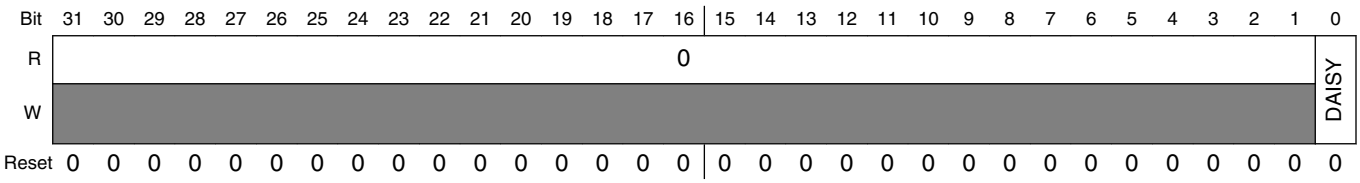


IOMUXC_SDMA_EVENTS_15_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sdma, In Pin: events[15] 0 Selecting Pad: DISP0_DAT17 for Mode: ALT4. 1 Selecting Pad: GPIO_18 for Mode: ALT2.

43.3.537 IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT (IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT)

Address: IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT is 53FA_8000h base + 870h offset = 53FA_8870h

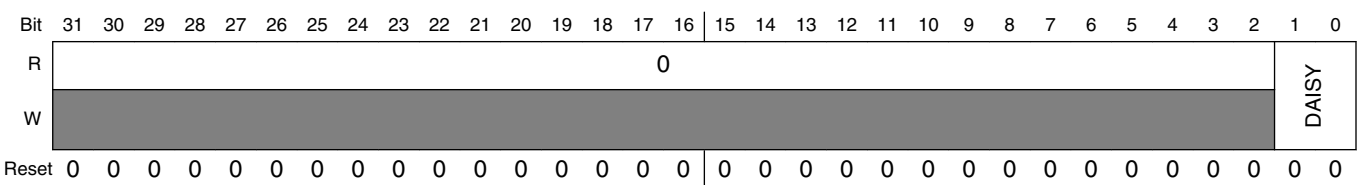


IOMUXC_SPDIF_SPDIF_IN1_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: spdif, In Pin: spdif_in1 0 Selecting Pad: KEY_COL3 for Mode: ALT3. 1 Selecting Pad: GPIO_16 for Mode: ALT5.

43.3.538 IOMUXC_UART1_IPP_UART_RTS_B_SELECT_INPUT (IOMUXC_UART1_IPP_UART_RTS_B_SELECT_INPUT)

Address: IOMUXC_UART1_IPP_UART_RTS_B_SELECT_INPUT is 53FA_8000h base + 874h offset = 53FA_8874h

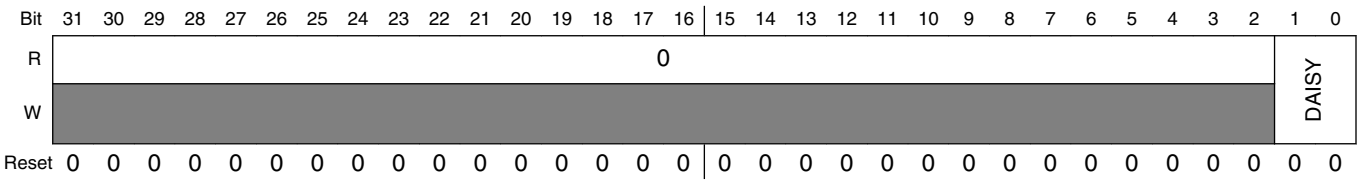


IOMUXC_UART1_IPP_UART_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart1, In Pin: ipp_uart_rts_b 00 Selecting Pad: EIM_D19 for Mode: ALT6. 01 Selecting Pad: EIM_D20 for Mode: ALT6. 10 Selecting Pad: PATA_RESET_B for Mode: ALT3. 11 Selecting Pad: PATA_IORDY for Mode: ALT3.

43.3.539 IOMUXC_UART1_IPP_UART_RXD_MUX_SELECT_INPUT (IOMUXC_UART1_IPP_UART_RXD_MUX_SELECT_INPUT)

Address: IOMUXC_UART1_IPP_UART_RXD_MUX_SELECT_INPUT is 53FA_8000h base + 878h offset = 53FA_8878h

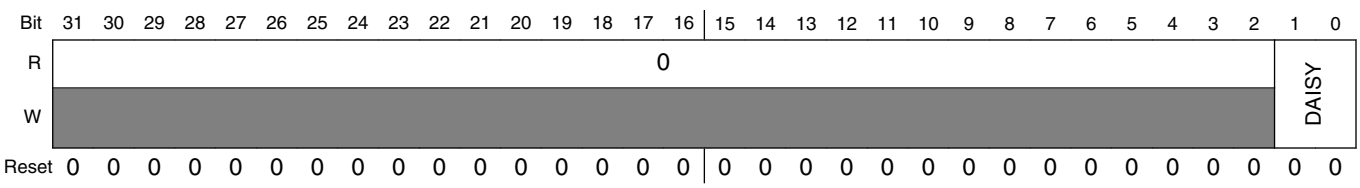


IOMUXC_UART1_IPP_UART_RXD_MUX_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart1, In Pin: ipp_uart_rxd_mux 00 Selecting Pad: CSI0_DAT10 for Mode: ALT2. 01 Selecting Pad: CSI0_DAT11 for Mode: ALT2. 10 Selecting Pad: PATA_DIOW for Mode: ALT3. 11 Selecting Pad: PATA_DMACK for Mode: ALT3.

43.3.540 IOMUXC_UART2_IPP_UART_RTS_B_SELECT_INPUT (IOMUXC_UART2_IPP_UART_RTS_B_SELECT_INPUT)

Address: IOMUXC_UART2_IPP_UART_RTS_B_SELECT_INPUT is 53FA_8000h base + 87Ch offset = 53FA_887Ch



IOMUXC_UART2_IPP_UART_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart2, In Pin: ipp_uart_rts_b 00 Selecting Pad: EIM_D28 for Mode: ALT2. 01 Selecting Pad: EIM_D29 for Mode: ALT2.

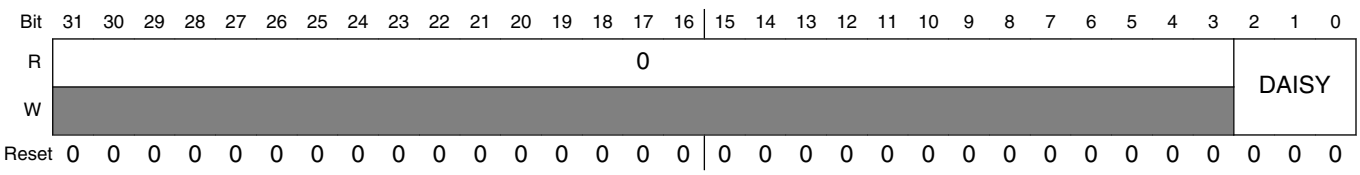
Table continues on the next page...

IOMUXC_UART2_IPP_UART_RTS_B_SELECT_INPUT field descriptions (continued)

Field	Description
	10 Selecting Pad: PATA_INTRQ for Mode: ALT3. 11 Selecting Pad: PATA_DIOR for Mode: ALT3.

43.3.541 IOMUXC_UART2_IPP_UART_RXD_MUX_SELECT_INPUT (IOMUXC_UART2_IPP_UART_RXD_MUX_SELECT_INPUT)

Address: IOMUXC_UART2_IPP_UART_RXD_MUX_SELECT_INPUT is 53FA_8000h base + 880h offset = 53FA_8880h

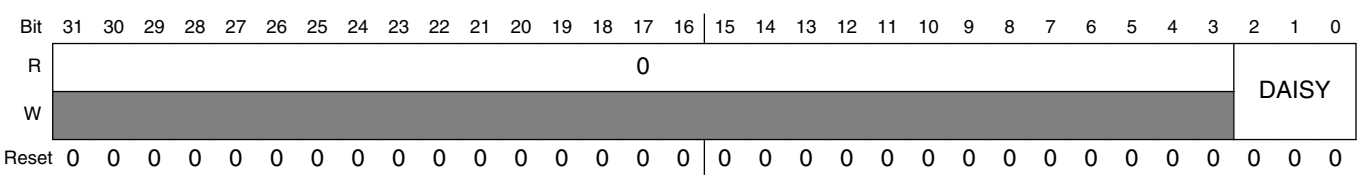


IOMUXC_UART2_IPP_UART_RXD_MUX_SELECT_INPUT field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart2, In Pin: ipp_uart_rxd_mux 000 Selecting Pad: EIM_D26 for Mode: ALT2. 001 Selecting Pad: EIM_D27 for Mode: ALT2. 010 Selecting Pad: PATA_DMARQ for Mode: ALT3. 011 Selecting Pad: PATA_BUFFER_EN for Mode: ALT3. 100 Selecting Pad: GPIO_7 for Mode: ALT4. 101 Selecting Pad: GPIO_8 for Mode: ALT4.

43.3.542 IOMUXC_UART3_IPP_UART_RTS_B_SELECT_INPUT (IOMUXC_UART3_IPP_UART_RTS_B_SELECT_INPUT)

Address: IOMUXC_UART3_IPP_UART_RTS_B_SELECT_INPUT is 53FA_8000h base + 884h offset = 53FA_8884h



IOMUXC_UART3_IPP_UART_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart3, In Pin: ipp_uart_rts_b 000 Selecting Pad: EIM_D23 for Mode: ALT2. 001 Selecting Pad: EIM_EB3 for Mode: ALT2. 010 Selecting Pad: EIM_D30 for Mode: ALT2. 011 Selecting Pad: EIM_D31 for Mode: ALT2. 100 Selecting Pad: PATA_DA_1 for Mode: ALT4. 101 Selecting Pad: PATA_DA_2 for Mode: ALT4.

43.3.543 IOMUXC_UART3_IPP_UART_RXD_MUX_SELECT_INPUT (IOMUXC_UART3_IPP_UART_RXD_MUX_SELECT_INPUT)

Address: IOMUXC_UART3_IPP_UART_RXD_MUX_SELECT_INPUT is 53FA_8000h base + 888h offset = 53FA_8888h

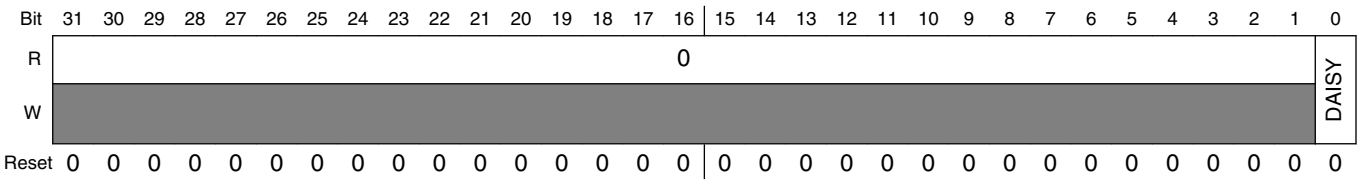
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																														DAISY	
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_UART3_IPP_UART_RXD_MUX_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart3, In Pin: ipp_uart_rxd_mux 00 Selecting Pad: EIM_D24 for Mode: ALT2. 01 Selecting Pad: EIM_D25 for Mode: ALT2. 10 Selecting Pad: PATA_CS_0 for Mode: ALT4. 11 Selecting Pad: PATA_CS_1 for Mode: ALT4.

43.3.544 IOMUXC_UART4_IPP_UART_RTS_B_SELECT_INPUT (IOMUXC_UART4_IPP_UART_RTS_B_SELECT_INPUT)

Address: IOMUXC_UART4_IPP_UART_RTS_B_SELECT_INPUT is 53FA_8000h base + 88Ch offset = 53FA_888Ch

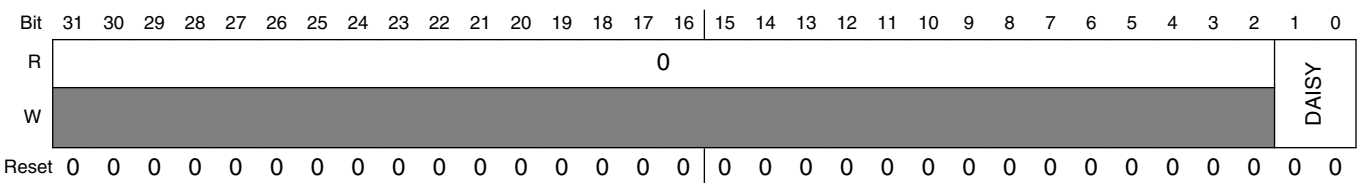


IOMUXC_UART4_IPP_UART_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart4, In Pin: ipp_uart_rts_b 0 Selecting Pad: CSI0_DAT16 for Mode: ALT2. 1 Selecting Pad: CSI0_DAT17 for Mode: ALT2.

43.3.545 IOMUXC_UART4_IPP_UART_RXD_MUX_SELECT_INPUT (IOMUXC_UART4_IPP_UART_RXD_MUX_SELECT_INPUT)

Address: IOMUXC_UART4_IPP_UART_RXD_MUX_SELECT_INPUT is 53FA_8000h base + 890h offset = 53FA_8890h

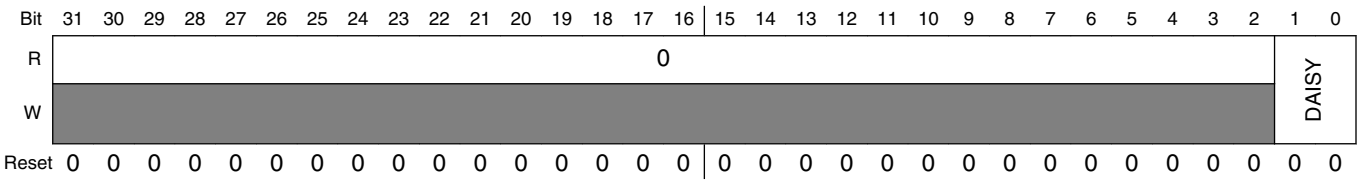


IOMUXC_UART4_IPP_UART_RXD_MUX_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart4, In Pin: ipp_uart_rxd_mux 00 Selecting Pad: KEY_COL0 for Mode: ALT4. 01 Selecting Pad: KEY_ROW0 for Mode: ALT4. 10 Selecting Pad: CSI0_DAT12 for Mode: ALT2. 11 Selecting Pad: CSI0_DAT13 for Mode: ALT2.

43.3.546 IOMUXC_UART5_IPP_UART_RTS_B_SELECT_INPUT (IOMUXC_UART5_IPP_UART_RTS_B_SELECT_INPUT)

Address: IOMUXC_UART5_IPP_UART_RTS_B_SELECT_INPUT is 53FA_8000h base + 894h offset = 53FA_8894h

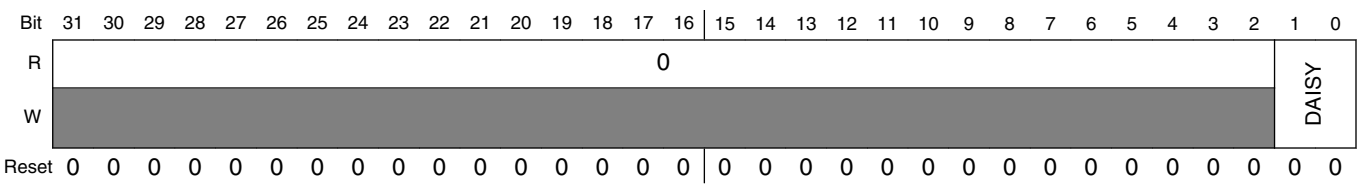


IOMUXC_UART5_IPP_UART_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart5, In Pin: ipp_uart_rts_b 00 Selecting Pad: KEY_COL4 for Mode: ALT4. 01 Selecting Pad: KEY_ROW4 for Mode: ALT4. 10 Selecting Pad: CSI0_DAT18 for Mode: ALT2. 11 Selecting Pad: CSI0_DAT19 for Mode: ALT2.

43.3.547 IOMUXC_UART5_IPP_UART_RXD_MUX_SELECT_INPUT (IOMUXC_UART5_IPP_UART_RXD_MUX_SELECT_INPUT)

Address: IOMUXC_UART5_IPP_UART_RXD_MUX_SELECT_INPUT is 53FA_8000h base + 898h offset = 53FA_8898h



IOMUXC_UART5_IPP_UART_RXD_MUX_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: uart5, In Pin: ipp_uart_rxd_mux 00 Selecting Pad: KEY_COL1 for Mode: ALT4. 01 Selecting Pad: KEY_ROW1 for Mode: ALT4.

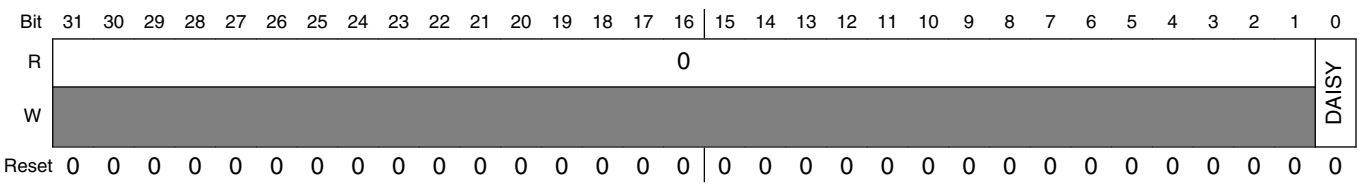
Table continues on the next page...

IOMUXC_UART5_IPP_UART_RXD_MUX_SELECT_INPUT field descriptions (continued)

Field	Description
	10 Selecting Pad: CSI0_DAT14 for Mode: ALT2. 11 Selecting Pad: CSI0_DAT15 for Mode: ALT2.

43.3.548 IOMUXC_USBOH3_IPP_IND_OTG_OC_SELECT_INPUT (IOMUXC_USBOH3_IPP_IND_OTG_OC_SELECT_INPUT)

Address: IOMUXC_USBOH3_IPP_IND_OTG_OC_SELECT_INPUT is 53FA_8000h base + 89Ch offset = 53FA_889Ch

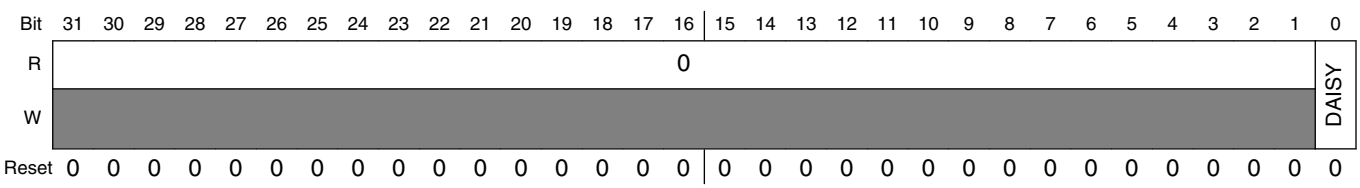


IOMUXC_USBOH3_IPP_IND_OTG_OC_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usboh3, In Pin: ipp_ind_otg_oc 0 Selecting Pad: KEY_COL4 for Mode: ALT5. 1 Selecting Pad: EIM_D21 for Mode: ALT6.

43.3.549 IOMUXC_USBOH3_IPP_IND_UH1_OC_SELECT_INPUT (IOMUXC_USBOH3_IPP_IND_UH1_OC_SELECT_INPUT)

Address: IOMUXC_USBOH3_IPP_IND_UH1_OC_SELECT_INPUT is 53FA_8000h base + 8A0h offset = 53FA_88A0h



IOMUXC_USBOH3_IPP_IND_UH1_OC_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IOMUXC_USBOH3_IPP_IND_UH1_OC_SELECT_INPUT field descriptions (continued)

Field	Description
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usboh3, In Pin: ipp_ind_uh1_oc 0 Selecting Pad: EIM_D30 for Mode: ALT6. 1 Selecting Pad: GPIO_3 for Mode: ALT6.

43.3.550 IOMUXC_USBOH3_IPP_IND_UH2_OC_SELECT_INPUT (IOMUXC_USBOH3_IPP_IND_UH2_OC_SELECT_INPUT)

Address: IOMUXC_USBOH3_IPP_IND_UH2_OC_SELECT_INPUT is 53FA_8000h base + 8A4h offset = 53FA_88A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DAISY
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_USBOH3_IPP_IND_UH2_OC_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: usboh3, In Pin: ipp_ind_uh2_oc 0 Selecting Pad: EIM_D19 for Mode: ALT7. 1 Selecting Pad: EIM_D30 for Mode: ALT7.

43.4 Functional Description

The IOMUXC Detailed Block Diagram is illustrated in Figure 43-552.

The IOMUXC is consist of 2 sub blocks:

- IOMUXC_REGISTERS: includes all the IOMUXC registers (6 main types are illustrated).
- IOMUXC_LOGIC: includes all the IOMUXC combinatorial logic (IP interface controls, address decoder, observability muxes).

The IOMUXC external signals and the IOMUXC sub-blocks connections internal signals are also detailed in this diagram. The following are the naming conventions used along with some explanation:

- <PAD> - any pad logic name.
- <GRP> - any pad group logic name.
- <FUNC> - one of the following pad control functions:
  - SRE (1 bit slew rate control).
  - DSE (2 bits drive strength control).
  - ODE (1 bit open drain control).
  - HYS (1 bit hysteresis control).
  - PULL_KEEP_CTL (4 bits pull up/down and keeper controls)
  - PUS (2 bits pull up/down configuration value)
  - PUE (1 bit pull/keep select)
  - PKE (1 bit enable/disable pull up, pull down or keeper capability)
  - DDR_MODE_SEL (1 bit ddr_mode control)
  - DDR_INPUT (1 bit ddr_input control)
- ips_addr[X:2] - since the IOMUXC registers are all 32 bits width address lines 0,1 are not used. X will be calculated automatically by the tool according to the final register number: X is smallest integer value which satisfy the constraint: 2^(X-1) >= (Total number of IOMUXC registers).

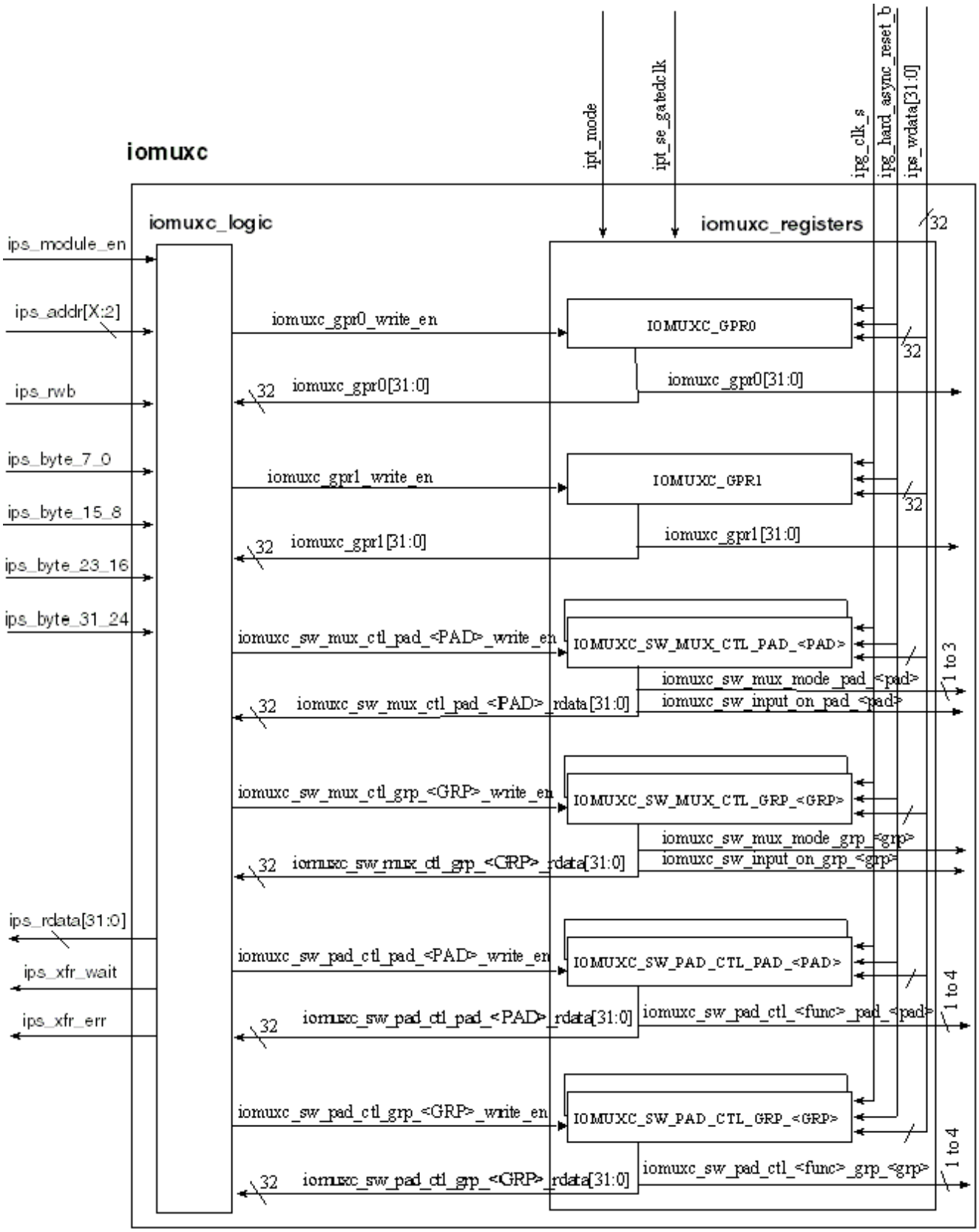


Figure 43-552. IOMUXC Block Diagram

The IOMUXC consists of a certain number of basic iomux_cell units, roughly equal to the number of pads in the SoC. If only one functional mode is required for a specific pad, then there is no need for IO Multiplexing and the signals can be connected directly from the block to the IO ring. An IOMUXC cell is required whenever 2 or more functional modes are needed for a specific pad, or when one functional mode and one test mode are required.

The basic iomux_cell design, which allows 2 levels of HW signal control (in ALT6 and ALT7 modes - ALT7 gets highest priority), is shown in Figure 43-553.

43.4.1 ALT6 and ALT7 Extended Muxing Modes

The ALT7 and ALT6 Extended Muxing Modes allow any signal in the system (Fuse, Pad input, JTAG, SW register, etc) to override any SW configuration and to force the ALT6/ ALT7 Muxing Mode.They also allow an IOMUXC SW register to control a group of pads.

43.4.2 SW Loopback through SION Bit

One option exists to override the regular pad functionality and force the input path to be active (ipp_ibe==1'b1), regardless of the value driven by the corresponding block. This can be done by setting the SION (SW Input On) bit in the IOMUXC_SW_MUX_CTL register (when available) to "1". It can be used for:

- LoopBack - Block X drives the pad and also receive pad value as an input.
- GPIO Capture - Block X drives the pad and the value is captured by GPIO.

NOTE

Software loopback is not affecting EMI pads.

43.4.3 Daisy Chain - Multi Pads Driving same Block Input Pin

In some cases, more than one pad may drive a single block input pin.

Such cases require adding one more level of I/O Multiplexing: all these input signals are muxed and a dedicated SW controlled register controls the mux in order to select the required input path. This means that a pad involved in such "Daisy Chain" situation may require 2 SW configuration commands: one for selecting the mode for this pad and one for defining it as the input path.

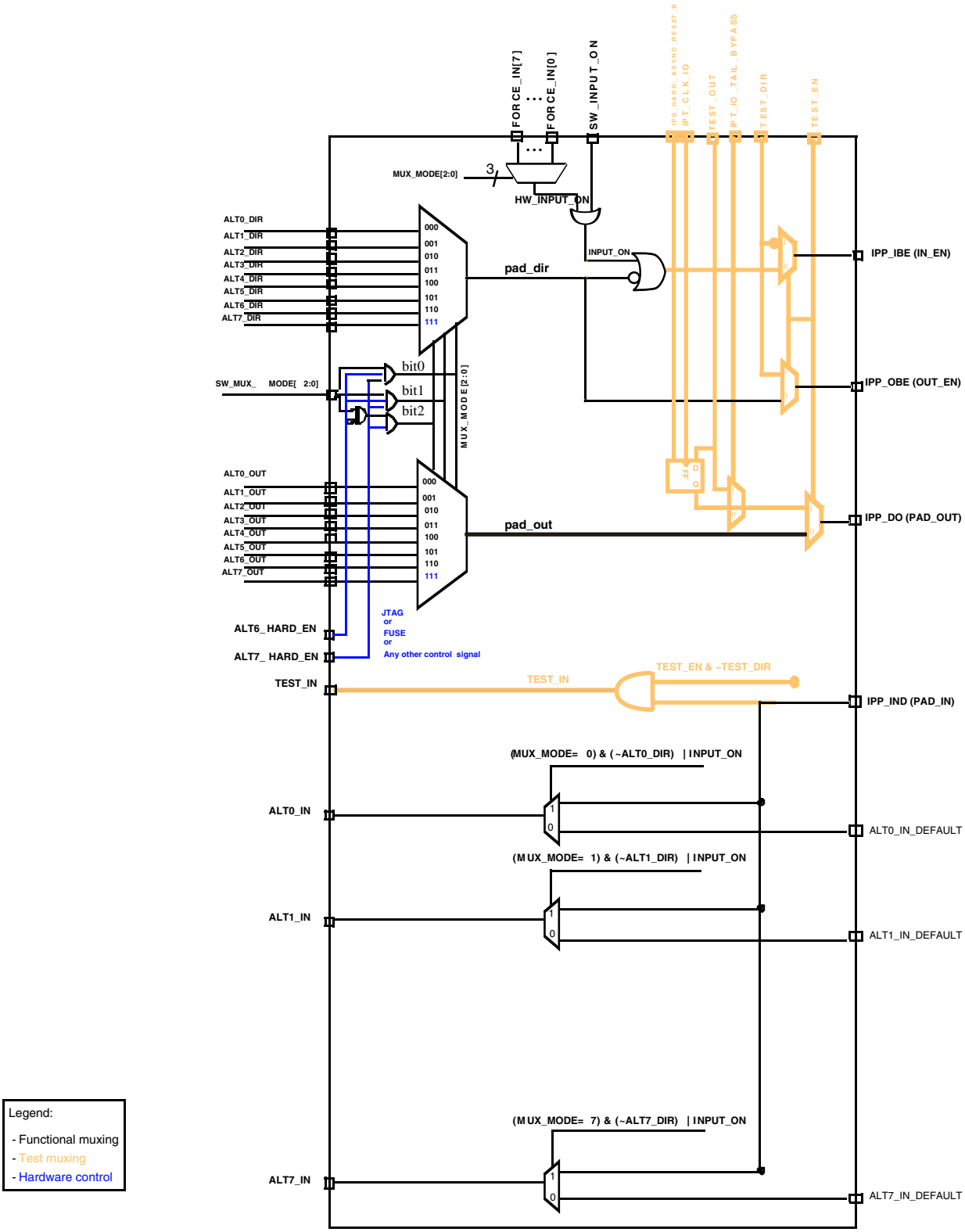


Figure 43-553. IOMUXC Cell Block Diagram

43.4.4 Interrupts

There are no interrupts that are generated by the IOMUX Controller (IOMUXC).

Chapter 44
IEEE 1588 Precision Time Protocol Assist (IPTP)

44.1 Introduction

The objective of the IEEE1588 standard is to specify a protocol to synchronize independent clocks running on separate nodes of a distributed measurement and control system to a high degree of accuracy and precision. The clocks communicate with each other over a communication network. The protocol generates a master-slave relationship among the clocks in the system. Within a given subnet of a network, there will be a single master clock. All clocks ultimately derive their time from a clock known as the grandmaster clock.

The protocol will enable synchronization between heterogeneous systems that include clocks of various inherent precision, resolution, and stability. The protocol will support system wide synchronization accuracy in the sub-microsecond range with minimal network and local clock computing resources.

Measurement and control applications use distributed system technologies such as network communication, local computing, and distributed objects . Many of these applications will be enhanced by having an accurate system wide sense of time achieved by having local clocks in each sensor, actuator, or other system devices. Existing protocols for clock synchronization are not optimal for these applications. For example, Network Time Protocol (NTP) targets large distributed computing systems with millisecond synchronization requirements.

Although IPTP (IEEE1588 Precision Time Protocol Assist) allows software-only implementations, hardware-assisted implementations deliver more precise clock synchronization.

The simplest IPTP implementations include ordinary applications at the top of the network protocol stack and generate time stamps at the application level. Typically, this implementation incurs the largest protocol stack delay fluctuation, thus yielding the least accuracy as the largest amount of error is introduced into the time stamp.

Hardware-assisted methods achieve the greatest accuracy due to the fact that they generate time stamps at the physical layer (as close to the wire as possible). Network-protocol-delay fluctuations, for these implementations, typically range from sub-microseconds to nanoseconds.

In this sub-block, the IPTP implementation is a hardware-assisted method. The hardware assist includes a time stamp unit to recognize PTP frames and transfer relevant time stamps, and a real-time clock with high resolution that is accurate to the sub-nanosecond.

44.2 IPTP Block Diagram

Figure 44-1 shows the Time Stamp Unit (TSU) and the IPTP Real Time Clock (RTC) sub-blocks integrated with the ARM platform.

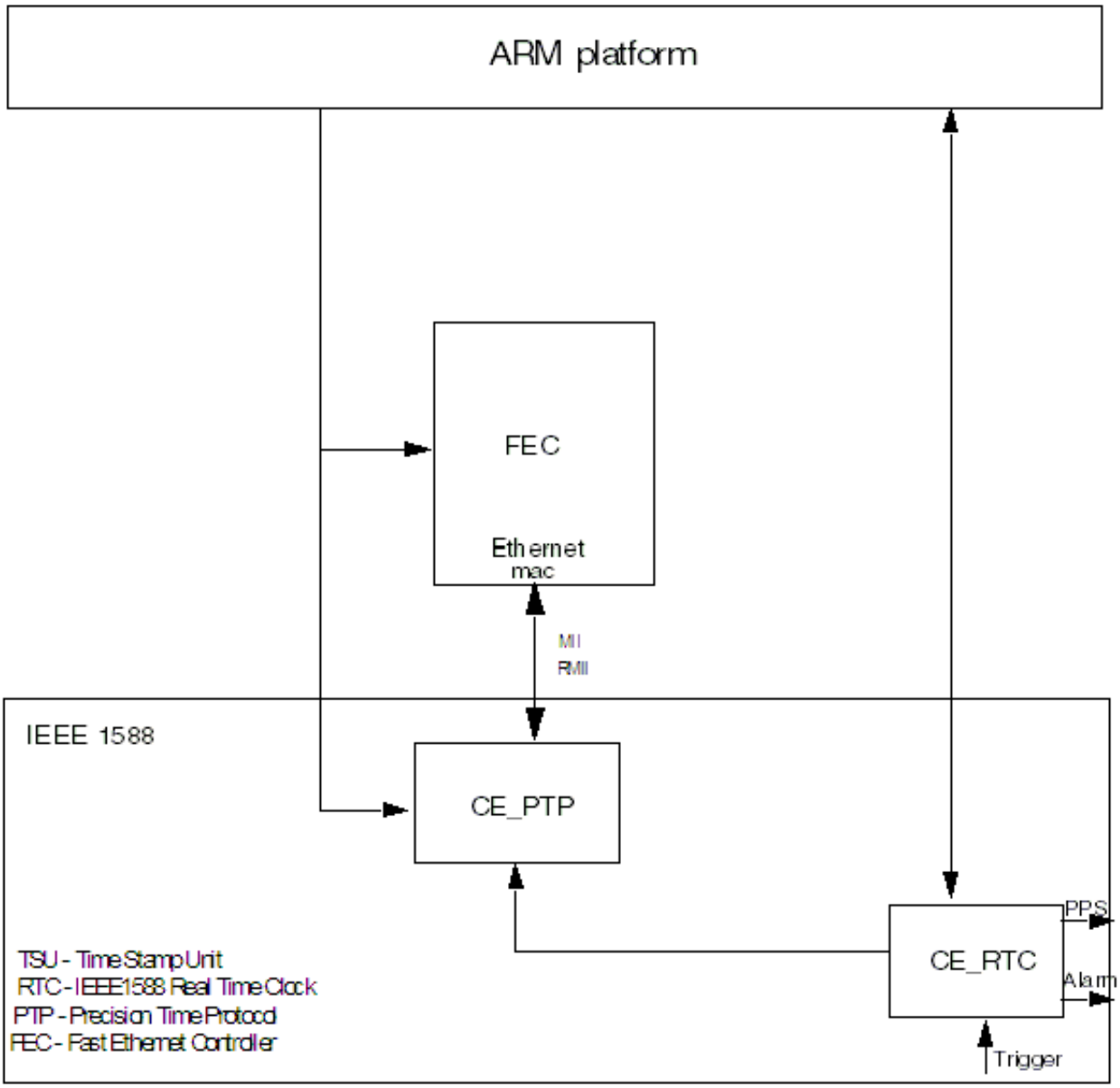


Figure 44-1. IPTP block diagram

44.3 Time Stamp Unit (TSU) Key Features

- Support IPTP V2
- Supports Ethernet systems
- Support IPTP time stamping
- Recognition of PTP frames on receive side
  - Sync type
  - Delay_Req type
  - Delay_Resp type
  - Follow_Up type

    - Management type
    - Pdelay_Req
    - Pdealy_Resp
    - Announce
    - Pdelay_Resp_Follow_Up
    - Signaling
- Support PTP frame with VLAN tag (single VLAN)
- Support the detection of a new Ethernet type (IPTP V2)
- Supports MII/RMII Physical I/F
- Support Ethernet frame monitoring on full and half duplex modes
- Supports slave and master mode
- Support single- or multi- IPTP masters
- Support out-of-band mode of operation
    - Out-of-band (time stamp is read through host I/F)
- Time stamp captured close to the physical line
- Support the ability to timestamp Tx and Rx packets based on signals that are generated by the PHY upon detection of Start-Of-Frame appearing on the line
- Supports configured values for the PTP header fields
- Supports configured values for fields offset
- Supports interrupt notification due to following:
    - Rx PTP frame detection (for all the event messages)
    - Tx PTP frame transmission which was marked by the software as a PTP frame
    - Rx and Tx time stamp overrun error

44.4 IPTP Real Time Clock (RTC) Key Features

- Support single IPTP RTC.
- Support timer frequency compensation.
- Support timer offset update.
- Support configured source of clock.
- Time stamp capture on two general purpose external triggers through new GPIO connection.
    - Maskable interrupts on GPIO time stamp trigger
    - Programmable polarity for both output trigger signals
- Two 64-bit alarm (future time) registers that hold the value of the future time.
    - Maskable interrupts on alarm
    - Programmable polarity for both alarm (future time) output signals
- One 64-bit fixed period interval (FIPER) start register. Used to define the starting time of PPS signals generation
- Three programmable timer output pulse period phase aligned with IPTP timer clock

  - Pulse Per Second (PPS) signals are aligned to frequency and phase
  - Maskable interrupts associated with each pulse
- Separate maskable timer interrupt event register
- Phase aligned adjustable (divide by N) clock output

44.5 IPTP Implementation Assumptions

The IPTP V2 standard makes several assumptions about the environment in which it operates. The following assumptions must be met to ensure correct operation of this implementation:

- The network must support multicast and unicast communication
- It must be possible to prevent multicast messages from propagating beyond a subnet
- Each clock implementing the protocol must meet the physical performance requirements, including oscillator frequency and time adjustment range.
- A clock's stated properties, including its stratum and identifier, must accurately describe the clock.

The following assumptions must be met to achieve optimal clock synchronization performance:

- A clock may contain asymmetric delays in its time stamp mechanism or protocol path. If these asymmetries are not negligible, they must be correctly accounted for.
- Network delay between master and slave on a subnet must be constant
- Boundary clocks must be used to synchronize across subnets
- The time stamps used in PTP are generated as close to the physical layer as practical for a given clock implementation. In cases where the most accurate time stamps can be generated only after a message has actually been transmitted, the actual value is communicated in the Follow_Up message from the master clock.
- The computing power of clocks implementing the protocol must be great enough, and the number of clocks per subnet must be small enough, to meet the standard constraints.
- The inherent stability of a clock's oscillator must be adequate. (The oscillator serving as the time base for a PTP clock has an accuracy and stability such that a second measured in the time base of the PTP clock is within +/- 0.01% of the SI second over the rated temperature range of the PTP clock.)

44.6 Modes of Operation

- Out-of-band - This mode is determined in the TSMR register. All the time stamps are transferred through the host interface, using dedicated registers. T

transmitter's time stamps after last indication in the buffer descriptor, and the receiver's time stamps when a PTP bit is set in the RxBD. The software is responsible for setting the PTP bit in the TxBD. In both receiver and transmitter the hardware can set an interrupt before accessing the time stamps.

- PTP frame parsing options - All of the relevant parsing values and fields can be configured using TSPDR register, retaining high flexibility.
- Pulse per second - The FIPER registers (IPTP_TMR_FIPER) can be configured by software in order to generate the pulse per second outputs.
- Alarm mode - The IPTP_TMR_ALARM register can be configured by software as an alarm (future time) value. When the IPTP RTC reaches this value, an interrupt is generated to the core, and a dedicated output trigger signal will be asserted.
- Input trigger - The IPTP_TMR_ETTS register configures a synchronous trigger to the IPTP RTC in order to capture a time stamp.
- Source clock select - This mode is configured in the Timer Control Register (IPTP_TMR_CTRL). There are two options for the source clock that can be connected to the IPTP RTC.
- Time Stamp Point - The IPTP_PTP_TSMR register can be configured to sample the IPTP RTC according to SFD detection or according to an external time stamp lock trigger.

44.7 Time Stamp Unit (TSU)

The TSU logic implementation supports out-of-band mode of operation. As mentioned in the preceding sections, the hardware assumption is that the IPTP is running on an Ethernet system.

In out-of-band method, the reporting is done using dedicated registers per interface. The software will read the transmitter's time stamps after the "last" indication of the TxBD, and the receiver's time stamps when a PTP bit is set in the RxBD. If there is sufficient time to guarantee that the software will be able to read the time stamp register between PTP frames, then the out-of-band method is sufficient.

The TSU logic supports MII/RMII Physical I/F.

In order to snapshot a precise time stamp, the hardware either observes the ports on the interfaces between the Media Access Controller (MAC) devices and the physical layer devices (PHY) to detect an SFD, or receives an external lock trigger and captures the timestamp on the transmit and receive sides.

Figure 44-2 shows the time stamp point of a PTP frame with SFD detection.

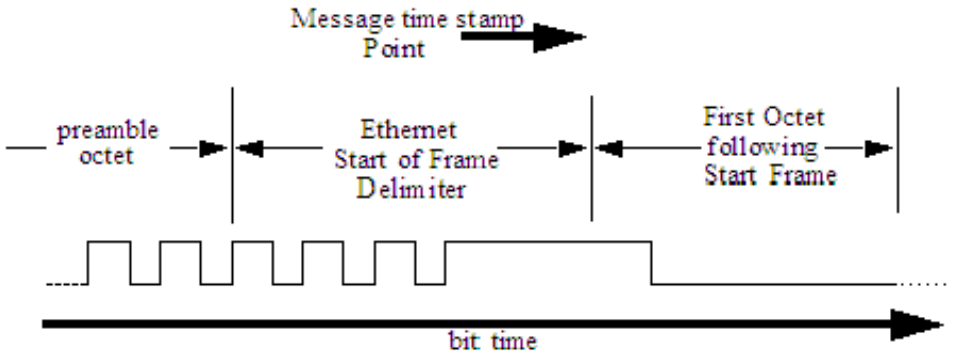


Figure 44-2. Message time stamp point with SFD detection

In addition, the TSU detects a Precise Time Protocol (PTP) frame of a specific type and generates an indication to the ARM platform and to the MAC.

In the IEEE1588 V2 specification, there are 10 types of PTP frames:

- Sync message
- Follow_Up message
- Delay_Req message
- Delay_Resp message
- Management message
- Pdelay_Req message
- Pdelay_Resp message
- Announce message
- Pdelay_Resp_Follow_Up message
- Signaling message

Those 10 messages are divided into two groups:

- PTP event
- PTP general

The PTP event group consists of Sync, Delay_Req, Pdelay_Req, and Pdealy_Resp messages. These messages should be time stamped and can cause an interrupt generation. The PTP general group consists of the rest of the frames; these frames transfer time stamps and determine general system parameters.

In order to detect a PTP frame see Table 44-32 and Table 44-34 for the parsing fields and values.

44.7.1 PTP Event Interrupts

In the case of out-of-band mode, the software can use the TSU interrupts as a trigger to read the time stamps registers. The software should clear the event register and read the time stamp before the next PTP frame detection. If a new PTP frame arrives and the relevant bit in the event register is still set, an overrun indication will be asserted.

44.8 IPTP Real Time Clock (RTC)

In a distributed control system containing multiple clocks, individual clocks tend to drift apart due to instabilities inherent in source oscillators and environmental conditions such as temperature, air circulation, mechanical stresses, vibration, aging, etc. Hence, some kind of correction is necessary to synchronize individual clocks to maintain the notion of global time, which is accurate to some requisite clock resolution.

In order to achieve this goal, the IPTP RTC supports the following:

- The nominal increment is chosen according to the nominal oscillator frequency
- The drift is compensated by slightly increasing or decreasing the increment

Figure 44-3 illustrates the block diagram of the IPTP RTC.

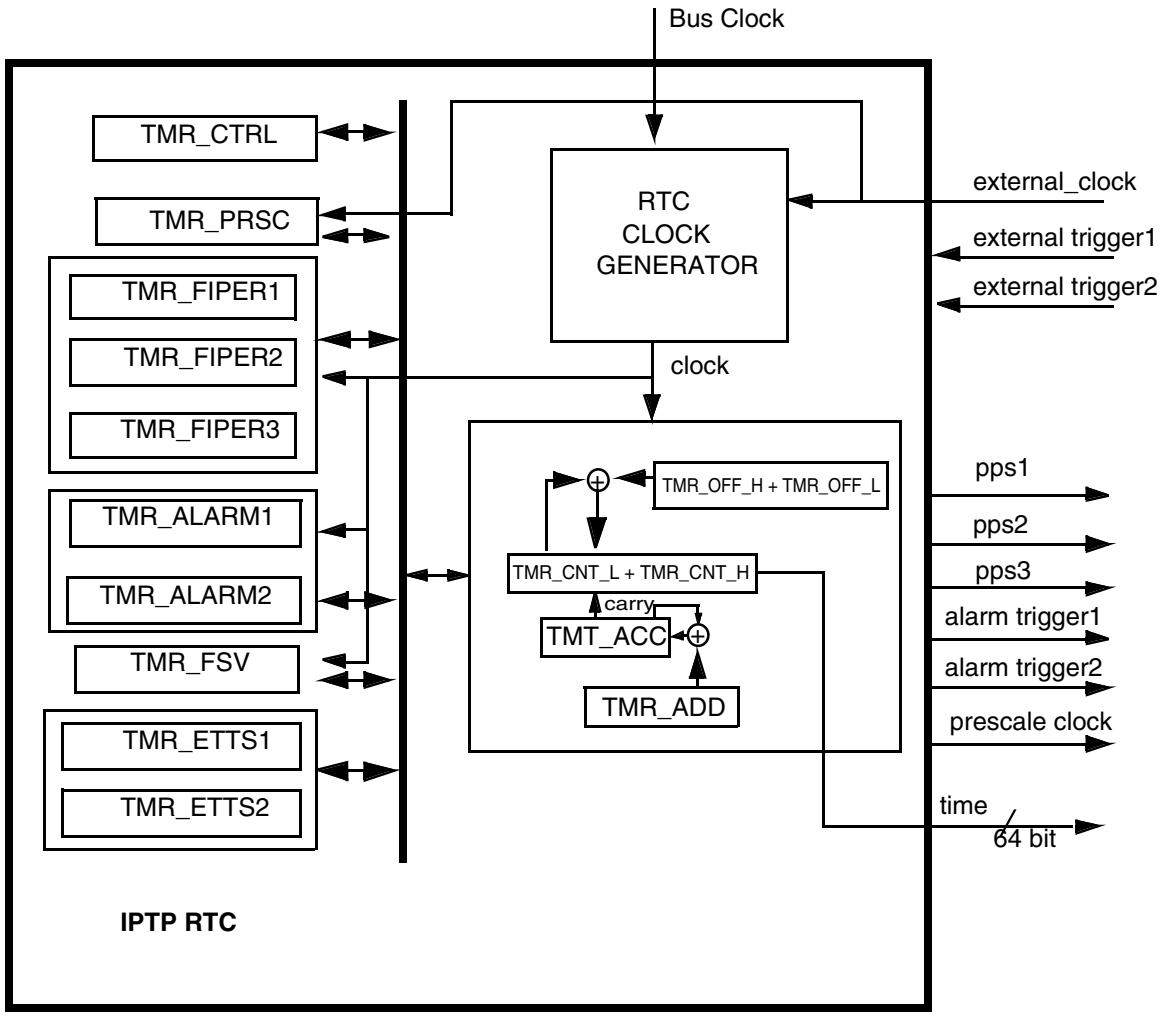


Figure 44-3. IPTP RTC block diagram

All of the constituents of the frequency compensated clock operate at the frequency of the external clock source. The frequency compensation value contained in the addend register is added to the accumulator once every external clock. The clock counter is incremented whenever the accumulator asserts an overflow pulse signal. Therefore, the nominal frequency at which the clock counter is incremented is the frequency that the software determined by updating the addend with a frequency compensation value.

The update of the frequency compensation value can be done every cycle of clock synchronization by Sync and delay messages.

Table 44-1 shows examples of possible IPTP RTC parameters. The results have been calculated according to the this equation:

frequency_compensation_value = 2^32/frequency_ratio.

Table 44-1. Values Example

Frequency of the Clock	Frequency of the Counter	Frequency Ratio	Frequency Compensation Value
200MHz	150MHz	1.33	0xc07b301
200MHz	190MHz	1.05	0xf3cf3cf
200MHz	70MHz	2.86	0x5982af7

In addition to the frequency compensation, the RTC enables updating the offset of the timer counter. The offset value is derived from the time stamps transferred on PTP frames.

The software uses the offset register in order to add the correction value. The timer counter adds the value of the offset register if a write transaction to the relevant address is detected. The values in the offset register and in the addend register are represented in 2's complement.

The standard demands sub-microseconds resolution. The IPTP RTC contains a 64-bit timer register and a 32-bit addend register. If the clock reaches 200MHz, by configuring the addend to 0xFFFF_FFFF, the counter will tick every 5ns.

IPTP RTC supports the following features:

- Two alarm output triggers, asserted if the timer value reaches the alarm (future time) register value
- Three pulse per second output signals, generated by configuring the FIPER registers. The signals are aligned in phase and frequency to the timer, and edge aligned to the IPTP timer reference output clock (prescale clock). The PPSs signal width is according to the period of the selected timer clock or according to the width of the IPTP timer reference output clock
- Two input triggers in order to capture time stamps in a dedicated register
- IPTP timer reference output clock which is a divided output clock, by configuring the prescale register
- Input of a time stamp lock trigger to sample the IPTP RTC according to an external trigger

44.8.1 RTC Clock Sources

The IPTP RTC enables three sources of input clock: an external oscillator, DPLL-3, DPLL-4, or the FEC_PHY clock.

All the clocks should meet the performances and requirements of the s

44.8.2 Prescale Output Clock and Pulse per Second Edge Alignment

In order to generate pulse per second signals that are edge aligned to the output prescale clock, as shown in Figure 44-4, the software configuration should be according to the following:

1. The ratio between the prescale clock divider value and the FIPER value should be devisable by the clock period, see Timer Fixed Interval Period 1 Regtister (IPTP_TMR_FIPER1).
2. Configure the TMR_FSV register to a value which is less by 3 clock periods than the required FIPER enabling time.
3. Note that each time when writing a new value to TMR_FSV, also write to the FIPERx register with the required PPS interval value.

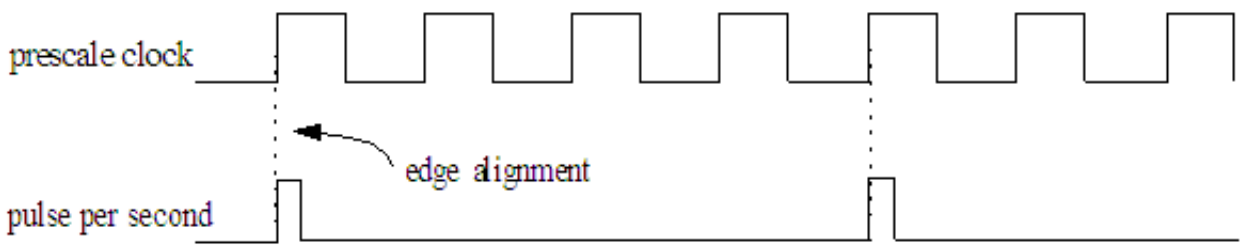


Figure 44-4. PPS and Prescale Clock Edge Alignment

44.9 PTP Frame Reception

The following describes the main actions performed for each operation mode when a PTP frame is received.

44.9.1 Out-of-Band Mode

In order to work in out-of-band mode the FEC should be configured as follows:

- FEC in Ethernet Mode
- Set IPTP mode in UPSMR[PTPE]

The Rx TSU is monitoring the received Ethernet frame on the physical line. When SFD is detected or external time stamp lock trigger is received, the IPTP Rx TSU locks the real time value of the IPTP clock.

If the Rx TSU parse unit has detected a PTP frame, it will notify the Ethernet MAC that a PTP frame has been detected, and the Ethernet controller will set RxBD[RPTP]. In addition, the Rx TSU will set the corresponding bit in the event register and generate an interrupt to the ARM platform.

When the software detects that a PTP frame has arrived, either by getting an interrupt or by polling the RxBD[RPTP], it will read the dedicated time stamp register (IPTP_TMR_UCx_RXTS_H/L) through the host interface.

In case the PTP frame is marked with CRC, Overrun, or Non-octet receive errors, the interrupt in the TMR_PEVENT will not be set and the software should drop this frame regardless of the indication in the RxBD[RPTP].

44.10 PTP Frame Transmission

The following describes the main actions performed for each operation mode when a PTP frame is transmitted.

44.11 Cycle Delay from Time Stamp Location

The following table describes the delay on the Ethernet line, in clock cycles, for each of the interfaces between the Media Access Controller (MAC) devices and the physical layer devices (PHY) in Tx and Rx.

Table 44-2 describes the delay on the Ethernet line.

Table 44-2. Delay on Ethernet Line

MII		RMII10		RMII100	
Tx	Rx	Tx	Rx	Tx	Rx
0	1	1	3-4	0.5	4

44.12 Initialization Sequence

44.12.1 RTC Mode Registers

1. Initialize the TMR_ALARMn registers to 0xFFFF_FFFF
2. Initialize the TMR_FSV register to 0xFFFF_FFFF
3. Program the IPTP timer clock period in IPTP_TMR_CTRL[tclk_period]
4. Select the timer clock source in the IPTP_TMR_CTRL[clksel]
5. Program the IPTP_TMR_TEMASK register
6. Write 0xFFFF_FFFF to IPTP_TMR_TEVENT to clear the RTC Event register
7. Program the TMR_ADD according to the needed drift compensation needed value

44.12.1.1 Enable Sequence

1. Enable the RTC by setting IPTP_TMR_CTRL[TE]
2. Enable the PTP by setting IPTP_PTP_TSMR[EN]
3. Enable the FEC.

44.13 Programmable Registers

All IPTP registers are 32-bits wide, located on 32-bit address boundaries, and should only be accessed as 32-bit quantities.

All addresses used in this chapter are offsets from RTC Base and PTP Base. (See "Memory Map" chapter.)

IPTP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FD_4000	Timer Control Register (IPTP_TMR_CTRL)	32	R/W	0000_0000h	44.13.1/ 2588
63FD_4004	Timer Events Register (IPTP_TMR_TEVENT)	32	R/W	0000_0000h	44.13.2/ 2590
63FD_4008	Timer Mask Register (IPTP_TMR_TEMASK)	32	R/W	0000_0000h	44.13.3/ 2592
63FD_400C	Timer Counter Low Register (IPTP_TMR_CNT_L)	32	R/W	0000_0000h	44.13.4/ 2593
63FD_4010	Timer Counter High Register (IPTP_TMR_CNT_H)	32	R/W	0000_0000h	44.13.5/ 2594
63FD_4014	Timer Addend Register (IPTP_TMR_ADD)	32	R/W	0000_0000h	44.13.6/ 2595

Table continues on the next page...

IPTP memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FD_4018	Timer Accumulator Register (IPTP_TMR_ACC)	32	R	0000_0000h	44.13.7/ 2596
63FD_401C	Timer Prescale Register (IPTP_TMR_PRSC)	32	R/W	0000_0000h	44.13.8/ 2596
63FD_4020	Timer Offset Low Register (IPTP_TMR_OFF_L)	32	R/W	0000_0000h	44.13.9/ 2597
63FD_4024	Timer Offset High Register (IPTP_TMR_OFF_H)	32	R/W	0000_0000h	44.13.10/ 2597
63FD_4028	Alarm 1 Time Low Register (IPTP_TMR_ALARM1_L)	32	R/W	0000_0000h	44.13.11/ 2598
63FD_402C	Alarm 1 Time High Register (IPTP_TMR_ALARM1_H)	32	R/W	0000_0000h	44.13.12/ 2598
63FD_4030	Alarm 2 Time Low Register (IPTP_TMR_ALARM2_L)	32	R/W	0000_0000h	44.13.13/ 2599
63FD_4034	Alarm 2 Time High Register (IPTP_TMR_ALARM2_H)	32	R/W	0000_0000h	44.13.14/ 2599
63FD_4038	Timer Fixed Interval Period 1 Regtister (IPTP_TMR_FIPER1)	32	R/W	FFFF_0000h	44.13.15/ 2600
63FD_403C	Timer Fixed Interval Period 2 Regtister (IPTP_TMR_FIPER2)	32	R/W	FFFF_0000h	44.13.16/ 2601
63FD_4040	Timer Fixed Interval Period 3 Regtister (IPTP_TMR_FIPER3)	32	R/W	FFFF_0000h	44.13.17/ 2602
63FD_4044	External Trigger Time Stamp 1 Low Register (IPTP_TMR_ETTS1_L)	32	R/W	0000_0000h	44.13.18/ 2603
63FD_4048	External Trigger Time Stamp 1 High Register (IPTP_TMR_ETTS1_H)	32	R/W	0000_0000h	44.13.19/ 2603
63FD_404C	External Trigger Time Stamp 2 Low Register (IPTP_TMR_ETTS2_L)	32	R/W	0000_0000h	44.13.20/ 2604
63FD_4050	External Trigger Time Stamp 2 High Register (IPTP_TMR_ETTS2_H)	32	R/W	0000_0000h	44.13.21/ 2604
63FD_4054	FIPER Start Low Register (IPTP_TMR_FSV_L)	32	R/W	0000_0000h	44.13.22/ 2605
63FD_4058	FIPER Start High Register (IPTP_TMR_FSV_H)	32	R/W	0000_0000h	44.13.23/ 2605
63FF_C000	Time Stamp Unit Parsing Definitons Register 1 (IPTP_PTP_TSPDR1)	32	R/W	0800_1100h	44.13.24/ 2605
63FF_C004	Time Stamp Unit Parsing Definitons Register 2 (IPTP_PTP_TSPDR2)	32	R/W	0140_013Fh	44.13.25/ 2606
63FF_C008	Time Stamp Unit Parsing Definitons Register 3 (IPTP_PTP_TSPDR3)	32	R/W	0001_0302h	44.13.26/ 2607

Table continues on the next page...

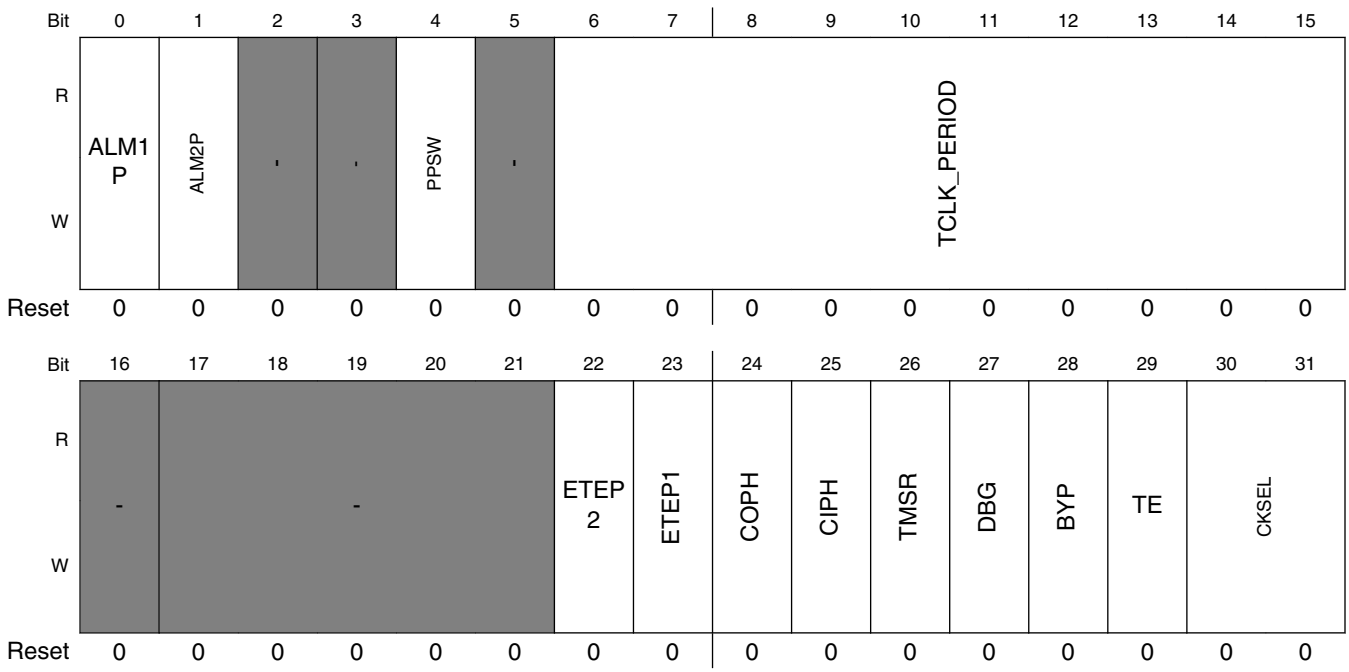
IPTP memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FF_C00C	Time Stamp Unit Parsing Definitons Register 4 (IPTP_PTP_TSPDR4)	32	R/W	0400_8100h	44.13.27/ 2608
63FF_C010	Time Stamp Unit Parsing Offset Values (IPTP_PTP_TSPOV)	32	R/W	0C17_244Ah	44.13.28/ 2609
63FF_C014	Time Stamp Unit Mode Register (IPTP_PTP_TSMR)	32	R/W	0000_0000h	44.13.29/ 2611
63FF_C018	Timer PTP Event Register (IPTP_PTP_TMR_PEVENT)	32	R/W	0000_0000h	44.13.30/ 2613
63FF_C01C	Timer PTP Mask Register (IPTP_PTP_TMR_PEMASK)	32	R/W	0000_0000h	44.13.31/ 2616
63FF_C020	Timer Stamp Unit Receiver TIme High (IPTP_TMR_UC_RXTS_H)	32	R	0000_0000h	44.13.32/ 2618
63FF_C030	Timer Stamp Unit Receiver TIme Low (IPTP_TMR_UC_RXTS_L)	32	R	0000_0000h	44.13.33/ 2619
63FF_C040	Time Stamp Unit Transmitter Time High (IPTP_TMR_UC_TXTS_H)	32	R	0000_0000h	44.13.34/ 2619
63FF_C050	Time Stamp Unit Transmitter Time Low (IPTP_TMR_UC_TXTS_L)	32	R	0000_0000h	44.13.35/ 2620
63FF_C060	Time Stamp Unit Parsing Definitons Register 5 (IPTP_PTP_TSPDR5)	32	R/W	0010_2030h	44.13.36/ 2620
63FF_C064	Time Stamp Unit Parsing Definitons Register 6 (IPTP_PTP_TSPDR6)	32	R/W	8090_A0B0h	44.13.37/ 2621
63FF_C068	Time Stamp Unit Parsing Definitons Register 7 (IPTP_PTP_TSPDR7)	32	R/W	C0D0_0000h	44.13.38/ 2622
63FF_C160	1588_ACC_PTP_Event Register (IPTP_1588_ACC_PTP_Event)	32	R/W	0000_0000h	44.13.39/ 2623
63FF_C164	1588_ACC_PTP_Mask Register (IPTP_1588_ACC_PTP_Mask)	32	R/W	0000_0000h	44.13.40/ 2626

44.13.1 Timer Control Register (IPTP_TMR_CTRL)

The IPTP_TMR_CTRL defines control fields relevant to the IPTP timer. IPTP_TMR_CTRL is a writable register in order to reset, configure, and initialize the IPTP timer clock. Only one of these registers is active for the entire Engine.

Address: IPTP_TMR_CTRL is 63FD_4000h base + 0h offset = 63FD_4000h



IPTP_TMR_CTRL field descriptions

Field	Description
0 ALM1P	Alarm1 output polarity default value is 0 0 active high output 1 active low output
1 ALM2P	Alarm2 output polarity default value is 0 0 active high output 1 active low output
2 -	RESERVED
3 -	RESERVED
4 PPSW	Pulse Per Second width output clock

Table continues on the next page...

IPTP_TMR_CTRL field descriptions (continued)

Field	Description
	NOTE: When IPTP_TMR_PRSC[PRSC_OCK] is set to 16'h1 this bit has no meaning default value is 0 0 PPS signal width is identical to the width selected timer clock 1 PPS signal width is identical to the width of the IPTP timer reference
5 -	RESERVED
6–15 TCLK_PERIOD	IPTP timer reference clock period. The timer clock counter will increment by TCLK_PERIOD every time the accumulator register overflows. This clock period must be larger than the clock period of the timer reference clock. This field can be programmed to 10'h1 to count clock ticks for applications that do not desire to add the clock period. default value is 10'h0 **NOTE:** It is recommended to work with clock ticks when the oscillator period can not be represented by an integer number
16 -	RESERVED
17–21 -	RESERVED
22 ETEP2	External trigger 2 edge polarity default value is 0 **NOTE:** When DLPB2 is set to 1, this bit is not relevant 0 time stamp on the rising edge of the external trigger 1 time stamp on the falling edge of the external trigger
23 ETEP1	External trigger 1 edge polarity default value is 0 **NOTE:** When DLPB1 is set to 1, this bit is not relevant 0 time stamp on the rising edge of the external trigger 1 time stamp on the falling edge of the external trigger
24 COPH	Generated clock (prescale) output phase default value is 0 0 non-inverted divided clock is output 1 inverted divided clock is output
25 CIPH	External oscillator input clock phase default value is 0 0 non-inverted frequency tuned timer input clock 1 inverted frequency tuned timer input clock
26 TMSR	Timer soft reset. When enabled, it resets all the timer registers and state machines. default value is 0

Table continues on the next page...

IPTP_TMR_CTRL field descriptions (continued)

Field	Description
	0 normal operation 1 places entire timer in reset except control and configuration registers
27 DBG	Debug mode. When set, allows writing to the read only timer registers default value is 0 0 normal operation 1 enable debug mode
28 BYP	Bypass drift compensated clock default value is 0 0 64-bit clock counter is incremented on the accumulator overflow (both phase and frequency are fixed) 1 64-bit clock counter is directly driven from the external oscillator ignoring accumulator overflow (only phase is fixed)
29 TE	IPTP Timer enable. If not enabled, all the timer registers and state machines are disabled. default value is 0 0 timer not enabled 1 timer enabled and resumes normal operation
30–31 CKSEL	Selection of the source clock for the IPTP timer clock 00 external clock source: This can be DPLL-3, DPLL-4, FEC_PHY clock or external oscillator. 01 IPG clock 1x Reserved for block

44.13.2 Timer Events Register (IPTP_TMR_TEVENT)

The IPTP timer implementation requires several interrupt event signals. It needs to add a new interrupt output line for the timer, independent of the existing Tx, Rx and Error interrupts. The IPTP timer interrupt does not require any interrupt coalescing. Software may poll this register at any time to check for pending interrupts. If an event occurs and its corresponding enable bit is set in the event mask register (EMASK), the event also causes a hardware interrupt at the Programmable Interrupt Controller (PIC). A bit in the timer event register is cleared by writing a 1 to that bit position.

Address: IPTP_TMR_TEVENT is 63FD_4000h base + 4h offset = 63FD_4004h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	-						ETS2	ETS1	-						ALM1	ALM2	FS	-							PPS1	PP2	PP3	-				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPTP_TMR_TEVENT field descriptions

Field	Description
0–5 -	RESERVED
6 ETS2	External trigger 2 time stamp sampled default value is 0 0 external trigger time stamp not sampled 1 external trigger time stamp sampled
7 ETS1	External trigger 1 time stamp sampled default value is 0 0 external trigger time stamp not sampled 1 external trigger time stamp sampled
8–13 -	RESERVED
14 ALM1	Current time equaled alarm 1 time register default value is 0 0 alarm time has not be reached yet 1 alarm time has been reached
15 ALM2	Current time equaled alarm 2 time register default value is 0 0 alarm time has not been reached yet 1 alarm time has been reached
16 FS	Current time equaled the FIPER Start Value (FSV) default value is 0 0 FIPER value has not been reached yet 1 FIPER value has been reached
17–23 -	RESERVED
24 PP1	Indicates that a periodic pulse has been generated based on FIPER1 register. default value is 0 0 periodic pulse not generated 1 periodic pulse generated
25 PP2	Indicates that a periodic pulse has been generated based on FIPER2 register. default value is 0 0 periodic pulse not generated 1 periodic pulse generated
26 PP3	Indicates that a periodic pulse has been generated based on FIPER3 register. default value is 0

Table continues on the next page...

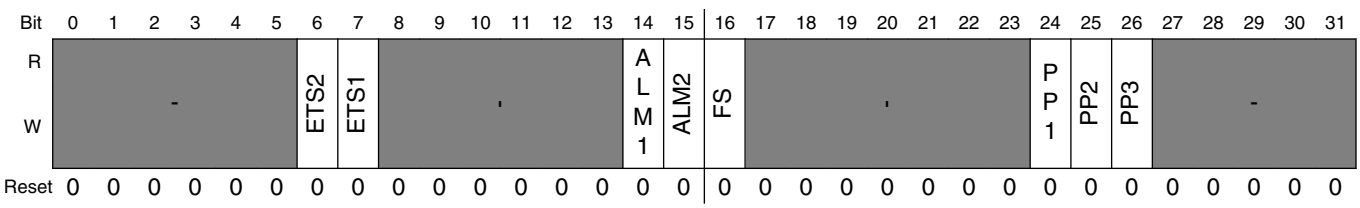
IPTP_TMR_TEVENT field descriptions (continued)

Field	Description
	0 periodic pulse not generated 1 periodic pulse generated
27–31 -	RESERVED

44.13.3 Timer Mask Register (IPTP_TMR_TEMASK)

The IPTP timer implementation requires several interrupt event signals. It needs to add a new interrupt output line for the timer, independent of the existing Tx, Rx and Error interrupts. The IPTP timer interrupt does not require any interrupt coalescing. Software may poll this register at any time to check for pending interrupts. If an event occurs and its corresponding enable bit is set in the event mask register (EMASK), the event also causes a hardware interrupt at the Programmable Interrupt Controller (PIC). A bit in the timer event register is cleared by writing a 1 to that bit position.

Address: IPTP_TMR_TEMASK is 63FD_4000h base + 8h offset = 63FD_4008h



IPTP_TMR_TEMASK field descriptions

Field	Description
0–5 -	RESERVED
6 ETS2	External trigger 2 time stamp sampled default value is 0 0 external trigger time stamp not sampled 1 external trigger time stamp sampled
7 ETS1	External trigger 1 time stamp sampled default value is 0 0 external trigger time stamp not sampled 1 external trigger time stamp sampled
8–13 -	RESERVED
14 ALM1	Current time equaled alarm 1 time register default value is 0

Table continues on the next page...

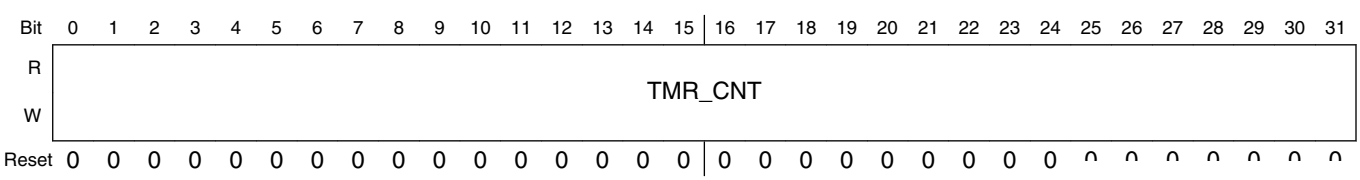
IPTP_TMR_TEMASK field descriptions (continued)

Field	Description
	0 alarm time has not be reached yet 1 alarm time has been reached
15 ALM2	Current time equaled alarm 2 time register default value is 0 0 alarm time has not been reached yet 1 alarm time has been reached
16 FS	Current time equaled the FIPER Start Value (FSV) default value is 0 0 FIPER value has not been reached yet 1 FIPER value has been reached
17–23 -	RESERVED
24 PP1	Indicates that a periodic pulse has been generated based on FIPER1 register. default value is 0 0 periodic pulse not generated 1 periodic pulse generated
25 PP2	Indicates that a periodic pulse has been generated based on FIPER2 register. default value is 0 0 periodic pulse not generated 1 periodic pulse generated
26 PP3	Indicates that a periodic pulse has been generated based on FIPER3 register. default value is 0 0 periodic pulse not generated 1 periodic pulse generated
27–31 -	RESERVED

44.13.4 Timer Counter Low Register (IPTP_TMR_CNT_L)

The timer counter register (IPTP_TMR_CNT_L) represents accurate time in terms of clock ticks or in nano-seconds. This is a read/write register. Writing to these registers will override the previous time.

Address: IPTP_TMR_CNT_L is 63FD_4000h base + Ch offset = 63FD_400Ch



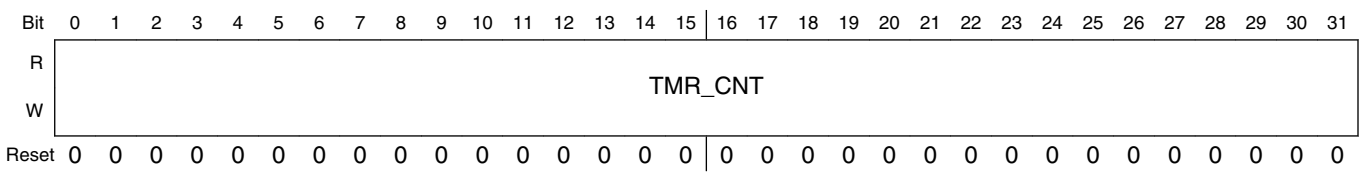
IPTP_TMR_CNT_L field descriptions

Field	Description
0–31 TMR_CNT	H/L value of the current time counter. Current time is calculated by adding TMROFF_H/L with the TMR_CNT_H/L counter. This register can be written through the register writes. Writes to the TMR_CNT_L register copies the written value into the shadow TMR_CNT_L register. Writes to the TMR_CNT_H register copies the values written into the shadow TMR_CNT_H register. Contents of the shadow registers are copied into the TMR_CNT_L and TMR_CNT_H registers following a write into the TMR_CNT_H register. Writes to these registers have precedence over the timer increment. The TMR_CNT_L register must be written first. Reads from the TMR_CNT_L register copies the entire 64-bit clock time of the read enable into the TMR_CNT_H/L shadow registers. Read instruction from the TMR_CNT_H register reads the value stored in the TMR_CNT_H shadow register. In order to get the correct 64-bit TMR_CNT_H/L counter values, the TMR_CNT_L register must be read first.

44.13.5 Timer Counter High Register (IPTP_TMR_CNT_H)

The timer counter register (PTP_TMR_CNT_H) represents accurate time in terms of clock ticks or in nano-seconds. This is a read/write register. Writing to these registers will override the previous time.

Address: IPTP_TMR_CNT_H is 63FD_4000h base + 10h offset = 63FD_4010h



IPTP_TMR_CNT_H field descriptions

Field	Description
0–31 TMR_CNT	H/L value of the current time counter. Current time is calculated by adding TMROFF_H/L with the TMR_CNT_H/L counter. This register can be written through the register writes. Writes to the TMR_CNT_L register copies the written value into the shadow TMR_CNT_L register. Writes to the TMR_CNT_H register copies the values written into the shadow TMR_CNT_H register. Contents of the shadow registers are copied into the TMR_CNT_L and TMR_CNT_H registers following a write into the TMR_CNT_H register. Writes to these registers have precedence over the timer increment. The TMR_CNT_L register must be written first. Reads from the TMR_CNT_L register copies the entire 64-bit clock time of the read enable into the TMR_CNT_H/L shadow registers. Read instruction from the TMR_CNT_H register reads the value stored in the TMR_CNT_H shadow register. In order to get the correct 64-bit TMR_CNT_H/L counter values, the TMR_CNT_L register must be read first.

44.13.6 Timer Addend Register (IPTP_TMR_ADD)

The Timer Drift Compensation Addend Register (IPTP_TMR_ADD) is used to hold the timer frequency compensation value (FreqCompensationValue). The nominal frequency of the clock counter is determined by the frequency division ratio (FreqDivRatio) and the clock frequency (FreqClock). This register is programmed with 232/FreqDivRatio. Frequency division ratio (FreqDivRatio) is the ratio between the frequency of the oscillator (TimerOsc) and the desired clock frequency (NominalFreq). FreqDivRatio is a design constant chosen to be greater than 1.0001. The ADDEND value is added to the 32-bit accumulator register at every rising edge of the oscillator clock (TimerOsc). The clock counter is incremented at every carry pulse of the accumulator.

Address: IPTP_TMR_ADD is 63FD_4000h base + 14h offset = 63FD_4014h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	ADDEND																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

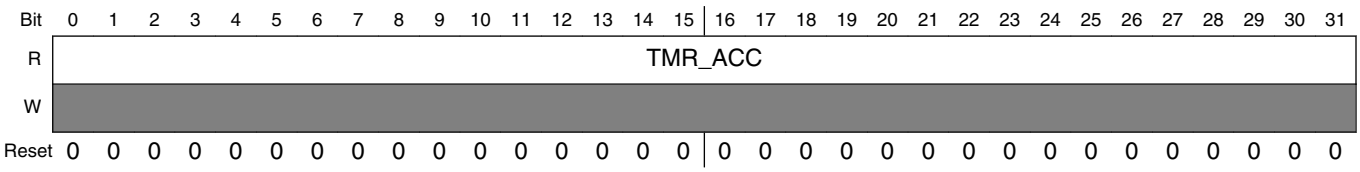
IPTP_TMR_ADD field descriptions

Field	Description
0–31 ADDEND	Timer drift compensation addend register value. It is programmed with a value of 2^32/FreqDivRatio. For example: FreqOsc = 50 MHz FreqClock = 40 MHz FreqDivRatio = 1.25 RTCAD = ceil(2^32/1.25) = 0xCCCC_CCCD default value is 32'h0

44.13.7 Timer Accumulator Register (IPTP_TMR_ACC)

The Timer Accumulator Register accumulates the value of the addend register into it. An overflow pulse of the accumulator is used to increment the timer clock (IPTP_TMR_CNT) by IPTP_TMR_CTRL[TCLK_PERIOD]. This register is read only. This register is writable only in debug mode (IPTP_TMR_CTRL[DBG]).

Address: IPTP_TMR_ACC is 63FD_4000h base + 18h offset = 63FD_4018h



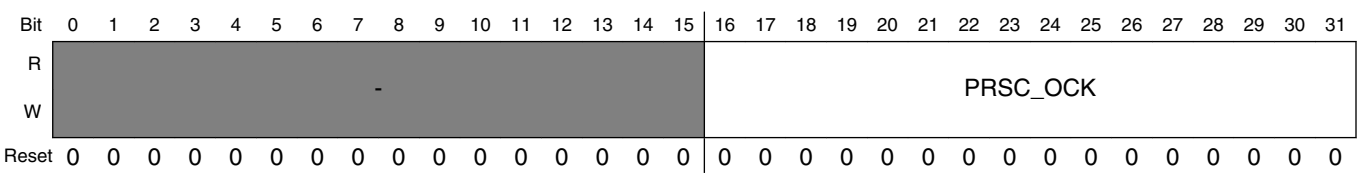
IPTP_TMR_ACC field descriptions

Field	Description
0–31 TMR_ACC	32bit timer accumulator value

44.13.8 Timer Prescale Register (IPTP_TMR_PRSC)

The Timer Prescale Register is used to adjust output clock frequency. 1588_ACC_PTP_Mask Register Timer Prescale Register shows the IPTP_TMR_PRSC register.

Address: IPTP_TMR_PRSC is 63FD_4000h base + 1Ch offset = 63FD_401Ch



IPTP_TMR_PRSC field descriptions

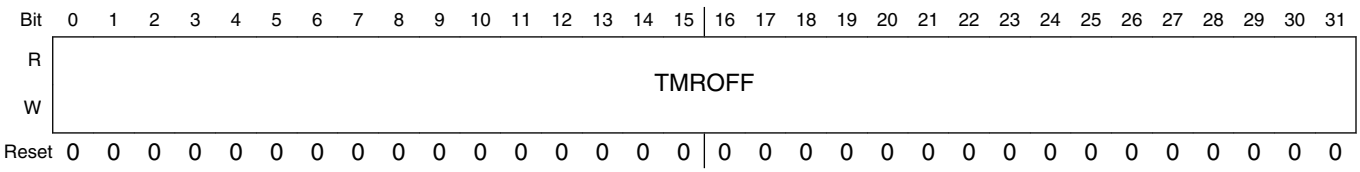
Field	Description
0–15 -	RESERVED
16–31 PRSC_OCK	Output clock division/prescale factor. Output clock is generated by dividing the timer clock by this number. Setting the prescaler value to 16'h1 will generate an output clock in the same frequency as the selected timer clock. default value is 16'h0

44.13.9 Timer Offset Low Register (IPTP_TMR_OFF_L)

The timer offset register is used to update the timer to the current time by adding its value to the clock counter

.

Address: IPTP_TMR_OFF_L is 63FD_4000h base + 20h offset = 63FD_4020h



IPTP_TMR_OFF_L field descriptions

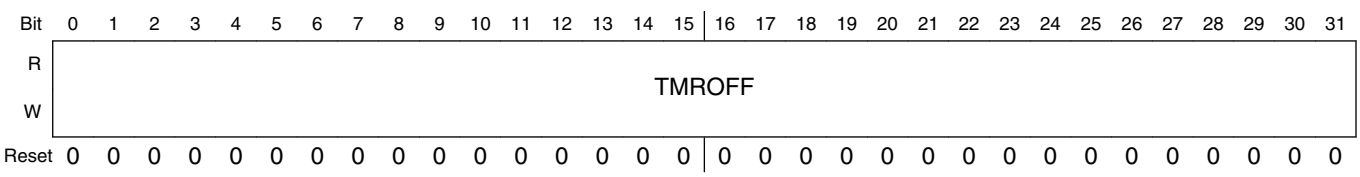
Field	Description
0–31 TMROFF	The TMROFF value, which is determined by the software calculations, updates the timer clock counter (TMR_CNT) with its value. The update is executed when the hardware detects write transaction to the offset high register. default value is 32'h0

44.13.10 Timer Offset High Register (IPTP_TMR_OFF_H)

The timer offset register is used to update the timer to the current time by adding its value to the clock counter

.

Address: IPTP_TMR_OFF_H is 63FD_4000h base + 24h offset = 63FD_4024h



IPTP_TMR_OFF_H field descriptions

Field	Description
0–31 TMROFF	The TMROFF value, which is determined by the software calculations, updates the timer clock counter (TMR_CNT) with its value. The update is executed when the hardware detects write transaction to the offset high register. default value is 32'h0

44.13.11 Alarm 1 Time Low Register (IPTP_TMR_ALARM1_L)

The Alarm (future time) Comparator Register (IPTP_TMR_ALARM1_L) holds alarm time for comparison with the current timer clock counter (IPTP_TMR_CNT)

Address: IPTP_TMR_ALARM1_L is 63FD_4000h base + 28h offset = 63FD_4028h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	TMR_ALARMn																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPTP_TMR_ALARM1_L field descriptions

Field	Description
0–31 TMR_ALARMn	The IPTP RTC interrupt is generated when the value of the RTC counter (TMR_CNT_L/TMR_CNT_H) is greater than or equal to TMR_ALARMn[TMR_ALARM]. This value should be configured before enabling the timer. default value is 32'h0

44.13.12 Alarm 1 Time High Register (IPTP_TMR_ALARM1_H)

The Alarm (future time) Comparator Register (IPTP_TMR_ALARM1_H) holds alarm time for comparison with the current timer clock counter (IPTP_TMR_CNT)

Address: IPTP_TMR_ALARM1_H is 63FD_4000h base + 2Ch offset = 63FD_402Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	TMR_ALARMn																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

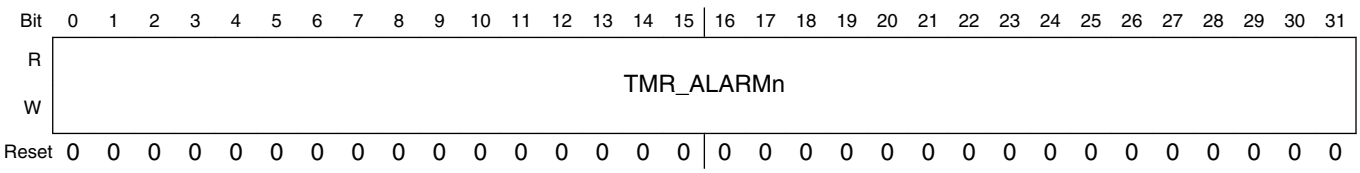
IPTP_TMR_ALARM1_H field descriptions

Field	Description
0–31 TMR_ALARMn	The IPTP RTC interrupt is generated when the value of the RTC counter (TMR_CNT_L/TMR_CNT_H) is greater than or equal to TMR_ALARMn[TMR_ALARM]. This value should be configured before enabling the timer. default value is 32'h0

44.13.13 Alarm 2 Time Low Register (IPTP_TMR_ALARM2_L)

The Alarm (future time) Comparator Register (IPTP_TMR_ALARM_H/ IPTP_TMR_ALARM_L) holds alarm time for comparison with the current timer clock counter (IPTP_TMR_CNT)

Address: IPTP_TMR_ALARM2_L is 63FD_4000h base + 30h offset = 63FD_4030h



IPTP_TMR_ALARM2_L field descriptions

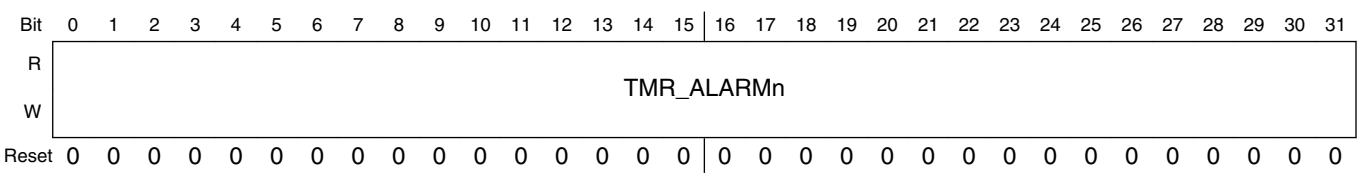
Field	Description
0–31 TMR_ALARMn	The IPTP RTC interrupt is generated when the value of the RTC counter (TMR_CNT_L/TMR_CNT_H) is greater than or equal to TMR_ALARMn[TMR_ALARM]. This value should be configured before enabling the timer. default value is 32'h0

44.13.14 Alarm 2 Time High Register (IPTP_TMR_ALARM2_H)

The Alarm (future time) Comparator Register (IPTP_TMR_ALARM_H/ IPTP_TMR_ALARM_L) holds alarm time for comparison with the current timer clock counter (IPTP_TMR_CNT)

.

Address: IPTP_TMR_ALARM2_H is 63FD_4000h base + 34h offset = 63FD_4034h



IPTP_TMR_ALARM2_H field descriptions

Field	Description
0–31 TMR_ALARMn	The IPTP RTC interrupt is generated when the value of the RTC counter (TMR_CNT_L/TMR_CNT_H) is greater than or equal to TMR_ALARMn[TMR_ALARM]. This value should be configured before enabling the timer. default value is 32'h0

44.13.15 Timer Fixed Interval Period 1 Regtister (IPTP_TMR_FIPER1)

The Timer Fixed Interval Period Pulse Generator Register is used to generate periodic pulses. This register is reset with 0xFFFF_FFFF to prevent any false pulse upon initialization. The down count register loads the value programmed in the Fixed Period Interval Register (FIPER). The FIPER must be programmed before the timer is enabled and before the FIPER start value (FSV), which triggers the start of the down counting, is configured. At every tick of the FreqOsc, the counter decrements by the value of IPTP_TMR_CTRL[TCLK_PERIOD]. It generates a pulse when the down counter value reaches zero regardless of the timer value. It reloads the down counter in the cycle following a pulse. The software is responsible for loading these registers with desired period intervals. There are three FIPER registers.

Address: IPTP_TMR_FIPER1 is 63FD_4000h base + 38h offset = 63FD_4038h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	FIPER																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPTP_TMR_FIPER1 field descriptions

Field	Description
0–31 FIPER	The FIPER value represents the rate of pulses generated at the output. The PPS signal is set with respect to the prescale clock. When the prescale value is different than one, the conditions below must be met: In order to allow PPS alignment to phase and frequency, the following equations must result in an integer value: (FIPER_VALUE + TMR_CTRL[TCLK_PERIOD]) / (TMR_PRSC[PRSC_OCK] *TMR_CTRL[TCLK_PERIOD]) FIPER_VALUE > TMR_CTRL[TCLK_PERIOD] * TMR_PRSC[PRSC_OCK] The FIPER start value must be configured to a value which is 3*TMR_CTRL[TCLK_PERIOD] less than the targeted value Assuming PPS should be set every 1 sec. (10^9 nsec), timer frequency is 100MHZ, and the TMR_CTRL[TCLK_PERIOD] is 10, then the configured value should be: {10^9/ TMR_CTRL[TCLK_PERIOD]}hex - {TMR_CTRL[TCLK_PERIOD]}hex = 5F5E100-{TMR_CTRL[TCLK_PERIOD]}hex= 5F5E0F6 In order to keep tracking the prescale output clock, each time before enabling the FIPER, the FIPER must be reset by writing a new value to the register. default value is 0

44.13.16 Timer Fixed Interval Period 2 Regtister (IPTP_TMR_FIPER2)

The Timer Fixed Interval Period Pulse Generator Register is used to generate periodic pulses. This register is reset with 0xFFFF_FFFF to prevent any false pulse upon initialization. The down count register loads the value programmed in the Fixed Period Interval Register (FIPER). The FIPER must be programmed before the timer is enabled and before the FIPER start value (FSV), which triggers the start of the down counting, is configured. At every tick of the FreqOsc, the counter decrements by the value of IPTP_TMR_CTRL[TCLK_PERIOD]. It generates a pulse when the down counter value reaches zero regardless of the timer value. It reloads the down counter in the cycle following a pulse. The software is responsible for loading these registers with desired period intervals. There are three FIPER registers.

Address: IPTP_TMR_FIPER2 is 63FD_4000h base + 3Ch offset = 63FD_403Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	FIPER																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPTP_TMR_FIPER2 field descriptions

Field	Description
0–31 FIPER	The FIPER value represents the rate of pulses generated at the output. The PPS signal is set with respect to the prescale clock. When the prescale value is different than one, the conditions below must be met: In order to allow PPS alignment to phase and frequency, the following equations must result in an integer value: (FIPER_VALUE + TMR_CTRL[TCLK_PERIOD]) / (TMR_PRSC[PRSC_OCK] *TMR_CTRL[TCLK_PERIOD]) FIPER_VALUE > TMR_CTRL[TCLK_PERIOD] * TMR_PRSC[PRSC_OCK] The FIPER start value must be configured to a value which is 3*TMR_CTRL[TCLK_PERIOD] less than the targeted value Assuming PPS should be set every 1 sec. (10^9 nsec), timer frequency is 100MHZ, and the TMR_CTRL[TCLK_PERIOD] is 10, then the configured value should be: {10^9/ TMR_CTRL[TCLK_PERIOD]}hex - {TMR_CTRL[TCLK_PERIOD]}hex = 5F5E100-{TMR_CTRL[TCLK_PERIOD]}hex= 5F5E0F6 In order to keep tracking the prescale output clock, each time before enabling the FIPER, the FIPER must be reset by writing a new value to the register. default value is 0

44.13.17 Timer Fixed Interval Period 3 Regtister (IPTP_TMR_FIPER3)

The Timer Fixed Interval Period Pulse Generator Register is used to generate periodic pulses. This register is reset with 0xFFFF_FFFF to prevent any false pulse upon initialization. The down count register loads the value programmed in the Fixed Period Interval Register (FIPER). The FIPER must be programmed before the timer is enabled and before the FIPER start value (FSV), which triggers the start of the down counting, is configured. At every tick of the FreqOsc, the counter decrements by the value of IPTP_TMR_CTRL[TCLK_PERIOD]. It generates a pulse when the down counter value reaches zero regardless of the timer value. It reloads the down counter in the cycle following a pulse. The software is responsible for loading these registers with desired period intervals. There are three FIPER registers.

Address: IPTP_TMR_FIPER3 is 63FD_4000h base + 40h offset = 63FD_4040h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	FIPER																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

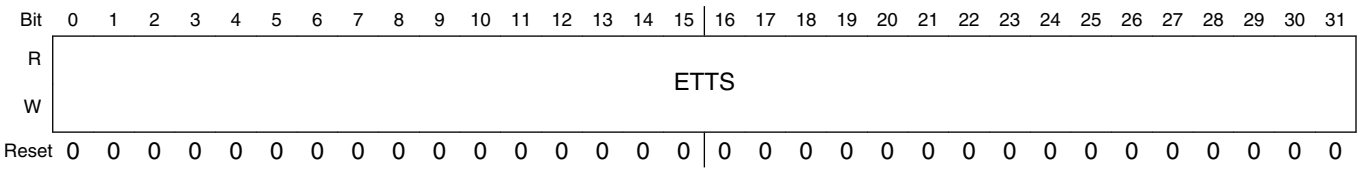
IPTP_TMR_FIPER3 field descriptions

Field	Description
0–31 FIPER	The FIPER value represents the rate of pulses generated at the output. The PPS signal is set with respect to the prescale clock. When the prescale value is different than one, the conditions below must be met: In order to allow PPS alignment to phase and frequency, the following equations must result in an integer value: (FIPER_VALUE + TMR_CTRL[TCLK_PERIOD]) / (TMR_PRSC[PRSC_OCK] *TMR_CTRL[TCLK_PERIOD]) FIPER_VALUE > TMR_CTRL[TCLK_PERIOD] * TMR_PRSC[PRSC_OCK] The FIPER start value must be configured to a value which is 3*TMR_CTRL[TCLK_PERIOD] less than the targeted value Assuming PPS should be set every 1 sec. (10^9 nsec), timer frequency is 100MHZ, and the TMR_CTRL[TCLK_PERIOD] is 10, then the configured value should be: {10^9/ TMR_CTRL[TCLK_PERIOD]}hex - {TMR_CTRL[TCLK_PERIOD]}hex = 5F5E100-{TMR_CTRL[TCLK_PERIOD]}hex= 5F5E0F6 In order to keep tracking the prescale output clock, each time before enabling the FIPER, the FIPER must be reset by writing a new value to the register. default value is 0

44.13.18 External Trigger Time Stamp 1 Low Register (IPTP_TMR_ETTS1_L)

The general purpose External Trigger Time Stamp Register (IPTP_TMR_ETTS1_L) holds the time stamp at the programmable edge of the external trigger. There are only two of these registers.

Address: IPTP_TMR_ETTS1_L is 63FD_4000h base + 44h offset = 63FD_4044h



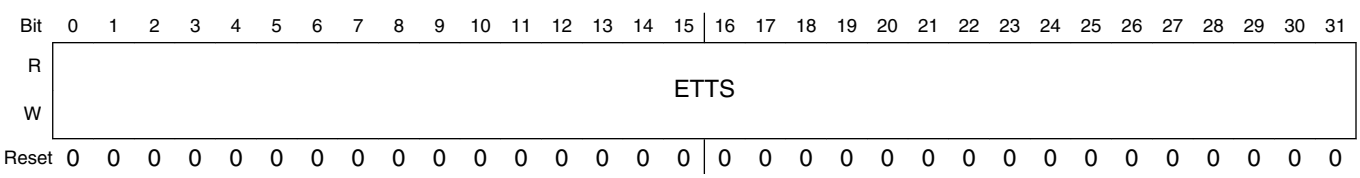
IPTP_TMR_ETTS1_L field descriptions

Field	Description
0–31 ETTS	Time stamp value at the programmable edge of the external trigger. default value is 32'h0

44.13.19 External Trigger Time Stamp 1 High Register (IPTP_TMR_ETTS1_H)

The general purpose External Trigger Time Stamp Register 1 High (IPTP_TMR_ETTS1_H) holds the time stamp at the programmable edge of the external trigger. There are only two of these registers.

Address: IPTP_TMR_ETTS1_H is 63FD_4000h base + 48h offset = 63FD_4048h



IPTP_TMR_ETTS1_H field descriptions

Field	Description
0–31 ETTS	Time stamp value at the programmable edge of the external trigger. default value is 32'h0

44.13.20 External Trigger Time Stamp 2 Low Register (IPTP_TMR_ETTS2_L)

The general purpose External Trigger Time Stamp Register (IPTP_TMR_ETTS2_L) holds the time stamp at the programmable edge of the external trigger. There are only two of these registers.

Address: IPTP_TMR_ETTS2_L is 63FD_4000h base + 4Ch offset = 63FD_404Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	ETTS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPTP_TMR_ETTS2_L field descriptions

Field	Description
0–31 ETTS	Time stamp value at the programmable edge of the external trigger. default value is 32'h0

44.13.21 External Trigger Time Stamp 2 High Register (IPTP_TMR_ETTS2_H)

The general purpose External Trigger Time Stamp Register 2 High (IPTP_TMR_ETTS2_H) holds the time stamp at the programmable edge of the external trigger. There are only two of these registers.

Address: IPTP_TMR_ETTS2_H is 63FD_4000h base + 50h offset = 63FD_4050h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	ETTS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPTP_TMR_ETTS2_H field descriptions

Field	Description
0–31 ETTS	Time stamp value at the programmable edge of the external trigger. default value is 32'h0

44.13.22 FIPER Start Low Register (IPTP_TMR_FSV_L)

The FIPER Start Register Value indicates to the three FIPERs when to start the down counting.

Address: IPTP_TMR_FSV_L is 63FD_4000h base + 54h offset = 63FD_4054h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	TMR_FSV																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPTP_TMR_FSV_L field descriptions

Field	Description
0–31 TMR_FSV	The FIPER Start value is used as an indicator to the FIPER when to start the down counting. default value is 32'h0

44.13.23 FIPER Start High Register (IPTP_TMR_FSV_H)

The FIPER Start Register Value indicates to the three FIPERs when to start the down counting.

Address: IPTP_TMR_FSV_H is 63FD_4000h base + 58h offset = 63FD_4058h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	TMR_FSV																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPTP_TMR_FSV_H field descriptions

Field	Description
0–31 TMR_FSV	The FIPER Start value is used as an indicator to the FIPER when to start the down counting. default value is 32'h0

44.13.24 Time Stamp Unit Parsing Definitons Register 1 (IPTP_PTP_TSPDR1)

In order to detect a PTP frame, the TSU parses several fields in the Ethernet frame. The IPTP_PTP_TSPDR1 register is used to determine the values of the Ethernet type field and for UDP/IPv4 or UDP/IPv6 only, the IPT field.

The values for the parsing can be modified.

NOTE

The default values are according to IEEE1588 V1 Time Stamp Unit Parsing Offset Values (IPTP_PTP_TSPOV). For IEEE1588 V2 the relevant field should be configured according to table Table 44-33.

Address: IPTP_PTP_TSPDR1 is 63FD_4000h base + 28000h offset = 63FF_C000h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	ETT																IPT								-							
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0

IPTP_PTP_TSPDR1 field descriptions

Field	Description
0–15 ETT	Ethernet Type Field. This field indicates the protocol used in the data field of the frame For IPv4 or IPv6, this field should be configured to 0x0800 For PTP over IEEE 802.3/Ethernet, this field should be configured to 0x88F7 default value is 0x0800
16–23 IPT	IP Type Field. For PTP over UDP/IPv4 or PTP over UDP/IPv6, this field should be configured to 0x11 default value is 0x11
24–31 -	RESERVED

44.13.25 Time Stamp Unit Parsing Definitons Register 2 (IPTP_PTP_TSPDR2)

In order to detect a PTP frame, the TSU parses several fields in the Ethernet frame. The IPTP_PTP_TSPDR2 register is used to determine the values of the UDP port field for UDP/IPv4 or UDP/IPv6 only. For any other protocol, this register is ignored.

The values for the parsing can be modified.

Address: IPTP_PTP_TSPDR2 is 63FD_4000h base + 28004h offset = 63FF_C004h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	DPNGE																DPNEV															
Reset	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1

IPTP_PTP_TSPDR2 field descriptions

Field	Description
0–15 DPNGE	Destination port field value, represents PTP general For PTP over UDP/IPv4 or PTP over UDP/IPv6 the recommended value according to IEEE1588 standard (1588_ACC_PTP_Mask Register Time Stamp Unit Parsing Offset Values) is 0x0140. default value is 0x0140
16–31 DPNEV	Destination port field value, represents PTP event For PTP over UDP/IPv4 or PTP over UDP/IPv6 the recommended value according to IEEE1588 standard (1588_ACC_PTP_Mask Register Time Stamp Unit Parsing Offset Values) is 0x013F default value is 0x013F

44.13.26 Time Stamp Unit Parsing Definitons Register 3 (IPTP_PTP_TSPDR3)

In order to detect a PTP frame, the TSU parses several fields in the Ethernet frame.

The IPTP_PTP_TSPDR3 register is used to determine the values of the PTP control field.

The values for the parsing can be modified and are used only when IPTP_PTP_TSMR[PTPV] = 0. When IPTP_PTP_TSMR[PTPV] = 1, this register is ignored.

Address: IPTP_PTP_TSPDR3 is 63FD_4000h base + 28008h offset = 63FF_C008h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	SYCTL								DRCTL								DRPCTL								FUCTL							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0

IPTP_PTP_TSPDR3 field descriptions

Field	Description
0–7 SYCTL	Control field value represents Sync message. The recommended value according to IEEE1588 standard (1588_ACC_PTP_Mask Register Time Stamp Unit Parsing Offset Values) is 0x00. default value is 0x00
8–15 DRCTL	Control field value represents Delay_Req message. The recommended value according to IEEE1588 standard (1588_ACC_PTP_Mask Register Time Stamp Unit Parsing Offset Values) is 0x01. default value is 0x01
16–23 DRPCTL	Control field value represents Delay_Resp message. The recommended value according to IEEE1588 standard (1588_ACC_PTP_Mask Register Time Stamp Unit Parsing Offset Values) is 0x03.

Table continues on the next page...

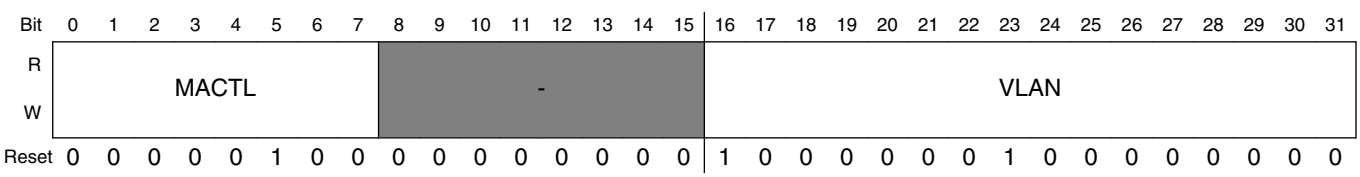
IPTP_PTP_TSPDR3 field descriptions (continued)

Field	Description
	default value is 0x03
24–31 FUCTL	Control field value represents Follow_Up message. The recommended value according to IEEE1588 standard (1588_ACC_PTP_Mask Register Time Stamp Unit Parsing Offset Values) is 0x02. default value is 0x02

44.13.27 Time Stamp Unit Parsing Definitons Register 4 (IPTP_PTP_TSPDR4)

The IPTP_PTP_TSPDR4[MACTL] field is used to determine the values of the PTP control field and is in continuance to the IPTP_PTP_TSPDR3 register. The values for the parsing can be modified and are used only when IPTP_PTP_TSMR[PTPV] = 0. When IPTP_PTP_TSMR[PTPV] = 1, this field is ignored. In addition, IPTP_PTP_TSPDR4 register provides a configured value to detect a VLAN field. This value can be detected during bytes 12-13 and cause a shifting of 4 bytes for all other parsing offsets. In case of a VLAN frame, the offsets should be configured exactly as for non VLAN frame. This field is relevant for both IPTP_PTP_TSMR[PTPV] = 0 and IPTP_PTP_TSMR[PTPV] = 1.

Address: IPTP_PTP_TSPDR4 is 63FD_4000h base + 2800Ch offset = 63FF_C00Ch



IPTP_PTP_TSPDR4 field descriptions

Field	Description
0–7 MACTL	Control field value represents Management message. The recommended value according to IEEE1588 standard (1588_ACC_PTP_Mask Register Time Stamp Unit Parsing Offset Values) is 0x04. default value is 0x04
8–15 -	RESERVED
16–31 VLAN	VLAN tag value (0x8100) default value is ox8100

44.13.28 Time Stamp Unit Parsing Offset Values (IPTP_PTP_TSPOV)

In order to detect a PTP frame, the TSU parses several fields in the Ethernet frame. TSU offset values register (IPTP_PTP_TSPOV) is used to determine the offset of each field in the Ethernet frame. In case the structure of the frame has been changed or manipulated, the field's offsets can be configured.

The parser allows to detect a PTP header as defined in IEEE1588 V1 and V2. In order to retain high flexibility, all of the parsing values and their offsets can be configured

.

1588_ACC_PTP_Mask Register Time Stamp Unit Parsing Offset Values describes the structure of the PTP frame according to IEEE1588 V1 (IPTP_PTP_TSMR[PTPV]=0)

Table 44-32. PTP Frame Structure according to IEEE1588 V1

Layer Name	Octet Number	Field Name	Field Value
IP	12-13	Ethernet Type Field	0x0800
UDP	23	IP Type Field.	0x11
PTP Event	36-37	UDP port	0x013F
PTP General	36-37	UDP port	0x0140
Sync	74	Control	0x00
Delay_Req	74	Control	0x01
Follow_Up	74	Control	0x02
Delay_Resp	74	Control	0x03
Management	74	Control	0x04

1588_ACC_PTP_Mask Register Time Stamp Unit Parsing Offset Values describes the structure of the PTP frame according to IEEE1588 V2 (IPTP_PTP_TSMR[PTPV]=1) when working with PTP over UDP/IPv4 or PTP over UDP/IPv6.

NOTE

Note: X signifies the PTP common message header offset.

Table 44-33. PTP Frame Structure according to IEEE1588 V2 PTP over UDP/IPv4 or PTP over UDP/IPv6

Layer Name	Octet Number	Field Name	Field Value according to standard
IP	12-13	Ethernet Type Field	0x0800
UDP	23	IP Type Field.	0x11
PTP Event	36-37	UDP port	0x013F

Table continues on the next page...

Table 44-33. PTP Frame Structure according to IEEE1588 V2 PTP over UDP/IPv4 or PTP over UDP/IPv6 (continued)

Layer Name	Octet Number	Field Name	Field Value according to standard
PTP General	36-37	UDP port	0x0140
Sync	X	Message Type	0x0
Delay_Req	X	Message Type	0x1
Pdelay_Req	X	Message Type	0x2
Pdelay_Resp	X	Message Type	0x3
Follow_Up	X	Message Type	0x8
Delay_Resp	X	Message Type	0x9
Pdelay_Resp_Follow_Up	X	Message Type	0xA
Announce	X	Message Type	0xB
Signaling	X	Message Type	0xC
Management	X	Message Type	0xD

1588_ACC_PTP_Mask Register Time Stamp Unit Parsing Offset Values describes the structure of the PTP frame according to IEEE1588 V2 (IPTP_PTP_TSMR[PTPV]=1) when working with PTP over IEEE 802.3/Ethernet.

NOTE

X signifies the PTP common message header offset.

Table 44-34. PTP Frame Structure according to IEEE1588 V2 PTP over IEEE 802.3/Ethernet

Layer Name	Octet Number	Field Name	Field Value according to standard
EtherType	12-13	Ethernet Type Field	0x88F7
Sync	X	Message Type	0x0
Delay_Req	X	Message Type	0x1
Pdelay_Req	X	Message Type	0x2
Pdelay_Resp	X	Message Type	0x3
Follow_Up	X	Message Type	0x8
Delay_Resp	X	Message Type	0x9
Pdelay_Resp_Follow_Up	X	Message Type	0xA
Announce	X	Message Type	0xB
Signaling	X	Message Type	0xC
Management	X	Message Type	0xD

Address: IPTP_PTP_TSPOV is 63FD_4000h base + 28010h offset = 63FF_C010h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	ETTOF								IPTOF								UDOF								PTOF							
Reset	0	0	0	0	1	1	0	0	0	0	0	1	0	1	1	1	0	0	1	0	0	1	0	0	0	1	0	0	1	0	1	0

IPTP_PTP_TSPOV field descriptions

Field	Description
0–7 ETTOF	Ethernet type offset (0x0C) default value is 0x0C
8–15 IPTOF	IP type offset (0x17) **NOTE:** When PTP_TSMR[ENUDP] = 0 this field is ignored. default value is 0x17
16–23 UDOF	UDP port offset (0x24) **NOTE:** When PTP_TSMR[ENUDP] = 0 this field is ignored. default value is 0x24
24–31 PTOF	Control field offset (when PTP_TSMR[PTPV]=0) (0x4A) MessageType field offset (when PTP_TSMR[PTPV]=1) default value is 0x4A

44.13.29 Time Stamp Unit Mode Register (IPTP_PTP_TSMR)

The IPTP_PTP_TSMR register is used to determine the modes of operation of the TSUs

.

Address: IPTP_PTP_TSMR is 63FD_4000h base + 28014h offset = 63FF_C014h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	'												OPMODE1	OPMODE2	OPMODE3	OPM ODE4
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	-							PTPV	ENU DP	ENTR N	-		EN1	EN2	EN3	EN4
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPTP_PTP_TSMR field descriptions

Field	Description
0–11 -	RESERVED
12 OPMODE1	TSU mode of operation. **NOTE:** If OPMODE is set, the Ethernet receive soft preamble mode should be set as well (MACCFG2[SRP]) default value is 0 0 out-of-band 1 Reserved
13 OPMODE2	RESERVED
14 OPMODE3	RESERVED
15 OPMODE4	RESERVED
16–22 -	RESERVED
23 PTPV	IPTP version of operation default value is 0 0 IEEE1588 V1 1 IEEE1588 V2
24 ENUDP	Enable UDP **NOTE:** This field is valid only when PTPV= 1 default value is 0 0 UDP disabled 1 UDP enabled
25 ENTRN	Enable transportSpecific **NOTE:** This field is valid only when PTPV= 1 and ENUDP= 0 default value is 0 0 Disable parsing of the transportSpecific field in the common message header 1 Enable parsing of the transportSpecific field in the common message header
26–27 -	RESERVED
28 EN1	default value is 0 0 Time Stamp Unit Disabled 1 Time Stamp Unit Enabled
29 EN2	RESERVED

Table continues on the next page...

IPTP_PTP_TSMR field descriptions (continued)

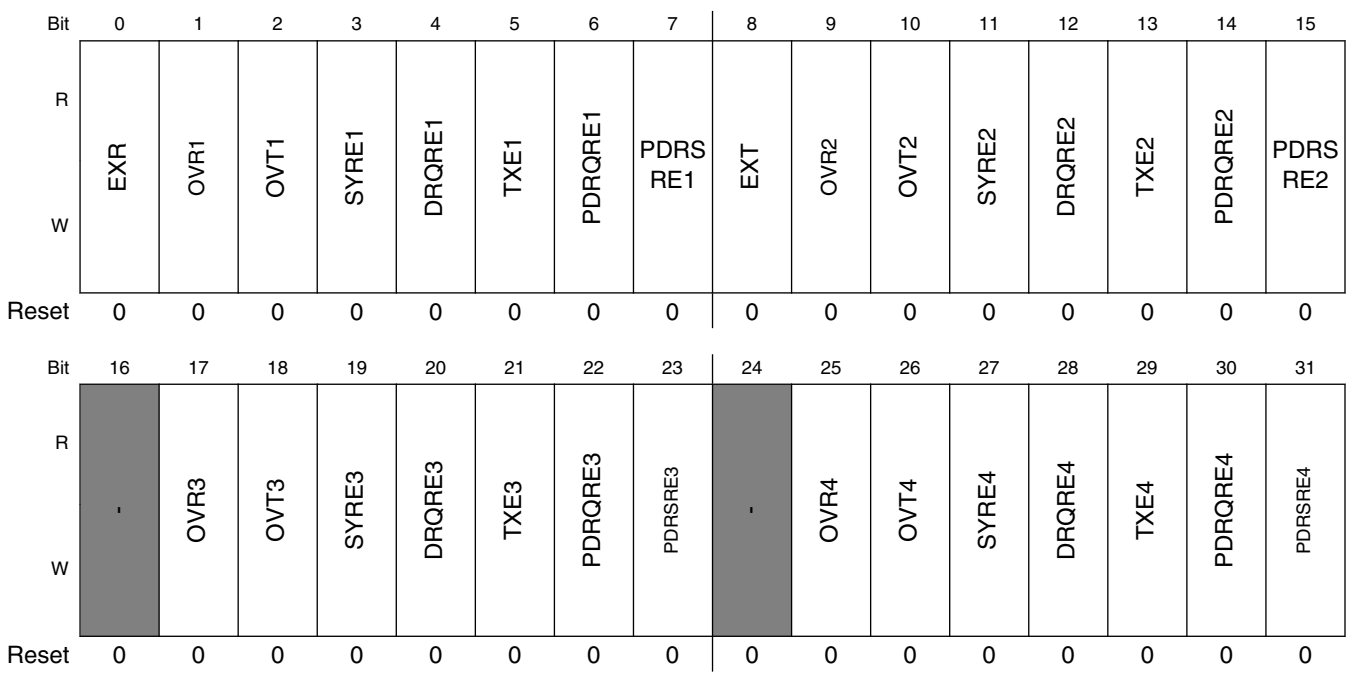
Field	Description
30 EN3	RESERVED
31 EN4	RESERVED

44.13.30 Timer PTP Event Register (IPTP_PTP_TMR_PEVENT)

The IPTP_PTP_TMR_PEVENT is used as the TSU event register. The IPTP_PTP_TMR_PEVENT reports events recognized on the Ethernet channel and generates interrupts. After event recognition, the TSU sets the corresponding IPTP_PTP_TMR_PEVENT bit. The IPTP_PTP_TMR_PEVENT bits are cleared by writing ones; writing zeros does not affect bit values. All unmasked bits must be cleared before the ARM Platform clears the internal interrupt request.

Interrupts generated by the TSU event register (IPTP_PTP_TMR_PEVENT) can be masked in the TSU mask register (IPTP_PTP_TMR_PEMASK), which has the same bit format as IPTP_PTP_TMR_PEVENT. If a IPTP_PTP_TMR_PEMASK bit = 1, the corresponding interrupt in the event register is enabled. If the bit is 0, the interrupt is masked.

Address: IPTP_PTP_TMR_PEVENT is 63FD_4000h base + 28018h offset = 63FF_C018h



IPTP_PTP_TMR_PEVENT field descriptions

Field	Description
0 EXR	External trigger was not received to receiver from PHY. No time stamp taken default value is 0
1 OVR1	Overrun occurred in receiver time stamp register. default value is 0
2 OVT1	Overrun occurred in transmitter time stamp register. default value is 0
3 SYRE1	Sync frame has been received. default value is 0
4 DRQRE1	Delay_req frame has been received. default value is 0
5 TXE1	PTP frame has been transmitted. default value is 0
6 PDRQRE1	Pdelay_Req frame has been received. default value is 0
7 PDRSRE1	Pdelay_Resp frame has been received. default value is 0
8 EXT	External trigger was not received to transmitter from PHY. No time stamp taken default value is 0
9 OVR2	RESERVED
10 OVT2	RESERVED
11 SYRE2	RESERVED
12 DRQRE2	RESERVED
13 TXE2	RESERVED
14 PDRQRE2	RESERVED
15 PDRSRE2	RESERVED
16 -	RESERVED
17 OVR3	RESERVED
18 OVT3	RESERVED

Table continues on the next page...

IPTP_PTP_TMR_PEVENT field descriptions (continued)

Field	Description
19 SYRE3	RESERVED
20 DRQRE3	RESERVED
21 TXE3	RESERVED
22 PDRQRE3	RESERVED
23 PDRSRE3	RESERVED
24 -	RESERVED
25 OVR4	RESERVED
26 OVT4	RESERVED
27 SYRE4	RESERVED
28 DRQRE4	RESERVED
29 TXE4	RESERVED
30 PDRQRE4	RESERVED
31 PDRSRE4	RESERVED

44.13.31 Timer PTP Mask Register (IPTP_PTP_TMR_PEMASK)

Interrupts generated by the TSU event register (IPTP_PTP_TMR_PEVENT) can be masked in the TSU mask register (IPTP_PTP_TMR_PEMASK), which has the same bit format as IPTP_PTP_TMR_PEVENT. If a IPTP_PTP_TMR_PEMASK bit = 1, the corresponding interrupt in the event register is enabled. If the bit is 0, the interrupt is masked.

Address: IPTP_PTP_TMR_PEMASK is 63FD_4000h base + 2801Ch offset = 63FF_C01Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	EXR	OVR1	OVT1	SYRE1	DRQRE1	TXE1	PDRQRE1	PDRS RE1	EXT	OVR2	OVT2	SYRE2	DRQRE2	TXE2	PDRQRE2	PDRS RE2
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	-	OVR3	OVT3	SYRE3	DRQRE3	TXE3	PDRQRE3	PDRS RE3	-	OVR4	OVT4	SYRE4	DRQRE4	TXE4	PDRQRE4	PDRS RE4
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPTP_PTP_TMR_PEMASK field descriptions

Field	Description
0 EXR	External trigger was not received to receiver from PHY. No time stamp taken default value is 0
1 OVR1	Overrun occurred in receiver time stamp register. default value is 0
2 OVT1	Overrun occurred in transmitter time stamp register. default value is 0
3 SYRE1	Sync frame has been received. default value is 0
4 DRQRE1	Delay_req frame has been received. default value is 0
5 TXE1	PTP frame has been transmitted. default value is 0
6 PDRQRE1	Pdelay_Req frame has been received. default value is 0
7 PDRSRE1	Pdelay_Resp frame has been received. default value is 0

Table continues on the next page...

IPTP_PTP_TMR_PEMASK field descriptions (continued)

Field	Description
8 EXT	External trigger was not received to transmitter from PHY. No time stamp taken default value is 0
9 OVR2	RESERVED
10 OVT2	RESERVED
11 SYRE2	RESERVED
12 DRQRE2	RESERVED
13 TXE2	RESERVED
14 PDRQRE2	RESERVED
15 PDRSRE2	RESERVED
16 -	RESERVED
17 OVR3	RESERVED
18 OVT3	RESERVED
19 SYRE3	RESERVED
20 DRQRE3	RESERVED
21 TXE3	RESERVED
22 PDRQRE3	RESERVED
23 PDRSRE3	RESERVED
24 -	RESERVED
25 OVR4	RESERVED
26 OVT4	RESERVED
27 SYRE4	RESERVED
28 DRQRE4	RESERVED

Table continues on the next page...

IPTP_PTP_TMR_PEMASK field descriptions (continued)

Field	Description
29 TXE4	RESERVED
30 PDRQRE4	RESERVED
31 PDRSRE4	RESERVED

44.13.32 Timer Stamp Unit Receiver TIme High (IPTP_TMR_UC_RXTS_H)

TheTime Stamp Unit Receiver Time High (IPTP_TMR_UC_TXTS_H) register is shown in 1588_ACC_PTP_Mask Register Timer Stamp Unit Receiver TIme High.

The IPTP_TMR_UC_TXTS_L/TMR_UC_TXTS_H registers are used to capture the time stamp of a recognized PTP frame on the transmitter lines. The TSU samples the IPTP RTC 64-bit counter immediately after detecting the message time stamp point (See MTSP bit in IPTP_TSMR_CTRL). The sampled value becomes valid only after one of the frame events. In this case, the ARM platform reads the value in order to calculate the synchronization's commands.

IPTP_TMR_UC_RXTS_L/IPTP_TMR_UC_RXTS_H have the same purpose on the receiver lines

Address: IPTP_TMR_UC_RXTS_H is 63FD_4000h base + 28020h offset = 63FF_C020h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	TSVAL																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPTP_TMR_UC_RXTS_H field descriptions

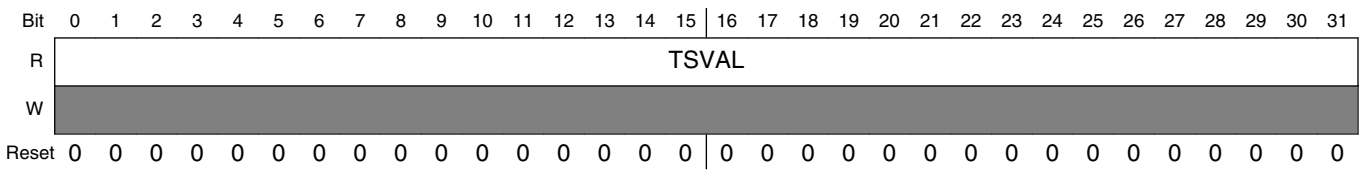
Field	Description
0–31 TSVAL	Time stamp value of the PTP packet

44.13.33 Timer Stamp Unit Receiver TIme Low (IPTP_TMR_UC_RXTS_L)

The IPTP_TMR_UC_TXTS_L/TMR_UC_TXTS_H registers are used to capture the time stamp of a recognized PTP frame on the transmitter lines. The TSU samples the IPTP RTC 64-bit counter immediately after detecting the message time stamp point (See MTSP bit in IPTP_TSMR_CTRL). The sampled value becomes valid only after one of the frame events. In this case, the ARM platform reads the value in order to calculate the synchronization's commands.

IPTP_TMR_UC_RXTS_L/IPTP_TMR_UC_RXTS_H have the same purpose on the receiver lines.

Address: IPTP_TMR_UC_RXTS_L is 63FD_4000h base + 28030h offset = 63FF_C030h



IPTP_TMR_UC_RXTS_L field descriptions

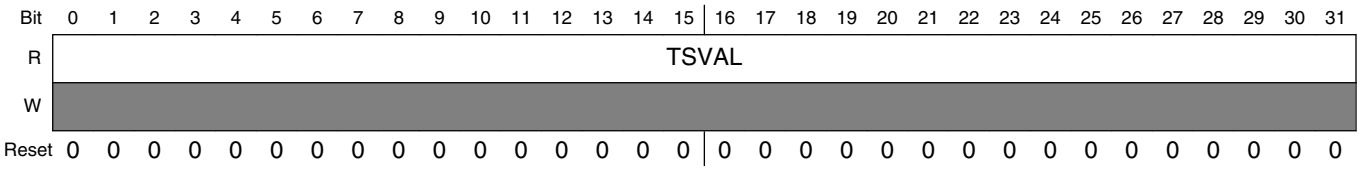
Field	Description
0–31 TSVAL	Time stamp value of the PTP packet

44.13.34 Time Stamp Unit Transmitter Time High (IPTP_TMR_UC_TXTS_H)

The IPTP_TMR_UC_TXTS_L/TMR_UC_TXTS_H registers are used to capture the time stamp of a recognized PTP frame on the transmitter lines. The TSU samples the IPTP RTC 64-bit counter immediately after detecting the message time stamp point (See MTSP bit in IPTP_TSMR_CTRL). The sampled value becomes valid only after one of the frame events. In this case, the ARM platform reads the value in order to calculate the synchronization's commands.

IPTP_TMR_UC_RXTS_L/IPTP_TMR_UC_RXTS_H have the same purpose on the receiver lines.

Address: IPTP_TMR_UC_TXTS_H is 63FD_4000h base + 28040h offset = 63FF_C040h



IPTP_TMR_UC_TXTS_H field descriptions

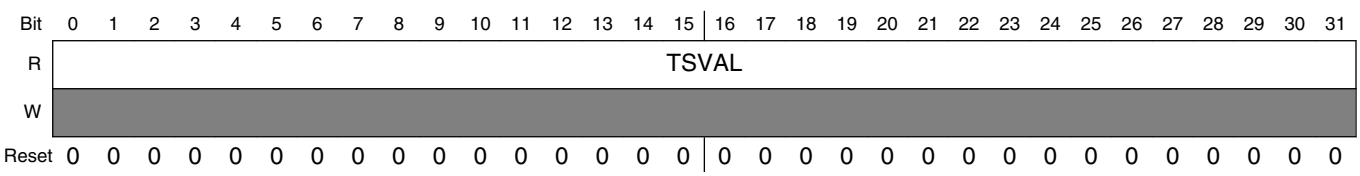
Field	Description
0–31 TSVAL	Time stamp value of the PTP packet

44.13.35 Time Stamp Unit Transmitter Time Low (IPTP_TMR_UC_TXTS_L)

The IPTP_TMR_UC_TXTS_L/TMR_UC_TXTS_H registers are used to capture the time stamp of a recognized PTP frame on the transmitter lines. The TSU samples the IPTP RTC 64-bit counter immediately after detecting the message time stamp point (See MTSP bit in IPTP_TSMR_CTRL). The sampled value becomes valid only after one of the frame events. In this case, the ARM platform reads the value in order to calculate the synchronization's commands.

IPTP_TMR_UC_RXTS_L/IPTP_TMR_UC_RXTS_H have the same purpose on the receiver lines

Address: IPTP_TMR_UC_TXTS_L is 63FD_4000h base + 28050h offset = 63FF_C050h



IPTP_TMR_UC_TXTS_L field descriptions

Field	Description
0–31 TSVAL	Time stamp value of the PTP packet

44.13.36 Time Stamp Unit Parsing Definitons Register 5 (IPTP_PTP_TSPDR5)

The IPTP_PTP_TSPDR5 register is used to determine the values of the PTP MessageType field.

The values for the parsing can be modified and are used only when IPTP_PTP_TSMR[PTPV] = 1. When IPTP_PTP_TSMR[PTPV] = 0, this register is ignored.

Address: IPTP_PTP_TSPDR5 is 63FD_4000h base + 28060h offset = 63FF_C060h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	SYMSG				-				DRMSG				-				PDRMSG				-				PDRPMSG				-			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0

IPTP_PTP_TSPDR5 field descriptions

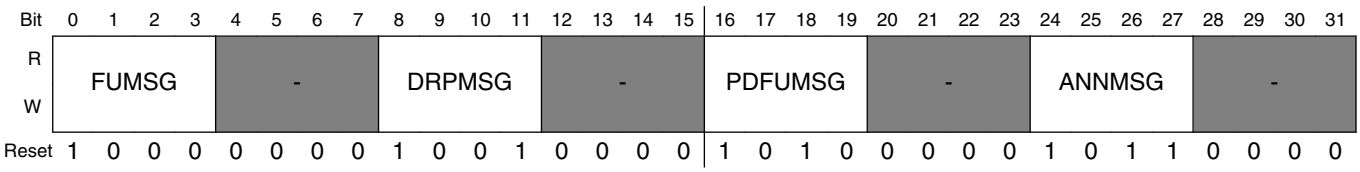
Field	Description
0–3 SYMSG	Message Type field value represents Sync message. The recommended value according to IEEE1588 standard (Table 44-34) 0x0. default value is 0x0
4–7 -	RESERVED set to 0x0
8–11 DRMSG	Message Type field value represents Delay_Req message. The recommended value according to IEEE1588 standard (Table 44-34) 0x1. default value is 0x1
12–15 -	RESERVED set to 0x0
16–19 PDRMSG	Message Type field value represents Pdelay_Req message. The recommended value according to IEEE1588 standard (Table 44-34) 0x2. default value is 0x2
20–23 -	RESERVED set to 0x0
24–27 PDRPMSG	Message Type field value represents Pdelay_Resp message. The recommended value according to IEEE1588 standard (Table 44-34) 0x3. default value is 0x3
28–31 -	RESERVED set to 0x0

44.13.37 Time Stamp Unit Parsing Definitons Register 6 (IPTP_PTP_TSPDR6)

The IPTP_PTP_TSPDR6 register is used to determine the values of the PTP MessageType field and is in continuance with the IPTP_PTP_TSPDR5 register.

The values for the parsing can be modified and are used only when IPTP_PTP_TSMR[PTPV] = 1. When IPTP_PTP_TSMR[PTPV] = 0, this register is ignored.

Address: IPTP_PTP_TSPDR6 is 63FD_4000h base + 28064h offset = 63FF_C064h



IPTP_PTP_TSPDR6 field descriptions

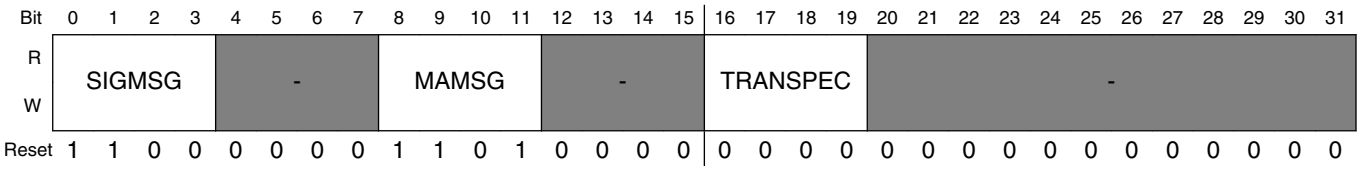
Field	Description
0–3 FUMSG	Message Type field value represents Follow_Up message. The recommended value according to IEEE1588 standard (Table 44-34) is 0x8. default value is 0x8
4–7 -	RESERVED set to 0x0
8–11 DRPMSG	Message Type field value represents Delay_Resp message. The recommended value according to IEEE1588 standard (Table 44-34) is 0x9. default value is 0x9
12–15 -	RESERVED set to 0x0
16–19 PDFUMSG	Message Type field value represents Pdelay_Resp_Follow_Up message. The recommended value according to IEEE1588 standard (Table 44-34) is 0xA. default value is 0xA
20–23 -	RESERVED set to 0x0
24–27 ANNMSG	Message Type field value represents Announce message. The recommended value according to IEEE1588 standard (Table 44-34) is 0xB. default value is 0xB
28–31 -	RESERVED set to 0x0

44.13.38 Time Stamp Unit Parsing Definitons Register 7 (IPTP_PTP_TSPDR7)

The IPTP_PTP_TSPDR7 register is used to determine the values of the PTP MessageType field and is in continuance with the IPTP_PTP_TSPDR6 register.

The values for the parsing can be modified and are used only when IPTP_PTP_TSMR[PTPV] = 1. When IPTP_PTP_TSMR[PTPV] = 0, this register is ignored.

Address: IPTP_PTP_TSPDR7 is 63FD_4000h base + 28068h offset = 63FF_C068h



IPTP_PTP_TSPDR7 field descriptions

Field	Description
0–3 SIGMSG	Message Type field value represents Signaling message. The recommended value according to IEEE1588 standard (Table 44-34) is 0xC. default value is 0xC
4–7 -	RESERVED set to 0x0
8–11 MAMSG	Message Type field value represents Management message. The recommended value according to IEEE1588 standard (Table 44-34) is 0xD. default value is 0xD
12–15 -	RESERVED set to 0x0
16–19 TRANSPEC	transportSpecific field value represents a subtype of the Ethertype. **NOTE:** This field is parsed only when PTP_TSMR[ENTRN]=1, PTP_TSMR[PTPV] = 1, and PTP_TSMR[ENUDP] = 0. default value is 0x0
20–31 -	RESERVED set to 0x0

44.13.39 1588_ACC_PTP_Event Register (IPTP_1588_ACC_PTP_Event)

The IPTP_1588_ACC_PTP_EVENT register reports events recognized on the Ethernet channel and generates interrupts. After event recognition, the TSU sets the corresponding event bit. The event bits are cleared by writing ones. Writing zeros does not affect bit values. All unmasked bits must be cleared before the ARM platform clears the internal interrupt request.

Interrupts generated by the event register can be masked in the mask register (IPTP_1588_ACC_PTP_MASK), which has the same bit format as the event register. If a mask bit = 1, the corresponding interrupt in the event register is enabled. If the bit is 0, the interrupt is masked.

Address: IPTP_1588_ACC_PTP_Event is 63FD_4000h base + 28160h offset = 63FF_C160h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	OVR1	OVT1	SYRE1	DRQRE1	GTXE1	PDRQRE1	PDRSRE1	OVR2	OVT2	SYRE2	DRQRE2	GTXE2	PDRQRE2	PDRSRE2	OVR3	OVT3
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	SYRE3	DRQRE3	GTXE3	PDRQRE3	PDRSRE3	OVR4	OVT4	SYRE4	DRQRE4	GTXE4	PDRQRE4	PDRSRE4				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPTP_1588_ACC_PTP_Event field descriptions

Field	Description
0 OVR1	Overrun occurred in receiver time stamp register.
1 OVT1	Overrun occurred in transmitter time stamp register.
2 SYRE1	Sync frame has been received.
3 DRQRE1	Delay_req frame has been received.
4 GTXE1	(GLOBCOAL+1) PTP frames have been transmitted.
5 PDRQRE1	Pdelay_req frame has been received.
6 PDRSRE1	Pdelay_resp frame has been received.
7 OVR2	RESERVED
8 OVT2	RESERVED
9 SYRE2	RESERVED
10 DRQRE2	RESERVED

Table continues on the next page...

IPTP_1588_ACC_PTP_Event field descriptions (continued)

Field	Description
11 GTXE2	RESERVED
12 PDRQRE2	RESERVED
13 PDRSRE2	RESERVED
14 OVR3	RESERVED
15 OVT3	RESERVED
16 SYRE3	RESERVED
17 DRQRE3	RESERVED
18 GTXE3	RESERVED
19 PDRQRE3	RESERVED
20 PDRSRE3	RESERVED
21 OVR4	RESERVED
22 OVT4	RESERVED
23 SYRE4	RESERVED
24 DRQRE4	RESERVED
25 GTXE4	RESERVED
26 PDRQRE4	RESERVED
27 PDRSRE4	RESERVED
28–31 -	RESERVED

44.13.40 1588_ACC_PTP_Mask Register (IPTP_1588_ACC_PTP_Mask)

The IPTP_1588_ACC_PTP_EVENT register reports events recognized on the Ethernet channel and generates interrupts. After event recognition, the TSU sets the corresponding event bit. The event bits are cleared by writing ones. Writing zeros does not affect bit values. All unmasked bits must be cleared before the ARM platform clears the internal interrupt request.

Interrupts generated by the event register can be masked in the mask register (IPTP_1588_ACC_PTP_MASK), which has the same bit format as the event register. If a mask bit = 1, the corresponding interrupt in the event register is enabled. If the bit is 0, the interrupt is masked.

Address: IPTP_1588_ACC_PTP_Mask is 63FD_4000h base + 28164h offset = 63FF_C164h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	OVR1	OVT1	SYRE1	DRQRE1	GTXE1	PDRQRE1	PDRSRE1	OVR2	OVT2	SYRE2	DRQRE2	GTXE2	PDRQRE2	PDRSRE2	OVR3	OVT3
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	SYRE3	DRQRE3	GTXE3	PDRQRE3	PDRSRE3	OVR4	OVT4	SYRE4	DRQRE4	GTXE4	PDRQRE4	PDRSRE4				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPTP_1588_ACC_PTP_Mask field descriptions

Field	Description
0 OVR1	Overrun occurred in receiver time stamp register.
1 OVT1	Overrun occurred in transmitter time stamp register.
2 SYRE1	Sync frame has been received.
3 DRQRE1	Delay_req frame has been received.
4 GTXE1	(GLOBCOAL+1) PTP frames have been transmitted.
5 PDRQRE1	Pdelay_req frame has been received.

Table continues on the next page...

IPTP_1588_ACC_PTP_Mask field descriptions (continued)

Field	Description
6 PDRSRE1	Pdelay_resp frame has been received.
7 OVR2	RESERVED
8 OVT2	RESERVED
9 SYRE2	RESERVED
10 DRQRE2	RESERVED
11 GTXE2	RESERVED
12 PDRQRE2	RESERVED
13 PDRSRE2	RESERVED
14 OVR3	RESERVED
15 OVT3	RESERVED
16 SYRE3	RESERVED
17 DRQRE3	RESERVED
18 GTXE3	RESERVED
19 PDRQRE3	RESERVED
20 PDRSRE3	RESERVED
21 OVR4	RESERVED
22 OVT4	RESERVED
23 SYRE4	RESERVED
24 DRQRE4	RESERVED
25 GTXE4	RESERVED
26 PDRQRE4	RESERVED

Table continues on the next page...

IPTP_1588_ACC_PTP_Mask field descriptions (continued)

Field	Description
27 PDRSRE4	RESERVED
28–31 -	RESERVED

Chapter 45
Image Processing Unit (IPU)

45.1 Overview

The IPU v3M is planned to be a part of the video and graphics subsystem in an application processor. This is an enhanced version of the IPUv1.

The goal of the IPU is to provide comprehensive support for the flow of data from an image sensor and/or to a display device. This support covers all aspects of these activities:

- Connectivity to relevant devices - cameras, displays, graphics accelerators, TV encoders and decoders.
- Related image processing and manipulation: sensor image signal processing, display processing, image conversions, etc.
- Synchronization and control capabilities (to avoid tearing artifacts).

This integrative approach leads to several significant advantages:

- Automation: The involvement of the ARM platform in image management is minimized. In particular, display refresh/update and a camera preview (displaying the input from an image sensor) can be performed completely autonomously. The resulting benefits are reducing the overhead due to SW-HW synchronization, freeing the ARM platform to perform other tasks and reduced power consumption (when the ARM core is idle and can be powered down).
- Optimal data path: Access to system memory is minimized. In particular, significant processing can be performed on-the-fly while receiving data from an image sensor and/or sending data to a display. System memory is used essentially only when a change in pixel order or frame rate is needed. The resulting benefits are reduced load on the system bus and further reduction of power consumption.
- Resource sharing: Maximal HW reuse for different applications, resulting with the support of a wide range of requirements with minimal HW.

The HW reuse mentioned above is enabled by a sophisticated configurability of each HW block. This configurability also allows the support of a wide range of external devices, data formats and operation modes. The resulting flexibility is important also because the support requirements are evolving significantly, so expected future changes need to be anticipated and accounted for.

The following further principles guided the choice of support provided by the IPU:

- For key applications that DESERVE and NEED HW support (for acceleration or low power), provide the best support (leading to an optimal implementation).
- For additional applications that can benefit from the HW, consider cost vs. benefit of making minor modifications/extensions to support them.
- For all other relevant applications (to be supported by SW), verify that their support is not degraded.
- Whenever possible, let the operating system (and its windowing system) act as it would without the IPU.

45.2 Architecture

A simplified block diagram of the IPU is shown. The role of each block is described in IPU.

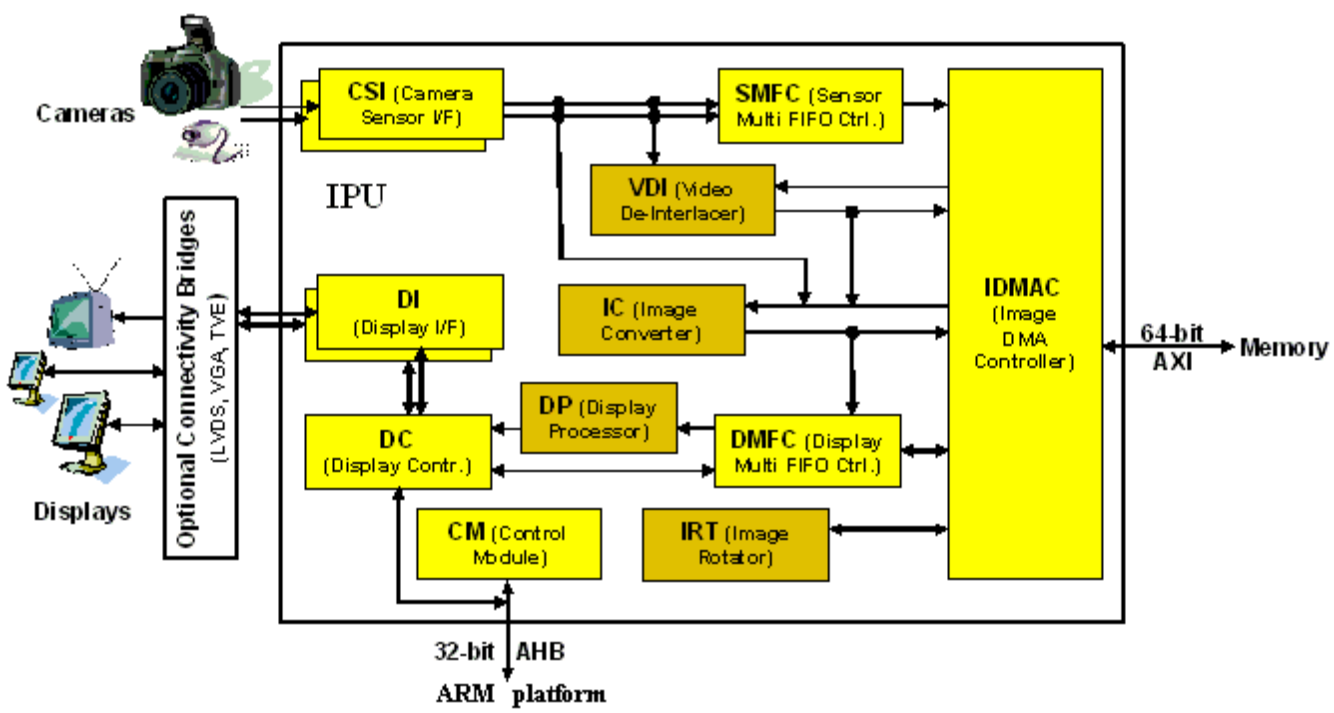


Figure 45-1. IPU Block Diagram

Table 45-1. IPU - Block Description

Block	Description
CSI - Camera Sensor Interface	Controls a camera port; provides interface to an image sensor or a related device. IPU includes 2 such blocks.
DI - Display Interface	Provides interface to displays, display controllers and related devices. IPU includes 2 such blocks.
DC - Display Controller	Controls the display ports.
DP - Display Processor	Performs the processing required for data sent to display.
IC - Image Converter	Performs resizing, color conversion/correction, combining with graphics, and horizontal inversion.
VDIC - Video De Interlacer	Performs video de interlacing (Interlaced -> progressive) or combining.
IRT - Image Rotator	Performs rotation (90 or 180 degrees) and inversion (vertical/horizontal).
IDMAC - Image DMA Controller	Controls the memory port; transfers data to/from system memory.
DMFC - Display Multi FIFO Controller	Controls FIFOs for IDMAC channels related to the display system.
CM - Control Module	Provides control and synchronization.

45.3 Features And Functionality

45.3.1 External Ports

IPU has the following ports:

- Two camera ports - each controlled by a CSI sub-block, providing a connection to image sensors and related devices.
- Two display ports - each controlled by a DI sub-block, providing a connection to displays and related devices.
- Memory port - AXI (AHB V3.0) master, controlled by the IDMAC - providing connection to the system memory.
- AHB-lite slave port, providing connection to the ARM Platform (and to any other master connected to the ARM's cross-bar switch).
- Additional ports for control and debug.

45.3.1.1 Camera Ports

The role of these ports is to receive input from image sensors (or TV decoders) and to provide support for time-sensitive control signals to the camera. (Non-time-sensitive controls; configuration, reset are performed by the ARM platform through I2C I/F or GPIO signals).

Each of the camera ports includes the following features:

- Direct connectivity to most relevant image sensors and to TV decoders.
- Interface types
  - Parallel interface
    - Up to 20-bit input data bus.
    - A single value in each cycle, except for special cases listed in the table below (comments column).
    - Programmable polarity.
- The data formats
  - Interleaved color components, up to 16 bits per value (component).
  - The supported formats are listed in the table below.

Table 45-2. Data Formats Supported By The Camera Port

Format	Resolution	On-The-Fly Processing	Direct path to memory	Comments
Bayer RGB	8 bits/value	No	8- or 16-bit values	
	10 bits/value	No	Written to the MSB of a 16-bit word	
	16 bits/value	Yes		
Full RGB or YUV 4:4:4	444/555 mode	Yes, starting with color extension to 8 bits/ sample	Yes	In parallel I/F: through an 8-bit or 16-bit bus
	565 mode			In parallel I/F: through an 8-bit or 16-bit bus
	8 bits/value (888 mode)	Yes	Yes	
	8-16 bits/value	No	8- or 16-bit components are written to the MSB of a 16-bit word 10 bits/ value can also be packed in a 32-bit word	
YUV 4:2:2 Component order: UY1VY2... or Y1UY2V...	8 bits/value	Yes	Yes	In parallel I/F: through an 8-bit bus (such as BT.656) or 16-bit bus (such as BT.1120)
	9-10 bits/value	No	Written to the MSB of a 16-bit word	In parallel I/F: through a 10-bit bus (such as BT.656) or 20-bit bus (such as BT.1120)
	16 bits/value	No	Written to the MSB of a 16-bit word	
Gray scale	8 bits/value	Yes	Yes	
	16 bits/value	No	Written to the MSB of a 16-bit word	

Table continues on the next page...

Table 45-2. Data Formats Supported By The Camera Port (continued)

Format	Resolution	On-The-Fly Processing	Direct path to memory	Comments
Generic data		No	Yes In a parallel I/F, if wider than 8 bits, each bus word is written to the MSB of a 16-bit word	May be used for any other formant, such as JPEG/MPEG4

- Scan order: progressive or interlaced data (expected only for YUV 4:2:2) is sent directly to system memory, where it can be read back for further processing.
- Frame size: up to 8192 x 4096 pixels
- Synchronization: video mode
  - The sensor is the master of the pixel clock (PIXCLK) & synchronization signals
  - Synchronization signals are received using either of the following methods:
    - Dedicated control signals -VSYNC, HSYNC - with programmable pulse width & polarity
    - Controls embedded in the data stream, following loosely the BT.656 protocol, with flexibility in code values and location.
- Synchronization : still image capture
  - The image capture is triggered by the ARM platform or by an external signal (such as a mechanical shutter).
  - Synchronized strobes are generated for up to 6 outputs - the sensor and camera peripherals (such as flash, mechanical shutter).
- Additional features
  - Frame rate reduction, by the periodic skipping of frames
    - The supported reduction ratios are: m:n, where m,n<=5
    - This is supported independently for the different destinations - IC, SMFC.
  - Window-of-interest selection
  - Pre-flash - for red-eye reduction and for measurements (such as focus) in low-light conditions

Several sensors can be connected to each of the CSIs. Simultaneous functionality (sending data) is supported as follows:

- Two sensors can send data independently, each through a different port.
- Only one of the (non-generic) streams can be transferred to theVDIC or IC for on-the-fly processing, while the others are sent directly to system memory.

The input rate supported by the camera port is as follows:

- Parallel interface:

  - Average: up to 135M values/sec Bayer - 90 M pixels/sec (e.g., 9M pixels @ 15 fps) YUV 4:2:2 - 45 M pixels/sec (e.g., 3M pixels @ 15 fps) YUV 4:4:4 or RGB - 30 M pixels/sec (e.g., 3M pixels @ 10 fps)
  - Peak: up to 180 values/sec (to account for up to 35% blanking intervals)
- When two or more sensors are used simultaneously, the total rate supported is as above.
- On-the-fly processing (as listed in the table above) may be restricted to a lower input rate. See Processing for further details.
- I/O pads may also be restricted to a lower input rate.

45.3.1.2 Display Ports

The role of these ports is to communicate with display devices, either directly or through a controller (such as a graphics accelerator) or a bridge (such as a TV encoder or an LVDS interface bridge).

45.3.1.2.1 Access Modes

Two access modes are supported.

Synchronous access

In this mode, the IPU transfers a two-dimensional block of pixels to the display device, in synchronization with the screen refresh cycle.

This mode has a dual role:

- For a RAM-less display or a TV screen, this mode is used to perform the screen refresh process from a display buffer in system memory.
- For a "smart" display, this mode is used to transfer a rectangular block of pixels to the display's screen and, in some cases, also to the display buffer
  - The transferred block may be only part of the screen (the rest of the screen being refreshed by the integrated controller, from the internal buffer). Moreover, a mask can be used to transfer to the display only parts of the block, such as a window partly hidden by other windows.
  - If the block is transferred only to the screen, the transfer rate must be equal to the refresh rate. If, however, the transfer is also to the display's memory, the rate can be reduced to the rate at which the input buffer is updated.

In all cases (including the last one), the IPU sends to the display all the synchronization signals controlling the screen refresh and the block transfer is synchronized with these signals. This synchronization means that tearing effects are avoided when using this mode.

Asynchronous Access:

This is the main mode used for communicating with an external display controller (possibly in a smart display or a graphics accelerator). In this mode, the IPU performs random access - read/write - to the memory and registers of the controller.

Two types of addressing methods are supported

- Generic linear addressing of pixels and generic data
- 2-dimensional (X/Y) addressing of pixels

The following access types are provided:

- Data transfer to the external device, after on-the-fly processing in the
- Data transfer (DMA) - read/write - between the host's system memory and the external device, through the IPU's memory port (controlled by the IDMAC), such as the transfer of a rectangular block of pixels (possibly full screen).
- Host access - read/write - to an external device, through the AHB-slave port
  - Access types
    - Direct access - emulating a directly-addressed access (see below) This includes burst access (incremental; up to 8 words/burst)
    - Low-level access - leaving to the host the explicit generation of the access protocol
  - The possible accessing modules include the ARM platform and the system DMA controller (as well as any other AHB master connected to the ARM's cross-bar switch).

Transfer of video/graphics data stream to controller's display buffer is performed using one of the first two modes above. Unlike in the synchronous mode, this process is not tightly-synchronized with the screen refresh cycle. However, a loose synchronization - to avoid tearing - is still possible: the appropriate timing for the transfer can be derived from the VSYNC signal of the screen refresh - either generated by the IPU's display controller or received from the external controller.

The asynchronous access requires the specification of an address. The display interface uses "indirect addressing", namely, there is no address bus, and the address, as well as control and configuration commands, are embedded in the data stream. The access procedure - including writing addresses and commands - is managed autonomously by the interface, in one of two ways:

- Automatic emulation of transparent access, following microcode ("access template") generated by the ARM Platform. This mechanism is very flexible, supporting a wide variety of devices.
- Streaming commands/addresses from a buffer stored (by the ARM Platform) in system memory.

Note that direct access requires the use of the first method - automatic emulation.

The Interface

The display interface is very flexible and supports a wide variety of devices from major manufacturers. The following interface types are provided (in each of the two display ports)

- Parallel video interface (for synchronous access) - up to 24-bit data bus.
  - Control protocol - follows Sharp HR and generic TFT definitions
  - Supports BT.656 (8-bit) and BT.1120 (16-bit) protocols
  - Supports HDTV standards SMPTE274 (1080i/p) and SMPTE296 (720p)
- A parallel bidirectional bus interface (for asynchronous access) - up to 32-bit data bus.
  - Compatible with MIPI-DBI standard.
  - Control protocol - either system-80 or system-68K The timing and polarity of the signals are programmable.
  - Byte-enable - optional, for a 16-bit interface
  - Burst access For direct access, the burst is determined by the corresponding signal in the AHB interface.
- A serial interface - 3-wire, 4-wire and 5-wire (two flavors) (for asynchronous access)

The supported formats for pixel data are

- RGB - color depth fully configurable; up to 8 bits/value (color component)
- YUV 4:2:2, 8 bits/value (for TV encoder)

In the parallel interfaces, the data bus has up to 32 bits. Non-trivial mapping of pixels to the bus is restricted to the 24 LSB's. This mapping is fully configurable and very flexible. In the serial interfaces, the data is mapped in the same way as in the parallel interfaces and then serialized.

The interface also supports "generic data". Such data is transferred - byte-by-byte, without modification - between the system memory and the display device (through a serial interface or 8/16-bit parallel interface). Non-conventional pixel formats can be supported by considering them as "generic data".

For the interface clock, there are the following options (independently for each port)

- Derived from the IPU internal clock (master mode)
- Provided by an external source (slave mode)

The transfer rate supported:

- For single port (for on-chip interfaces):

- 170 Mega-Accesses/Sec if the DI clock is derived from an external to IPU source (like another PLL)
- 180 Mega-Accesses/Sec if the DI clock is derived from the IPU clock (HSP_CLK)
- When off-chip interfaces are involved the rate may be limited by IO capabilities. Please refer to the device's data-sheet for exact numbers.

For synchronous access with one cycle/pixel, this enables, e.g (including 35% blanking intervals)

- 1080p (1920x1080) @ 60 fps
- WSXGA+ (1680x1050) @ 60 fps
- The combined rate for the two ports is up to 180 MP/sec

The interface includes the following additional features:

- Screen size: up to 4096 x 2048 pixels, programmable by software.
- Scan Order: progressive or interlaced
- Synchronization
  - Programmable horizontal and vertical synchronization output signals (for synchronous access)
  - Data enabling output signal
- Software contrast control using 8-bit programmable pulse-width modulation (PWM) Two dedicated PWM outputs are provided

Connecting To Display Devices

IPU allows the connectivity to multiple display devices. In particular, it supports the following setup:

- Primary LCD display; can be smart, dumb (RAM-less) or dual-port; may use the parallel interface or (through an integrated bridge) LVDS interface.
- Second LCD display; can be smart or dumb (RAM-less); may use parallel or serial interface or (through an integrated bridge) LVDS interface.
- TV Output: either digital parallel output, or (through an integrated TV encoder bridge) analog output.

Each of the above connections has independent settings - interface timing, access template, chip-select, etc.

Simultaneous functionality of the above devices is possible in each of the following ways:

- Two devices can be accessed (synchronously or asynchronously) independently, each through a different port: each using any of the available interfaces.

- Two devices can time-share asynchronous accesses through the legacy serial & parallel interfaces, using the CS signals.
- An asynchronous access can be performed during vertical blanking intervals of a synchronous access (screen refresh; to the same or other device).

The possibilities for simultaneous functionality by time-sharing the legacy interfaces in a single port are summarized in the following table.

Table 45-3. Simultaneous Functionality of Display Port By Time-Sharing Legacy Interfaces

	Second/Third Display Type		
Primary Display Type	**Smart Display Serial Interface**	**Smart Display Parallel Interface Asynchronous access**	**RAM-less display or TV screen**
Smart Display Parallel Interface Asynchronous access	Yes	Yes	Yes; access to the smart display is restricted to blanking intervals
Dual Port Smart Display Synchronous access	Yes	Yes, access to the secondary display is restricted to blanking intervals	Not Available
TFT RAM-less Display Or TV screen	Yes	Yes, access to the smart display is restricted to blanking intervals	Not Available
Graphics Accelerator	Yes	Yes, if the accelerator supports a chip select functionality	Not Available

45.3.1.3 Memory Port

The memory port is an AXI (AHB V3.0) master port, used to read/write data - typically two-dimensional blocks from/to system memory.

The interface supports the following features

- Clock rate up to 200 MHz (equal to the internal clock)
- 64-bit data bus
- The supported data formats are listed in the table below.

Table 45-4. Data Formats Supported By The Memory Interface

Format	Resolution	Input/Output	Comments
Non-interleaved YUV (in three separate buffers)	8 bits/value	both	4:4:4, 4:2:2, 4:2:0 formats
Partially-interleaved YUV (in two separate buffers)	8 bits/value	both	4:2:2, 4:2:0 formats Y buffer and UV buffer
Interleaved YUV (all color components in a single buffer)	8 bits/value	both	4:4:4 format (YUV...) 4:2:2 format (UY1VY2... or UY2VY1... or Y1UY2V...or Y2UY1V...)
Interleaved true color	8, 12,16, 18, 24, 32 bits/ pixel 0 - 8 bits per R/G/B/A value	both	Flexible component location A: translucency value (only in input)
Coded color (using a palette)	4,8 bits/pixel	input only	
Gray scale	8 bits/pixel	both	
	4 bits/pixel	input only	Transferred to display port
Generic data (Transparent M)	8 bits/unit	both	E.g.: From CSI; to DI; Compressed data to/from DP Translucency for combining

- The pixel formats are translated to/from a uniform internal format: RGBA/YUVA 8:8:8:8
- The supported ordering of bytes and pixels is little endian. For 4 bits/pixel, big endian is also supported.
- Addressing modes include:
  - Sequential access (to a contiguous memory buffer) - for generic data.
  - Raster-scan within a two-dimensional window of a video/display buffer - for both pixel and generic data.
  - Raster-scan of two-dimensional blocks within a two-dimensional window (for rotation of pixel data)
- Additional features
  - Scan order: progressive or interlaced Interlaced access is supported for fields which are stored either in separate memory buffers or with rows interleaved in a single buffer.
  - Reordered scan, implementing inversion and rotation.
    - Rotation and horizontal inversion - only when transferring two-dimensional blocks (to/from the IRT)
    - Vertical inversion - also in row-by-row raster-scan

- Scrolling
  - Applications Panning within a frame Frame scrolling
  - Not supported for non-interleaved and partially-interleaved formats
  - Resolution Vertical: single pixel Horizontal: 18 BPP - 4 pixels; 12, 4 BPP or YUV 422 - 2 pixels; other formats - one pixel
- Conditional read (for combining): fully-transparent or hidden pixels are not read.
  - This is supported, for graphics. by reading the transparency (alpha) from a separate buffer
- Input/output FIFOs (in the SMFC, DMFC and in the processing sub-blocks) - size adjusted to provide resilience for latency of up to 1500 cycles.

45.3.1.4 Processing

The IPU processes rectangular blocks of pixels. The processing is performed in these sub-blocks - VDIC, DP, IC and IRT.

(see the IPU block diagram and Table 45-1).

45.3.1.4.1 Processing flows

Several time-shared data flows are supported, as described in the following table.

Table 45-5. Time-Shared Data Flows through the IPU

Name	Number	Type	Flow	Target	Restrictions
Display Refresh/ Update	5 flows (at most two of them of type DS1)	DS1	Fmem -> DP -> Display	Synchronous Access (e.g. display refresh; controlled by the DI)	
		DS2	Fmem -> DP -> Display	Asynchronous Access (e.g. display update)	
		DS3	Fmem <-> Display	Generic Data Transfer	
	1 flow	DS4	ARM Platform<-> Display	Direct Access	
Video Playback	flows	PL1	Bmem -> VDIC -> IC -> Bmem -> IRT -> Fmem + DSx	Main option	
		PL2	Fmem -> IRT -> Bmem -> IC -> DP	Low power (branching to DSx, as a video plane)	Large enough window No other video flows
		PL3	Fmem -> VDIC -> IC -> DP	Low power (branching to DSx, as a video plane)	Interlaced source Large enough window No other video flows

Table continues on the next page...

Table 45-5. Time-Shared Data Flows through the IPU (continued)

Name	Number	Type	Flow	Target	Restrictions
Camera Pre-view	1 flow (VF2 may be used also as a playback flow)	VF1	Sensor -> IC -> Bmem -> IRT -> Fmem+DSx	main option	Single progressive input
		VF2	Sensor -> Fmem -> VDIC -> IC -> Bmem -> IRT -> Fmem+ DSx	two inputs and/or interlaced input	When the VDIC is used, one of the three input fields can go directly from the sensor to the VDIC. In that case the sensor output goes to the memory via the VDIC and not the SMFC
		VF3	Sensor -> IC -> DP	Low power Smart Display Tearing-less (branching to DS2, as a video plane)	Single progressive input Refresh rate = 2x sensor frame rate Large enough window No other video flows
		VF4	Sensor -> IC -> Fmem + DS1	Low power RAM-less Display Single Display Buffer (in internal memory) Tearing-less	Single progressive input Refresh rate = 2x sensor frame rate Large enough window No other video flows
Video Record	1 flow	RCx	IC -> Bmem -> IRT -> Fmem	(branching from VFx)	
Graphic Overlays	2 flows	GF1	Fmem -> IC	(combining with the main flow	
	2 flow	GF2	Fmem -> DP		

Comments

- System memory usage - legend
  - Fmem: frame double-buffer (page-flip) in system memory (typically external)
  - Bmem: two possibilities
    - A frame double buffer, as above
    - A band (4-256 rows) double-buffer (page-flip) in system memory (could be internal)
  - Direct arrow between two processing stages represents an internal pipeline
- Time-sharing
  - IC can time-share tightly three flows: one VFx, one RCx and one PLx (with independent processing parameters)
  - DP can time-share one DS1 flow and a one DS2 flow (each with different destinations and independent processing parameters)

    - Direct access to display (DS4) time-shares tightly the display port with other active DSx flows.
    - Other time-sharing (between PLx; between DS2&DS3; in IRT) is frame-by-frame
- Any of the processing stages in the above flows can be skipped.
- Triggering and synchronization
    - Flow segments starting from a sensor are triggered and synchronized by input from the sensor
    - Flow segments ending with display refresh are triggered and synchronized by the refresh control mechanism in the DI.
    - Flow segments starting and ending in system memory, are triggered either by the double buffering mechanism or by explicit configuration and are processed continuously without delay, at a rate determined by the available resources. These flow segments have a lower priority than the sensor/display-driven flows.

The functionality of each of the processing blocks is described below.

45.3.1.4.2 Display Processor (DP)

The Display Processor performs all the processing required for data sent to a display.

- Input: from the IC and/or from system memory
    - Order: rows, progressive or interlaced
    - Format: YUVA/RGBA, non-decimated, 8 bits/value
- Processing chain
    - Combining 2 video/graphics planes
    - Overlaying a simple HW cursor 32 x 32 pixels, uniform color; may be combined logically with the full plane.
    - Color conversion/correction - linear (multiplicative & additive) Programmable; including:
        - YUV <-> RGB, YUV<->YUV conversions where YUV stands for any one of the color formats defined in the MPEG-4 standard
        - Adjustments: brightness, contrast, color saturation, etc.
        - Special effects: gray-scale, color inversion, sephia, blue-tone, etc.
        - Color-preserving clipping, for gamut mapping
        - Hue-preserving gamut mapping - for minimal color distortion
        - Applied to the output of combining or to one of the inputs
    - Gamma correction and contrast stretching - programmable piecewise-linear map
- Output: to display (through the DC)
    - Rate: up to 180M pixels/sec (e.g., WSXGA (1680x10508) @ 75 fps + 35% blanking intervals)
    - Format: YUV/RGB, non-decimated, 8 bits/value

The DP processes a single data flow at any given time, but supports up to three data flows by time sharing.

- A Primary flow:
  - The input is loaded periodically using a timer (e.g. for a synchronous access)
  - Optionally, frames are skipped if the content has not changed (as appropriate for a smart display)
- Two secondary flows:
  - Asynchronous; processed when the DP is not needed for the primary flow (during blanking intervals or when a primary frame is skipped).
  - The two secondary flows are switched frame-by-frame.

45.3.1.5 Video De-Interlacer or Combiner (VDIC)Video De-Interlacer (VDI)

The Video De-Interlacer and Combiner has two operation modes

- De-interlacing: converts an interlaced video stream to progressive order.
- Combining: combines two video/graphics planes and a background color

45.3.1.5.1 De interlacing in the VDIC

The Video De-Interlacer converts an interlaced video stream to progressive order, using a high-quality 3-field motion-adaptive filter.

- Video source - SDTV: 480i30 (720x480 @ 30 fps) or 576i25 (720x576 @ 25 fps) and HDTV: 1080i30 (1920x1080 @ 30 fps)
- Input: - three consecutive fields
  - Source
    - The most recent field may come from the CSI or from system memory
    - The other two fields are read from memory
  - Field size: up to 968x1024 pixels (may be a vertical stripe of a wider field; e.g. 1920 pixels)
  - Pixel format: YUV 4:2:2/4:2:0, 8 bits/value
- Output: progressive frame
  - Destination: to system memory or to the Image Converter.
  - Frame size: up to 968X2048 pixels
  - Rate: up to 180 MP/sec (e.g., 1920x1080 @ 85 fps)
  - Format: same as input format

The de-interlacing is performed using a high-quality 3-field filter which is motion adaptive:

- For slow motion - retains the full resolution (of both top and bottom fields)
- For fast motion - prevents motion artifacts

The VDIC supports a single video stream at any given time.

45.3.1.5.2 Combining in the VDIC

- Input for combining: two progressive video/graphics planes
  - Source: system memory
  - Plane size: up to 1920x1200 pixels.
  - Pixel format: RGB/YUV 4:2:2, 8 bits/value

In this mode:

- The two input planes are read from system memory, using the previous field and next field input FIFOs.
- Their relative height, up/down is configurable
- Each of them may cover only part of the output frame. For the remaining part of the frame, i.MX 6Dual/6Quad uses the following values:
  - Down plane: a 24-bit background color, stored in an internal register
  - Up plane: a transparent pixel
- The combining method is identical to the one in the DP, IC.

45.3.1.5.3 Image Converter (IC)

The Image Converter performs various operations on a video stream.

- Input: from sensor or from system memory
  - Frame size: up to 4096x4096 pixels
  - Rate: up to150M pixels/sec (e.g.,4 MP @ 30 fps + 35% blanking intervals)
  - Order: rows, progressive. If resizing is not used, interlaced order is also acceptable.
  - Pixel format: YUV/RGB, 8 bits/value
- Processing chain:
  - Resizing
    - Fully flexible resizing ratio Maximal downsizing ratio: 16:1 Subject to this limitation, any N->M resizing can be performed
    - Independent horizontal and vertical resizing ratios.
  - Color conversion/correction - linear (multiplicative & additive) Programmable; including:
    - YUV <-> RGB, YUV<->YUV conversions where YUV stands for any one of the color formats defined in the MPEG-4 standard
    - Adjustments: brightness, contrast, color saturation...
    - Special effects: gray-scale, color inversion, sephia, blue-t[illegible]

    - Combining with a graphics plane (e.g. application-specific overlay)
    - Horizontal inversion
- Output: to system memory or (for a single active flow) to a display device (through the DP).
    - Frame size: up to 1024x1024 pixels
    - Rate: up to75Mpixels/sec (e.g.,1920x1080 @ 30 fps)
    - Order: rows, progressive. If resizing is not used, interlaced order is also acceptable.
    - Format: YUV/RGB, 8 bits/value

The IC supports three time-shared data flows: record, camera preview and playback (the first two share a common input).

45.3.1.5.4 Image Rotator (IRT)

- Input/output: from/to system memory
    - Rate: up to 75M pixels/sec
    - Order: raster scan of 8x8-pixel blocks
    - Format: YUV/RGB, non-decimated, 8 bits/value
- Transformation: combination of the following
    - 90-degree rotation
    - Horizontal inversion
    - Vertical inversion

45.3.1.6 Automatic Procedures

The IPU is equipped with powerful control and synchronization capabilities to perform its tasks with minimal involvement of the ARM Core and minimal use of memory.

In particular, it includes:

- An integrated DMA controller with an AXI master port, allowing autonomous access to the system memory.
- An integrated display controller, performing screen refresh of a RAM-less display.
- A page-flip double buffering mechanism, synchronizing read and write access to system memory, to prevent tearing effects.
- A double/triple buffer synchronization mechanism with a video/graphics source.
- Internal synchronization, e.g., between input from sensor and output to display

As a result, in most cases, the ARM platform is involved only when it also performs part of the processing (e.g. video coding). In particular, the following procedures are performed by the IPU completely autonomously:

- Screen refresh for RAM-less displays
- Update of the ("partial plane") display buffer used for screen refresh (located either in system memory or in an external display controller, e.g. of a smart display or graphics accelerator), when the content is generated in a different ("full plane") buffer.

Typically, there are extended periods of times in which there is no other activity in the system. The ARM platform, being idle, can be put to a low-power mode, reducing the power consumption and extending significantly the battery life.

The IPU supports several techniques to reduce further the power consumption of the display system:

- Dynamically optimized screen refresh rate (see Screen Refresh below)
- Optimized update of the display buffer (see Update Of The Display Buffer below)
- Dynamic backlight control, with low-light compensation by image enhancement

Further features and capabilities of the automatic procedures are outlined below

45.3.1.6.1 Screen Refresh

- The refresh rate may vary within a predefined range. Within this range, the rate is dynamically adjusted to the content update rate.
- An indication about the availability of new content is obtained as follows:
  - If the page-flip double buffering is used, the mechanism provides this indication
  - If only a single buffer is used (and incrementally updated), the IPU can receive an indication of a modification, either through a dedicated external "snooping" signal (e.g. from the EXTMC), or from the ARM platform (by setting an internal flag).
- The IPU counts the refresh cycles: the total and those with new content. The ARM platform can use these counters to optimize display management (e.g. switching display buffer compression on/off). The counters are reset by the ARM platform.
- The transferred data may be processed on the way, using the IC and DP.

45.3.1.6.2 Update Of The Display Buffer

- Conditional update The IPU can receive an external "snooping" signal indicating a modification of the full plane buffer (as during screen refresh above). It monitors the signal and, upon detection, it performs one of the following:
  - Performs an update, without any SW intervention
  - Interrupts the ARM core, that can initiate some more involved procedure (e.g. selective update)
- Automatic display of a changing image (animation) or moving image (scrolling) This is implemented by reading frames (from a full plane buffer) with incremental offset.

When the IPU reaches the last programmed frame, it can perform one of the following:
    - Return to the first frame, without any SW intervention
    - Interrupt the ARM platform, to generate the next content.
- The timing of the update can be adjusted to avoid tearing.
- The transferred data may be processed on the way, using the IC and DP

45.3.1.6.3 Camera Preview

- Tearing artifacts can be prevented by (automatic) page-flip double buffering in system memory
- Alternatively, the video stream from an image sensor can be sent directly to the display buffer used for screen refresh. The significance of this option is that only a single frame buffer is needed (and not two). This buffer may be located either in system memory or in an external display controller.
    - This option is useful, e.g., in a low frame rate, when tearing is not visible.
    - When tearing must be prevented, the refresh cycle in the display can be synchronized with the timing signals from the sensor (two refresh cycles for each input frame): the IPU receives a VSYNC signal from the sensor and generates from it synchronization signals for the display.

45.4 Functional Description

45.4.1 IPU detailed block diagram

The following figure is the IPU top level block diagram.

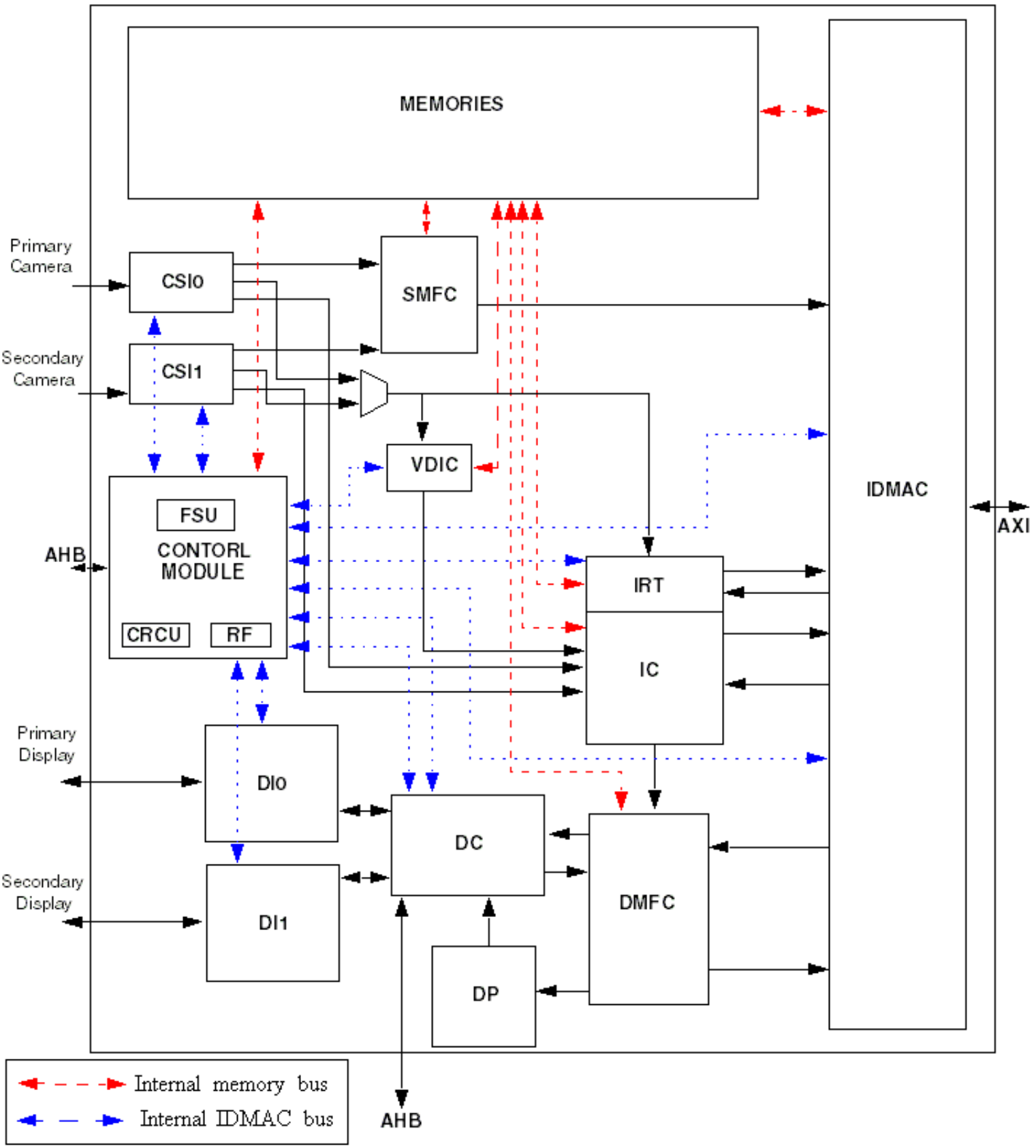


Figure 45-2. IPU Detailed Block Diagram

45.4.2 Image DMA Controller (IDMAC)

The following diagram is the IDMAC's block diagram.

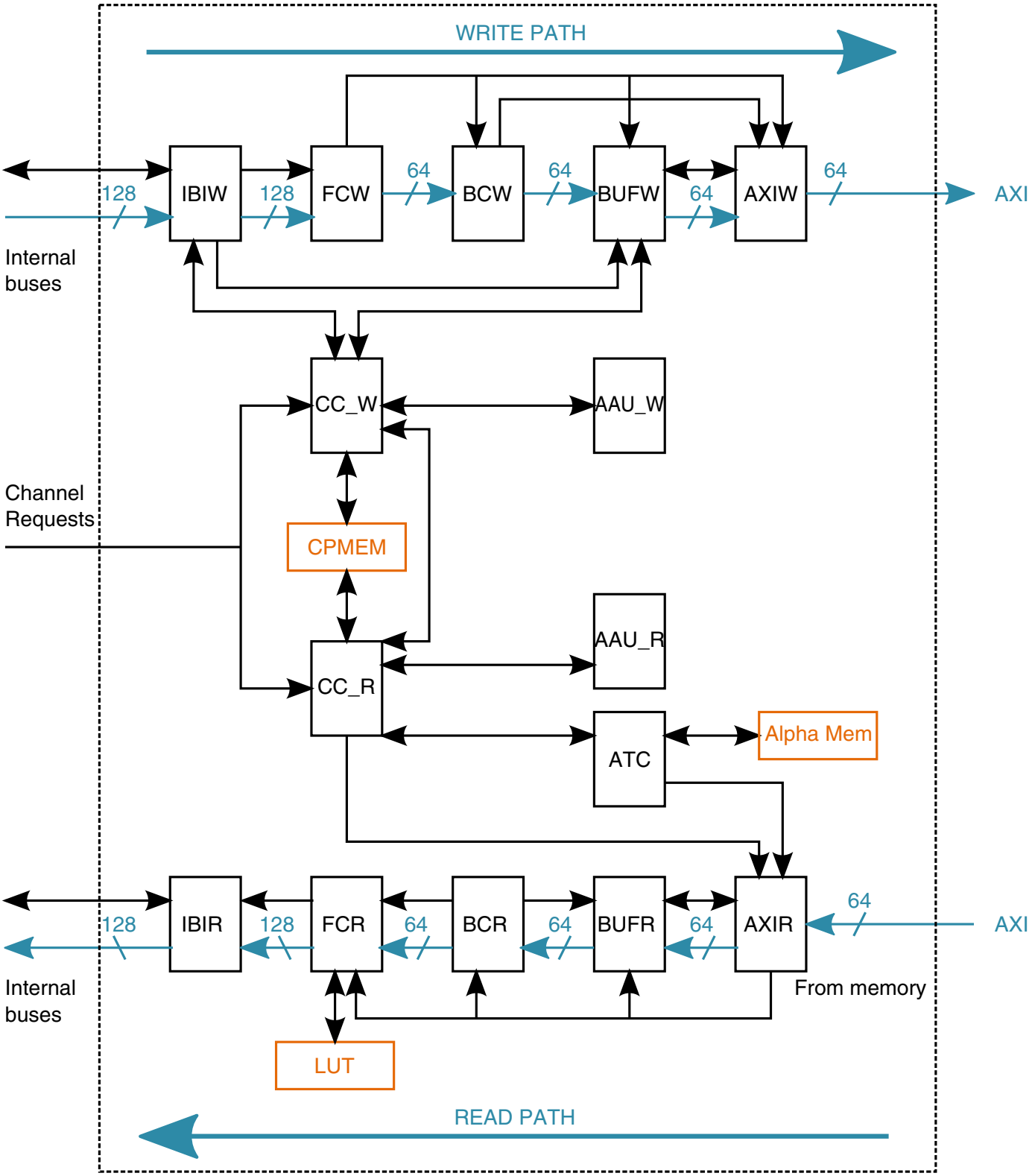


Figure 45-3. IDMAC Block Diagram

The following table describes the IDMAC's sub-block glossary

Table 45-6. IDMAC's sub modules glossary

Sub Module	Description
IBIW	Internal Bus Interface Write
IBIR	Internal Bus Interface Read
FCW	Format Converter Write
FCR	Format Converter Read
BCW	Buffer Controller Write
BCR	Buffer Controller Read
BUFW	Buffer Write
BUFR	Buffer Read
AXIW	AXI Write
AXIR	AXI Read
CC_W	Channel Control Write
CC_R	Channel Control Read
AAU_W	Address Arithmetic Unit Write
AAU_R	Address Arithmetic Unit Read
ATC	Alpha Transparency Controller
LUT	Look up table
CPMEM	Channel Parameter Memory

45.4.2.1 IDMAC's channels

The table below summarizes the IDMAC's channels.

Enabling a channel is done via the channel's corresponding IDMAC_CH_EN bit.

Table 45-7. IDMAC DMA channels list

Channel #	Source	Destination	Alternate Destination	Purpose	Data type
0	CSI (via SMFC)	Fmem		VF2 - Bayer; BPP>8; JPEG;	Generic or Pixel
1	CSI (via SMFC)	Fmem		VF2 - Bayer; BPP>8; JPEG;	Generic or Pixel
2	CSI (via SMFC)	Fmem		VF2 - Bayer; BPP>8; JPEG;	Generic or Pixel
3	CSI (via SMFC)	Fmem		VF2 - Bayer; BPP>8; JPEG;	Generic or Pixel
5	VDIC	Bmem	IC	VF1/VF2	Pixel

Table continues on the next page...

Table 45-7. IDMAC DMA channels list (continued)

Channel #	Source	Destination	Alternate Destination	Purpose	Data type
8	Fmem	VDIC		Previous field	Pixel
9	Fmem	VDIC		Current field	Pixel
10	Fmem	VDIC		Next field	Pixel
11	Bmem	IC		video plane for post processing task	Pixel
12	Bmem	IC		video plane for PrP tasks (view finder or encoding)	Pixel
13	VDIC	Fmem		Recent field from CSI	Pixel
14	Fmem	IC		graphics plane for PrP task (view finder or encoding)	Pixel
15	Fmem	IC		graphics plane for post processing task	Pixel
16	Reserved				
17	Fmem	IC		Transparency (alpha for channel 14)	Generic
18	Fmem	IC		Transparency (alpha for channel 15)	Generic
19	Fmem	VDIC		Transparency (alpha for channel 25)	Generic
20	IC	Bmem		Preprocessing data from IC (encoding task) to memory	Pixel
21	IC	Bmem	DMFC	Preprocessing data from IC (viewfinder task) to memory; This channel can be configured to send the data directly to the DMFC. This is done by programming the IC_DMFC_SEL bit.	Pixel
22	IC	Bmem		Postprocessing data from IC to memory	Pixel
23	Fmem	DP		DP primary flow - main plane	Pixel
24	Fmem	DP		DP secondary flow - main plane	Pixel
25	Fmem	VDIC		Plane #1 of the VDIC for combining	pixel
26	Fmem	VDIC		Plane #3 of the VDIC for combining	pixel
27	Fmem	DP		DP primary flow - auxiliary plane	Pixel
28	Fmem	DC		DC channel for both sync and async flows	Pixel
29	Fmem	DP		DP secondary flow - auxiliary plane	Pixel
30	Reserved				
31	Fmem	DP		Transparency (alpha for channel 27)	Generic
32	Reserved				
33	Fmem	DP		Transparency (alpha for channel 29)	Generic
34	Reserved				
35	Reserved				
36	Reserved				
37	Reserved				
38	Reserved				
39	Reserved				

Table continues on the next page...

Table 45-7. IDMAC DMA channels list (continued)

Channel #	Source	Destination	Alternate Destination	Purpose	Data type
40	DC	Fmem		DC read channel	Generic
41	Fmem	DC		DC async flow	Generic
42	Fmem	DC		DC command stream	Generic
43	Fmem	DC		DC command stream	Generic
44	Fmem	DC		DC output mask	Generic
45	Bmem	IRT		Rotation for post Encoding task	Pixel
46	Bmem	IRT		Rotation for viewfinder task	Pixel
47	Bmem	IRT		Rotation for post processing task	Pixel
48	IRT	Bmem		Rotation for Encoding task	Pixel
49	IRT	Bmem		Rotation for viewfinder task	Pixel
50	IRT	Bmem		Rotation for post processing task	Pixel
51	Fmem	DP		Transparency (alpha for channel 23)	Generic
52	Fmem	DP		Transparency (alpha for channel 24)	Generic
53-63	Reserved				

45.4.2.2 IBIW & IBIR - Internal bus interface for write and read

The Internal Bus Interface handles the internal IPU protocol communicating between the IDMAC and the IPU's sub modules. The IBIR handles channels that perform read from external memory. The IBIW handles channels that perform write accesses to external memory.

45.4.2.3 FCW & FCR - Format converter write and read

The format converter performs packing ("write direction") / unpacking ("read" direction) of pixels with programmable position and width of color components, decoding 4- or 8-bits coded pixels according to a loaded look-up table, panning of an image read from the system memory according to a panning offset (start pixel address).

The format converter supports formats with a pixel width of 4, 8, 12, 16, 18, 24 or 32 bits. The format converter unit handles two pixels simultaneously.

The IPU sub modules can handle only the formats, presented below. Each component is 8 bit:

Figure 45-4. IPU internal pixel formats

The pixel can be stored in the memory in the formats presented below. (R/G/B means that this could component can be R or G or B; A is the location of the alpha component).

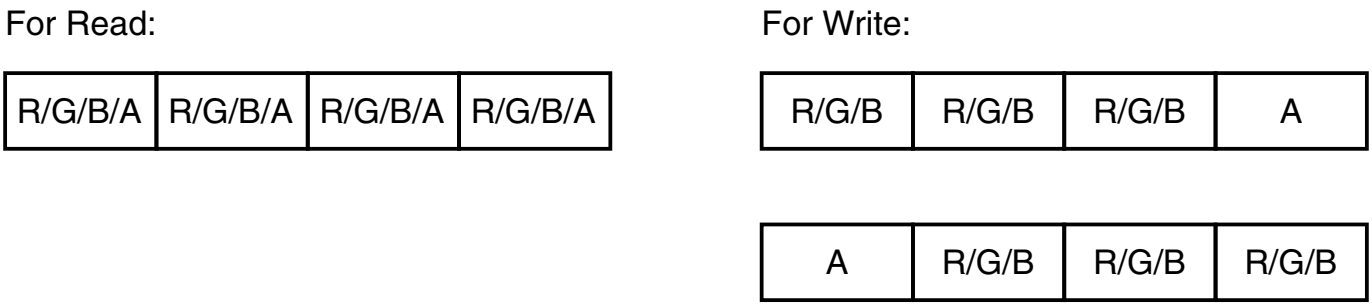


Figure 45-5. IPU external pixel formats

Formatting parameters are written in the channel parameter memory (CPMEM - Channel Parameter Memory). The following parameters are used:

- Offset OFS0 between MSB position of the color component 0 and MSB position of packed pixel. The color component 0 occupies the most significant bits of the unpacked pixel (mostly this is the R component). The OFS0 range is from 0 to 31.
- Color component 0 width WID0 minus 1.
- Offset OFS1 between MSB position of the color component 1 and MSB position of packed pixel. The color component 1 occupies the middle left bits of the unpacked pixel (mostly this is the G component). The OFS1 range is from 0 to 31.
- Color component 1 width WID1 minus 1.
- Offset OFS2 between MSB position of the color component 2 and MSB position of packed pixel. The color component 2 occupies the middle right bits of the unpacked pixel (mostly this is the B component). The OFS2 range is from 0 to 31.
- Color component 2 width WID2 minus 1.
- Offset OFS3 between MSB position of the color component 3 and MSB position of packed pixel. The color component 3 occupies the least significant bits of the

unpacked pixel (mostly this is the A component). The OFS3 range is from 0 to 31. For write specified DMA channels, the OFS3 value is set to 24 or 0 bits.

- In cases of read with separate alpha (alpha is located in a separate buffer in the system's memory than the pixel data), the alpha component size is defined according to WID3.

The figures below show examples of data packing and unpacking.

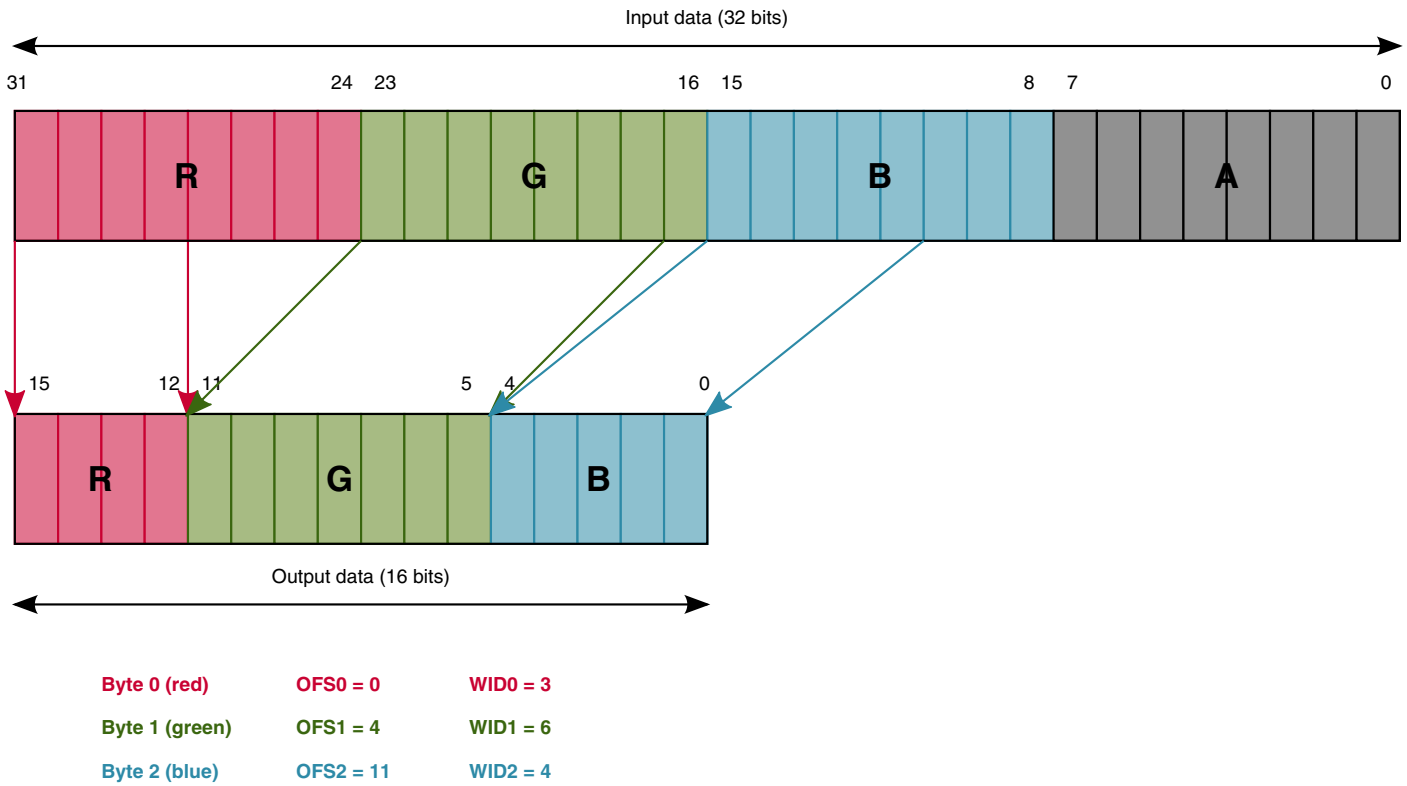


Figure 45-6. Data Packing Example

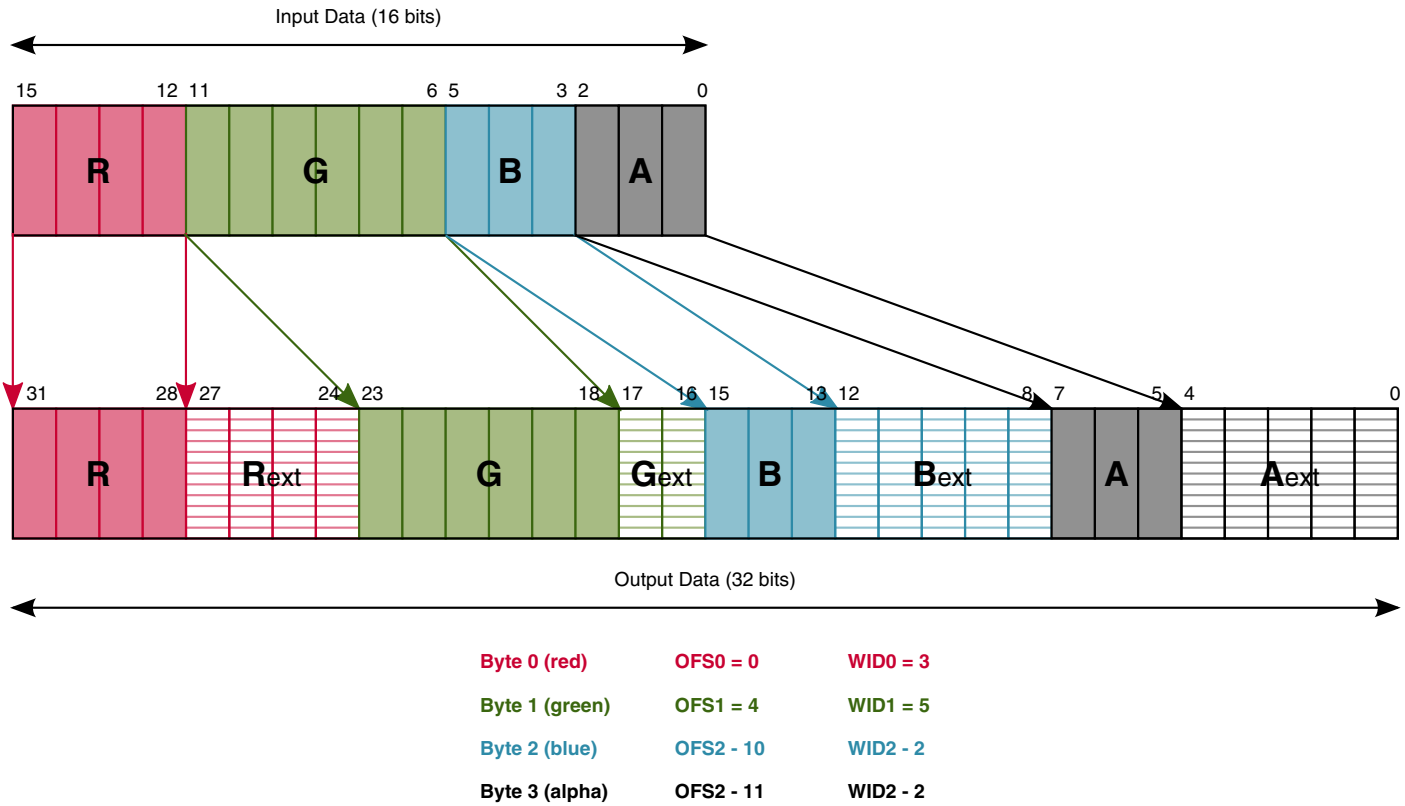


Figure 45-7. Data Unpacking Example

If read data has the coded format, it is decoded via the look-up table. The Look-Up Table Memory (LUT- Look Up Table) must be loaded at the IDMAC initialization step. The LUT output format must match the IPU internal format RGBA 8888 where R is placed in MSB and A is placed in LSB. The A field is used only for graphics data.

45.4.2.4 Buffering units

The buffering units (BUFW, BUFR) are used to store the data consisting of different coded color components.

- On write transactions (BUFW), before writing to memory & after the format has been coded.
- On read transactions (BUFR), after reading from memory & before the format has been decoded.

Each buffering unit includes 4 X 64 bytes buffers. Each buffer, for each direction, can handle any kind of color component (interleaved - Y / partial interleaved - Y, UV / non interleaved - Y, U, V).

The Write buffers are controlled by the buffer controller for write (BCW) and AXIW units.

The Read buffers are controlled by the buffer controller for read (BCR) and AXIR units.

45.4.2.4.1 Handling real time channels

The memory controller connected to the AXI bus of the IPU can use the AXI ID associated with each burst in order to distinguish between real time and non real time channels.

In order to do that, the user has to set the channel's ID according to the settings in the memory controller and set the priority of the channel according to its nature. The buffer controller (BCW/BCR) holds all the pending requests that won the arbitration. However, as the memory controller can distinguish between the real time channels and non real time channels within the IPU, there could be a situation where the real time requests are blocked as the IPU's queue is filled with non real time requests. To avoid that, the user can limit the number of non-real time requests in the queue.

The queue for read requests can handle up to 8 requests. The queue for write requests can handle up to 6 requests. The user can limit the number of non real time requests by setting the USED_BUFS_MAX_W for write requests and USED_BUFS_MAX_R for read requests. The feature that limits the number of requests is enabled by setting the USED_BUFS_EN_R bit for the read requests and USED_BUFS_EN_W for the write requests.

45.4.2.5 AXIW - AXI Write and AXIR - AXI Read

The AXI Master Interfaces are responsible for data transfer from/to the system memory. The Interface supports only 64-bits burst accesses of 1-8 words, with nonalignment of a byte resolution.

2 separate & independent AXI masters are used for "read" (AXIR) & "write" (AXIW), each can be programmed (via CPMEM) with 4 different IDs to support out-of-order accesses within bursts.

45.4.2.6 CC_W & CC_R - Channel Control Write and Read

The Channel Control unit is the main control unit of the IDMAC.

- It arbitrates the channels according to the priority.
- Controls the address arithmetic unit.

- Functions as a memory interface to the CPMEM. It reads the parameters from the CPMEM, prepares the controls accordingly and writes back updated parameters to the CPMEM.
- The read unit provides the parameters to the IBIR unit
- The write unit provides the parameters to the AXIW unit

The CC calculates all the parameters related to the access except the address and BS (burst size), which are calculated at the AAU.

The priority is set according to:

- The channel's corresponding bit in the IDMAC_CH_PRI_1 & IDMAC_CH_PRI_2 registers.
- The watermark signal generated from the sub module. The watermark signal is ignored unless the channel's corresponding IDMAC_WM_EN bit is set.
- Special priority for alpha channels

A priority value is calculated for each of the enabled channels according to the above conditions. Then, the CC unit selects between the channels with the same priority value in a round-robin fashion.

Table 45-8. Calculated priority value

alpha channel	Channel's priority bit	watermark signal	Priority Value
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	1
1	0	1	2
1	1	0	3
1	1	1	4

45.4.2.6.1 Locking the arbitration and reordering the AXI bursts

The performance of the overall system can be improved by sending AXI bursts of consecutive addresses. This can be done by reordering the AXI bursts in a way that accesses that belong to the same channel will be sent one after the other. This can be done by controlling the IDMAC_LOCK_# bits of the corresponding channel.

An IDMAC request that won the arbitration will be served for the next bursts according to the settings of the IDMAC_LOCK_# bits. The block that issued the request (DMFC on IPU) will assert the request only if it has enough room in its FIFO to accept the number

of bursts defined by the IDMAC_LOCK_# bits. IPU provides this capability to channels that serve real time screen refresh to synchronous display (23,27,28). In addition, it provides this capability to channels that may generate very short AXI bursts (IC and IRT)

45.4.2.7 AAU_W & AAU_R- Address Arithmetic Unit for Write and Read

The AAU_R & AAU_W units calculate the address in the system memory to be accessed by the These units also calculate the burst size (BS). The address calculation is done according to parameters stored in the CPMEM.

The following main addressing parameters are used:

- XB-Horizontal pixel position in frame
- YB-Vertical pixel position in frame
- SL-Stride line minus 1 (gap in bytes between two pixels in the same column in two consecutive rows).
- SX-Horizontal pixel scrolling offset
- SY-Vertical pixel scrolling offset
- EBA-Frame buffer base address in bytes (there are two such parameters to support double buffering)
- BPP-Bits per pixel
- FW-Frame width minus 1
- FH-Frame height minus 1

Relations between the addressing parameters and image frame are shown in the table below.

The system memory address in bytes is calculated as:

$$ADDR = EBA + (XB + SX) \cdot BPP + (YB + SY) \cdot (SL + 1)$$

with $0 < XB =< FW$ and $0 < YB =< FH$.

For non-interleaved formats the 4 LSB bits of SX are defined according to the IOX parameter.

When double buffering is used, the EBA0 is the base address of the buffer 0 and the EBA1 is the base address of the buffer 1. The IPU_CHA_CUR_BUF Register is a status register. It contains 1-bit pointers to the current working buffers for all IPU DMA channels. The IPU automatically toggles a pointer after completion of the current buffer processing. If the ARM platform is a data source for specific double-buffered channel, it should check this status bit in order to know what is the IPU current buffer. The ARM platform is allowed to write to the buffer only when a working DMA channel does not

use it. After the ARM platform has been fill the buffer, it has to set the corresponding bit in the IPU_CHA_BUF0_RDY and IPU_CHA_BUF1_RDY Registers. If needed, the ARM platform can only clear the pointer by writing 1 but not set it.

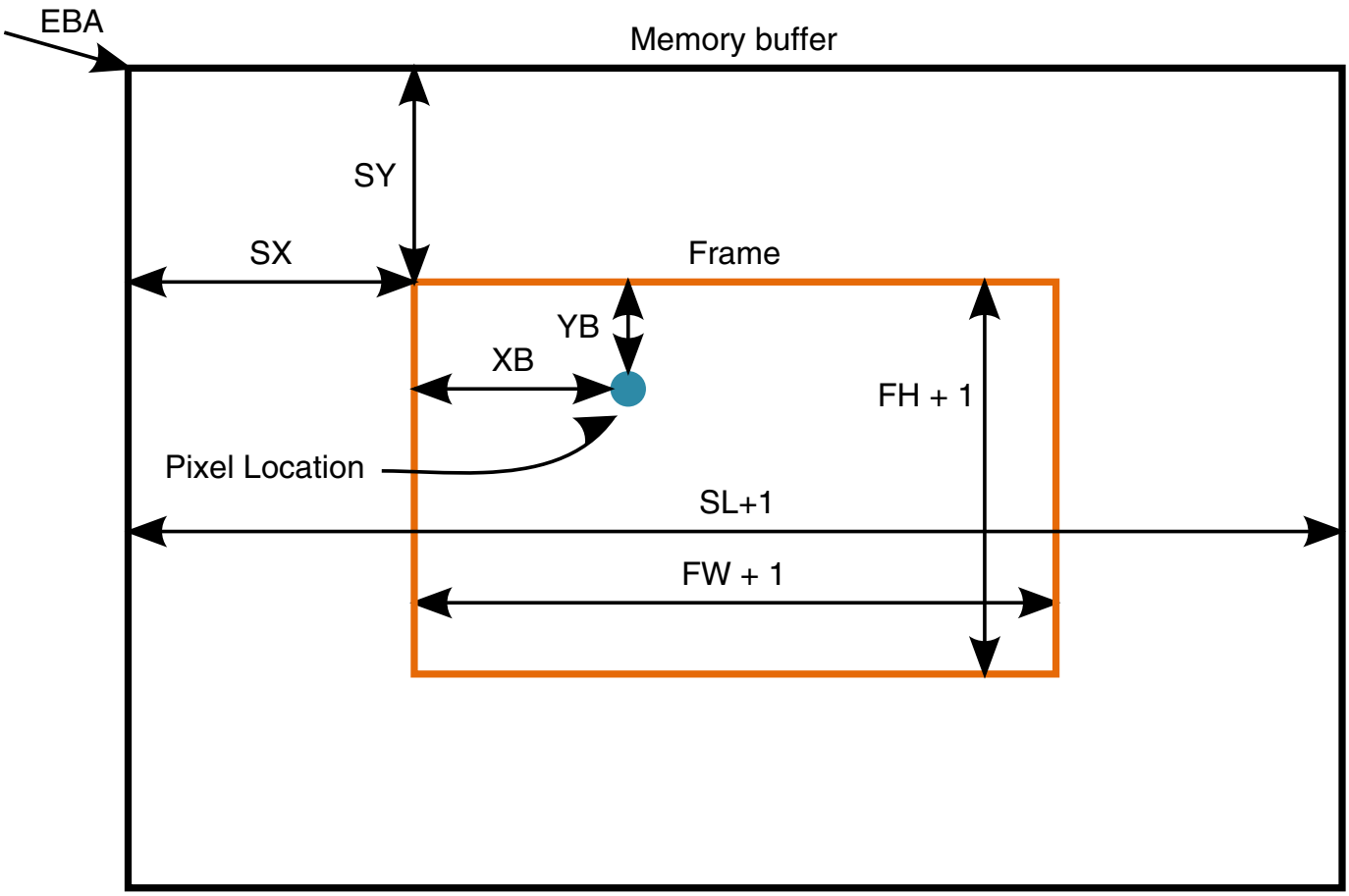


Figure 45-8. Addressing Parameters and Image Frame

The XB and YB coordinates are calculated according to addressing mode. There are two addressing modes:

- 2D mode
- Block mode

In 2D mode the pixel data is transferred to the memory row-by-row. There are two ways to use 2D mode: start from YB = 0 and finish at YB =< FH (YB is incremented) or start from YB = FH and finish at YB =< 0 (YB is decremented). The second option provides vertical flip of the image.

In block mode the frame is divided into blocks. This is needed for rotation or post-filtering, where the order used for data transfers is block-by-block. The order of the block transfer is according to the VF,HF and ROT bits in the IDMAC Channel Parameter

Memory. The order within the block is row-by-row where the block size is limited by the block width (BW) and block height (BH) parameters. The BW and BH parameters are set by IC rotation section and cannot be configured through the Channel Parameter Memory.

The Channel Control is responsible for the address calculation flow. It takes channel parameters from the Channel Parameter Memory, updates them and controls the Address Arithmetic Unit.

45.4.2.7.1 Scrolling support

Automatic display of a changing image (animation) or moving image (scrolling) is implemented by reading frames (from a background buffer) with incremental offset. Enabling the scrolling feature is done by setting the channel's corresponding SCE bit.

The scrolling step is controlled by channel's corresponding SDX and SDY parameters, and the scrolling direction is defined by the channel's corresponding SDRX and SDRY parameters. The maximum number of scrolled frames to be read is defined by the channel's corresponding SM parameter.

When the last programmed frame is reached (IDMAC's internal counter reached SM), IDMAC can perform one of the following (controlled by the SCC bit):

- Return to the first frame, without any SW intervention. The return point is defined by SX0 and SY0 parameters.
- Interrupt the ARM platform, to generate the next content.

45.4.2.8 ATC - Alpha Transparency Controller

The Alpha transparency controller (ATC) handles the alpha buffers on the external memory for cases where the pixel data and the alpha data are located on separate buffers (separate alpha mode). In that case the IDMAC reads the alpha data and the pixel data, merge them together and provides a pixel that includes the alpha information to the relevant sub module.

The ATC's main functions are:

- Generates requests for alpha channels, following a request to pixel data from the module.
- Maintain an internal alpha memory buffer for each channel.
- When there isn't enough alphas in the memory the ATC blocks the corresponding pixel channels.
- When there is a request of pixel channel the ATC load and accumulate its alphas in a register.

- ATC memory controller can manage 8 channels of alphas.
- ATC supports synchronous new frame before end of frame errors.

In order to configure the channel to use separate alpha the channel's corresponding IDMAC_SEP_AL bit should be set in addition to the ALU bit in the CPMEM of the corresponding channel.

The following figure is the ATC's block diagram.

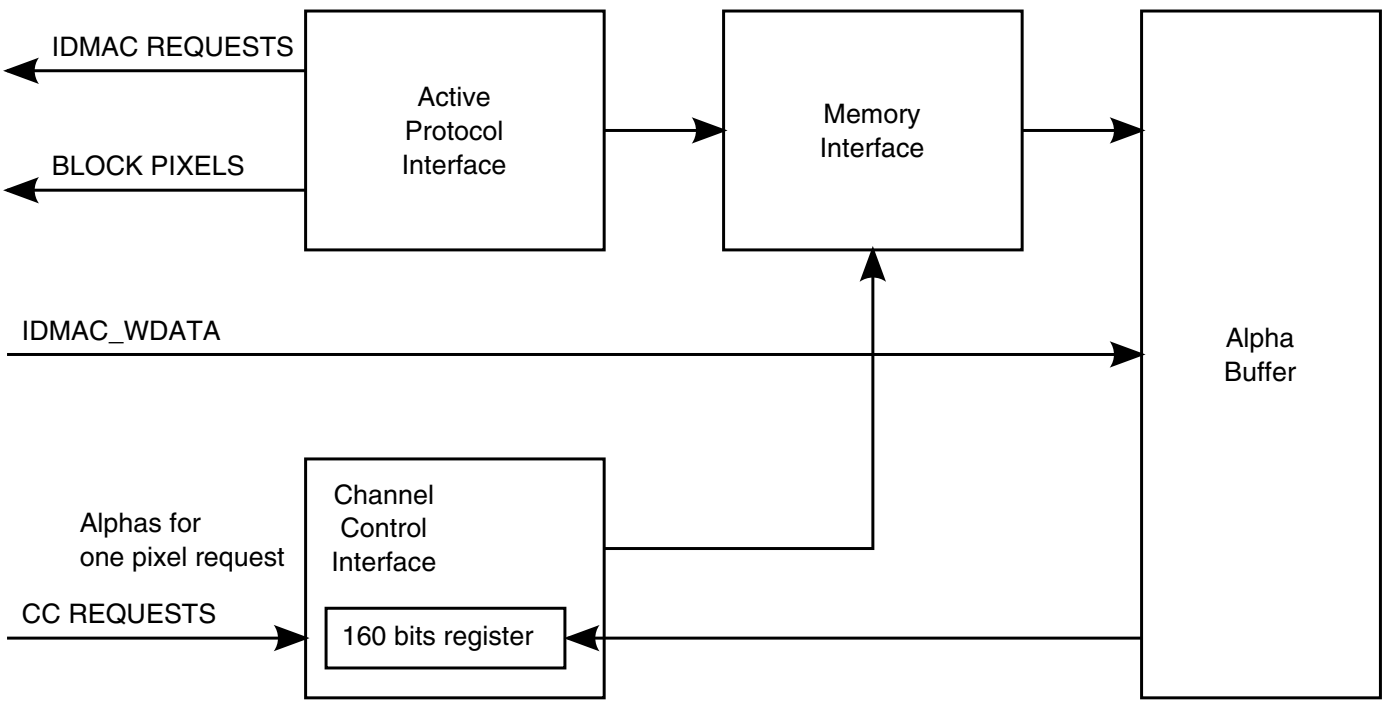


Figure 45-9. ATC block diagram

The ATC alpha buffer memory can hold up to 8 buffers of alphas. A pointer to a buffer in the ATC memory is defined according to the ALBM parameter in the CPMEM. The table below describes the relations between a data channel, an alpha channel and the pointer in the alpha buffer memory.

Table 45-9. Alpha channels mapping

data channel number	associated alpha channel number	Alpha buffer memory (ALBM)
14	17	0
15	18	1
27	31	2
29	33	3
23	51	4
24	52	5

Table continues on the next page...

Table 45-9. Alpha channels mapping (continued)

data channel number	associated alpha channel number	Alpha buffer memory (ALBM)
25	19	6

45.4.2.8.1 Conditional read

The alpha data can be used to reduce reads from the memory of pixels that are going to be transparent (alpha = 0). The conditional read feature is enabled by the CRE bit in the CPMEM.

If all of the corresponding alpha values for a single burst of pixels are equal to zero, the IDMAC will block the access to the external memory and provide a data of all zeros to the corresponding channel. This way some of the accesses to the memory can be prevented, thus reducing the load on the memory.

45.4.2.9 LUT- Look Up Table

When working in coded pixel format, the data read from the memory is the decoded value of pixel according to address given. In case of 8 bit code, the data read from the memory is the decoded value of the pixel according to the address given.

In case of 4 bit code configuration, The address of the 4 bit decoded values is set according to DEC_SEL parameter in the CPMEM

00 = addresses 0 to 15

01 = addresses 64 to 79

10 = addresses 128 to 143

11 = addresses 192 to 207

Table 45-10. Look-Up Table Memory Structure

Address	Word	DEC_SEL	Description
0	Word0	4 BPP = 00 8 BPP = don't care	Decoded Pixels [15:0] for both 8 and 4 bit coded configuration
...	...		
15	Word15		
16	Word16	don't care	Decoded Pixels [63:16] for 8 bit coded configuration only
...	...		
63	Word63		

Table continues on the next page...

Table 45-10. Look-Up Table Memory Structure (continued)

Address	Word	DEC_SEL	Description
64	Word64	4 BPP = 01 8 BPP = don't care	Decoded Pixels [79:64] for both 8 and 4 bit coded configuration
...	...		
79	Word79		
80	Word80	don't care	Decoded Pixels [127:80] for 8 bit coded configuration only
...	...		
127	Word127		
128	Word128	4 BPP = 10 8 BPP = don't care	Decoded Pixels [143:128] for both 8 and 4 bit coded configuration
...	...		
143	Word143		
144	Word144	don't care	Decoded Pixels [191:144] for 8 bit coded configuration only
...	...		
191	Word191		
192	Word192	4 BPP = 11 8 BPP = don't care	Decoded Pixels [207:192] for both 8 and 4 bit coded configuration
...	...		
207	Word207		
208	Word208	don't care	Decoded Pixels [255:208] for 8 bit coded configuration only
...	...		
255	Word255		

45.4.2.10 CPMEM - Channel Parameter Memory

The CPMEM holds the configuration parameters for each IDMAC channel. The CPMEM can holds the settings of 80 channels. Each channel's settings are defined by a two mega-words. Each mega-word is 160 bits wide.

The following diagram illustrates the CPMEM's structure.

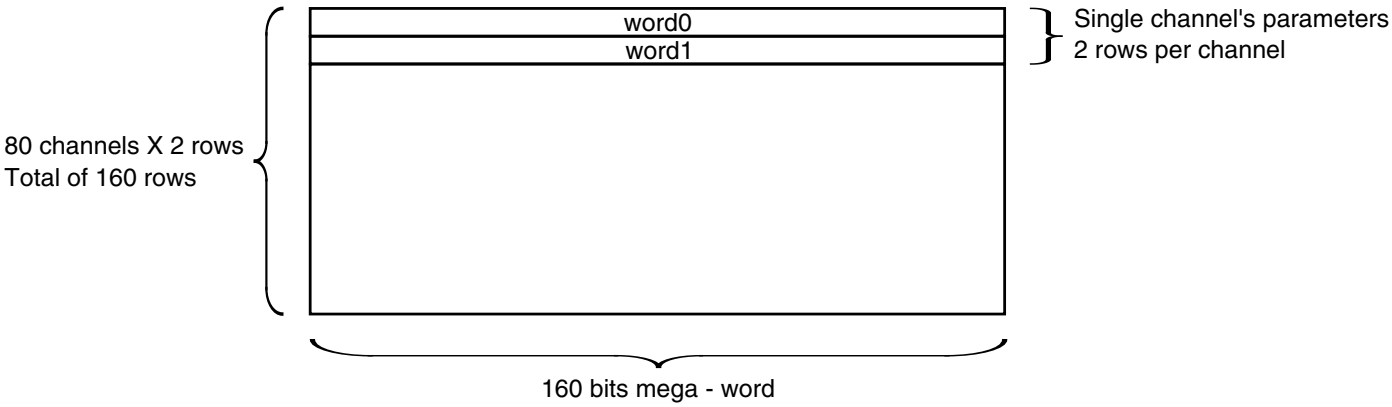


Figure 45-10. CPMEM structure

Each IDMAC channel can be configured to work in one of two different modes:

- Non Interleaved mode where the Y:U:V data is organized in 3 separate buffers in the system's memory
- Interleaved mode where the Y:U:V data is organized in a single buffer in the system's memory

The parameters and the way they are organized are different for each mode. The Pixel Format Select (PFS) value of the CPMEM words are used by the IPU to determine if the words should be interpreted as interleaved or non-interleaved format.

The following tables describe the IDMAC parameters and their organization in each mode.

45.4.2.10.1 CPMEM's words' structure for non interleaved mode

The table below describes the CPMEM's words' structure for non interleaved mode. Each CPMEM word consist of 160 bits. The bits that are not listed in the table below are reserved bits.

Table 45-11. Channel Parameters Memory for non-interleaved

Name	Mnemonic	Size	Location	Description
Word 0				
XV Virtual Coordinate	XV	10 bits	W0[9:0]	Variable coordinates for determining next block address. {X1,Y1} and {X2,Y2} coordinates will be determined according to {XV,YV} upon restart of channel. These coordinates are used for Y:U:V (Y pointer) and RGB formats.
YV Virtual Coordinate	YV	9 bits	W0[18:10]	

Table continues on the next page...

Table 45-11. Channel Parameters Memory for non-interleaved (continued)

XB inner Block Coordinate	XB	13 bits	W0[31:19]	Variable coordinates for determining address within the block. These coordinates are used for Y:U:V (Y pointer) and RGB formats. Need 24 bits for 2D transfer support.
YB inner Block Coordinate	YB	12 bits	W0[43:32]	
New Sub Block	NSB_B	1 bit	W0[44]	This bit determines if the next value for {XB,YB} should be taken from {XB,YB} saved in channel parameter memory or from new {x1,y1}/{x2,y2}.
Current Field	CF	1 bit	W0[45]	CF = 0 Current field is even CF = 1 Current field is odd
Mem U Buffer Offset	UBO	22 bits	W0[67:46]	Double buffer destination address offset for Y:U:V (U pointer) formats.The actual physical address value is divided by 8 (i.e. this parameter includes bits [24:3] of the actual address)
Mem V Buffer Offset	VBO	22 bits	W0[89:68]	Double buffer destination address offset for Y:U:V (V pointer) format. The actual physical address value is divided by 8 (i.e. this parameter includes bits [24:3] of the actual address)
Initial Offset X	IOX	4 bits	W0[93:90]	The IOX parameter, is the offset in pixels for a frame that starts at a non aligned address. for 42x formats must be even.
Reduce Double Read or Writes	RDRW	1 bits	W0[94:94]	This bit is relevant for YUV4:2:0 formats. For read channels: U and V components are not read from odd rows. (read - supported only for the VDIC) For write channels: U and V components are not written to odd rows. (write - supported for all write channels)
Scan Order	SO	1 bit	W0[113]	SO = 0 Scan order is progressive SO = 1 Scan order is interlaced

Table continues on the next page...

Table 45-11. Channel Parameters Memory for non-interleaved (continued)

Band Mode	BNDM	3 bits	W0[116:114]	BNDM = 000 bands disable. BNDM = 001 bands enable. Band height = 4 lines. BNDM = 010 bands enable. Band height = 8 lines. BNDM = 011 bands enable. Band height = 16 lines. BNDM = 100 bands enable. Band height = 32 lines. BNDM = 101 bands enable. Band height = 64 lines. BNDM = 110 bands enable. Band height = 128 lines. BNDM = 111 bands enable. Band height = 256 When working in band mode, the channel's corresponding IDMAC_BNDM_EN bit has to be set.
Block Mode	BM	2 bits	W0[118:117]	BM = 00 block mode disable. BW = FW, BH = FH BM = 01 block mode enable. BW = 8, BH = 8 BM = 10 block mode enable. BW = 16, BH = 16 (this mode is reserved for future use) BM = 11 not used
Rotation	ROT	1 bit	W0[119]	ROT = 0 -> No rotation ROT = 1 -> 90 degree rotation clockwise
Horizontal Flip	HF	1 bit	W0[120]	HF = 0 -> No flip HF = 1 -> Horizontal flip enable
Vertical Flip	VF	1 bit	W0[121]	VF = 0 -> No flip VF = 1 -> Vertical flip enable
Threshold Enable	THE	1 bit	W0[122]	THE = 0 -> Threshold disable THE = 1 -> Threshold enable
Conditional Access Polarity	CAP	1 bit	W0[123]	CAP = 0 -> If conditional bit in CM register is low skip the access CAP = 1 -> If conditional bit in CM register is high skip the access. This mode is reserved for future use.
Conditional Access Enable	CAE	1 bit	W0[124]	CAE = 0 -> Conditional access disable CAE = 1 -> Conditional access enable This mode is reserved for future use.

Table continues on the next page...

Table 45-11. Channel Parameters Memory for non-interleaved (continued)

Frame Width	FW	13 bits	W0[137:125]	Number of pixels in one row, of the channel frame. FW 000000000000 = 0001 pixels 000000000001 = 0002 pixels ..... 111111111111 = 8192 pixels
Frame Height	FH	12 bits	W0[149:138]	Number of pixels in one column, of the channel frame. FH 000000000000 = 0001 line 000000000001 = 0002 lines ..... 111111111111 = 4096 lines For progressive YUV 4:2:0 (non interleaved and partial interleaved formats) the FH value should be a multiple of 2. For interlaced YUV 4:2:0 (non interleaved and partial interleaved formats) the FH value should be a multiple of 4.
Word 1				
Ext Mem Buffer 0 Address	EBA0	29 bits	W1[28:0]	1st double buffer destination address for RGB and Y:U:V (Y pointer) formats. This parameter must not be changed if the corresponding channel is enabled. The actual physical address value is divided by 8 (i.e. this parameter includes bits [31:3] of the actual address)
Ext Mem Buffer 1 Address	EBA1	29 bits	W1[57:29]	2nd double buffer destination address for RGB and Y:U:V (Y pointer) formats. This parameter must not be changed if the corresponding channel is enabled. The actual physical address value is divided by 8 (i.e. this parameter includes bits [31:3] of the actual address)
Interlace Offset	ILO	20 bits	W1[77:58]	2nd double buffer destination address for RGB and Y:U:V (Y pointer) formats. An interlaced data stored in the memory can be read as a consecutive progressive only if FW*BPP is a multiplication of 8. The actual physical address value is divided by 8 (i.e. this parameter includes bits [22:3] of the actual address). For YUV420 formats, the ILO is relevant only to the Y component as the U and V components do not exist for the even lines. This value is signed

Table continues on the next page...

Table 45-11. Channel Parameters Memory for non-interleaved (continued)

Number of Pixels in Whole Burst Access	NPB	7 bits	W1[84:78]	Number of pixels per burst access. The following are valid numbers of pixels in a memory burst access according to the BPP parameter: NPB 0000111 = 08 pixels in each burst 0001111 = 16 pixels in each burst ..... 0111111 = 64 pixels in each burst Range: 16BPP => 1 -> 32 pixels (for YUV444 NI) 08BPP => 1 -> 64 pixels (For YUV420 NI/PI and YUV422 NI/PI) In NI/PI formats the NPB has to be a multiplication of 8
Pixel Format Select	PFS	4 bits	W1[88:85]	4'h0 = non-interleaved 4:4:4 4'h1 = non-interleaved 4:2:2 4'h2 = non-interleaved 4:2:0 4'h3 = partial interleaved 4:2:2 4'h4 = partial interleaved 4:2:0 4'h5 to 4'hF = NA
Alpha Used	ALU	1 bit	W1[89]	1 = the alpha associated with the data of this channel resides on another channel (separate buffer) 0 = the alpha associated with the data of this channel resides along with the pixel data (same buffer) The corresponding alpha channel must be enabled to assure correct behavior.
Alpha Channel Mapping	ALBM	2 bits	W1[92:90]	Alpha channel mapping - This parameter is a pointer to a buffer in the ATC memory. This parameter is relevant only to data channels that are associated with a separate alpha buffer (like graphic plane channels). The parameter should be programmed on the data channels' ALBM. Setting this parameter to any other channel has no meaning. See Table 45-9 for exact ALBM mapping.
AXI Id	ID	2 bits	W1[94:93]	AXI protocol id
Threshold	TH	7 bits	W1[101:95]	0000000 = 32 lines 0000001 = 64 lines ..... 1111111 = 4096 lines

Table continues on the next page...

Table 45-11. Channel Parameters Memory for non-interleaved (continued)

Stride Line	SLY	14 bits	W1[115:102]	Address vertical scaling factor in bytes for memory access. Also number of maximum bytes in the "Y" component row according to memory limitations. SLY 00000000000000 = 00001 bytes 00000000000001 = 00002 bytes ..... 11111111111111 = 16384 bytes
Width3	WID3	3 bits	W1[127:125]	Fourth color component size of the input-unpacking/ output-packing pixel. WID3 000 = 1 bits 001 = 2 bits ..... 111 = 8 bits As this is a non-interleaved format, this field is relevant only to the alpha associated with this pixel channel.
Stride Line	SLUV	14 bits	W1[141:128]	Address vertical scaling factor in bytes for memory access. Also number of maximum bytes in the "U" or "V" component row according to memory limitations. SLUV 00000000000000 = 00001 bytes 00000000000001 = 00002 bytes ..... 11111111111111 = 16384 bytes
Conditional Read Enable	CRE	1 bit	W1[149:149]	This bit enables the conditional read feature.

45.4.2.10.2 CPMEM's words' structure for interleaved mode

The table below describes the CPMEM's words' structure for interleaved mode. Each CPMEM word consist of 160 bits. The bits that are not listed in the table below are reserved bits

Table 45-12. Channel Parameters Memory for interleaved

Name	Mnemonic	Size	Location	Description
Word 0				

Table continues on the next page...

Table 45-12. Channel Parameters Memory for interleaved (continued)

XV Virtual Coordinate	XV	10 bits	W0[9:0]	Variable coordinates for determining next block address. {X1,Y1} and {X2,Y2} coordinates will be determined according to {XV,YV} upon restart of channel. These coordinates are used for Y:U:V (Y pointer) and RGB formats.
YV Virtual Coordinate	YV	9 bits	W0[18:10]	
YB inner Block Coordinate	XB	13 bits	W0[31:19]	Variable coordinates for determining address within the block. These coordinates are used for Y:U:V (Y pointer) and RGB formats. Need 24 bits for 2D transfer support.
XB inner Block Coordinate	YB	12 bits	W0[43:32]	
New Sub Block	NSB_B	1 bit	W0[44]	This bit determines if the next value for {XB,YB} should be taken from {XB,YB} saved in channel parameter memory or from new {x1,y1}/{x2,y2}.
Current Field	CF	1 bit	W0[45]	CF = 0 Current field is even CF = 1 Current field is odd
Scroll X counter	SX	12 bits	W0[57:46]	Holds the temporary count for the Scroll X in between frame For interleaved YUV4:2:2 formats the SX should be a multiple of 2.
Scroll Y counter	SY	11 bits	W0[68:58]	Holds the temporary count for the Scroll Y in between frame
Number of Scroll	NS	10 bits	W0[78:69]	This variable holds the total number of Scrolls
Scroll Delta X	SDX	7 bits	W0[85:79]	Frame start row offset, compared to last frame. SDX 0000000 = 00 pixels 0000001 = 01 pixels 0000010 = 02 pixels ..... 1111110 = 30 pixels 1111111 = 127 pixels For interleaved YUV4:2:2 formats the SDX should be a multiple of 2.
Scroll Max	SM	10 bits	W0[95:86]	Frame maximum row and column increment offset in frame. SM 0000000000 = 0001 0000000001 = 0002 ..... 1111111111 = 1024

Table continues on the next page...

Table 45-12. Channel Parameters Memory for interleaved (continued)

Scrolling Configuration	SCC	1 bit	W0[96]	Determines if scrolling will continue from zero when NS counter has reached SM or stop at the current value for SX and SY for the next frames to come. SCC 0 => Scrolling will stop at NS = SM 1 => Scrolling will start from "0" at NS = SM
Scrolling Enable	SCE	1 bit	W0[97]	SCE = 0 Scrolling disable SCE = 1 Scrolling enable
Scroll Delta Y	SDY	7 bits	W0[104:98]	Frame start column offset, compared to last frame. SDY 0000000 = 00 pixels 0000001 = 01 pixels 0000010 = 02 pixels ..... 1111110 = 30 pixels 1111111 = 127 pixels
Scroll Horizontal Direction	SDRX	1 bit	W0[105]	Determines if the next frame will move right or left compared to the current frame. SDRX 0 => Next frame will be right of current 1 => Next frame will be left of current
Scroll Vertical Direction	SDRY	1 bit	W0[106]	Determines if the next frame will move down or up compared to the current frame. SDRY 0 => Next frame will be down of current 1 => Next frame will be up of current
Bits Per Pixel	BPP	3 bits	W0[109:107]	3'h0 = 32 Bits per pixel 3'h1 = 24 Bits per pixel 3'h2 = 18 Bits per pixel 3'h3 = 16 Bits per pixel 3'h4 = 12 Bits per pixel 3'h5 = 08 Bits per pixel 3'h6 = 04 Bits per pixel

Table continues on the next page...

Table 45-12. Channel Parameters Memory for interleaved (continued)

Decode Address Select	DEC_SEL	2 bits	W0[111:110]	Upon 4BPP, selects between two look-up tables DEC_SEL 00 = addresses 0 to 15 01 = addresses 64 to 79 10 = addresses 128 to 143 11 = addresses 192 to 207
Access Dimension	DIM	1 bit	W0[112]	DIM = 0 Access Dimension is 2d DIM = 1 Access Dimension is 1d
Scan Order	SO	1 bit	W0[113]	SO = 0 Scan order is progressive SO = 1 Scan order is interlaced
Band Mode	BNDM	3 bits	W0[116:114]	BNDM = 000 bands disable. BNDM = 001 bands enable. Band height = 4 lines. BNDM = 010 bands enable. Band height = 8 lines. BNDM = 011 bands enable. Band height = 16 lines. BNDM = 100 bands enable. Band height = 32 lines. BNDM = 101 bands enable. Band height = 64 lines. BNDM = 110 bands enable. Band height = 128 lines. BNDM = 111 bands enable. Band height = 256 When working in band mode, the channel's corresponding IDMAC_BNDM_EN bit has to be set.
Block Mode	BM	2 bits	W0[118:117]	BM = 00 block mode disable. BW = FW, BH = FH BM = 01 block mode enable. BW = 8, BH = 8 BM = 10 block mode enable. BW = 16, BH = 16 (this mode is reserved for future use) BM = 11 not used
Rotation	ROT	1 bit	W0[119]	ROT = 0 -> No rotation ROT = 1 -> 90 degree rotation clockwise
Horizontal Flip	HF	1 bit	W0[120]	HF = 0 -> No flip HF = 1 -> Horizontal flip enable
Vertical Flip	VF	1 bit	W0[121]	VF = 0 -> No flip VF = 1 -> Vertical flip enable
Threshold Enable	THE	1 bit	W0[122]	THE = 0 -> Threshold disable THE = 1 -> Threshold flip enable

Table continues on the next page...

Table 45-12. Channel Parameters Memory for interleaved (continued)

Conditional Access Polarity	CAP	1 bit	W0[123]	CAP = 0 -> If conditional bit in CM register is low skip the access CAP = 1 -> If conditional bit in CM register is high skip the access
Conditional Access Enable	CAE	1 bit	W0[124]	CAE = 0 -> Conditional access disable CAE = 1 -> Conditional access enable
Frame Width	FW	13 bits	W0[137:125]	Number of pixels in one row, of the channel frame. FW 000000000000 = 0001 pixels 000000000001 = 0002 pixels ..... 111111111111 = 8192 pixels For interleaved YUV4:2:2 formats the FW should be a multiple of 2.
Frame Height	FH	12 bits	W0[149:138]	Number of pixels in one column, of the channel frame. FH 000000000000 = 0001 line 000000000001 = 0002 lines ..... 111111111111 = 4096 lines
Word 1				
Ext Mem Buffer 0 Address	EBA0	29 bits	W1[28:0]	1st double buffer destination address for RGB and Y:U:V (Y pointer) formats. This parameter must not be changed if the corresponding channel is enabled.The actual physical address value is divided by 8 (i.e. this parameter includes bits [31:3] of the actual address)
Ext Mem Buffer 1 Address	EBA1	29 bits	W1[57:29]	2nd double buffer destination address for RGB and Y:U:V (Y pointer) formats. This parameter must not be changed if the corresponding channel is enabled.The actual physical address value is divided by 8 (i.e. this parameter includes bits [31:3] of the actual address)

Table continues on the next page...

Table 45-12. Channel Parameters Memory for interleaved (continued)

Interlace Offset	ILO	20 bits	W1[77:58]	2nd double buffer destination address for RGB and Y:U:V (Y pointer) formats. An interlaced data stored in the memory can be read as a consecutive progressive only if FW*BPP is a multiplication of 8.The actual physical address value is divided by 8 (i.e. this parameter includes bits [22:3] of the actual address). This value is signed
Number of Pixels in Whole Burst Access	NPB	7 bits	W1[84:78]	Number of pixels per burst access. The following are valid numbers of pixels in a memory burst access according to the BPP parameter: NPB 0000000 = 01 pixels in each burst 0000001 = 02 pixels in each burst ..... 1111111 = 128 pixels in each burst Range: 32BPP => 1 -> 16 pixels 24BPP => 1 -> 20 pixels 16BPP => 1 -> 32 pixels 12BPP => 1 -> 40 pixels 08BPP => 1 -> 64 pixels 04BPP => 1 -> 128 pixels
Pixel Format Select	PFS	4 bits	W1[88:85]	4'h0 to 4'h4 = NA 4'h5 = Code (LUT) 4'h6 = Generic data 4'h7 = RGB (& also YUV interleaved 4:4:4) 4'h8 = interleaved 4:2:2 Y1U1Y2V1[1] 4'h9 = interleaved 4:2:2 Y2U1Y1V1[2] 4'hA = interleaved 4:2:2 U1Y1V1Y2[3] 4'hB = interleaved 4:2:2 U1Y2V1Y1[4] 4'hC to 4'hF = NA
Alpha Used	ALU	1 bit	W1[89]	1 = the alpha associated with the data of this channel resides on another channel (separate buffer) 0 = the alpha associated with the data of this channel resides along with the pixel data (same buffer) The corresponding alpha channel must be enabled to assure correct behavior.

Table continues on the next page...

Table 45-12. Channel Parameters Memory for interleaved (continued)

Alpha Channel Mapping	ALBM	3 bits	W1[92:90]	Alpha channel mapping - This parameter is a pointer to a buffer in the ATC memory. This parameter is relevant only to data channels that are associated with a separate alpha buffer (like graphic plane channels). The parameter should be programmed on the data channels' ALBM. Setting this parameter to any other channel has no meaning. See Table 45-9 for exact ALBM mapping.
AXI Id	ID	2 bits	W1[94:93]	AXI protocol id; IPU is targeted to an AXI slave that can handle up to 2 requests with 2 different IDs + one request with a third ID. In case that IPU is going to be used on a system that can handle more than 2 requests with different IDs, the number of different IDs programmed in the CPMEM for different channels is limited for 2. This limitation is relevant for read channels only. For write channels there's no such limitation
Threshold	TH	7 bits	W1[101:95]	0000000 = 32 lines 0000001 = 64 lines ..... 1111111 = 4096 lines
Stride Line	SL	14 bits	W1[115:102]	Address vertical scaling factor in bytes for memory access. Also number of maximum bytes in row according to memory limitations. SL 00000000000000 = 00001 bytes 00000000000001 = 00002 bytes ..... 11111111111111 = 16384 bytes
Width0	WID0	3 bits	W1[118:116]	First color component size of the input-unpacking/ output-packing pixel. WID0 000 = 1 bits 001 = 2 bits ..... 111 = 8 bits This field is relevant only for interleaved RGB format (PFS = 4'h7)

Table continues on the next page...

Table 45-12. Channel Parameters Memory for interleaved (continued)

Width1	WID1	3 bits	W1[121:119]	Second color component size of the input-unpacking/ output-packing pixel. WID1 000 = 1 bits 001 = 2 bits ..... 111 = 8 bits This field is relevant only for interleaved RGB format (PFS = 4'h7)
Width2	WID2	3 bits	W1[124:122]	Third color component size of the input-unpacking/ output-packing pixel. WID2 000 = 1 bits 001 = 2 bits ..... 111 = 8 bits This field is relevant only for interleaved RGB format (PFS = 4'h7)
Width3	WID3	3 bits	W1[127:125]	Fourth color component size of the input-unpacking/ output-packing pixel. WID3 000 = 1 bits 001 = 2 bits ..... 111 = 8 bits This field is relevant only for interleaved RGB format (PFS = 4'h7)

Table continues on the next page...

Table 45-12. Channel Parameters Memory for interleaved (continued)

Offset0	OFS0	5 bits	W1[132:128]	Number of bits between MSB of pixel and MSB of color component, on input. 1 states that the color component will be the first color component aligned to MSB of output pixel. OFS0 00000 = No offset 00001 = u => 1 bit sleft, p => 1 bit sright ..... 11111 = u => 31 bit sleft, p => 31 bit sright * u = unpacking * p = packing * sleft = shift left * sright = shift right This field is relevant only for interleaved RGB format (PFS = 4'h7)
Offset1	OFS1	5 bits	W1[137:133]	Number of bits between MSB of pixel and MSB of color component on input. 2 states that the color component will be the second color component aligned to MSB output pixel. OFS1 00000 = No offset 00001 = u => 1 bit sleft, p => 1 bit sright ..... 11111 = u => 31 bit sleft, p => 31 bit sright * u = unpacking * p = packing * sleft = shift left * sright = shift right This field is relevant only for interleaved RGB format (PFS = 4'h7)

Table continues on the next page...

Table 45-12. Channel Parameters Memory for interleaved (continued)

Offset2	OFS2	5 bits	W1[142:138]	Number of bits between MSB of pixel and MSB of color component on input. 3 states that the color component will be the third color component aligned to MSB output pixel. OFS2 00000 = No offset 00001 = u => 1 bit sleft, p => 1 bit sright ..... 11111 = u => 31 bit sleft, p => 31 bit sright * u = unpacking * p = packing * sleft = shift left * sright = shift right This field is relevant only for interleaved RGB format (PFS = 4'h7)
Offset3	OFS3	5 bits	W1[147:143]	Number of bits between MSB of pixel and MSB of color component on input. 4 states that the color component will be the fourth color component aligned to MSB output pixel. OFS3 00000 = No offset 00001 = u => 1 bit sleft, p => 1 bit sright ..... 11111 = u => 31 bit sleft, p => 31 bit sright * u = unpacking * p = packing * sleft = shift left * sright = shift right This field is relevant only for interleaved RGB format (PFS = 4'h7)
Select SX SY Set	SXYS	1 bit	W1[148:148]	This bit selects between the settings on: SC_CORD and SC_CORD1
Conditional Read Enable	CRE	1 bit	W1[149:149]	This bit enables the conditional read feature.
Decode Address Select bit[2]	DEC_SEL2	1 bit	W1[150:150]	This field is reserved

1. Y1U1Y2V1 means byte0 = bits [7:0] =Y1; byte1 = bits [15:8] =U1; byte2 = bits [23:16] =Y2; byte3 = bits [31:24] = V1
2. Y2U1Y1V1 means byte0 =Y2; byte1 =U1; byte2 =Y1; byte3 = V1
3. U1Y1V1Y2 means byte0 =U1; byte1 =Y1; byte2 =V1; byte3 = Y2
4. U1Y2V1Y1 means byte0 =U1; byte1 =Y2; byte2 =V1;byte3 = Y1

45.4.2.10.3 Accessing the CPMEM for programming

Each IDMAC's channel's parameters are located on 2 CPMEM entries. Each Entry is 160 bit. The CPMEM is memory mapped and is accessible via the AHB bus. The AHB bus's accesses are 32bit wide. A CPMEM entry is composed of 5x32bit words. The next CPMEM entry starts at the next 8x32bit words (0x0, 0x20,0x40, etc.).

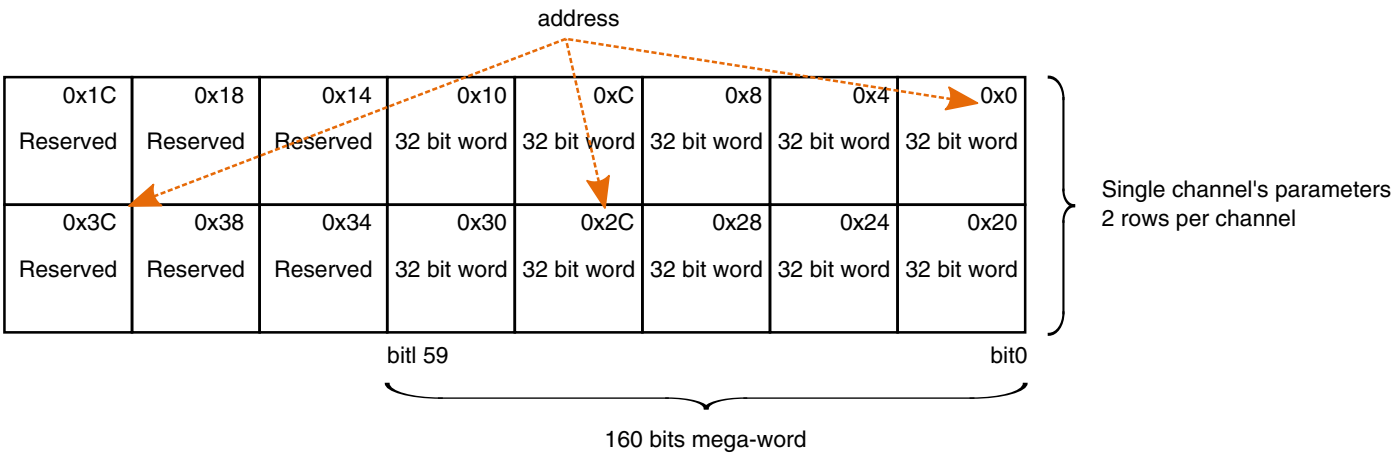


Figure 45-11. CPMEM's word structure

45.4.2.10.4 Alternate IDMAC settings

Some of the IDMAC channels support alternate flow. This means that a physical IDMAC channel can use alternate set of parameters. Switching between the flows is controlled by the CM.

The primary flow the IDMAC's settings are read from the channel's corresponding entry in the CPMEM. The alternate settings can be stored in another entry on the CPMEM. The pointer to the alternate entry on the CPMEM is stored on the physical channel's corresponding IDMAC_SUB_ADDR parameters.

45.4.2.11 IDMAC's modes of operation

45.4.2.11.1 Rotation modes

Rotation is performed by the IDMAC and the Rotation unit inside the IC.

The frame is partitioned into 8X8 pixels blocks. The IC reorders the pixels within a block. The IDMAC reorders the block. The reordering is done according to the ROT, VF & HF parameters in the CPMEM.

The following diagram illustrate various options for reordering of blocks

- ROTATE means that the ROT bit is set
- HORIZONTAL means that the HF bit is set
- VERTICAL means that the VF bit is set

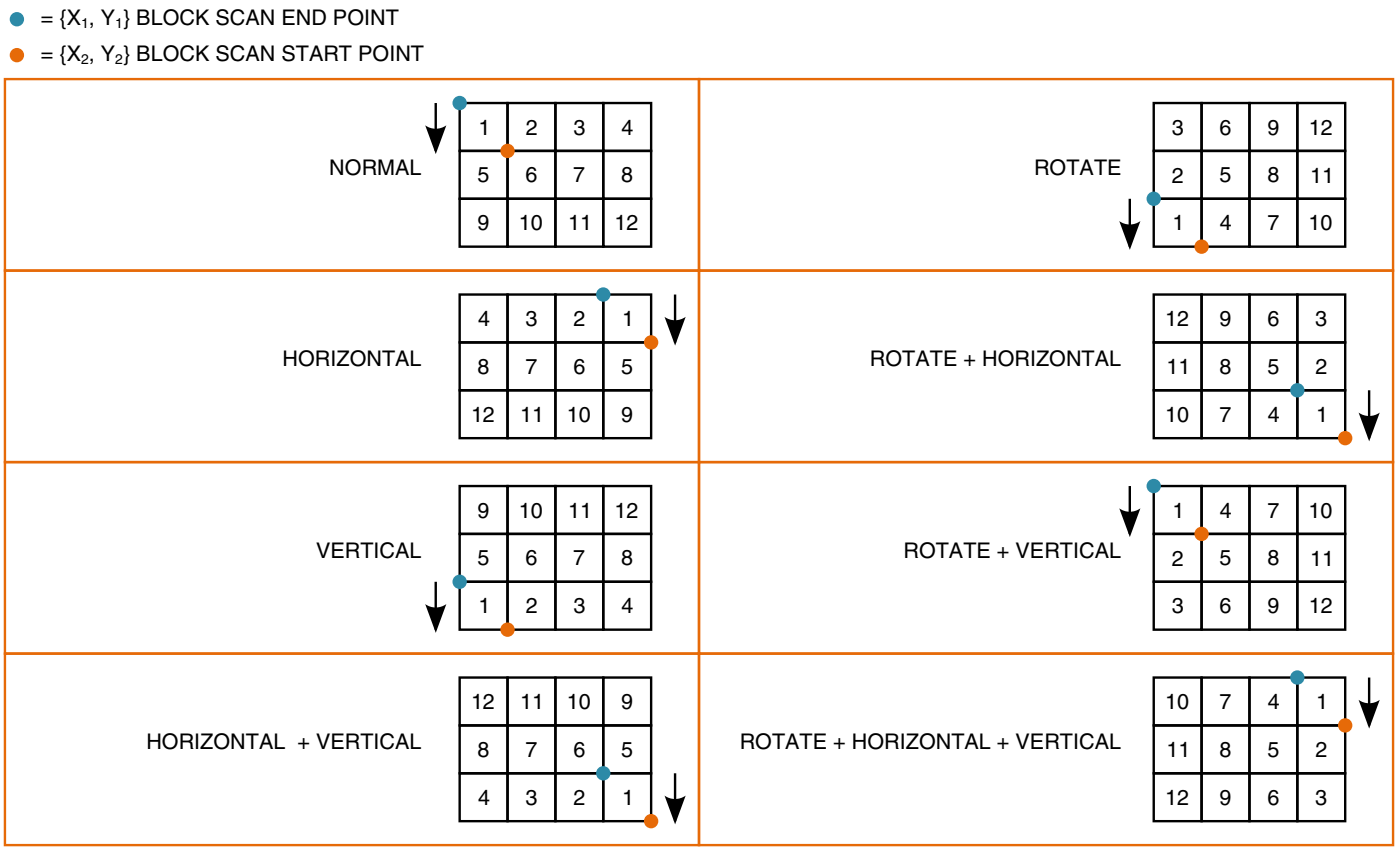


Figure 45-12. Rotation options

45.4.2.11.2 Frame size

The IPU supports various non-interleaved modes; the Frame Height (FH) and Frame Width (FW).

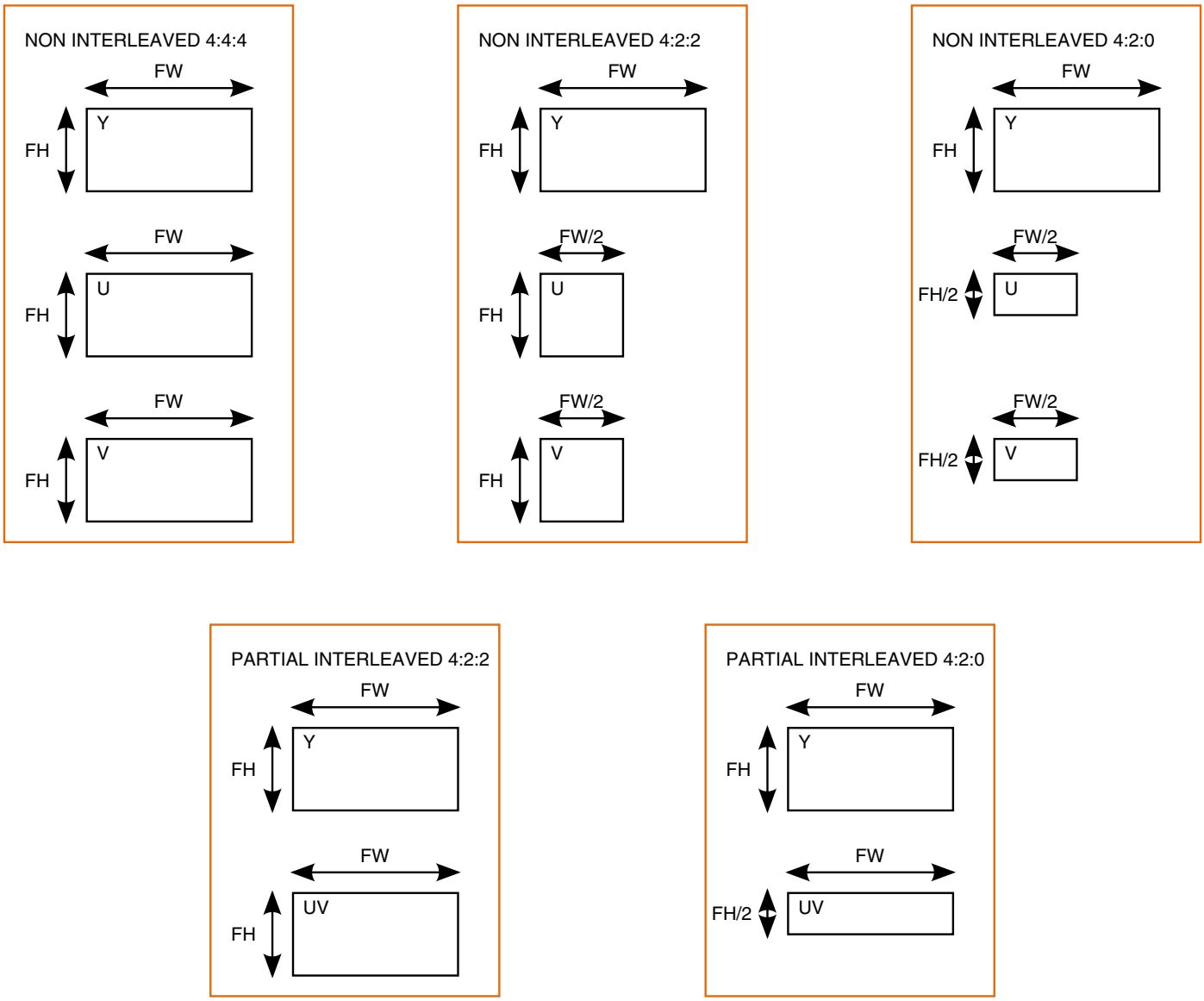


Figure 45-13. Frame size in various modes

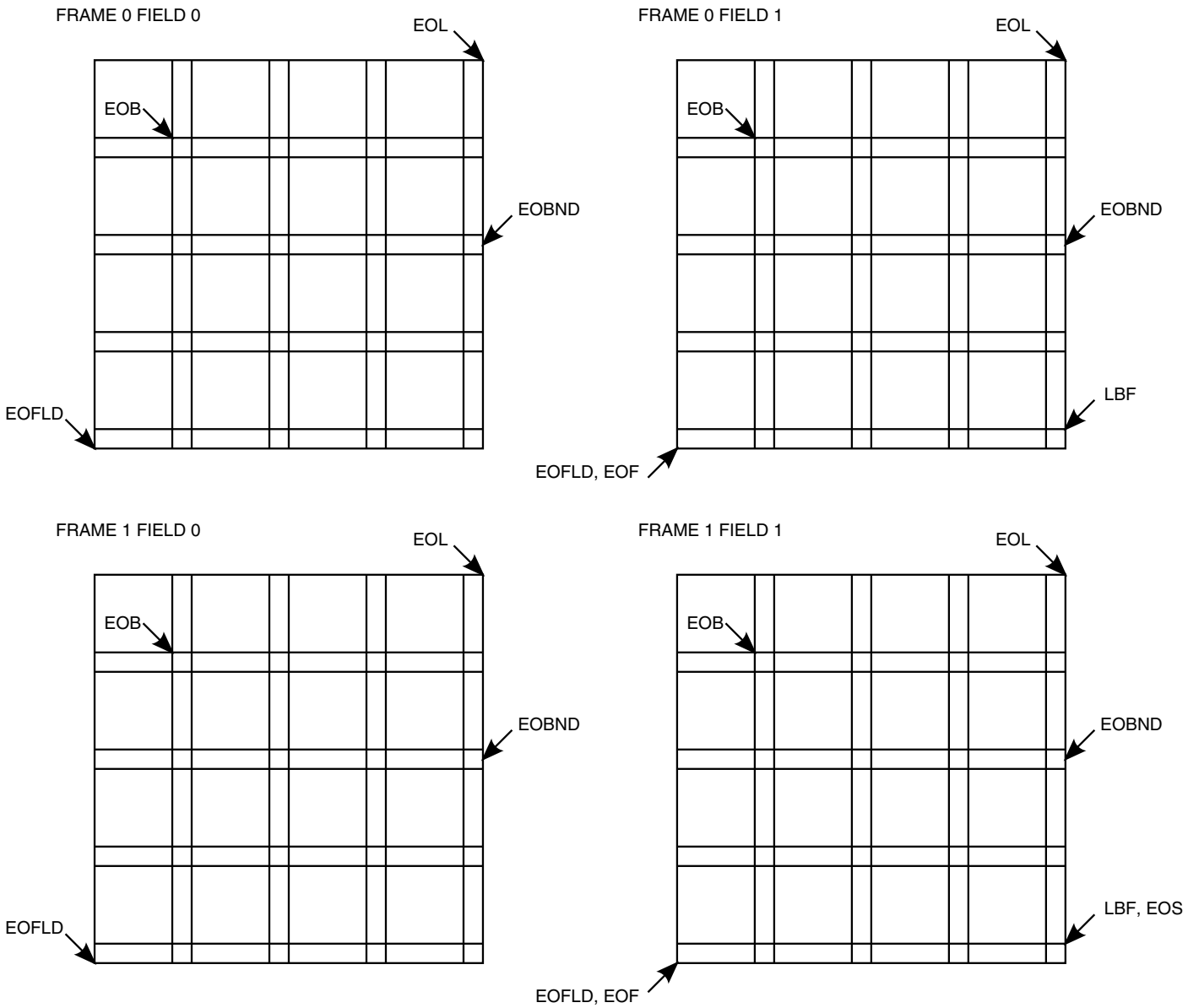


Figure 45-14. Frame's controls

45.4.2.12 IDMAC's restriction

The data must be received from the system's memory through the AXIR interface ("read" direction) "in-order" within a single burst. The entire burst can occur "out-of-order."

45.4.2.13 IDMAC's Endianness support

Byte Endianness - only LE (little endian) is supported

Pixel Endianness - both LE & BE are supported, only for read direction (only 4 BPP case is meaningful, supported only for the "read" direction)

45.4.2.14 IDMAC's internal events

Some of the IDMAC's internal signals can be used for monitor the progress of flows. These bits can be polled by software. Some of these bits can be used to trigger an interrupt or an SDMA event.

The following table describes the available events and their meaning

Table 45-13. IDMAC's internal events

IDMAC event's type	Monitored on	Event's meaning
end of frame	IDMAC_EOF	This is the channel's end of frame indication. This indication is asserted once the entire frame was read/written via the IDMAC. This indication is normally used as an interrupt or SDMA event
new frame acknowledge	IDMAC_NFACK	This indication means that the IDMAC acknowledge the new frame's request from the module. It can be used to track the starting point of a flow or a frame.
new frame before end of frame error	IDMAC_NFB4EOF_ERR	This error indication may indicate on data lost. This indication is asserted when a new frame starts before the completion of the previous frame. For example if a real time input (from camera) was not written properly to the memory due to FIFO full condition.
end of scroll	IDMAC_EOS	end of scroll; This indication is asserted when the scroll counter finished counting its pre defined value
end of band	IDMAC_EOBND	This is the end of band indication. Any time IDMAC complete reading/writing a band it will assert this indication. This is useful to manually control a flow via channels working in band mode.
treshold	IDMAC_TH	Threshold crossing indication. The treshold is defined according to the TH parameter in the IDMAC.
channel busy	IDMAC_CH_BUSY	This signal is asserted when a channel is between NFACK event to EOF event. Negation of these indications is one of the conditions for low power modes handshake.

45.4.3 Camera Sensor Interface (CSI)

The IPU has 2 identical camera sensor interfaces (CSI). The CSI description below refers to a single CSI.

45.4.3.1 CSI Block Diagram

This figure shows the CSI Block Diagram.

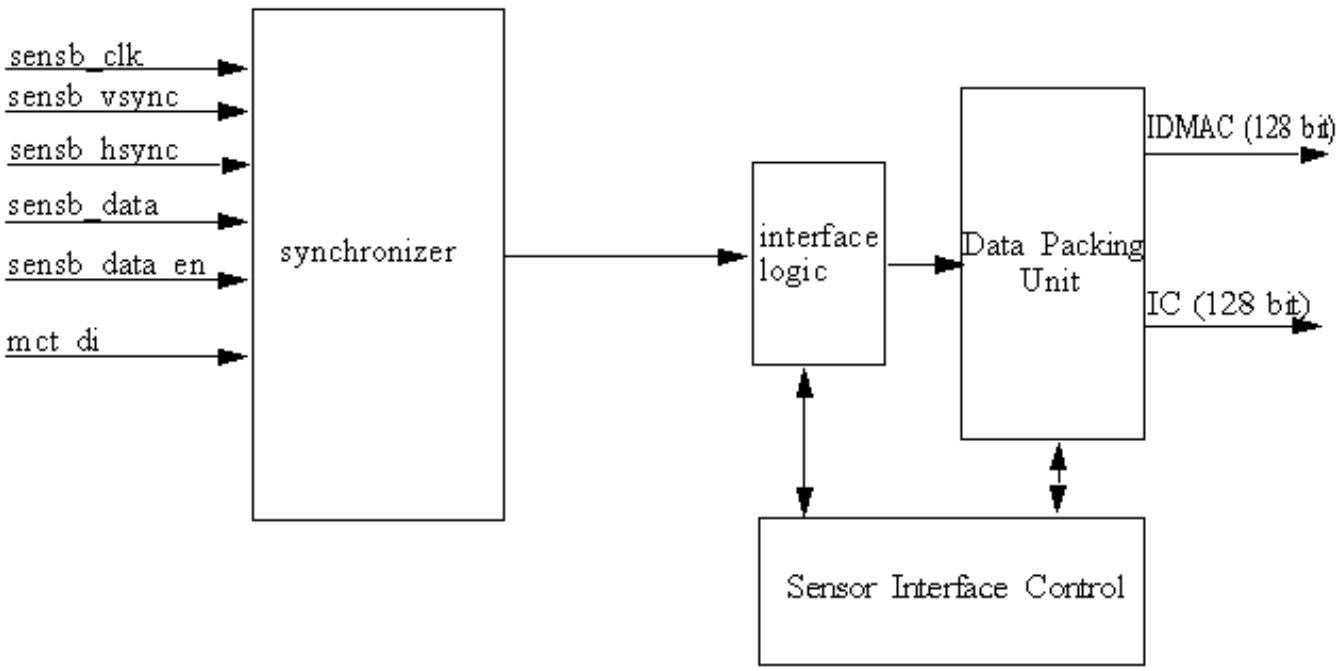


Figure 45-15. CSI Block Diagram

The CSI consists of synchronizer, interface logic, Data packing unit and Sensor Interface Control. The CSI is controlled via the peripheral bus registers. All programming parameters for the CSI are double buffered with synchronous change at the frame start.

The CSI gets data from the sensor, synchronizes the data and the control signals to the IPU clock (HSP_CLK), and transfer it according to configuration of DATA_DEST register to one or more of the following: IC, SMFC.

45.4.3.2 CSI Interface

CSI supports two types of interfaces. The interface is determined via the DATA_SOURCE register.

45.4.3.2.1 Parallel interface

In parallel interface a single value arrives in each clock, except when working in BT.1120 mode, in which two values arrive in each cycle. Each value can be 8-16 bit wide according to configuration of DATA_WIDTH. If DATA_WIDTH is configured to N, then 20-N LSB bits are ignored.

CSI can work with several data formats according to SENS_DATA_FORMAT configuration. In case the data format is YUV, the output of the CSI is always YUV444 (even if the data arrives in YUV422 format).

The polarity of the inputs can be configured using the registers SENS_PIX_CLK_POL, DATA_POL, HSYNC_POL and VSYNC_POL.

45.4.3.3 TEST MODE

When TEST_GEN_MODE register is configured to 1, the TEST MODE which is a debugging mode, is operated.

The CSI generates the frame by itself and sends it to one of the destination units. The sent frame is a chess board composed of black and configured color squares. The configured color is set with the registers PG_B_VALUE, PG_G_VALUE and PG_R_VALUE. The data can be sent in different frequencies according to the configuration of DIV_RATIO register.

CSI Test Mode requires the following CSIx_SENS_CONF settings:

CSIx_EXT_VSYNC = 0x1 (External VSYNC mode)

CSIx_DATA_WIDTH = 0x1 (8 bits per color)

CSIx_SENS_DATA_FORMAT = 0x0 (Full RGB or YUV444)

CSIx_PACK_TIGHT = 0x0 (Each component is written as a 16 bit word where the MSB is written to bit #15, color extension is done for the remaining leat significant bits.)

CSIx_SENS_PRTCL = 0x1 (Non-gated clock sensor timing/data mode)

CSIx_SENS_PIX_CLK_POL = 0x1 Pixel clock is inverted before applied to internal circuitry

CSIx_DATA_POL = 0x0 (Data lines are directly applied to internal circuitry.)

CSIx_HSYNC_POL = 0x0 (HSYNC is directly applied to internal circuitry)

CSIx_VSYNC_POL = 0x0 (VSYNC is directly applied to internal circuitry)

45.4.3.4 Sensor Image Frame Relations

The figure below illustrates the generalized relations between image frames produced by a sensor and accepted by the CSI. Generally, four frame definitions exist. The virtual frame A starts with the VSYNC signal. After vertical blanking starts frame B. The HSYNC signal indicates boundaries of the frame B. The frame B includes both the active sensor frame C and horizontal blanking intervals. A size of the blanking intervals depends on sensor type and programming. The size of the frame sent by the sensor (the actual pixels) has to be configured in the registers SENS_FRM_WIDTH and SENS_FRM_HEIGHT. The CSI selects a window (the frame D) inside the frame C by skipping rows and columns according to parameters defined in registers HSC, VSC, ACT_FRM_HEIGHT and ACT_FRM_WIDTH. frame D is sent to the rest of the IPU.

NOTE

The following limitation must exist:

SENS_FRM_HEIGHT >= VSC + ACT_FRM_HEIGHT

SENS_FRM_WIDTH >= HSC + ACT_FRM_WIDTH

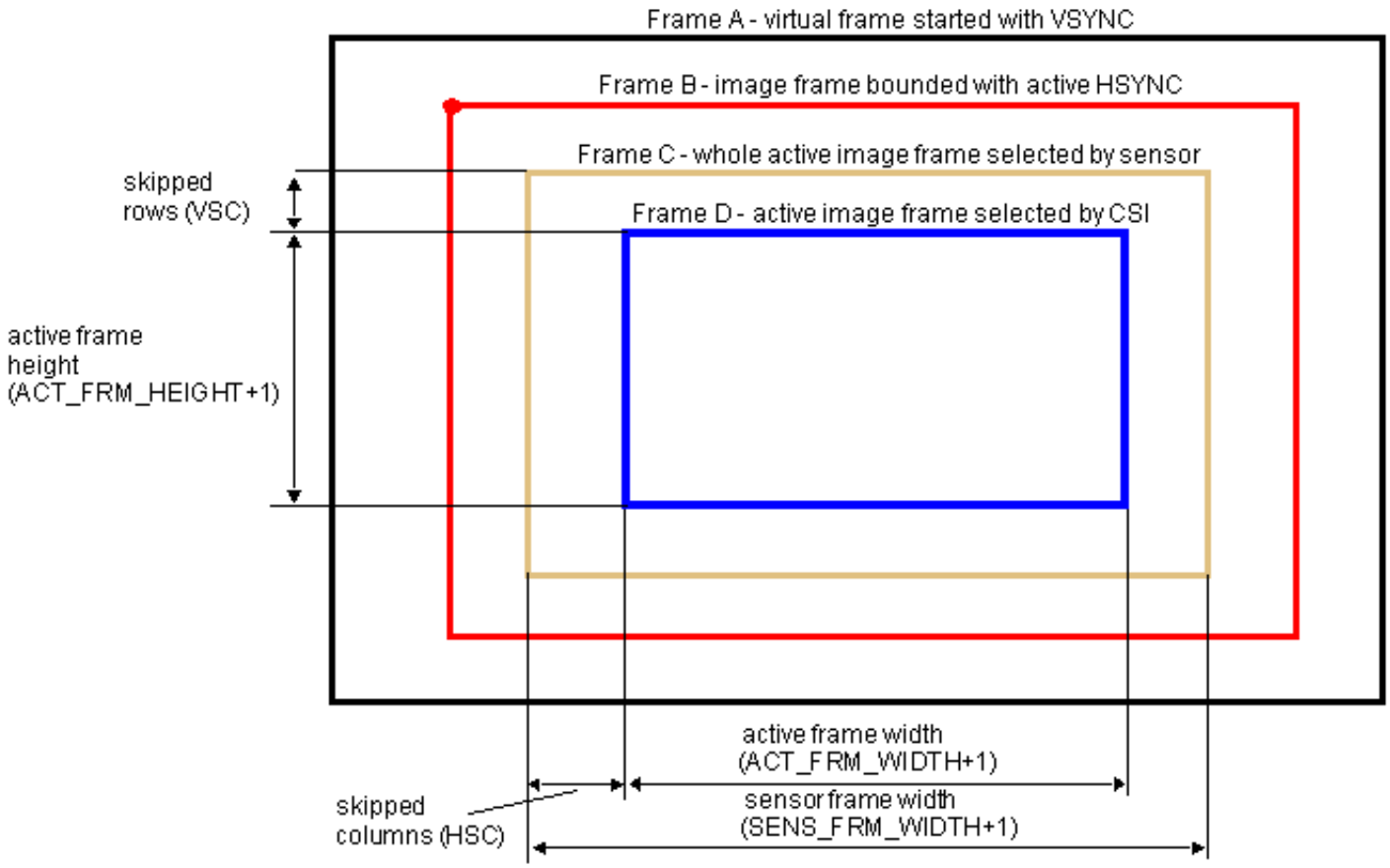


Figure 45-16. Sensor Image Frames

45.4.3.5 Timing/Data mode protocols

CSI can work in several timing/data mode protocols, according to SENS_PRTCL configuration.

45.4.3.5.1 gated mode

In this mode VSYNC is used to indicate beginning of a frame, HSYNC is used to indicate beginning of a raw. Sensor clock is ticking all the time.

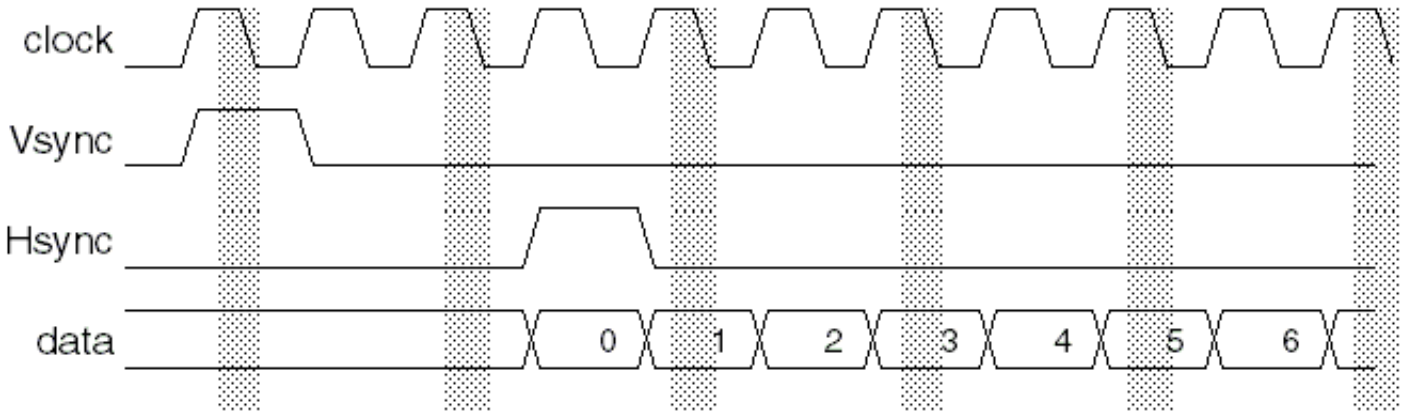


Figure 45-17. gated mode

45.4.3.5.2 non-gated mode

In this mode VSYNC is used to indicate beginning of a frame. Sensor clock is ticking only when data is valid. HSYNC is not used.

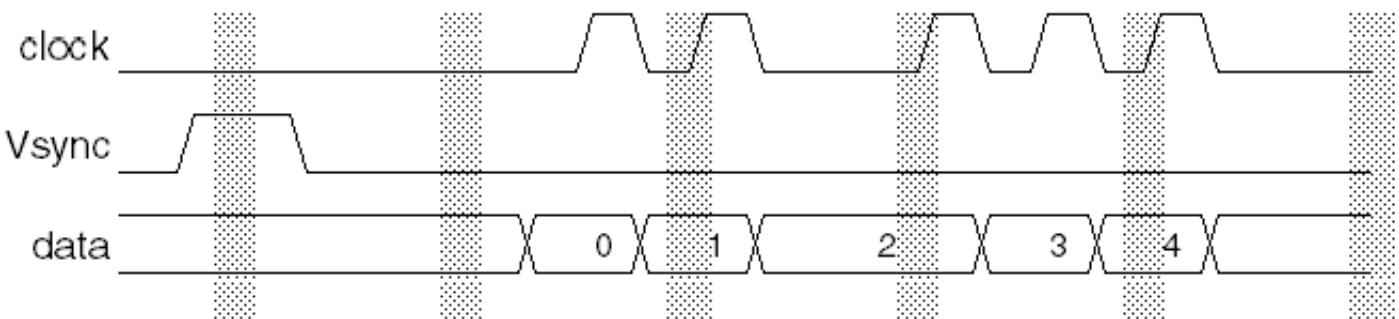


Figure 45-18. non-gated mode

45.4.3.5.3 BT.656 mode

In this mode the CSI works in compliance with recommendation ITU-R BT.656. The timing reference signals (frame start, frame end, line start, line end) are embedded in the data bus input. Each timing reference signal consists of a four word se

three words are fixed and configured in the CCIR_PRECOM register. The fourth word contains information defining field, the state of field blanking and the state of line blanking. these states are configured in registers CCIR_CODE_1 (for field 0) and CCIR_CODE_2 (for field 1).

In this mode in each cycle one value of data arrives

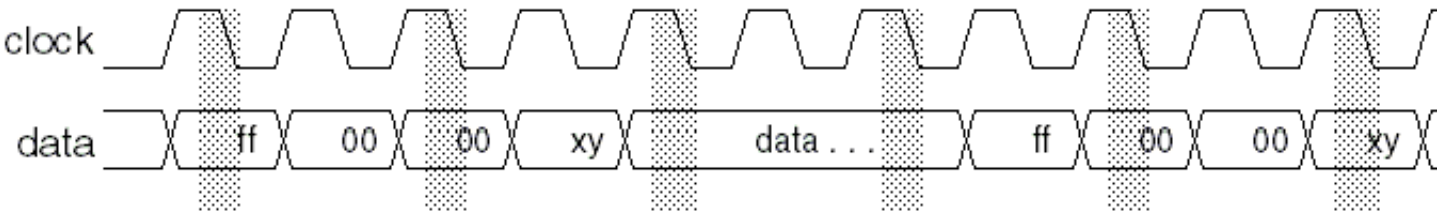


Figure 45-19. BT.656 mode

In this diagram the first three words are 0xff, 0x00, 0x00. The fourth word is XY and it includes the timing reference.

45.4.3.5.4 BT.1120 mode

In this mode the CSI works in compliance with recommendation ITU-R BT.1120. The timing reference signals (frame start, frame end, line start, line end) are embedded in the data bus input. Each timing reference signal consists of a four word sequence. The first three words are fixed and configured in the CCIR_PRECOM register. The fourth word contains information defining field, the state of field blanking and the state of line blanking. these states are configured in registers CCIR_CODE_1 (for field 0) and CCIR_CODE_2 (for field 1).

In this mode, the CSI can also work in DDR mode - data arrives on every edge of the clock. In addition, in each cycle two value of data arrive.

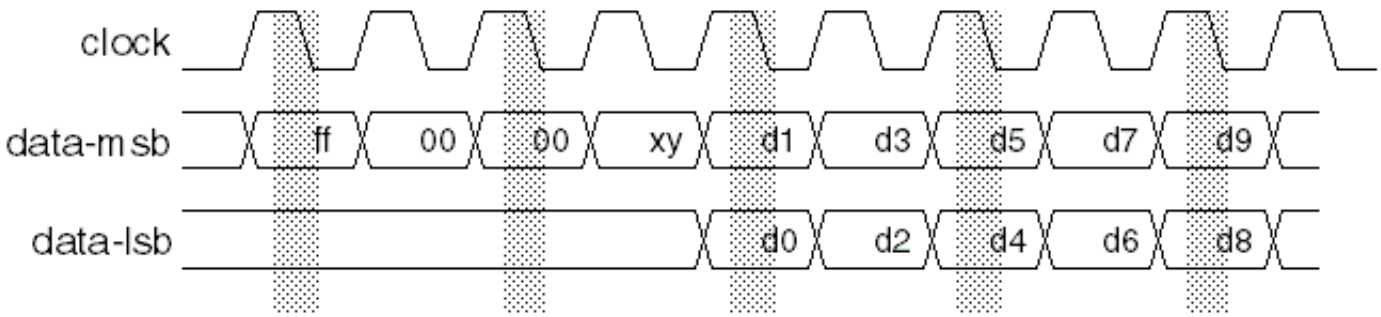


Figure 45-20. BT.1120 mode - SDR mode

In the above diagram each data arrives with the positive edge of the clock.

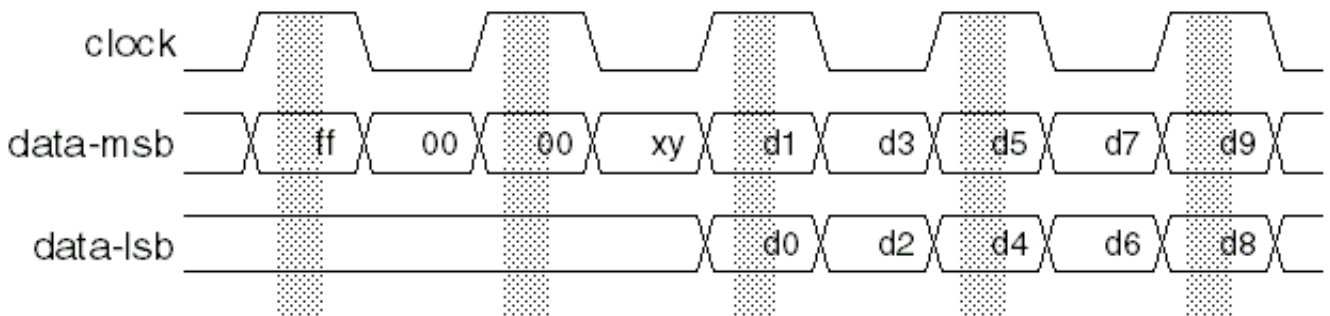


Figure 45-21. BT.1120 mode - DDR mode

In the above diagram, data arrives in the positive edge and the negative edge of the clock.

45.4.3.6 Packing to memory

The data bus output to the SMFC is 128-bit.

The following table shows the how the CSI performs the packing before sending the data to the SMFC.

The data bus output to the SMFC is 128-bit. The following table shows the how the CSI performs the packing before sending the data to the SMFC.

Table 45-14. Packing Unit

data format	component size	companded	regular packing	tight packing
Bayer, Generic data, JPEG	8	{8DC[1] ,8DC,8DC,...,8DC}	{8D,8D,8D,...,8D[2] }	NA
	9-16	{8DC,8DC,8DC,...,8DC}	{16DE[3] ,16DE,...,16DE}	NA
RGB, YUV	8	{8DC,8DC,8DC,8R}	{8D,8D,8D,8R}	NA
	9-10	{8DC,8DC,8DC,8R}	{16DE,16DE,16DE,16R[4] }	{10DE,10DE,10DE,2R}
	11-16	{8DC,8DC,8DC,8R}	{16DE,16DE,16DE,16R}	{10DT[5] ,10DT,10DT,2R}

1. DC - data after being companded
2. D - data arrived from sensor.
3. DE - data after being extended.
4. R- reserved bits
5. DT - data after being truncated.

The tight packing functionality is enabled when PACK_TIGHT register is set. It is only used when data format is RGB or YUV and the data width is bigger than 8.

45.4.3.7 Skipping frames

Some of the frames that are sent to the SMFC can be skipped. skipped frames are ignore by the CSI and are not sent to the corresponding unit. Using SKIP_SMFC and MAX_RATIO_SKIP_SMFC registers the user can define the frames for the SMFC that will be skipped.

45.4.3.8 16 bit camera support

Devices that support 16 bit data bus can be connected to the CSI. This can be done in one of the following ways.

16 bit YUV422

In this mode the CSI receives 2 components per cycle. The CSI is programmed to accept the data as 16 bit generic data. The captured data will be stored in the memory through the SMFC. The IDMAC needs to be programmed to store 16bit generic data. When the data is read back from the memory for further processing in the IPU it will be read as YUV422 data.

16 bit RGB as generic data

In this mode the CSI receives 3 components per cycle. If the external device is 24bit - the user can get connect a 16 bit sample of it (such as RGB565) The CSI is programmed to accept the data as 16 bit generic data. The captured data will be stored in the memory through the SMFC. The IDMAC needs to be programmed to store 16bit generic data. When the data is read back from the memory for further processing in the IPU it will be read as 16 bit RGB data. The IDMAC's mapping unit will be used

to remap the 16 bit data to the internal 24bpp RGB format In this mode on the fly processing is can't be performed. The data has to be sent to the memory first and then further processed by the IPU.

16 bit RGB565

This is the only mode that allows on the fly processing of 16 bit data. In this mode the CSI is programmed to receive 16 bit generic data. In this mode the interface is restricted to be in "non-gated mode" and the CSI#_DATA_SOURCE bit has to be set If the external device is 24bit - the user can connect a 16 bit sample of it (RGB565 format). The IPU has to be configured in the same way as the case of CSI#_SENS_DATA_FORMAT=RGB565

45.4.3.9 CSI Restrictions

The frequency of the sensor clock must not be greater than the IPU clock (HSP_CLK)

SENS_FRM_HEIGHT >= VSC + ACT_FRM_HEIGHT

SENS_FRM_WIDTH >= HSC + ACT_FRM_WIDTH

45.4.4 Sensor Multi FIFO Controller (SMFC)

The Sensor Multifile Controller used as buffer between CSI and IDMAC. Two masters (CSIs) can be connected to SMFC. Both masters can be active simultaneously.

Each master can send up to 4 frames, distinguished by csi_id bus. The frame can be mapped to one of four IDMAC channels via SMFC mapping registers. Each DMA channel have dedicated FIFO.

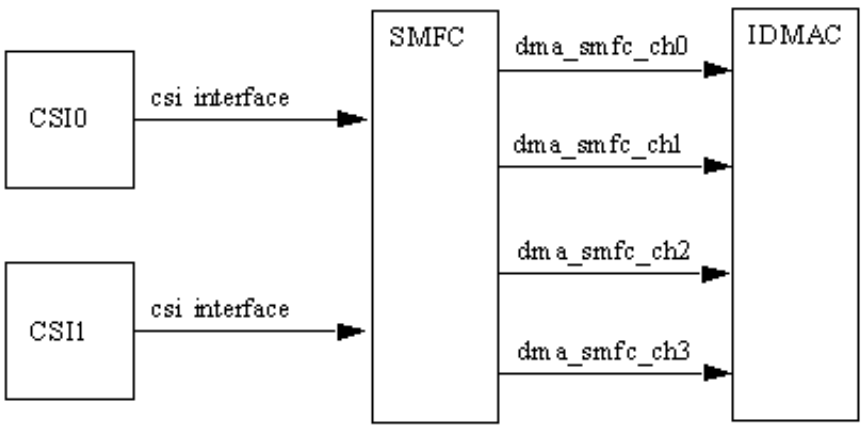


Figure 45-22. SMFC data flow

45.4.4.1 SMFC's Features

- Support two CSI masters and four DMA channels
- Automatic FIFO size calculation

45.4.4.2 SMFC's Functional description

SMFC supports up to four DMA channels. Each channel has a dedicated FIFO controller, as shown in the following figure. Sampled data and frame ID are kept in the buffers until the buffer is selected by Round Robin Priority Mechanism. Then, the content of ID buffer is compared to CH#_MAP bits and the corresponding FIFO controller is activated. As a result, the content of the buffer is copied to the RAM. The wptr and the "base" are used to calculate the location in the RAM. The rptr are following after wptr during dma_active signal, initiated by DMA.

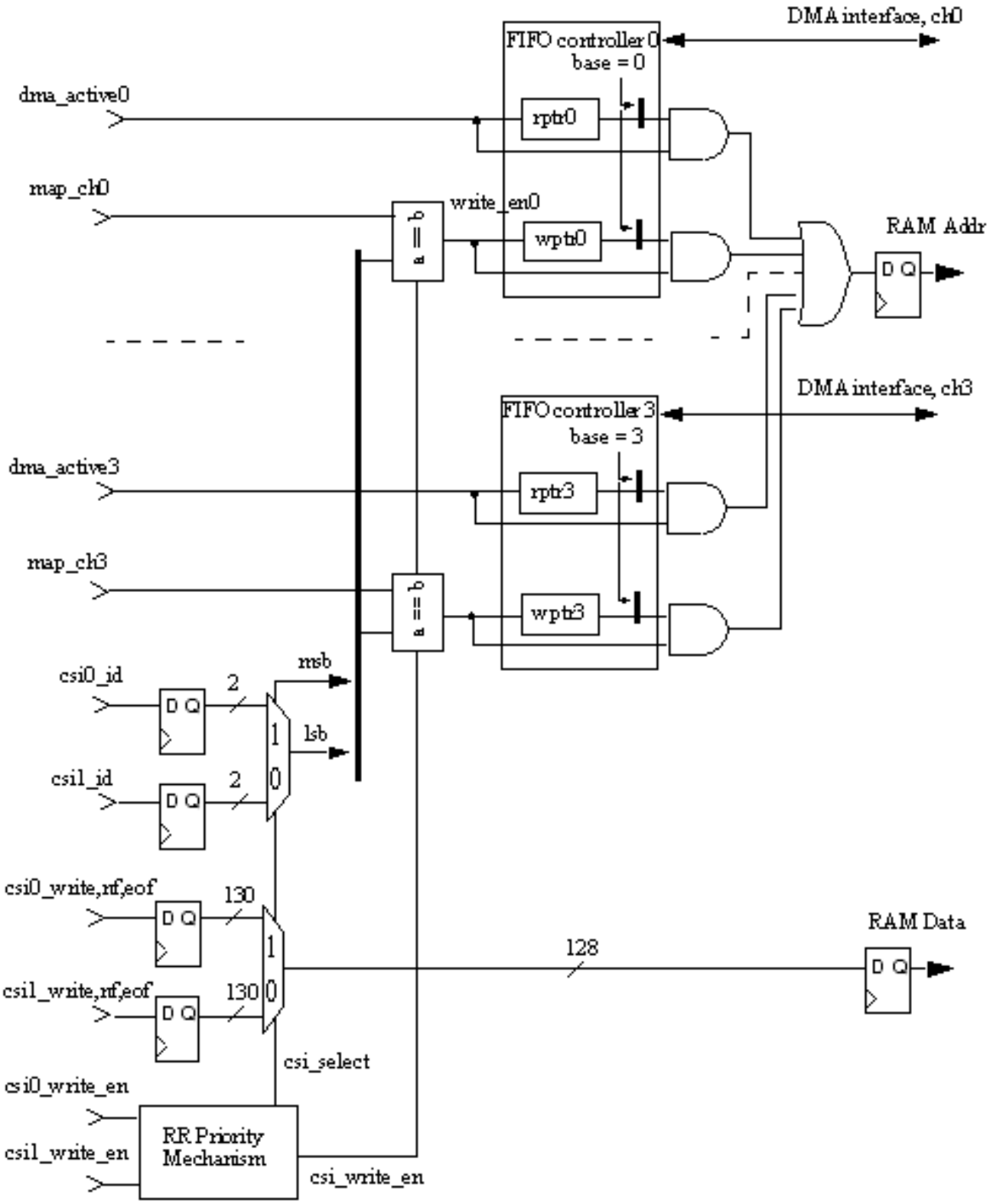


Figure 45-23. Sensor Multi FIFO Controller

Four FIFOs are implemented with single RAM due to the fact that the channels can't be active simultaneously. The memory space of SMFC is divided into four equal sectors. Each FIFO has a fixed base address - "base". The "base" used as MSB for corresponding pointer in order to calculate absolute address of the RAM. FIFO size of channels 1 and 3 are fixed and is equal to size of one sector. FIFO size of channels 0 and 2 depend from other channels as shown in the table below. Other configuration are not allowed.

Number of DMA channels required	enable/disable of channels 3,2,1,0	FIFO size (sectors) per channels 3,2,1,0
1	0 0 0 1	0 0 0 4
2	0 1 0 1	0 2 0 2
3	1 1 0 1	1 1 0 2
4	1 1 1 1	1 1 1 1

NOTE

Channels should not be enabled after activation of channel 0 or/ and channel This can cause to overlapping of FIFO areas and other malfunctions.

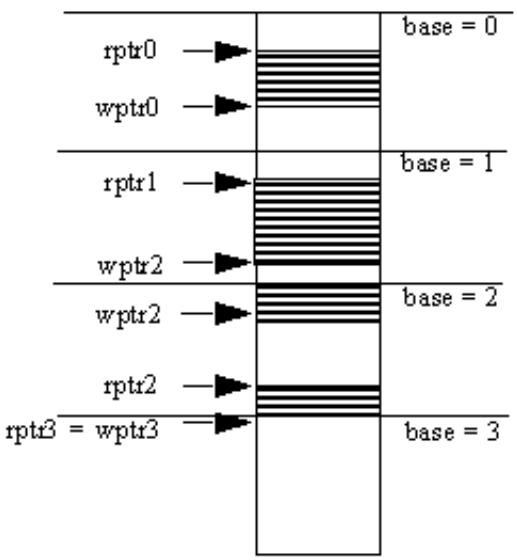


Figure 45-24. SMFC memory map when DMA channels 3,2,1,0 are enabled

45.4.4.2.1 SMFC Master interface.

SMFC Master interface is shown in the following figure.

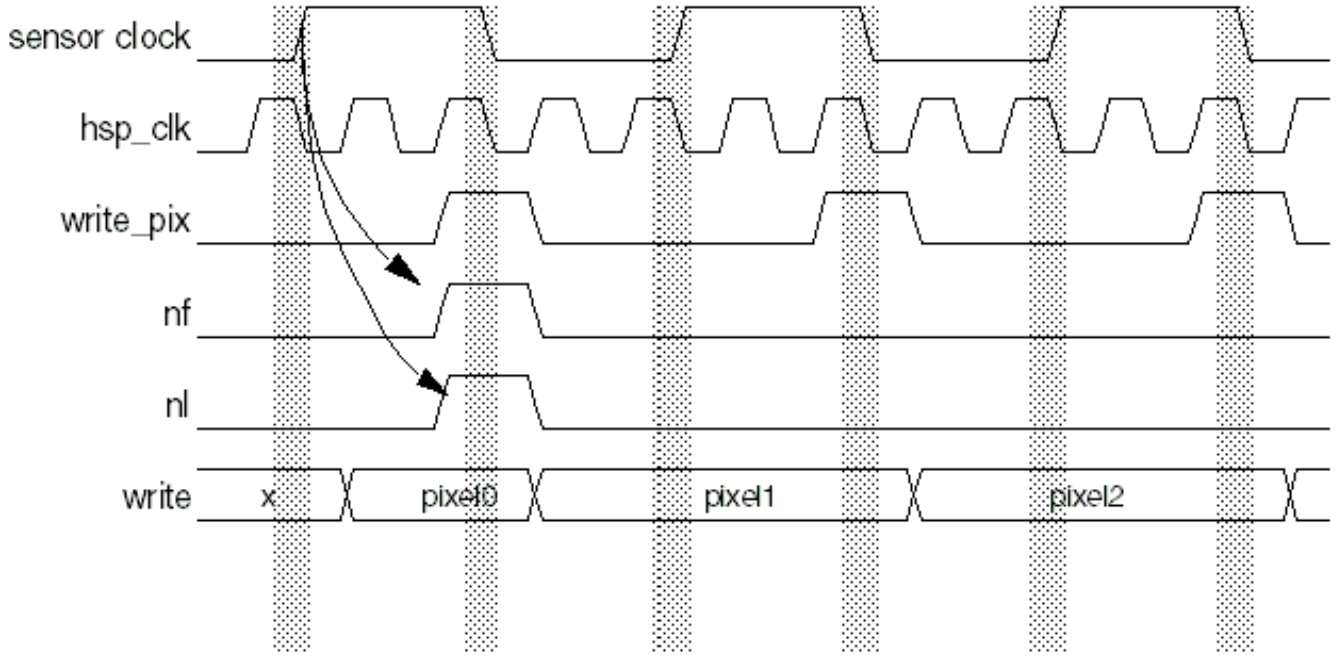


Figure 45-25. Timing diagram of SMFC slave interface

Sensor clock can be asynchronous to IPU clock (hsp_clk). The CSI synchronize data arrived from the sensor to hsp_clk. The csi_write signal indicates when data on csi_pix bus can be sampled by hsp_clk clock.

45.4.4.2.2 Restrictions

1. DMA channels should not be enabled after activation of channel 0 or/and channel 2. This can cause to overlapping of FIFO area and other malfunctions.
2. Watermark set value should be greater than watermark clear value.
3. One frame should not be mapped to few DMA channel.

45.4.5 Image Converter

45.4.5.1 IC Block Diagram

The IC Block Diagram is shown in the following figure.

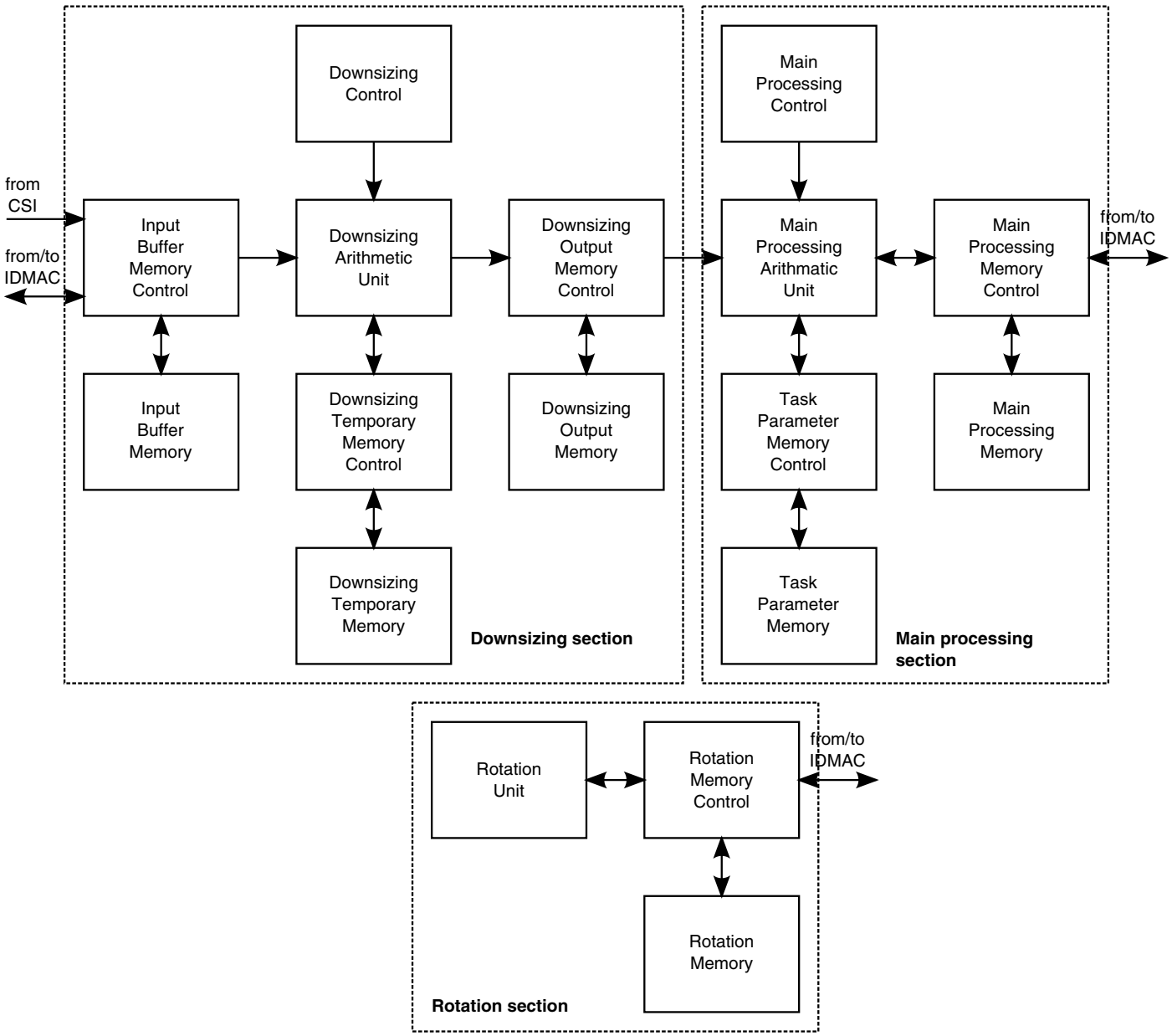


Figure 45-26. IC Block Diagram

The IC contains three processing sections: downsizing, main processing and rotation. The block is controlled via the peripheral bus registers. Some processing parameters should be written by the ARM core to the Task Parameter Memory. Writing to the memory is performed via the AHB bus.

45.4.5.2 Processing tasks

Each of the three processing sections performs up to three processing tasks with time sharing:

1. Preprocessing task for encoding.
2. Preprocessing task for displaying image from sensor (viewfinder).
3. Postprocessing task.

The tasks are performed by single hardware. The ARM platform configures each task before enabling it.

Tasks switching is transparent for the ARM platform. The time unit for task switching in the downsizing section corresponds to a processing time of one burst of eight pixels, in the main processing section - to a processing time of one image line, in the rotation section - to a processing time of one image frame.

All three tasks include similar operations controlled by commands. Task configuring consists in definition of commands for each task, as described in the following table.

Table 45-16. Task Commands

Command code	Command	Processing unit	Command parameters	Description
EN	Task Enable	DSU, MPU		Task will enabled from next frame. Task 1 is preprocessing for encoding. Task 2 is preprocessing for viewfinder. Task 3 is postprocessing.
	Downsizing	DSU	Downsizing ratio (GCR)	Downsizing ratio 1:1, 2:1, 4:1
	Resizing	MPU	Resizing ratio (GCR)	Resizing ratio from 2:1 to 1:M Resizing ratio = N:M; M =2^13; N = floor(M*(SI-1)/(SO-1)); SI - input size; SO - output size
CSC1	Color space conversion 1	MPU	Color conversion coefficients and offsets (TPM)	Color conversion matrix 1
GLOB_A	Global alpha	MPU		Used only for Task 2 and Task 3
CMB	Combining	MPU		Combining video with graphics. Used only for Task2 and Task3.

The ARM platform writes the commands to the IC_CONF Register. Because there is no double buffering for all the IC parameters, the ARM platform must disable a task before changing its parameters. After being disabled, the task is still allowed to complete its

current frame execution. At frame finish, the IPU sends an interrupt to the ARM platform indicating that the ARM platform can change task parameters. The ARM platform enables the task again and task execution is resumed from start of the next frame.

45.4.5.3 Downsizing Section

The sensor data from the CSI is written into a FIFO located in the Input Buffer Memory. Depending on programmed processing flow, the FIFO data can be sent to the system memory via the IDMAC or straight forward to the Downsizing Unit.

In the first case, the data is processed by the ARM platform and returned by the IDMAC to another FIFO located in the Input Buffer Memory.

For postprocessing, the IDMAC transfers a data from the system memory to the third FIFO. The data is read by the Downsizing Unit when the postprocessing is performed.

Each or three FIFOs has 128 pages. Each page can store one burst of 2 or 4 words which corresponds to 8 or 16 pixels. The size of the burst is defined according to the channel's corresponding CB#_BURST_SIZE bit. The memory word width is 128 bits. Each memory word contains color components of four adjacent pixels or 16 bytes of generic data (e.g. Bayer). Access to the FIFOs is controlled by the Input Buffer Memory Control.

The Downsizing Unit performs averaging and decimation of image pixels both in horizontal and vertical directions according to the following equations:

$$HP_{R,c} = \frac{1}{DS\_R\_H} \sum_{k=0}^{DS\_R\_H-1} IP_{r+k,c}$$

$$VP_{R,C} = \frac{1}{DS\_R\_V} \sum_{l=0}^{DS\_R\_V-1} HP_{R,c+l}$$

where IPr,c - the input pixel, HPR,c - the pixel after horizontal downsizing, VPR,C - the pixel after vertical downsizing, DS_R_H and DS_R_V - the horizontal and vertical downsizing ratios according to the IC_PRP_ENC_RSC, IC_PRP_VF_RSC and IC_PP_RSC Registers. The final calculation result is rounded to 8 bits.

Each of three downsizing tasks processes the data by bursts of 8 pixels. Normally, the current task runs until emptying the corresponding input FIFO. After finishing burst processing, the Downsizing Unit may switch between the current task and another task with higher priority, if the Input Buffer Memory has received a burst for this new task.

Averaging is performed firstly in the horizontal direction. All color components of a pixel are processed in parallel. After horizontal averaging has finished for a single output pixel, the new pixel value is added to the corresponding pixel value of a temporary row derived from previous averaging steps. This provides vertical averaging of the pixels. The temporary row is stored in the Downsizing Temporary Memory. The memory word width matches one accumulated pixel width (36 bits). There are three temporary rows stored in this memory - one per downsizing task.

After vertical averaging has been finished, the output row is written to the Downsizing Output Memory. The memory word of 48 bits includes two output pixels. For each task, the memory has a double buffer of one row. When the Downsizing Unit fills the foreground part of the double buffer, the Main Processing Unit takes pair or pixels from the background part. After the new downsized row has been ready, the foreground and background memory pointers are swapped.

45.4.5.4 Main Processing Section

The Main Processing Unit reads pairs of pixels from the Downsizing Output Memory background part. It processes the complete pixel row for the current task and after that switches to another task if the input data for this new task is ready.

For each task, the Main Processing Unit is able to perform the following sequence of operations:

1. Horizontal flipping the image (optional) performed with reading from the Downsizing Output Memory. Flipping is enabled via the VF, HF & ROT parameters of the corresponding DMA channels (Table 45-11 and Table 45-12) responsible for output of the task results. The preprocessing task for encoding uses the VF, HF & ROT parameters from IDMAC channel #20, the preprocessing task for viewfinder - from the IDMAC channel #21, the postprocessing task - from the IDMAC channel #22.
2. Horizontal resizing by bilinear interpolation between two adjacent pixels received from the Downsizing Output Memory according to the equation:

$$HP_{R,c} = IP_{r,c} + RS\_C\_H \cdot (IP_{r+1,c} - IP_{r,c})$$

where RS_C_H - the current horizontal resizing coefficient. The calculation result is rounded to 8 bits. The resizing coefficient is calculated as

$$RS\_C\_H = \left(\sum_{k=0}^{R-1} RS\_R\_H \right) \bmod(8196)$$

where RS_R_H - the horizontal resizing ratio from the IC_PRP_ENC_RSC, IC_PRP_VF_RSC and IC_PP_RSC Registers. The RS_R_H parameter is equal to a numerator N of the resizing ratio N:M with M =2^13.

The resulting row of the horizontal resizing is stored in the Task Parameter Memory.

3. Vertical resizing by bilinear interpolation between the current and previous results of horizontal resizing. Both current and previous results of horizontal resizing is stored in the Task Parameter Memory. Resizing is accomplished according to the equation:

$$VP_{R,C} = HP_{R,c} + RS\_C\_V \cdot (HP_{R,c+1} - HP_{R,c})$$

where RS_C_V - the current vertical resizing coefficient. The calculation result is rounded to 8 bits. The resizing coefficient is calculated as

$$RS\_C\_V = \left(\sum_{k=0}^{C-1} RS\_R\_V \right) \bmod(8196)$$

where RS_R_V - the horizontal resizing ratio from the IC_PRP_ENC_RSC, IC_PRP_VF_RSC and IC_PP_RSC Registers. The RS_R_V parameter is equal to a numerator N of the resizing ratio N:M with M =2^13.

At completion of vertical resizing, this row is updated - the current result of horizontal resizing replaces the previous one.

4. First color space conversion YUV to RGB or RGB to YUV with the conversion matrix CSC1. The conversion matrix coefficients are programmable. They are stored in the Task Parameter Memory. The conversion equations are:

$$Z_0 = 2^{SCALE-1} \cdot (X_0 \cdot C_{00} + X_1 \cdot C_{01} + X_2 \cdot C_{02} + A_0)$$
$$Z_1 = 2^{SCALE-1} \cdot (X_0 \cdot C_{10} + X_1 \cdot C_{11} + X_2 \cdot C_{12} + A_1)$$
$$Z_2 = 2^{SCALE-1} \cdot (X_0 \cdot C_{20} + X_1 \cdot C_{21} + X_2 \cdot C_{22} + A_2)$$

   where for YUV to RGB: X_0=Y, X_1=U, X_2=V, Z_0=R, Z_1=G, Z_2=B, for RGB to YUV: X_0=R, X_1=G, X_2=B, Z_0=Y, Z_1=U, Z_2=V.

   All the parameters of the conversion matrix are written by the ARM platform to the Task Parameter Memory (IC Task Parameter Memory). The final calculation result is limited according to the SAT_MODE parameter and rounded to 8 bits.

5. Combining video with graphics. There are the following combining options:
   - local alpha blending,
   - global alpha blending,
   - use of key color.

   If both alpha blending and color keying are enabled, color keying has higher priority (graphic pixels of the key color are fully transparent independently on the alpha value).

   Combining mode is selected via the IC_CONF Register. The combining equation is:

$$OP = IGP \cdot \alpha + IVP \cdot (1 - \alpha)$$

   where IGP - an input graphics pixel, IVP - an input video pixel, α=(A+floor(A/128))/256 - an alpha value, A - a global or local transparency parameter. The global A is written in the IC_CMBP_1 Register, the local A arrives together with the graphics pixel.

   A graphics pixel becomes transparent when color keying is enabled and a pixel color matches a key color (independently on an alpha parameter).

The graphics data is read from a FIFO located in the Main Processing Memory. The FIFO contains 32 pages of size of 8 pixels. The size of the burst is defined according to the channel's corresponding CB#_BURST_SIZE bit. The graphics pixel format in the FIFO is RGB or RGBA or YUV 4:4:4 or YUVA. The graphics data is loaded by the IDMAC to the FIFO from the system memory.

All the operation are executed by an unified processing unit sequentially. Steps 1 and 2 cannot be interrupted by another task. All other steps can be interrupted by a task with higher priority if an input row is ready for this task. Preprocessing tasks priority is higher than postprocessing task priority.

The processing unit consists of three identical parts for each color component. All three color components are processed in parallel. Each of the processing operations can be enabled or disabled by an appropriate command according to.

The processing results are written to an output FIFO located in the Main Processing Output Memory row-by-row. The FIFO contains 32 pages, each page can include one burst of 8 or 16 pixels. The size of the burst is defined according to the channel's corresponding CB#_BURST_SIZE bit. The IDMAC transfers the output bursts to the system memory or to the display via DMFC (Channel 21only). The Main Processing Memory contains three buffers for each tasks: the temporary row buffer, the graphics FIFO and the output FIFO. Each memory word (128 bits) stores 4 adjacent pixels in formats RGB or RGBA or YUV 4:4:4.

45.4.5.5 Rotation Section

The rotation section includes the Rotation Memory, which stores an input rectangular block of 8x8 pixels and an output FIFO containing four pages of 8 pixels each one. The Rotation Memory word width corresponds to four adjacent pixels - 96 bits. The input block is loaded to the memory by the IDMAC like to a FIFO.

The Rotation Unit rewrites pixels from the input block to the output FIFO with corresponding relocation of a pixel inside the block. Rotation and/or left/right flipping and/or up/down flipping are enabled separately for each of three tasks. Configuring the rotation and flipping options is performed via the VF, HF & ROT parameters of the corresponding DMA channels (Table 45-11 and Table 45-12) responsible for task data input. The preprocessing task for encoding uses the VF, HF & ROT parameters from the IDMAC channel #45, the preprocessing task for viewfinder - from the IDMAC channel #46, the postprocessing task - from the IDMAC channel #47.

Rotation and flip options are shown in the following table.

Table 45-17. Rotation and Flip Options

ROT	FLR	FUD	Image
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	

Table continues on the next page...

Table 45-17. Rotation and Flip Options (continued)

ROT	FLR	FUD	Image
1	0	1	
1	1	0	
1	1	1	

After finishing the rotation task, the IDMAC returns the output FIFO content to the system memory. When writing to the system memory, the IDMAC changes a location of the block relative to an input block location in order to provide proper rotation of the whole frame. Rotation tasks switching is performed after completion of rotation of the whole frame.

45.4.5.6 IC Task Parameter Memory

The following table presents IC task parameter memory details.

Table 45-18. IC Parameters

Address[1]	Word[2]	Parameter	Field	Description
x2008	Encoding CSC1 word0 and word1	C22	8:0	Coefficients of color conversion matrix1 for encoding task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx[3];
		C11	17:9	
		C00	26:18	
		A0	39:27	Offset of color conversion matrix1 for encoding task: Offset format is sxx.xxxxxxxxx
		SCALE	41:40	Scale of coefficients for color conversion matrix1 for encoding task: 0 --> coefficients *2 1 --> coefficients*1 2 --> coefficients*0.5 3 --> coefficients*0.25
		SAT_MODE	42:42	Saturation mode for color conversion matrix1 for encoding task: 0 --> (min, max) = (0, 255) 1 --> (min, max) = (16, 240) for Z1,Z2 1 --> (min, max) = (16, 235) for Z0
x2010	Encoding CSC1 word2 and word3	C20	8:0	Coefficients of color conversion matrix1 for encoding task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C10	17:9	
		C01	26:18	
		A1	39:27	Offset of color conversion matrix1 for encoding task: Offset format is sxx.xxxxxxxxx
x2018	Encoding CSC1 word4 and word5	C21	8:0	Coefficients of color conversion matrix1 for encoding task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C12	17:9	
		C02	26:18	
		A2	39:27	Offset of color conversion matrix1 for encoding task: Offset format is sxx.xxxxxxxxx

Table continues on the next page...

Table 45-18. IC Parameters (continued)

Address[1]	Word[2]	Parameter	Field	Description
x4028	Viewfinder CSC1 word0 and word1	C22	8:0	Coefficients of color conversion matrix1 for viewfinder task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C11	17:9	
		C00	26:18	
		A0	39:27	Offset of color conversion matrix1 for viewfinder task: Offset format is sxx.xxxxxxxxx
		SCALE	41:40	Scale of coefficients for color conversion matrix1 for viewfinder task: 0 -->coefficients *2 1--> coefficients*1 2--> coefficients*0.5 3-->coefficients*0.25
		SAT_MODE	42:42	Saturation mode for color conversion matrix1 for viewfinder task: 0 --> (min, max) = (0, 255) 1 --> (min, max) = (16, 240) for Z1,Z2 1 --> (min, max) = (16, 235) for Z0
x4030	Viewfinder CSC1 word2 and word3	C20	8:0	Coefficients of color conversion matrix1 for viewfinder task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C10	17:9	
		C01	26:18	
		A1	39:27	Offset of color conversion matrix1 for viewfinder task: Offset format is sxx.xxxxxxxxx
x4028	Viewfinder CSC1 word4 and word5	C21	8:0	Coefficients of color conversion matrix1 for viewfinder task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C12	17:9	
		C02	26:18	
		A2	39:27	Offset of color conversion matrix1 for viewfinder task: Offset format is sxx.xxxxxxxxx

Table continues on the next page...

Table 45-18. IC Parameters (continued)

Address[1]	Word[2]	Parameter	Field	Description
x6060	Postprocessing CSC1 word0 and word1	C22	8:0	Coefficients of color conversion matrix1 for viewfinder task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C11	17:9	
		C00	26:18	
		A0	39:27	Offset of color conversion matrix1 for viewfinder task: Offset format is sxx.xxxxxxxxx
		SCALE	41:40	Scale of coefficients for color conversion matrix1 for postprocessing task: 0 -->coefficients *2 1--> coefficients*1 2--> coefficients*0.5 3-->coefficients*0.25
		SAT_MODE	42:42	Saturation mode for color conversion matrix1 for postprocessing task: 0 --> (min, max) = (0, 255) 1 --> (min, max) = (16, 240) for Z1,Z2 1 --> (min, max) = (16, 235) for Z0
x6068	Postprocessing CSC1 word2 and word3	C20	8:0	Coefficients of color conversion matrix1 for postprocessing task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C10	17:9	
		C01	26:18	
		A1	39:27	Offset of color conversion matrix1 for post-processing task: Offset format is sxx.xxxxxxxxx
x6070	Postprocessing CSC1 word4 and word5	C21	8:0	Coefficients of color conversion matrix1 for post-processing task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C12	17:9	
		C02	26:18	
		A2	39:27	Offset of color conversion matrix1 for postprocessing task: Offset format is sxx.xxxxxxxxx

Table continues on the next page...

Table 45-18. IC Parameters (continued)

Address[1]	Word[2]	Parameter	Field	Description
x6078	Postprocessing CSC1 word0 and word1	C22	8:0	Coefficients of color conversion matrix2 for viewfinder task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C11	17:9	
		C00	26:18	
		A0	39:27	Offset of color conversion matrix2 for viewfinder task: Offset format is sxx.xxxxxxxxx
		SCALE	41:40	Scale of coefficients for color conversion matrix1 for viewfinder task: 0 -->coefficients *2 1--> coefficients*1 2--> coefficients*0.5 3-->coefficients*0.25
		SAT_MODE	42:42	Saturation mode for color conversion matrix2 for viewfinder task: 0 --> (min, max) = (0, 255) 1 --> (min, max) = (16, 240) for Z1,Z2 1 --> (min, max) = (16, 235) for Z0
x6080	Postprocessing CSC1 word2 and word3	C20	8:0	Coefficients of color conversion matrix2 for viewfinder task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C10	17:9	
		C01	26:18	
		A1	39:27	Offset of color conversion matrix2 for viewfinder task: Offset format is sxx.xxxxxxxxx
x6088	Postprocessing CSC1 word4 and word5	C21	8:0	Coefficients of color conversion matrix2 for viewfinder task: Z0 = X0*C00 + X1*C01 +X2*C02+A0; Z1 = X0*C10 + X1*C11 +X2*C12+A1; Z2 = X0*C20 + X1*C21 +X2*C22+A2; Coefficients format is s.xxxxxxxx;
		C12	17:9	
		C02	26:18	
		A2	39:27	Offset of color conversion matrix2 for viewfinder task: Offset format is sxx.xxxxxxxxx

1. The address documented in this table is the relative address within the TPM. This is the address that should be accessed when writing or reading from this memory via the AHB bus.
2. Each word is aligned to 64 bit accessible via the AHB bus in 2 separate 32bit accesses.
3. s - sign position, x - binary digit position

45.4.5.7 IC's DMA channels

The table below mas the IDMAC channels of the IC and the IRT to the corresponding tasks.

The IC's channel name is the name of the channel at IC level. The IC channel name is referred on the IC's programming model.

Table 45-19. IC's DMA Channels

IDMAC's channel number	IC's channel name	Read/ Write	Source	Destination	Processing Flow Purpose
20	CB0	Write	IC ENC	Memory	Preprocessing data from IC (encoding task) to memory
21	CB1	Write	IC VF	Memory/DMFC	Preprocessing data from IC (viewfinder task) to memory; This channel can be configured to send the data directly to the DMFC. This is done by programming the ic_dmfc_sel bit.
22	CB2	Write	IC PP	Memory	Postprocessing data from IC to memory
14	CB3	Read	Memory	IC VF	Graphics data for combining (viewfinder task)
15	CB4	Read	Memory	IC PP	Graphics data for combining (post-processing task)
11	CB5	Read	Memory	IC PP	Postprocessing data from memory
12	CB6	Read	Memory	IC VF	Preprocessing data from sensor stored in memory (for example Bayer)
5	CB7	Write	IC	Memory	Direct data from IC (sensor data) to memory
48	CB8	Write	ENC ROT	Memory	Preprocessing data after rotation (encoding task)
49	CB9	Write	VF ROT	Memory	Preprocessing data after rotation (viewfinder task)
45	CB10	Read	Memory	ENC ROT	Preprocessing data for rotation (encoding task)
46	CB11	Read	Memory	VF ROT	Preprocessing data for rotation (viewfinder task)
50	CB12	Write	PP ROT	Memory	Postprocessing data after rotation
47	CB13	Read	Memory	PP ROT	Postprocessing data for rotation

45.4.5.8 IC restrictions

- The input's frame width to the IC must be a multiplication of 8 pixels
- When performing resizing the frame width must be multiple of burst size - 8 or 16 pixels as defined by CB#_BURST_16 parameter.

45.4.5.9 IC bridge

The IC sub-block utilizes a single memory to serve read and write channels. These memories are the IBM, RM and MPM. The IDMAC has separate mechanism to handle read and write channels. As a result, a contention between read and write channels may occur on each one of the memories. In order to resolve the contention, an IC bridge sub-block is connected between the channels associated with this memory.

The bridge prioritizes a read channel over a write channel. In order to avoid starvation of the write channels, the user can limit the maximum consecutive requests of the same channel that will be served. The limitation is done by programming the memory's corresponding <>_brdg_max_rq field.

45.4.6 Display port

The display port handles all the IPU features targeted for controlling and sending data to the display. The display port consists of 4 modules.

DC - a display controller,

DP - a display processor,

DMFC - a display multi-FIFO controller

DI - a display interface. The DI is instantiated twice to provide two symmetrical display interfaces.

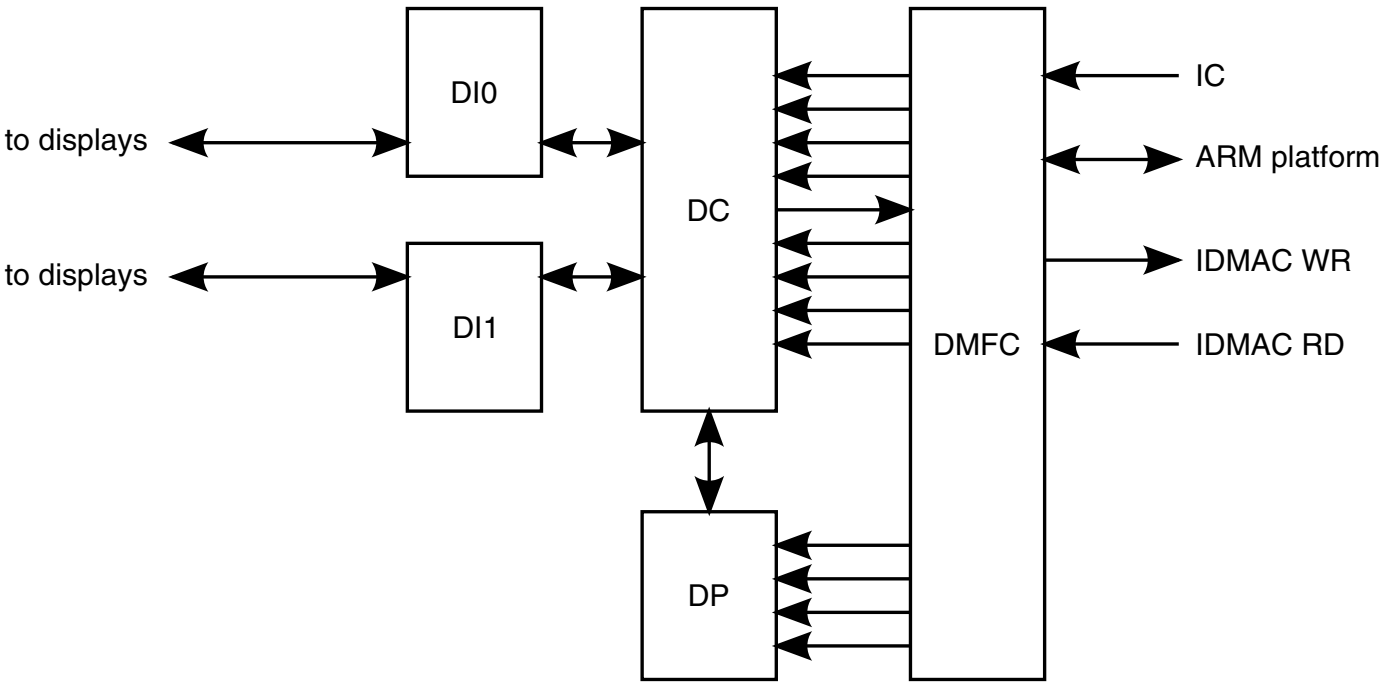


Figure 45-27. The display port

45.4.6.1 Display ports channels

The display port send data to the display over several channels. The data source may be the system's memory (via the IDMAC) the IC block and the ARM platform. The display port can read data from the display and send it to the ARM platform or IDMAC.

The data is routed over channels. The table below maps the IDMAC channels to DMFC, DC, DP channels and describes each channel.

Table 45-20. Display port channels

IDMAC's channel number	Display port's destination	DMFC channel number	DC channel number	Corresponding alpha channel	channel usage
21	DC	Programmable using dmfc_ic_in_port	Programmable using dmfc_ic_in_port	NA	This channel is coming from the IC module. When data is coming from the IC, the IC channel replaces one of the IDMAC's channels connected to the DMFC. When the IC_DMFC_SEL bit is set the output of the IC is routed to the DMFC. This channel can be routed to the DC channels 1,2,5B,5F,6B,6F. Routing this channel to the DC channel is done with the dmfc_ic_in_port bits. The DC's channel allocated to the IC channel can't be used for data coming from other source.

Table continues on the next page...

Table 45-20. Display port channels (continued)

IDMAC's channel number	Display port's destination	DMFC channel number	DC channel number	Corresponding alpha channel	channel usage
23	DP	5B	5	51	This channel is for the DP's primary flow. When a single plane is used, the data should be sent over this channel. When 2 planes are combined in this flow, the second plane should come on channel 27.
24	DP	6B	6	52	This channel is for the DP's secondary flow. When a single plane is used, the data should be sent over this channel. When 2 planes are combined in this flow, the second plane should come on channel 29.
27	DP	5F	5	31	This channel is for the DP's main flow. When a single plane is used, the data should be sent over channel 23 and this channel should not be used. When 2 planes are combined in this flow, one plane should come on channel 23 and the other plane on this channel.
28	DC	1	1	NA	This channel can serve sync and async flows. When channel 28 is connected to DI0, channel 23 must be connected to DI1 even if ch23 is not used. This is done by programming the PROG_DISP_ID_5 field
29	DP	6F	6	33	This channel is for the DP's secondary flow. When a single plane is used, the data should be sent over channel 24 and this channel should not be used. When 2 planes are combined in this flow, one plane should come on channel 24 and the other plane on this channel.
40	DC	0	0	NA	This is a read channel
41	DC	2	2	NA	This channel can serve only async flow
42	DC	1C		NA	Command stream. See Display port's restrictions
43	DC	2C		NA	Command stream. See Display port's restrictions
44	DC	3		NA	Mask channel. This mask channel can be associated with channel 23 or channel 28

45.4.6.2 Supported display interfaces

- The display port has 2 DI interfaces. Each interface can handle up to 3 displays.
- The total number of supported displays is 4.
- Each DI can handle up to 2 async interface - only one of them can be serial interface.
- Each DI can handle one synchronous interface. Asynchronous displays that are accessed in synchronous mode are considered synchronous interface.

45.4.6.2.1 Synchronous Interfaces

The DI supports the following synchronous display interfaces.

1. Synchronous generic interfaces to TFT dumb displays or RGB interfaces of smart displays.
2. Synchronous interfaces to Sharp displays.
3. Synchronous interfaces to TV encoders:
   - PAL
   - NTSC

   TV interfaces can operate in progressive or interlaced modes.
4. Synchronous interface to a graphic accelerator
5. BT.656
6. BT.1120

   BT.1120 and BT.655 support

   BT.1120 and BT.656 are supported. Only video data is supported, sending data during blanking intervals is not supported. The component size is always 8 bit.

45.4.6.2.2 Asynchronous Parallel Interfaces

The DI has a flexible asynchronous interface. The interface include 2 chip selects (CS) and 7 general purpose control signals.

The user can decide which of the 7 signals will be associated to each one of the asynchronous displays (up to 2 displays per DI). The using can configured some of the control signals to be shared by more than one display.

45.4.6.2.3 Asynchronous Serial Interfaces

See Serial display interface

45.4.6.3 Display port's bandwidth

When the IPU clock (HSP_CLK) is equal to 200 Mhz, the peak bandwidth supported by the display port is as follows

For on chip devices

- 170 Mega-Accesses/Sec if the DI clock is derived from an external to IPU source (like another PLL)
- 180 Mega-Accesses/Sec if the DI clock is derived from the IPU clock (HSP_CLK)

For off chip devices (Like an external LCD)

- 170 Mega-Accesses/Sec if the DI clock is derived from an external to IPU source (like another PLL)
- 180 Mega-Accesses/Sec if the DI clock is derived from the IPU clock (HSP_CLK)

An access can be a pixel, component or generic data.

45.4.6.4 Display Dual Mode

This mode is useful for smart display with synchronous interface. The data is sent to the display only when the data has changed or when the display processor's settings are changed. The physical interface to this display has synchronous interface's characteristics.

45.4.6.5 Display Errors

There are a few types of errors that may impact the image sent to the display. The IPU provides some automatic mechanisms, which allow the system to overcome these errors.

45.4.6.5.1 Data starvation errors

These types of errors may happen for synchronous displays if the data is ready at the DI to be sent to the display at the time point it is required. This anomaly may happen if the system is very loaded and the IDMAC was not able to read data from the external memory and provide it to the DMFC.

Frame boundary errors

In case that a new frame should be sent to the display but the data of the previous frame was not sent completely, the IPU will reset the display modules and flush the internal buffers, as the IPU expects that the new frame indication will be triggered at least 2 lines (blanking interval) before the actual data is to be sent. The should recover and be ready with the data of the new frame within the blanking interval period.

Error within a frame

In case that a pixel was missed within a frame i.e. the pixel did not arrived to the DI on time. The IPU has 2 ways to handle the situation. The mode is selected by configuring the DI#_ERR_TREATMENT bit.

Redo the last access. The IPU will keep sending the last access to the display. The IPU will drop any pixel that arrive till the end of the line. If the data of the next line arrives correctly the IPU will continue working normally as overcame the problem. In case that the data of the next line is also incorrect the IPU will perform the same procedure as described on frame boundary errors.

Freeze the clock. In this mode, the IPU freezes the clock sent to the display till the correct data arrives. In order to avoid a case where the clock is frozen forever and the system is stuck: when the clock is frozen, a watchdog timer starts counting. When the timer completed counting the IPU will perform the same procedure as described on frame boundary errors. The number of cycles to be counted by the watchdog timer is defined according to the DI#_WATCHDOG_MODE bits.

45.4.6.5.2 Anti tearing errors

In case of anti tearing errors, the IPU indicates about the error by asserting the corresponding DC_TEARING_ERR_# bit.

See also Antitearing control.

45.4.6.6 Display port's restrictions

- In case where 2 synchronous flows are used and additional asynchronous flow via DP is used. The asynchronous flow via DP can be targeted to the same DI that the synchronous flow via DP is targeted.
- There are only 2 command channels (42 and 42). Command channels are associated with data channels (24,28,41).
  - When channel 28 is associated with a command, the command stream will come from channel 42
  - When channel 41 is associated with a command, the command stream will come from channel 43
  - Channel 24 can be associated with a command stream only if channel 28 or channel 41 do not use a command stream. If channel 28 is not associated with a command stream then channel 24 can be associated with channel 42. If channel 41 is not associated with a command stream and channel 28 is associated with a command stream then channel 24 can be associated with channel 43.

- A channel that uses an alternate flow, cannot be associated with a command stream (ch. 24 or ch. 41)
- In case of a synchronous display using external clock, The DI where the synchronous display is connected to can be connected to another async display. But, it can support only the write direction. Read via the asynchronous interface cannot be performed if this DI uses external clock

45.4.7 DC - Display Controller

IPU handles few display flows supporting few displays. The IPU's flows' data sources can be the ARM platform, the system's memory, a camera or an external device connected on the display's port such as an external graphic accelerator. The data's destination can be any device connected on one of the DI ports.

The DC controls the flows coming to and from the DI port. The DC manages the flows, decides which flows are currently active and when each flow is activated. The arbitrates between the active flows, gets the data from the predefined source and distribute it to the correct DI.

The DC's core is the microcode. The microcode contains a set of routine. A routine is built of a set of commands stored in the template's (microcode) memory. For each event (like new frame, end of frame etc.) a specific routine is executed. The user writes the routines and map them to a specific events. The routine contains instructions to the DC about the way of handling the data/address/commands associated with the display. The routine may contain information about the data's mapping, about waveform's characteristics, and more.

The figure below shows the micro architecture diagram for the DC block

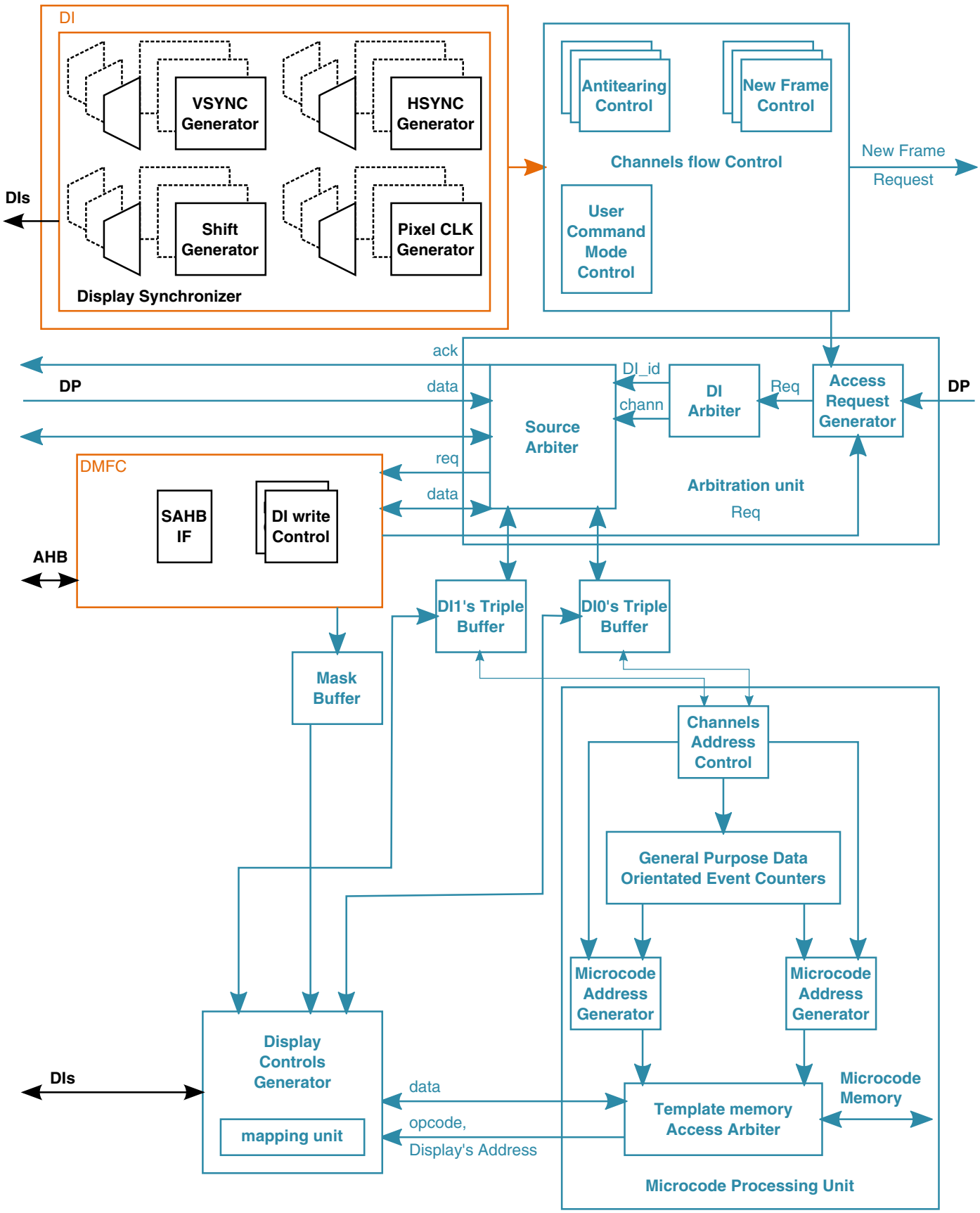


Figure 45-28. DC - Display Controller Block Diagrar

45.4.7.1 Channels flow control

45.4.7.1.1 New Frame control

The channels flow control schedules the flows handled by the DC

For asynchronous flows the scheduling is done according to a request coming from the Frame Synchronization Unit (FSU) on the control sub-block (CM).

For Synchronous flows the scheduling is done according to a trigger that is generated by a timer located on the one of the display's interface blocks (DI)

45.4.7.1.2 Antitearing control

Anti tearing mechanism uses a signal indicating on a display's refresh of a frame.

The supported tearing elimination triggers can be:

- An internally generated VSYNC signal
- A VSYNC signal generated by the display. The data source is the memory.
- A VSYNC signal coming from the CSI

The DC has the capability to avoid image tearing. For asynchronous flows where the source of the data is the system's memory or postprocessing, the DC monitors the position of a display's refresh pointer. Writing to the display is started only after crossing the window start point by the display refresh pointer. After that, writing to the display is allowed only when a write pointer does not advance beyond the refresh pointer. To provide anti-tearing mode, a window start time (in rows) must be defined in the PROG_START_TIME_1, PROG_START_TIME_2, PROG_START_TIME_5, PROG_START_TIME_6 registers for the corresponding channels.

The antitearing mechanism is limited to a case where only asynchronous flows are handle via the target DI.

In the case when tearing cannot be avoided (when the refresh rate is too high and the refresh pointer overtakes the write pointer after full refresh cycle), an error interrupt is generated. Tearing elimination mode can be disabled via the PROG_CHAN_TYP field for the corresponding channel.

When the data arrives directly from the camera via IC, tearing can be avoided only for the 2:1 ratio between a sensor scan rate and a display refresh rate. The additional conditions for tearing elimination in this case are to synchronize the sensor and display VSYNCs (by programming the VSYNC's settings on the DI) and full-screen image from the camera via IC channel.

45.4.7.1.3 User command mode control

A user may prepare in the system's memory a buffer that holds commends to be sent to the external device. The command buffer includes the same amount of lines as the data buffer. The line of commands are sent to the display line by line. A line of data is sent following each line of commands. This mode is activated by programming the PROG_CHAN_TYP of the corresponding channel.

The structure of the command is as follows:

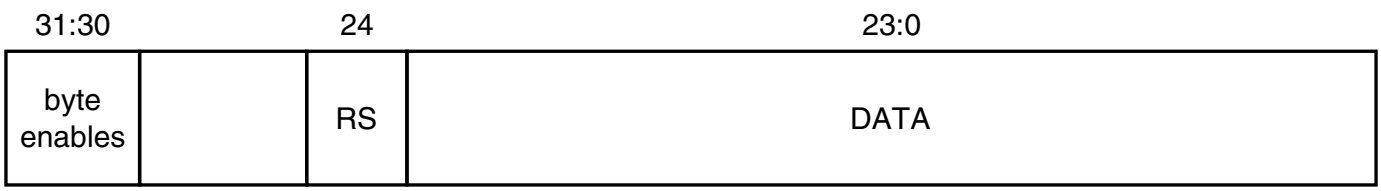


Figure 45-29. Structure of a command word

45.4.7.2 Arbitration Unit

This unit arbitrates the requests coming from the DMFC and from the DP and sends them to corresponding DI.

45.4.7.2.1 Access request generator

The requests coming from the DMFC or DP are sorted according to the target DI and then served according to a hard coded priority. The priority order is Sync flow, ARM platform access, IDMAC's Async flows.

45.4.7.2.2 DI arbiter

This units arbitrates between the DIs. The priority is hard coded. The priority order is Sync flow, ARM platform access, IDMAC's Async flows.

In case of 2 simultaneous requests to different DIs with the same priority the requests are served in a Round-Robin fashion. In case of 2 simultaneous Sync flow requests to different DI, the user can bypass the Round Robin mechanism and prioritize one DI on the other according to the SYNC_PRIORITY_1 & SYNC_PRIORITY_5 bits. Setting low priority to both of these channels is forbidden.

45.4.7.2.3 Source arbiter

Once the source of the request to be served was selected and the target DI was chosen, this unit routes the request and all the signals, associated with it, to the correct triple buffer and to the correct source of the data.

45.4.7.3 Microcode processing unit

The main control unit of the DC is the Microcode processing unit (MPU). The data coming to the DC may be associated with some additional information like new frame, new line, new address etc. The information is processed in the MPU. The MPU executes the associated routine. The routine includes a set of instructions of the actions to be performed by the DC and DI.

45.4.7.3.1 Channels address control

This unit controls the display's address for each channel. This unit is responsible for defining the next address to be accessed (by incrementing or jumping). Stores special events flags (like EOF, EOL etc.). Based on the display's address and the special events, the type of the routines to be executed is defined.

45.4.7.3.2 General purpose Data oriented events counters

A user may define up to 4 general purpose events. The events are triggered by a standard event (NF, NL). The standard event restarts a counter, when the counter completes counting the user's general purpose event is asserted. This event activates a routine like any other events

45.4.7.3.3 Microcode address generator

This unit calculates the physical address of the template memory where the event associated routine resides. In addition, these unit arbitrates between simultaneous events and select the event to be served.

The arbitration is done according to a user defined priority. The priority of each event is set according to the #_CHAN_PRIORITY_CHAN_# bits of each channel. There are 2 modes of arbitration:

- Serving all the pending events according to the priority
- Serving only the highest priority event while ignoring all the other events

The arbitration mode is defined according to the CHAN_MASK_DEFAULT bit of each channel.

45.4.7.3.4 Template's Memory Access Arbiter

This unit gets memory access requests from the 2 microcode address generator units and arbitrates between them in a Round Robin fashion. In case that only one of the requests belongs to a synchronous flow, this request is selected.

45.4.7.4 DC's Template structure

The template memory contains 256 template words. Each template word is a 42 bits words. Accessing a template word require 2 accesses (32bit each).

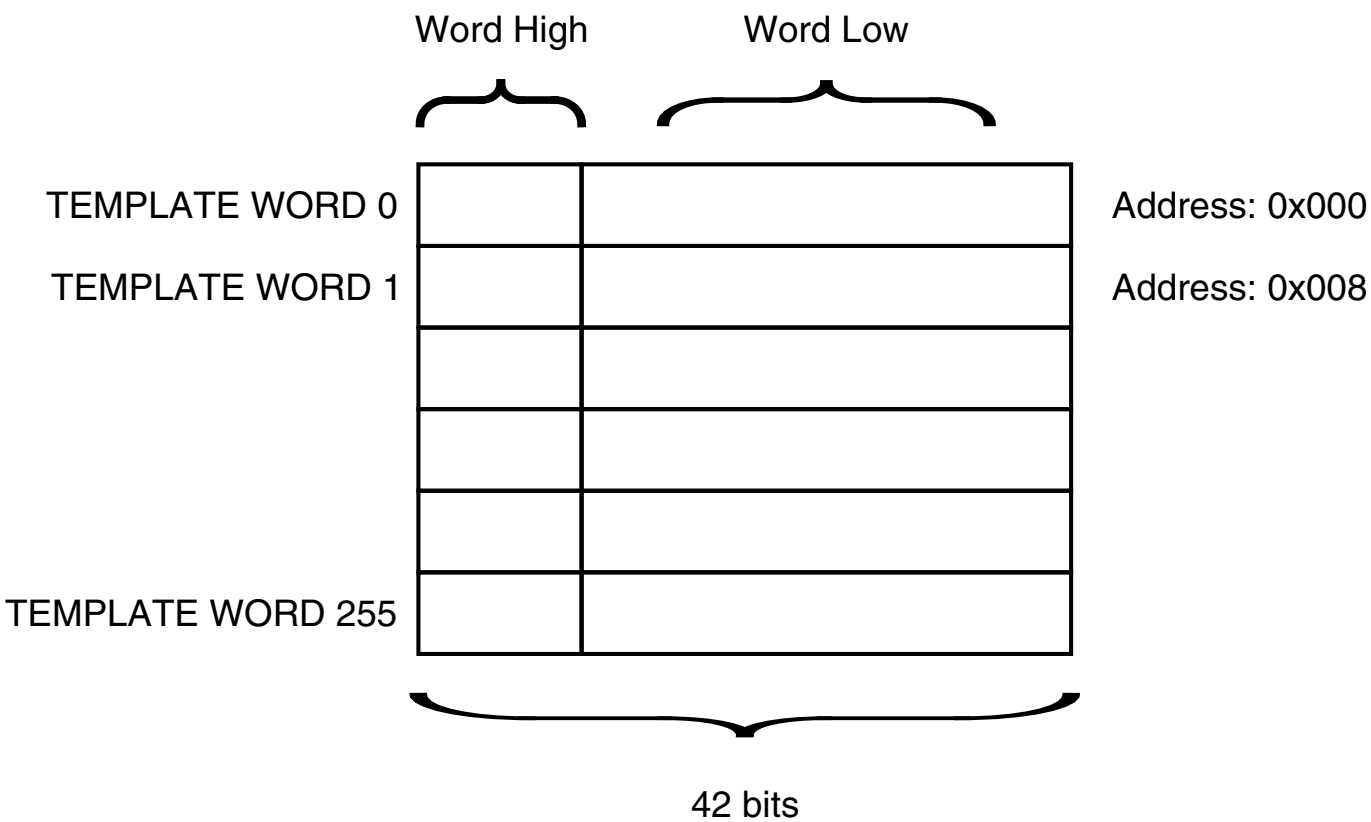


Figure 45-30. Template's structure

45.4.7.4.1 DC template's memory map

Table 45-21. DC template's memory map

0x1F80000 DC_MICROCODE_W0_L																
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Table continues on the next page...

Table 45-21. DC template's memory map (continued)

R	OPERAND												MAPPING			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAP PING	WAVEFORM				GLUELOGIC							SYNC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1F80004 DC_MICROCODE_W0_H																
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	STO P	OPCODE								
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 45-22. DC template's memory map

0x1F080000 DC_MICROCODE_W0_L																
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OPERAND												MAPPING			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAP PING	WAVEFORM				GLUELOGIC							SYNC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1F080004 DC_MICROCODE_W0_H																
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	STO P	OPCODE								
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DC template's fields description

Table 45-23. DC template's fields description

Field	Description
STOP	Stop bit - This bit should be set in order to indicate that the current command is the last command of the routine
OPCODE	The command's code
OPERAND	The command's operand - for some of the commands this field can hold a parameter associated with the command
MAPPING	The MAPPING field holds a pointer to a register holding 3 fields: MAPPING_PNTR_BYTE0_X, MAPPING_PNTR_BYTE1_X, MAPPING_PNTR_BYTE2_X. This pointers point to sets of OFFSET and MASK parameters that define the mapping scheme. MAPPING = 0 means that mapping is disabled. The value in this field should be incremented by 1 to get the correct X pointer value In order to point to MAPPING_PNTR_BYTE2_0, MAPPING_PNTR_BYTE1_0, MAPPING_PNTR_BYTE0_0 the user should write 1 to the MAPPING field
WAVEFORM	For data oriented output pins. The IPU has 4 waveform generator units. The IPU holds 12 sets of waveforms' configuration registers called DI0_DW_GEN_<i> and DI1_DW_GEN_<i> The WAVEFORM field defines which one of the 12 waveforms' configuration registers is used. The DI1_DW_GEN_X register holds a pointer to one of the 4 waveform generators units for each of the data oriented pins. 0 - The waveform of the data oriented output pins is not affected 1 -Points to DI0_DW_GEN_0 or DI1_DW_GEN_0 2 - Points to DI0_DW_GEN_1 or DI1_DW_GEN_1 ... 12 - Points to DI0_DW_GEN_11 or DI1_DW_GEN_11

Table continues on the next page...

Table 45-23. DC template's fields description (continued)

Field	Description
GLUELOGIC	For signals generated by waveform generator #3; This field provides extra flexibility on the signals waveform
	GLUELOGIC[6] - This bit defines if we are in clock mode (1) or CS mode(0). 1- clock mode When the command is related to the display clock's pin then only if we are in clock mode, GLUELOGIC[5:4] are valid. 0- CS mode When the command is related to the CS pin then only if we are in CS mode, GLUELOGIC[3:0] are valid.
	GLUELOGIC[5:4] - clock mode settings 00 - Freeze the display clock following the execution of the current command 01 - Freeze the display clock before the execution of the next command to be executed 10 - Enable (unfreeze) the display clock following the execution of the current command 11 - Enable (unfreeze) the display clock before the execution of the next command to be executed
	GLUELOGIC[3] - CS mode settings 1 - Once the signal is asserted then it remains asserted (high or low according to the polarity) 0 - No impact on the waveform
	GLUELOGIC[2] - CS mode settings 1 - Once the signal is negated then it remains negated (high or low according to the polarity) 0 - No impact on the waveform
	GLUELOGIC[1] - CS mode settings 1- The current waveform can be attached to the previous waveform. If the previous waveform was asserted and the current waveform start asserted the two waveforms will be attached so the signals' waveforms will be consecutive. This impact the behavior of the previous waveform. This can be done only if GLUELOGIC[0] of the previous waveform is set to 1 0 - No impact on the waveform
	GLUELOGIC[0] - CS mode settings 1 - this bit allows the next waveform to be attached to the current waveform.
SYNC	The data associated with this command should be synchronized to the DI's one of gen_time_sync generators' output. 0000 - No sync. The data is sent without any synchronization to any event 0001 - Sync with unit #1 0010 - Sync with unit #2 ... 1010 - Sync with unit #10 1011 - 1111 Reserved

DC template's command description

The diagram below illustrates the command processing flow. The command is being fetched from the microcode memory. The Opcode is decoded. According to the decoding several controls are sent to the processing unit. In addition the processing unit gets the pixel's data and the display's address. The processed data/ address is stored in an internal register. There are 2 types of commands

HOLD - In this type of commands the data/address is stored in the register and not sent to the DI. The data/address is held in the register for further processing in the following commands.

WRITE - In this type of commands the data/address is stored in the register and also sent to the DI.

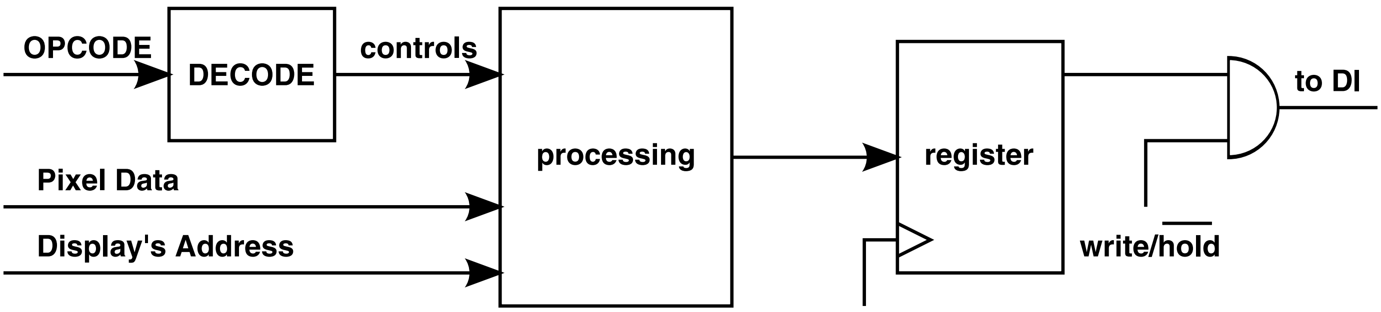


Figure 45-31. Opcode processing

The table below describes the DC's Commands.

Table 45-24. DC template's commands description

Command	COMMAND[41:0]																																									
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HLG	S	0	0	0	0	DATA																																0	0	0	0	0
	Hold 32bit word in an internal register for further processing. DATA is a general purpose data to be held.																																									
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRG	S	0	1	DATA																								WAVEFORM				GLUELOGIC							SYNC			
	Write 24bit word to the DI and Hold the word in register. DATA is a general purpose data to be written.																																									
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table continues on the next page...

Table 45-24. DC template's commands description (continued)

Command	COMMAND[41:0]																																								
HLOA	S	1	0	1	0	af	DATA																MAPPPING					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Hold the display's address in an internal register for further processing. af=0: No shift af=1: 8 bit right shift DATA is a 16bit general purpose data. This data is ORed with the 16 MSB of the mapped address.																																								
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WROA	S	1	1	1	0	af	DATA																MAPPPING					WAVEFORM				GLUELOGIC							SYNC			
	Write address to the display and Hold address in register. af=0: No shift af=1: 8 bit right shift																																								
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HLOD	s	1	0	0	0	0	DATA																MAPPPING					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Hold data in register. DATA is a 16bit general purpose data. This data is ORed with the 16 MSB of the mapped data coming from the data's source IDMAC or MCU.																																								
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WROD	s	1	1	0	0	0	DATA																MAPPPING					WAVEFORM				GLUELOGIC							SYNC			
	Write data to DI and Hold data in register. DATA is a 16bit general purpose data. This data is ORed with the 16 MSB of the mapped data coming from the data's source IDMAC or MCU.																																								
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HLOAR	S	1	0	0	0	1	1	1	af	0	0	0	0	0	0	0	0	0	0	0	0	0	MAPPPING					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Adding Mapped Address to held data and hold in an internal register. af=0: No shift af=1: 8 bit right shift																																								
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table continues on the next page...

Table 45-24. DC template's commands description (continued)

Command	COMMAND[41:0]																																								
WROAR	S	1	1	0	0	1	1	1	af	0	0	0	0	0	0	0	0	0	0	0	0	0	MAPPPING					WAVEFORM				GLUELOGIC							SYNC			
	Adding Mapped Address to held data. Write to DI and hold in register af=0: No shift af=1: 8 bit right shift																																								
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HLODR	S	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MAPPPING					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Adding Mapped Data to held data and hold in register.																																								
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table continues on the next page...

Table 45-24. DC template's commands description (continued)

<table>
<tr><th>Command</th><th colspan="42">COMMAND[41:0]</th></tr>
<tr><td rowspan="2">WRODR</td><td>S</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>M2</td><td>M1</td><td>M0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="5">MAPPPING</td><td colspan="4">WAVEFORM</td><td colspan="7">GLUELOGIC</td><td colspan="4">SYNC</td></tr>
<tr><td colspan="42">Adding Mapped Data to held data. Write to DI and hold in register.
M0:
0: a new data[7:0] before mapping
1: a previous access data [7:0] before mapping
M1:
0: a new data[15:8] before mapping
1: a previous access data [15:8] before mapping
M2:
0: a new data[31:16] before mapping
1: a previous access data [31:16] before mapping
If (M2 = M1 = M0 = 0) than the command performs:
Adding a new Mapped Data to a held data in an internal register and write it to display else a display's data will be combined from a new data and previous data according to M-flags. The combined data will be mapped according to MAPPING and sent to a display.
Examples:
previous_data = 0x89ABCDEF
new_data = 0x12345678
held_data (previous_data after mapping) = 0x0000EF
Current MAPPING mode: new_data & 0x00ffff00
Output:
If M2 = M1 = M0 = 0) than:
Output = new_data OR held_data = 0x3456EF
If M0 = 0, M1= 1,M2 = 1 than:
Output = MAPPING ({previous_data[31:8], new_data[7:0]}) = MAPPING(0x89ABCD78) = 0x00ABCD00)
If M0 = 0, M1= 1,M2 = 0 than:
Output = MAPPING ({new_data[31:16], previous_data[15:8], new_data[7:0]}) = MAPPING(0x1234CD78) = 0x0034CD00)
if M0 = 1, M1= 0, M2 = 0 than:
Output = MAPPING ({new_data[31:8], previous_data[7:0]}) = MAPPING(0x123456EF) = 0x00345600)</td></tr>
<tr><td></td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td rowspan="2">WRBC</td><td>S</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="5">MAPPPING</td><td colspan="4">WAVEFORM</td><td colspan="7">GLUELOGIC</td><td colspan="4">SYNC</td></tr>
<tr><td colspan="42">Merge 1 bit mask channel with mapped data. Hold in register and write to DI.
Mask data is coming from the DC's mask channel.</td></tr>
</table>

Table continues on the next page...

Table 45-24. DC template's commands description (continued)

Command	COMMAND[41:0]																																									
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WCLK	S	1	1	0	0	1	0	0	1	N_CLK_OPERAND													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Wait N_CLK_OPERAND clocks. N_CLK_OPERNAD is the number of DI_CLK cycles to wait.																																									
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WSTS-I	0	1	0	0	0	1	0	0	1	N_CLK_OPERAND													MAPPPING					WAVEFORM				GLUELOGIC							SYNC			
	Wait for Status I Read from the display and compare to a predefined PASSWORD. If the read data is equal to the PASSWORD, then continue. If not, redo the read & compare cycle. N_CLK_OPERAND is the number of DI_CLK cycles to wait before latching the data coming from the DI. DI_READ_DATA_ACK_VALUE_0 DI_READ_DATA_MASK_0																																									
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WSTS-II	0	1	0	0	0	1	0	1	0	N_CLK_OPERAND													MAPPPING					WAVEFORM				GLUELOGIC							SYNC			
	Wait for Status II Read from the display and compare to a predefined PASSWORD. If the read data is equal to the PASSWORD, then continue. If not, redo the read & compare cycle. N_CLK_OPERAND is the number of DI_CLK cycles to wait before latching the data coming from the DI. WSTS-II command must be followed by a WSTS-I command. This command is useful in case that the PASSWORD is read in 2 accesses. The comparison will be done only after the execution of the WSTS-I command.																																									
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WSTS-III	S	1	0	0	0	1	0	1	1	N_CLK_OPERAND													MAPPPING					WAVEFORM				GLUELOGIC							SYNC			
	Wait for Status III Read from the display and compare to a predefined PASSWORD. If the read data is equal to the PASSWORD, then continue. If not redo the read & compare cycle. N_CLK_OPERAND is the number of DI_CLK cycles to wait before latching the data coming from the DI. WSTS-III command must be followed by a WSTS-II command. This command is useful in case that the PASSWORD is read in 3 accesses. The comparison will be done only after the execution of the WSTS-I command. In case of a PASSWORD mismatch the read is done again. The entire cycle (WSTS-III -> WSTS-II -> WSTS-I) will be performed.																																									
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table continues on the next page...

Table 45-24. DC template's commands description (continued)

Command	COMMAND[41:0]																																								
RD	S	1	0	0	0	1	0	0	0	N_CLK_OPERAND													MAPPPING					WAVEFORM				GLUELOGIC							SYNC			
	Read data from DI N_CLK_OPERAND - means delay value in DI_CLK for display's data latching by DI, defined by user For serial display the N_CLK_OPERAND is fixed and should be set to 1 value.																																								
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WACK	S	1	0	0	0	1	1	0	1	0	N_CLK_OPERAND	0	0	0	0	0	WAVEFORM				GLUELOGIC							SYNC														
	Wait for acknowledge N_CLK_OPERAND - Number of DI_CLK cycles to count before monitoring the ACK received from the display.																																								
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table continues on the next page...

Table 45-24. DC template's commands description (continued)

Command	COMMAND[41:0]																																									
MSK	S	1	1	0	0	1	0	0	0	0	0	0	0	0	e0m	e1m	e2m	e3m	nfm	nlm	nfldm	eofm	eolm	eofldm	nadm	ncm	dm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Mask - Mask a specific event; In case of more than one pending events the user can mask some of the event and serve the others according to the priority. e0m - event 0 mask, defined by user e1m - event 1 mask, defined by user e2m - event 2 mask, defined by user e3m - event 3 mask, defined by user nfm - new frame mask, defined by user nlm - new line mask, defined by user nfldm - new field mask, defined by user eofm - end of frame mask, defined by user eolm - end of line mask, defined by user eofldm - end of field mask, defined by user nadm - new address mask, defined by user ncm - new channel mask, defined by user dm - data mask, defined by user																																									
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HMA	S	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ADDRESS								0	0	0	0	0
	HOLD_MICROCODE_ADDRESS - hold operand in special Microcode address register 0. This register is used to store a pointer in the microcode to be used by the BMA command ADDRESS is a microcode address in the microcode memory defined by the user.																																									
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HMA1	S	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ADDRESS								0	0	0	0	0
	HOLD_MICROCODE_ADDRESS - hold operand in special Microcode address register 1. This register is used to store a pointer in the microcode to be used by the BMA command ADDRESS is a microcode address in the microcode memory defined by the user.																																									
	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table continues on the next page...

Table 45-24. DC template's commands description (continued)

Command	COMMAND[41:0]
BMA	S 0 0 1 1 LF AF 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 N 0 0 SYNC
	BRANCH_MICROCODE_ADDRESS jump to Microcode address which is stored at an internal Microcode address register. The branch will be done to the ADDRESS stored with the HMA/HMA1 command. The BMA command manages the repeat cycles by one of the auto decrement counters (internal counters), the counter's id is defined by LF-bit (0/1) The N operand defines the N-number of routine's repetitions. In case of N=0. The BMA command functions as a JUMP command. As long as the internal counter didn't reached its predefined value, the microcode will branch to the pre defined address stored by the HMA/HMA1 commands. Selecting between HMA and HMA1 is done by the AF bit (it is calculated according to the "1 - AF" value). - That's the subroutine's microcode start address When the counter reached its predefined value, the microcode will branch to the pre defined address stored by the HMA/HMA1 commands. Selecting between HMA and HMA1 is done by the AF bit (it is calculated according to the "AF" value). - That's the subroutine's microcode return address BMA synchronization (SYNC field): The BMA command can be used for context switching between 2-channels. When switching between flows the user can run a routine for a new channel, which actually affects the previous flow. This feature is useful when a new flow breaks the current flow (post command for example): SYNC is used for display_id selection // 0 0 0 - current display // 0 0 1 - previous display // 1 0 0 - display 0 // 1 0 1 - display 1 // 1 1 0 - display 2 // 1 1 1 - display 3

45.4.7.5 Display controls' generator

This unit generates the control signals associated with the data that are sent to the DI block.

45.4.7.5.1 Bus Mapping Unit

The Bus Mapping Unit is responsible for programmable mapping of the input data and commands to the display interface format and vice versa. Address mapping is done by this unit as well (programmable via micro code).

The internal DI format for data and commands is a 24-bits word divided into three byte components (eight zeroes are added to MSB for 16-bits words from the DC). This word can be output or input in one, two, three or four cycles of the display clock.

The word coming from the memory is a 32bit word. The mapping operation is done on 24 bits only. The 24 bit are selected from the received 32 bit according to the W_SIZE_# field. The 24 bit input word is partitioned to a 3 bytes (3X8bits). Each byte can be mapped to any position at the 24bit output word. The exact position is set according to the MD_MASK & MD_OFFSET fields.

The MAPPING_PNTR_BYTE0_X, MAPPING_PNTR_BYTE1_X, MAPPING_PNTR_BYTE2_X fields holds the pointers for the MD_MASK & MD_OFFSET for each byte.

The MAPPING field holds a pointer to a register holding 3 fields: MAPPING_PNTR_BYTE0_X, MAPPING_PNTR_BYTE1_X, MAPPING_PNTR_BYTE2_X.

0 - no mapping i.e. 32 bits are sent as is.

1 - points to MAPPING_PNTR_BYTE0_0, MAPPING_PNTR_BYTE1_0, MAPPING_PNTR_BYTE2_0

2 - points to MAPPING_PNTR_BYTE0_1, MAPPING_PNTR_BYTE1_1, MAPPING_PNTR_BYTE2_1

...

30 - points to MAPPING_PNTR_BYTE0_29, MAPPING_PNTR_BYTE1_29, MAPPING_PNTR_BYTE2_29

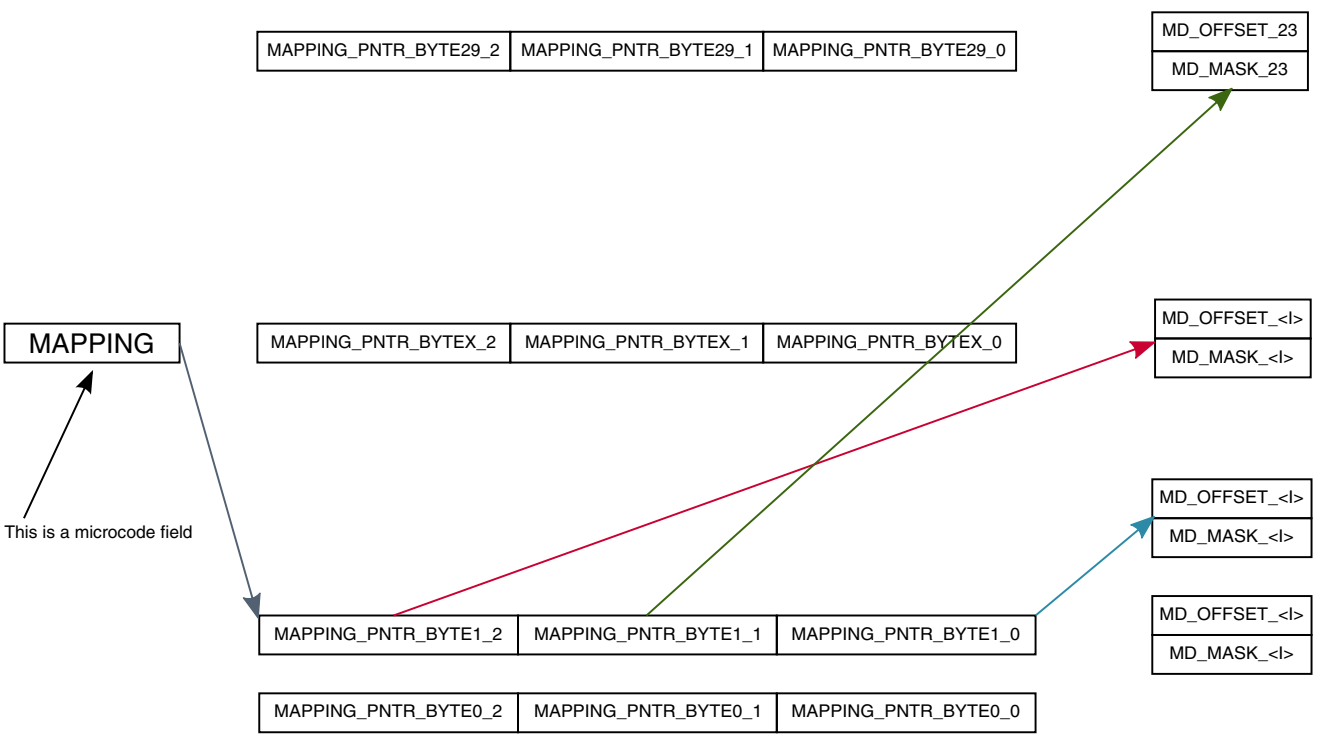
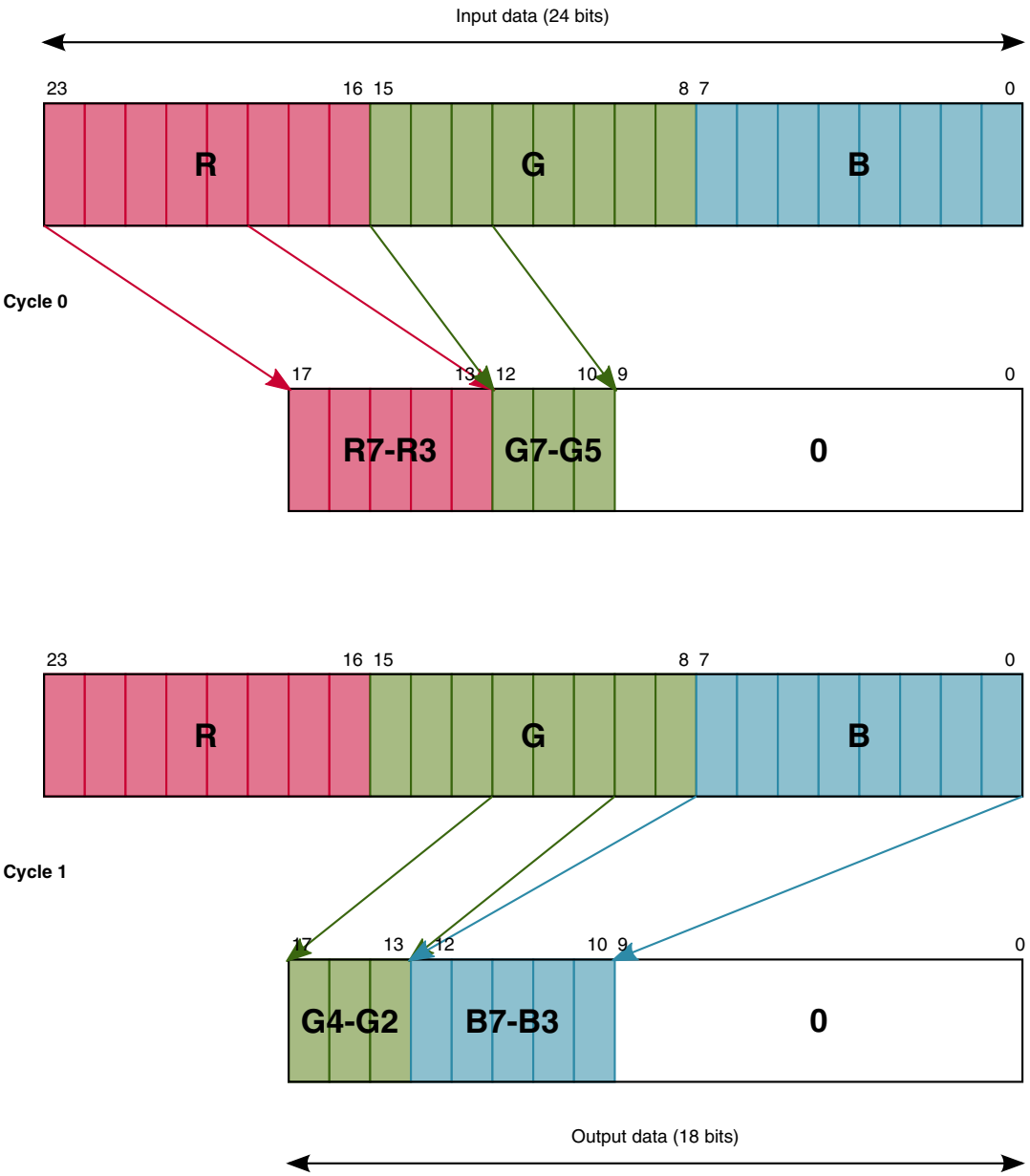


Figure 45-32. Mapping scheme

The mapping rule written in each of the Registers defines two types of parameters for the specific byte component and display:

1. Offsets of the byte component MSB relative to the output word LSB. Because the offsets can change dynamically, they are defined separately for the display clock cycles zero, one and two.
2. Numbers of the display clock cycles at which every bit of the byte component should be valid on the display bus. There are eight such 2-bit numbers in the Register.

Figure 45-33 presents an example of programming data packing for one of displays. Cycle 0 and Cycle1 in the diagram represent two separate atomic operations performed by the microcode template.



Cycle 0

MD_OFFSET_[R] - 0x11; MD_MASK_[R] - 0xF8

MD_OFFSET_[G] - 0xC; MD_MASK_[G] - 0xE0

MD_OFFSET[B] = 0x0; MD_MASK_[B] = 0x0

Cycle 1

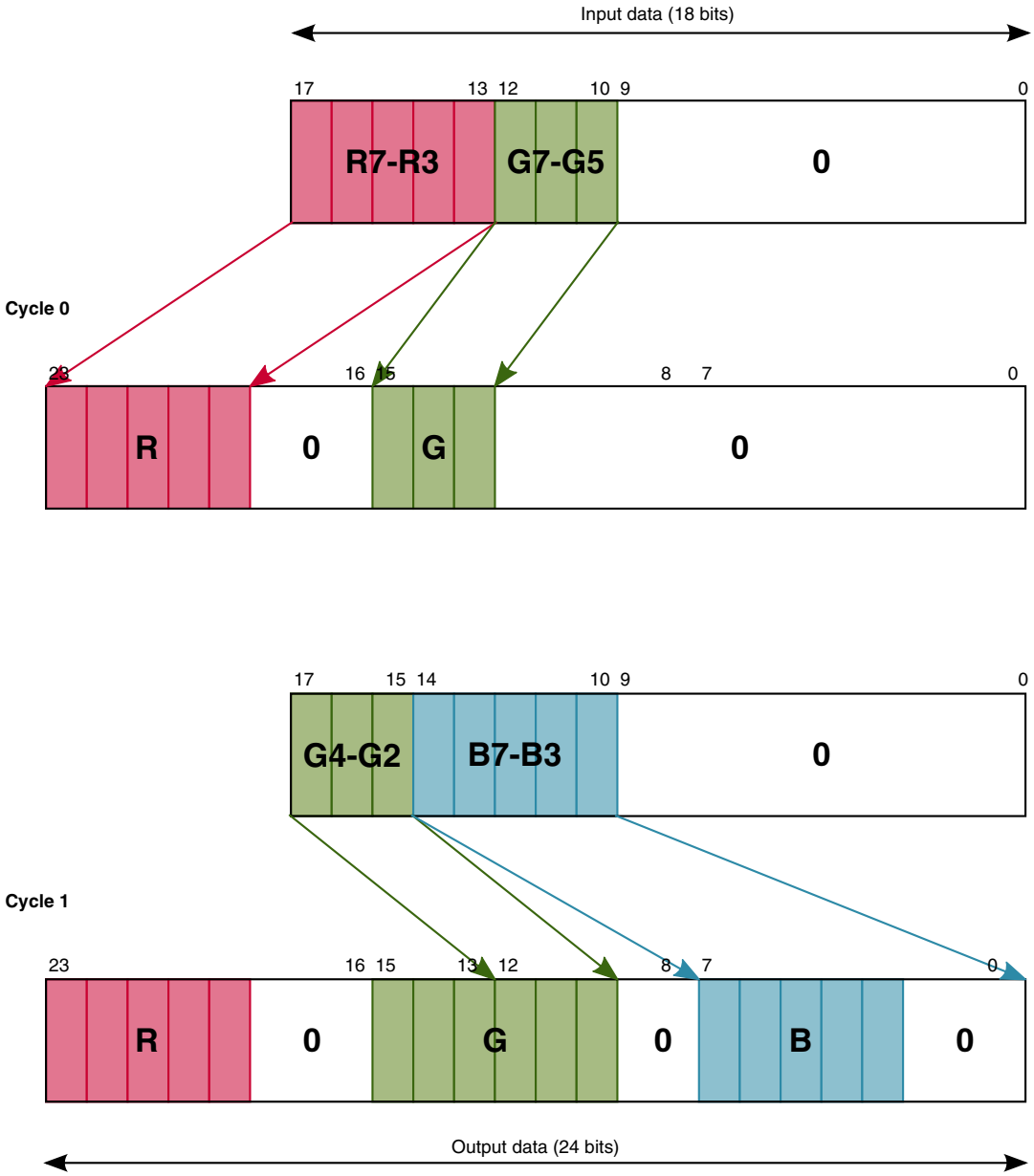
MD_OFFSET_[R] = 0x0; MD_MASK_[R] = 0x0

MD_OFFSET_[G] = 0x14; MD_MASK_[G] = 0x1C

MD_OFFSET_[B] = 0XE; MD_MASK_[B] = 0xF8

Figure 45-33. Example of Data Packing for Writing Data to th

The following figure presents an example of programming data packing for one of displays. Cycle 0 and Cycle1 in the diagram represent two separate atomic operations performed by the microcode template.



Cycle 0

MD_OFFSET_[R] - 0x11; MD_MASK_[R] - 0xF8

MD_OFFSET_[G] - 0xC; MD_MASK_[G] - 0xE0

MD_OFFSET[B] = 0x0; MD_MASK_[B] = 0x0

Cycle 1

MD_OFFSET_[R] = 0x0; MD_MASK_[R] = 0x0

MD_OFFSET_[G] = 0x14; MD_MASK_[G] = 0x1C

MD_OFFSET_[B] = 0XE; MD_MASK_[B] = 0xF8

Figure 45-34. Example of Data Unpacking for Reading Data from the Display

The same packing/unpacking registers are used for parallel and serial interface.

45.4.8 DMFC - Display Multi FIFO Controller

The following figure shows the block diagram for the DMFC block.

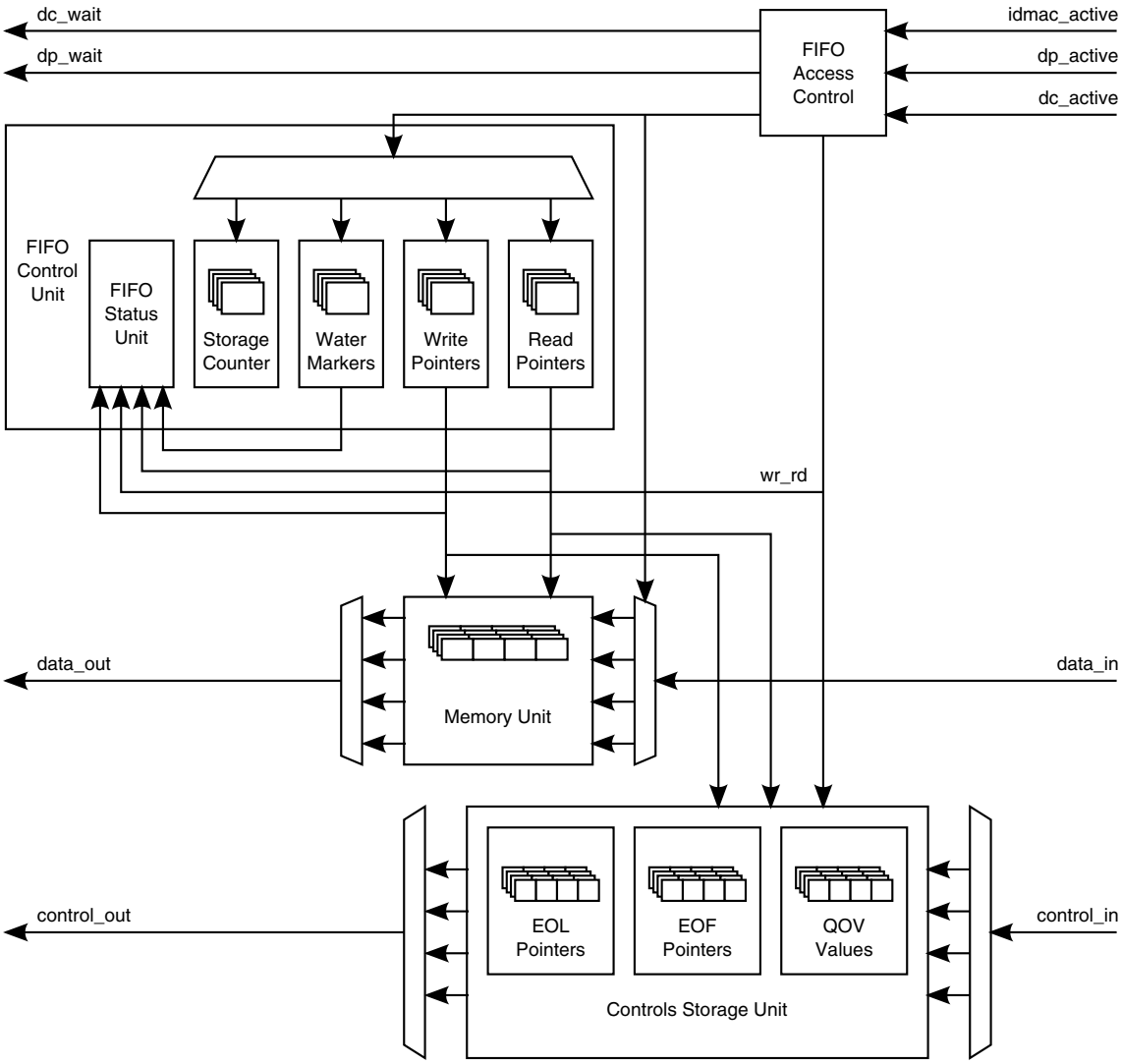


Figure 45-35. Display Multi Fifo Controller Block Diagram

The Display Multi Fifo Control manages Multi channels FIFOs. The DMFC serves the following clients:

- IDMAC - both read and write
- DP - read only
- DC - both read and write

- IC - write only
- AHB - both read and write

The DP and the DC read channels are physically attached to an IDMAC or an IC channel. As the IC has only one output channel connected to the DMFC. When the input is coming from the IC it replaces a channel that was physically attached to the IDMAC. The DMFC uses a single physical memory that serves the DP and DC read channels. The AHB accesses to the DC and the DC's write channel (read from display) use a separate physical memory.

45.4.8.1 DP and DC read channels

Each one of the DP and the DC read (read from memory) channels is physically attached to an IDMAC read channel. A portion of the physical memory is allocated for each channel. The DMFC arbitrates between channels according to a predefined priority.

The DMFC controls each of the FIFOs

- Assert a request any time there's available place on the FIFO.
- Make sure that there's available place on the FIFO to accept the coming data.
- Optionally assert a watermark indication to avoid starvation

45.4.8.1.1 FIFO allocation to channels

The physical memory is partitioned to 8 segments. For each channel the user has to define the start address at a segment's boundary using the DMFC_ST_ADDR parameter.

The size of the FIFO allocated to a channel is defined by the DMFC_FIFO_SIZE parameter. The user must allocate the FIFO and avoid overlapping between FIFOs.

The DMFC hold few special indications like EOF, EOL, EOFILD. The most important one is end of line (EOL). If the size of the FIFO is shorter than or equal to the IDMAC line's length (FW) than no special restrictions on the DMFC usage.

If the size of the FIFO is greater than the IDMAC line's length (FW) than the user has to be aware to the following restriction for each channel.

The DMFC has two operation modes which are distinguished by the wait4eot bit. For each channel the DMFC can store a maximum number of EOL indication. The maximum number of EOL indications of EOL is given in the table below.

Table 45-25. DMFC's number of EOL indications

IDMAC's Channel	Maximum lines on the FIFO
23	Up to 3 lines
27	Up to 2 lines
28	Up to 2 lines
Other	Up to 1 line

If the use case is that the number of EOL indications cannot exceed the maximum number of EOL indications than the user should have the wait4eot cleared.

If the use case is that the number of EOL indication can exceed the maximum number of EOL indications than the user should have the wait4eot set.

Having the wait4eot bit set has performance impact as the DMFC analyzes the data prior to sending it to the destination. In addition the DMFC cannot utilize the entire FIFO allocated to this channel.

The user need to specify the burst size of the IDMAC by setting the DMFC_BURST_SIZE field. This field must match the IDMAC settings. In case that the IDMAC's burst size is not a power-of-2 number, the value of this field should be rounded up to the nearest power-of-2 number. The burst size must not be greater than the FIFO's size.

45.4.8.1.2 Arbitration between channels

The arbitration between channels is fully hardware controlled. IDMAC has the highest priority. Then the synchronous channels. Then the asynchronous channels.

45.4.8.1.3 Watermark

The DMFC can generate a water mark signal for each channel. The watermark signal is sent to the IDMAC and dynamically increases the channels priority on the IDMAC's arbitration.

The watermark feature is enabled by the DMFC_WM_EN bit. The FIFO is partitioned to bursts. The user can set the watermark level at a burst boundary.

The watermark signal is set when the number of bursts on the FIFO + the number of already requested burst is smaller than the value specified on DMFC_WM_SET bit.

The watermark signal is cleared when the number of bursts on the FIFO + the number of already requested burst is greater than the value specified on DMFC_WM_CLR bit.

DMFC_WM_SET must be smaller than DMFC_WM_CLR.

45.4.8.2 IC interface

One of the IDMAC channels can be replaced by a flow coming from the IC using the DMFC_IC_IN_PORT. The user has to provide the IC's setting to the DMFC by programming the DMFC_IC_FRAME_WIDTH_RD, DMFC_IC_FRAME_HEIGHT_RD and DMFC_IC_FRAME_PPW_C fields. The burst size of the channel coming from the IC should always be programmed to 4 words.

45.4.8.3 DC write channel and AHB accesses

The second physical memory of the DMFC serves the

- IDMAC write channel (read from display)
- 2 AHB channels that can be read or write

The IDMAC write channel is programmed using the DMFC_RD_CHAN register in a similar way to the DC and DP read channels described above. The user has to provide the frame width and height for this channel and the pixel per word parameter.

The AHB channels are used from accesses via the AHB port to the display. The accesses are distributed between channels according to the MCU_T parameter.

45.4.9 DP - Display Processor

The display processor processes the image prior to sending it to the display. The main task performed by the DP is combining between 2 planes. The DP has 2 input FIFOs holding the data of full plane and the partial plane. In addition the DP performs some image enhancement functions like gamma correction, Color space conversion including Gamut mapping.

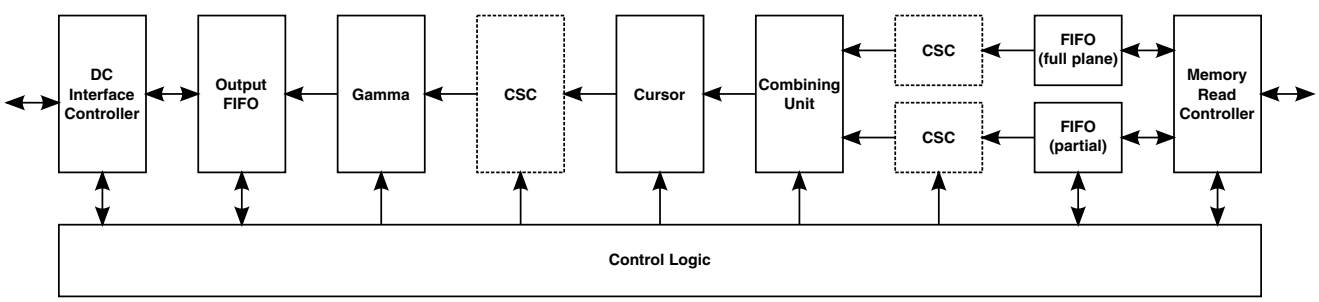


Figure 45-36. DP Micro architecture diagram

45.4.9.1 The DP programming model

The DP supports 3 flows. One sync flow and 2 Async flows. The DP holds 3 sets of registers. one set for each flow. Hence when referring to a register in this section the information is applicable to all the 3 sets. for example when referring to DP_COM_CONF, the information is applicable to DP_COM_CONF_SYNC, DP_COM_CONF_ASYNC0 and DP_COM_CONF_ASYNC1.

45.4.9.2 Displayed Planes

The following figure shows the planes displayed on a display.

There are full and partial planes. The partial plane's position is defined relatively to the upper left corner of the full plane (BGXP and BGYP). The size of the partial and full planes is defined on the corresponding IDMAC's channels' FW and FH parameters. The cursor position and parameters are set in the DP_CUR_POS register.

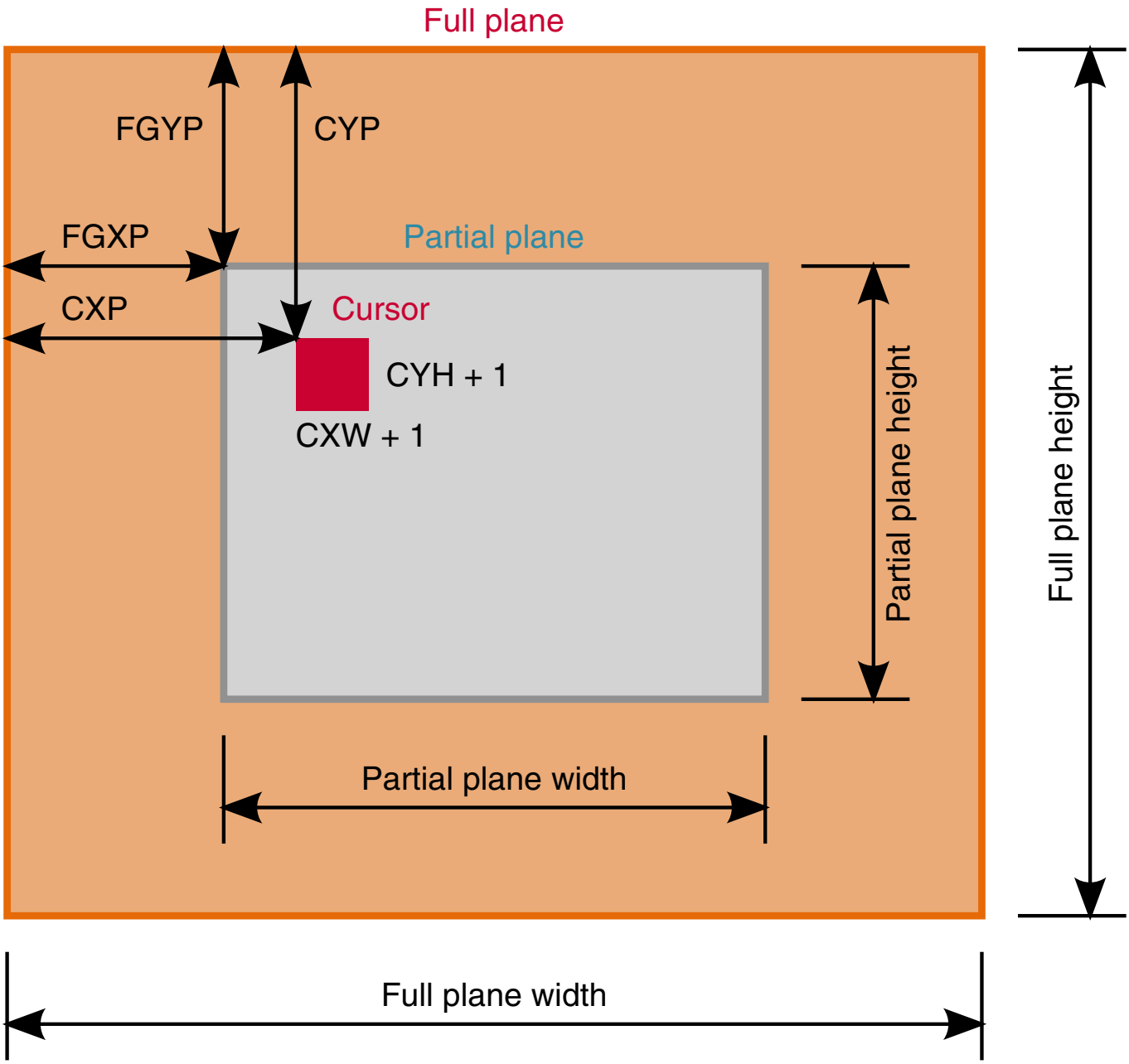


Figure 45-37. Displayed Planes

45.4.9.3 Combining Unit

The Combining Unit performs combining between the full and the partial planes. Each one of the planes may be graphics or video plane.

There are the following combining options:

- local alpha blending,
- global alpha blending,
- use of key color.
- order of the planes (full is presented over the partial plane and vice versa)

Combining mode is selected via the DP_COM_CONF Register. The combining equation is:

OP = BG*(1 - a) + FG*a

Where BG and FG are 2 input pixels; The DP_GWSEL bit defines if the BG is the pixel coming from the full plane or the partial plane.

a= (A+floor(A/128))/256 - an alpha value

A - a global or local transparency parameter.

The global A is written in the DP_GWAV field, the local A arrives together with the pixel.

A pixel becomes transparent when color keying is enabled and a pixel color matches a key color (independently on an alpha parameter). The color keying is defined on:DP_GWCKR, DP_GWCKG, DP_GWCKB

Combining takes 1 cycle per pixel. The Combining Unit outputs 24-bit words in the RGB/YUV format.

45.4.9.4 Cursor Generator

The Combining Unit output is passes through the Cursor Generator. The cursor's size and position are set via the DP_CUR_POS Register, a cursor color - via the DP_CUR_MAP Register. Different logic functions of combining the cursor with the image are supported as defined by the DP_COC field.

The cursor can be blinking. The blinking mechanism resides on the display controller sub-block. The blinking parameters are defined on the DC_BK_EN and DC_BKDVIV fields.

45.4.9.5 Color Space Conversion unit - CSC

The DP can get 2 input pixels from 2 different color spaces (YUV or RGB) and convert one of them to a common color space (YUV or RGB). In addition the 2 inputs can be of the same color space where the result is converted to another color space (YUV or RGB).

The DP has a single CSC unit that can be placed on one of 3 locations:

- At the output of one of the 2 input FIFOs
- At the output of the cursor generator.

Placing is done according to the DP_CSC_DEF field.

The color conversion implements a 3x3 matrix multiplication between the full RGB pixels and the color conversion constants, in order to obtain a YCC format image.

The conversion formula is:

x -> Clip(Round(S* 2^E)), S=Ax+B

where

A is a 3x3-dimensional matrix of weights, each a 10-bit signed number with 8 fractional digits

$$A = \begin{bmatrix} CSC\_A0 & CSC\_A1 & CSC\_A2 \\ CSC\_A3 & CSC\_A4 & CSC\_A5 \\ CSC\_A6 & CSC\_A7 & CSC\_A8 \end{bmatrix}$$

B is a 3-dimensional vector of offsets, each a 14-bit signed number with 2 fractional digits

$$B = \begin{bmatrix} CSC\_B0 & CSC\_B1 & CSC\_B2 \end{bmatrix}$$

S is a 3 dimensional vector of sums, each a 16-bit signed number with 4 fractional digits

S=Ax+B

E is an exponent, assuming one of the following values: -1,0,1,2 (allowing weights up to 8). The CSC_S parameters are encoded by 2 bits, please refer to the CSC_S parameter description.

$$E = [CSC\_S0\ CSC\_S1\ CSC\_S2]$$

A more explicit formula:

S[i] = (sum(A[i][j]*In[j]) >> 4) + (B[i] << 2) + (1 << (3-E[i]))

Out[i] = Clip(S[i] >> 4-E[i])

Where Clip() performs clipping to the range 0..255 (either per-component clipping or more sophisticated clipping that preserves the hue of the pixel)

45.4.9.5.1 Gamut mapping

When the color transformation produces colors outside the allowed range, they must be mapped back. This is called gamut mapping. The DP supports 2 clipping algorithms. (controlled by GAMUT_SAT_EN bit)

Hue preserving clipping algorithm is suitable only for RGB components. For YUV components, the per-component clipping algorithm is used.

Per component Clipping

This mapping is performed by clipping each of the components independently to its allowed range of values (the final value being uint8):

•Y: to 0..255 or 16..235 according to the SAT_MODE bit

•U/V: to 0.255 or 16..240 according to the SAT_MODE bit

Hue Preserving Clipping

Hue Preserving clipping is done in the following way

•First stage - eliminating negative values

```
N = min(R,G,B)
if (N<0) X-> X-N, where X=R,G,B
```

•At this stage, all components are non-negative and the MSB's beyond d=11 bits are ignored (assumed 0).

•Second stage - eliminating large values

```
M = max(R,G,B) (d-bit integer)
if (M>255)
```

```
M' = M >> (m-7), where m=8..d-1 is the index of the most-significant non-zero
bit in M
D' = Ceil(256*255/M') = 256..510 (since M' = 128..255)
Ceil(x) = the smallest integer which is not smaller than x
Implemented by a hard-wired 128x9-bit LUT
X -> min(255, (X*D')>>(m+1)), where X=R,G,B (8x9 multiplier)
else
X -> X
```

45.4.9.6 Gamma correction

The DP includes a gamma correction function. It is approximated by the piece-wise polynomial:

gammar = GAMMA_C<i> + ((R[4:0] * GAMMA_S<i>) >> 4 + 1)>>1

Where R is the input red component, that's a 9 bit input composed pixel_in*2+pixel_in[7].

Single approximation slope is used for Gamma Correction of red, green and blue color components. However the Gamma Correction block instantiated three times since processing for color components should be done in parallel. The gamma transform is also available for changing the contrast of the luminance component only.

The required Gamma correction slope for a specific display should by provided by the display manufacture. This information can be provided in various forms, as graph or formula. The gamma correction input pixel level (Gin) should be normalized to a maximum of 383. The gamma correction output pixel level (Gout) should be normalized to a maximum of 255. Then a following data should be collected:

Table 45-26. Gamma correction values

Gin	Gout
0	Gout0
2	Gout1
4	Gout2
8	Gout3
16	Gout4
32	Gout5
64	Gout6
96	Gout7
128	Gout8
160	Gout9
192	Gout10

Table continues on the next page...

Table 45-26. Gamma correction values (continued)

Gin	Gout
224	Gout11
256	Gout12
288	Gout13
320	Gout14
352	Gout15

Based on the table above the values of DP_GAMMA_S_SYNC<i> and DP_GAMMA_C_SYNC<i> fields for gamma correction control registers are calculated as following:

Table 45-27. Gamma correction values

i	DP_GAMMA_C_SYNC<i>	DP_GAMMA_S_SYNC<i>
0	Gout0	16*(Gout1-Gout0)
1	2*Gout1-Gout2	16*(Gout2-Gout1)
2	2*Gout2-Gout3	8*(Gout3-Gout2)
3	2*Gout3-Gout4	4*(Gout4-Gout3)
4	2*Gout4-Gout5	2*(Gout5-Gout4)
5	Gout5	Gout6-Gout5
6	Gout6	Gout7-Gout6
7	Gout7	Gout8-Gout7
8	Gout8	Gout9-Gout8
9	Gout9	Gout10-Gout9
10	Gout10	Gout11-Gout10
11	Gout11	Gout12-Gout11
12	Gout12	Gout13-Gout12
13	Gout13	Gout14-Gout13
14	Gout14	Gout15-Gout14
15	Gout15	255-Gout15

45.4.9.7 DC interface

The DC interface unit performs 2 tasks.

- Starts the flow via the DP by getting a request from the DC and once the DP is ready send the new frame request to the IDMAC.
- Control the DP's output FIFO and send the data to the DC using an handshake mechanism.

45.4.9.8 DP's flows management

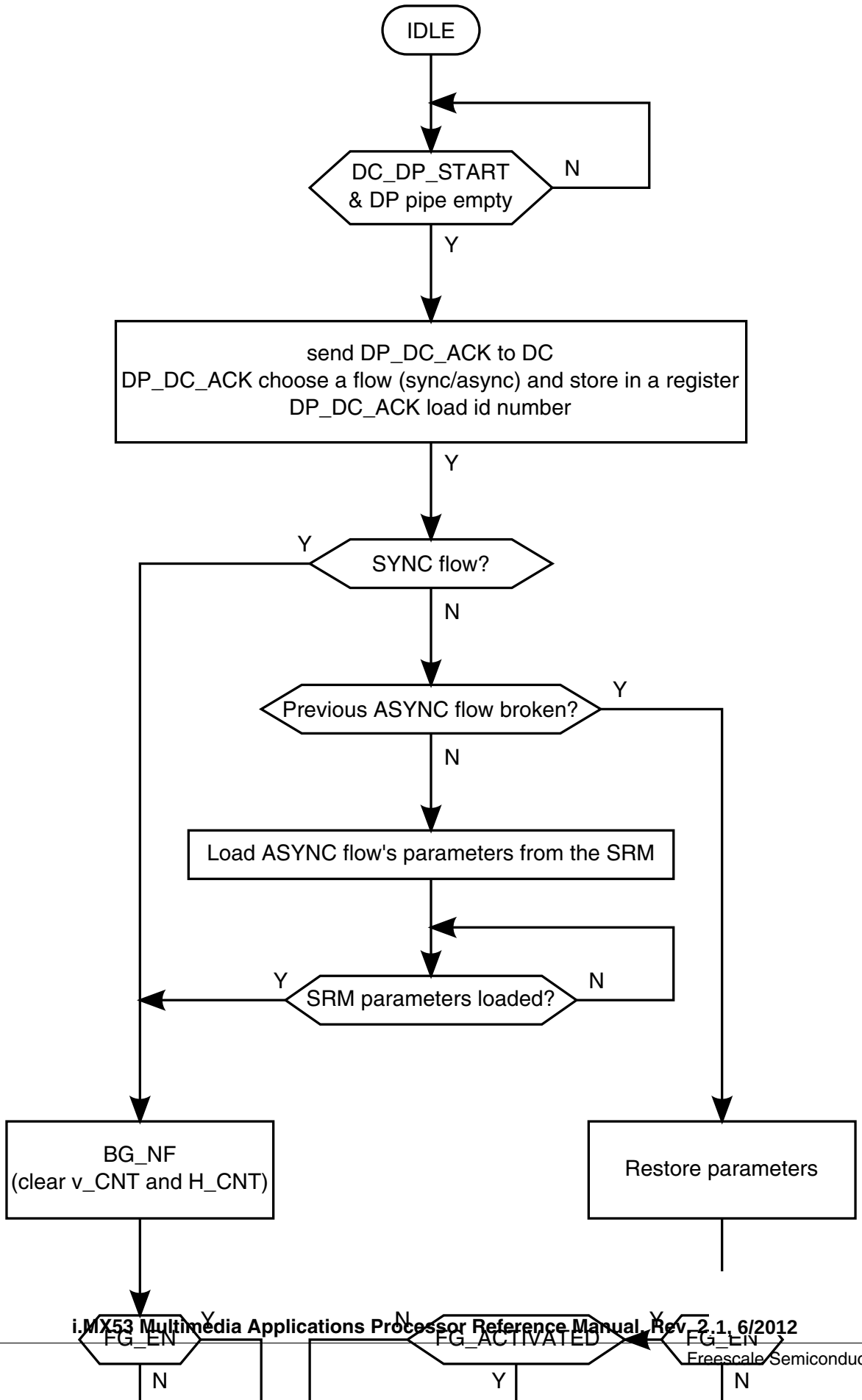
The DP can manage up to 3 flows: one synchronous flow and 2 asynchronous flows. However the DP can handle only one flow simultaneously. The DP has an automatic mechanism to control and switch between flows.

The DP's highest priority flow is the sync flow. An async flow can be executed during the blanking interval of the sync flow. An async flow can be stopped in order to serve the sync flow. The sync flow cannot be broken. The DP holds 3 sets of registers, one set for each flow. The registers are located in the SRM memory. According to the flow that needs to be executed the correct set of registers is loaded to the DP.

The figure below illustrates the flow management done by the DP.

A flow starts when a request from the DC arrives and the internal pipe is empty. For sync flows the full frame NF (new frame) indication is immediately. In case that a partial frame is also used the NF indication will be sent one row before the row that the partial plane is actually positioned.

In case of ASYNC flow the DP first check if the previous async flow was broken. If yes, the DP restores the last settings of the previous flow. If this is a new async flow the DP will first reload the flow's parameters from the SRM. In case that a partial frame is also used the NF indication will be sent one row before the row that the partial plane is actually positioned.


IDLE
DC_DP_START
& DP pipe empty
N
Y
send DP_DC_ACK to DC
DP_DC_ACK choose a flow (sync/async) and store in a register
DP_DC_ACK load id number
Y
Y
SYNC flow?
N
Previous ASYNC flow broken?
Y
N
Load ASYNC flow's parameters from the SRM
Y
SRM parameters loaded?
N
BG_NF
(clear v_CNT and H_CNT)
Restore parameters
Y
FG_EN
N
N
FG_ACTIVATED
Y
Y
FG_EN
N

45.4.9.9 DP debug unit

The DP supports synchronous and asynchronous flows using the same hardware. The asynchronous flow can be broken by the synchronous flow. The DP's debug unit provides the ability to know on which row exactly the async flow has been broken. This is done by providing an interrupt (with DP_DEBUG_CNT register) and providing row status flags (on DP_DEBUG_STAT register).

As the async flow can broken multiple times within a specific frame The user can control the breaking point the issues the debug event by programming BRAKE_CNT field.

45.4.9.10 Restriction

When both full and partial planes are processed, the full plane's minimal frame width is 13 pixels.

The Minimum frame height supported by the DP is 2 lines.

45.4.10 Display Interface (DI)

The DI provides arbitrates access to up to three displays with time multiplexing. It converts a data from the DC or the ARM platform (low level access for serial interface only) to a format suitable for the specific display interface.

The DI generates display clocks and other display control signals with programmable timings. The DI outputs data to or inputs from parallel and/or serial interfaces.

This block generates all the control signals sent to the display. The DC sends to the DI; the data for the display and a set of control signals. The controls coming from the DC are used in order to generate the control signals sent to the display. One exception is serial low level access (LLA), where the DC is bypassed and the data is coming directly from the ARM platform. The figure below is the DI's block diagram.

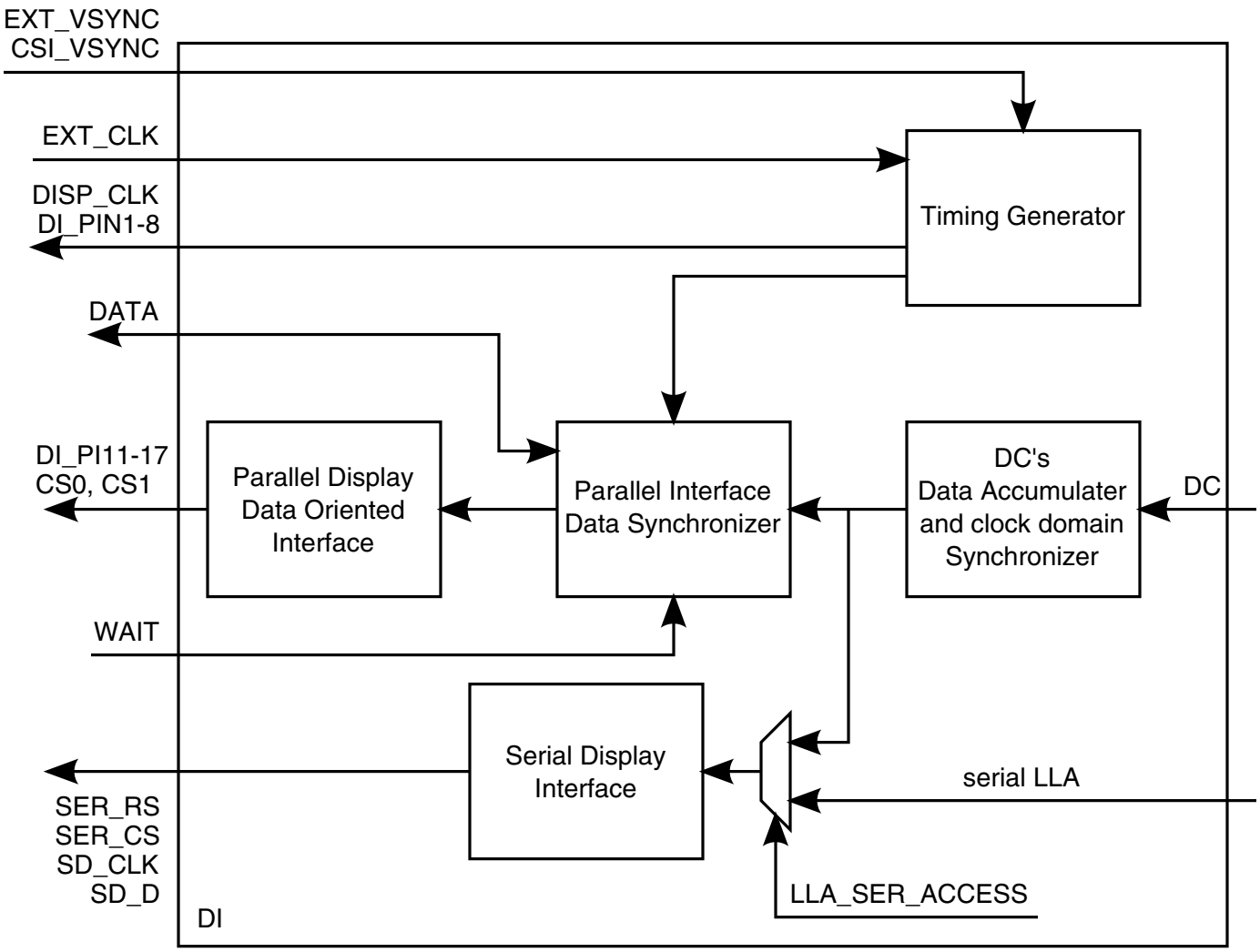


Figure 45-39. DI's block diagram

The display interface includes 2 groups of control signals:

- Time oriented signals - These type of signals are generated according to the DI's internal timers. These are free running signals that change their state according to a pre-defined waveform. For example VSYNC, HSYNC, display's clock (pixel clock) etc.
- Data oriented signals - The DC may add markers to data sent to the DI. These markers are used to indicate a specific attribute of the data (for example: end-of-line, end-of-frame, chip-select etc.). The marker coming from the DC triggers a specific waveform of one or more signals on the display's interface. The specific waveform will be seen on the bus along with the associated data. The markers may be synced to a time oriented signal. For example: attach the end-of-frame signal to the next VSYNC.

45.4.10.1 DC interface, data accumulator and clock domain synchronizer

The data accumulator is the DI's input buffer. It receives the data from the DC along with a set of control signals. The data accumulator receives the data from the DC's clock domain and synchronize it to the DI's clock domain.

45.4.10.2 Parallel interface data synchronizer and data oriented interface

The data accumulated in the DI's input buffer will be sent to the display according to the DI's internal events. Each event is generated by a counter. A tag is attached to each data by the DC's microcode using the SYNC field in the DC's microcode. The tag selects the event that the data will be synced to.

Once the corresponding event is generated, the data will be sent to the display. A data can be a pixel or a component (part of a pixel).The data synchronization occurs separately for each component. For asynchronous displays the tag will be equal to 0 i.e. the data is not synchronized to any event. The data will be sent out of the buffer immediately.

45.4.10.3 Timing generator

The timing generator is used for generating the waveforms' of each pin of the display's interface.

The timing generator is built of 10 counters. One counter functions as a time base This timer is called the BASE TIMER (BS). The other 9 counters are used in order to generate the control signals' waveforms. The last counter (counter #9) is special see Counter number 9.

The DI clock can be derived from IPU's clock (HSP_CLK) or from an external source (via the ipp_di_#_ext_clk pin). The clock's source is statically selected by configuring the DI#_CLK_EXT bit.

Figure 45-41 illustrates the main parameters of a waveform. A waveform's segment is built of 5 parameters. Each segment can be has 2 phase "ACTIVE PHASE" and "OFFSET PHASE".

- TIMEBASE - this is the base timer (fundamental timebase); all the other parameters are derived from this timer. The timebase is generated by counting DI clock cycles. The amount of cycles is defined according to DI#_DISP_CLK_PERIOD. This field

defines the Display interface clock period, This parameter contains an integer part (bits 11:4) and a fractional part (bits 3:0). It defines a fractional division ratio of the Di's source clock for generation of the display's interface clock. The timebase is generated from edge aligned pulses of the DI clock. The user can delay the timebase starting point by defining an offset to the timebase. This is done by configuring the DI#_DISP_CLK_OFFSET field. The offset is calculated from the point where the DI is enabled to the point where the timebase starts ticking. The display clock waveform is generated between 2 edges of the timebase. The waveform is defined according to the DI#_DISP_CLK_UP and DI#_DISP_CLK_DOWN. Each pin has a specific timebase that is derived from the fundamental timebase. The following figure illustrates the relations between the TIMEBASE and the DI's clock

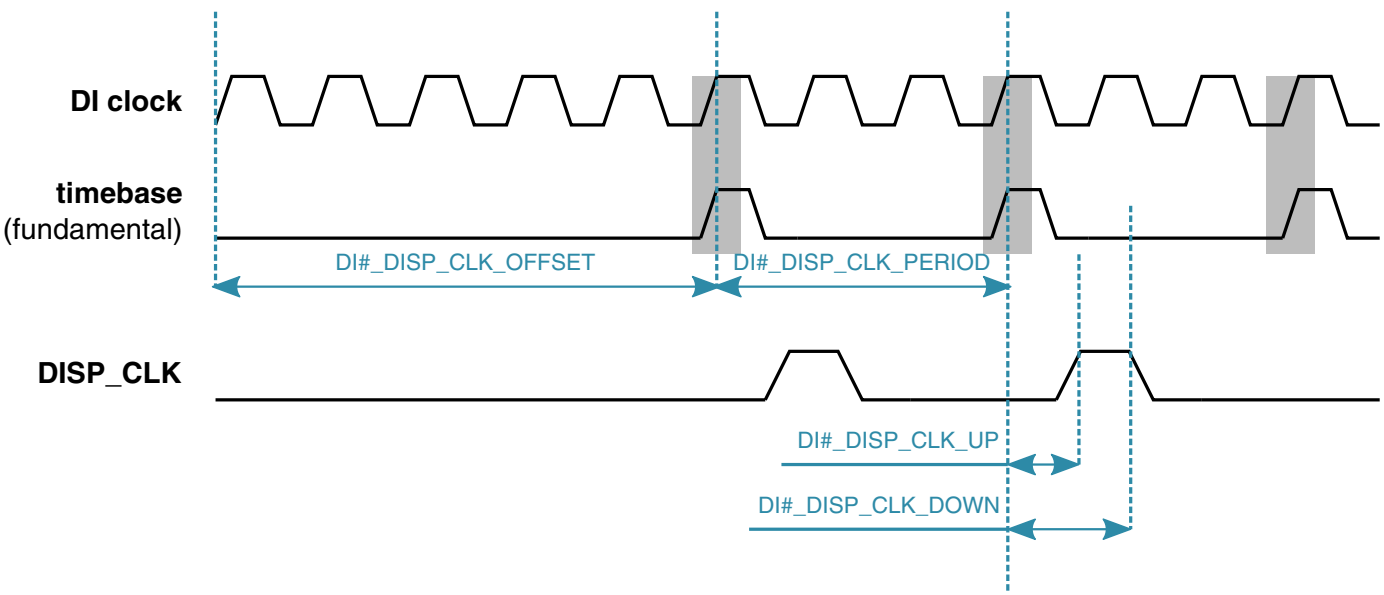


Figure 45-40. Timebase, DI's clock and display's clock relations

- OFFSET - this parameter defines when the length of the "OFFSET PHASE" it is defined by di#_offset_value_<N> field, where N is the counter's index.
- STEP - This parameter defines the length of the "ACTIVE PHASE"; it is defined by the di#_step_repeat_<N> field, where N is the counter's index. If the counter is in auto reload mode (di#_cnt_auto_reload_<N> bit is set) then the counter will be automatically reloaded forever. The value of di#_step_repeat_<N> is ignored in that case.
- RUN - The "ACTIVE PHASE" is partitioned to several "RUN sections"; this parameter defines the length of the "RUN section"; it is defined by the di#_run_value_m1_<N> field, where N is the counter's index.

- UP - Each "RUN section" contains the waveform of a single pulse. This parameter defines the offset from the beginning of the "RUN section" to the assertion of the signal; it is defined by the di#_cnt_up_<N> field, where N is the counter's index.
- DOWN - This parameter defines the offset from the beginning of the "RUN section" to the negation of the signal; it is defined by the di#_cnt_down_<N> field, where N is the counter's index. In case where DOWN < UP the waveform will have a 50% duty cycle.

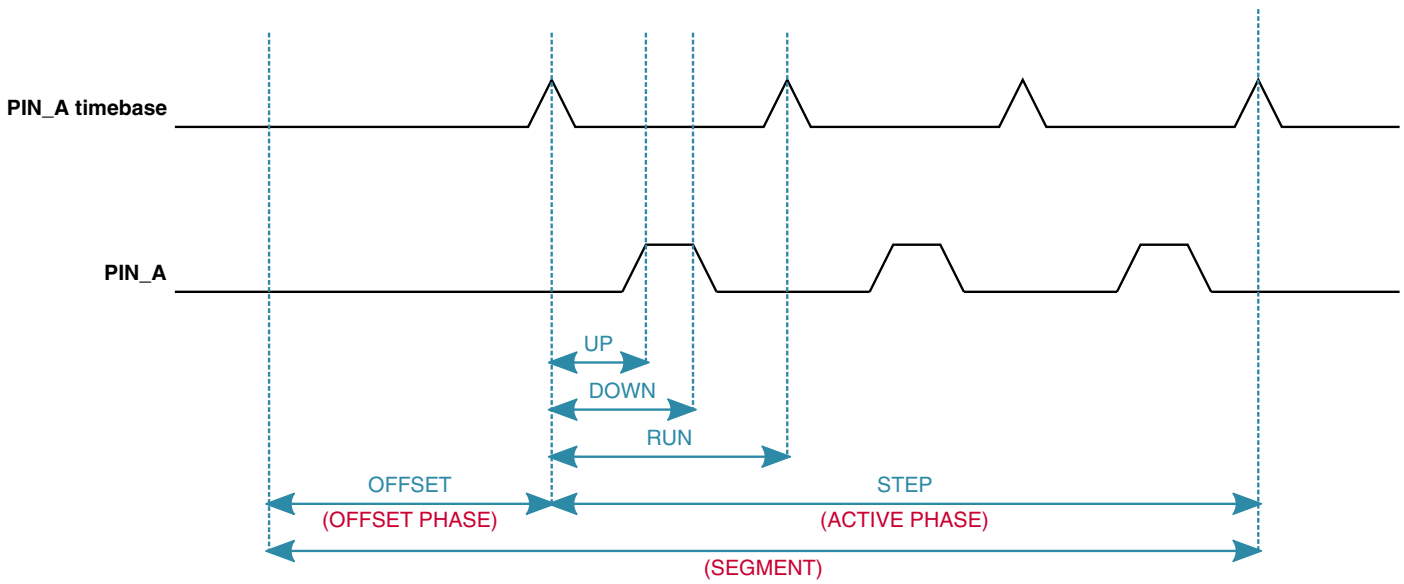


Figure 45-41. DI waveform's main parameters

45.4.10.3.1 Waveform concatenation

The DI provides the ability to derive the waveform from the fundamental timebase or from another PIN. In that case, one pin's waveform is used as another pin's timebase.

The following figure provides an example. PIN_A and PIN_C are derived from the fundamental timebase. However, PIN_B is derived from PIN_A's wavefore. The trigger is selected by DI#_RUN_RESOLUTION_<N>, where <N> is the index of the counter. A counter can be triggered by a counter with lowered index. For example: counter #5 can be triggered by counter #3 but can't be triggered by counter #7.

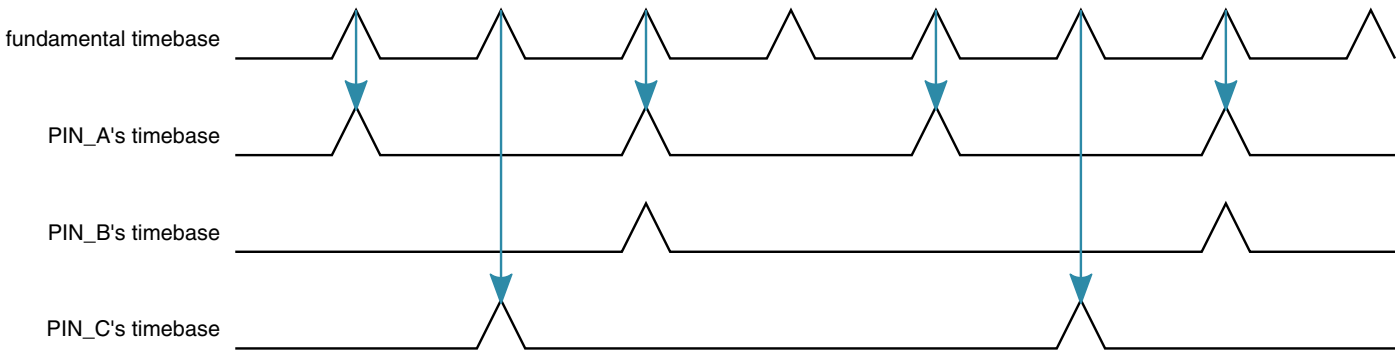


Figure 45-42. DI pins - Waveform's time bases concatenation

45.4.10.3.2 The basic counter

The DI has 9 counters. A counter is built of 3 units: timebase generator, waveform generator and polarity generator.

Figure 45-43 illustrates the counter's structure.

The timebase generator

The timebase generator gets 3 triggers. Clear, offset and run trigger. The Clear is the trigger that resets the counter. It is selected by programming the DI#_CNT_CLR_SEL_<N>.

The Offset trigger is the trigger used for counting the OFFSET_PHASE. The user can select the source of the trigger by programming the DI#_OFFSET_RESOLUTION_<N>. The offset's value is defined by programming the DI#_OFFSET_VALUE_<N>

The RUN trigger is the trigger used for counting the RUN period. The user can select the source of the trigger by programming the DI#_RUN_RESOLUTION_<N>. The RUN's value is defined by programming the DI#_RUN_VALUE_<N>

The timebase generator counts according to the DI_CLK or according to another counter's output. In order to use a source different than DI_CLK, the user should set the POLARITY_GEN_EN bits to 01.

Waveform generator

This unit generates the waveform. It gets the values of RUN, UP, DOWN and STEP and build the waveform accordingly. The waveform is counted according to the signal generated by the timebase generator.

Polarity generator

The waveform's polarity is controlled by 2 units. The static polarity is changed according to the POLARITY_<#> bit of each pin. This bit defines if the waveform of the pin is active high or active low.

The other unit controlling the polarity is the polarity generator. This unit is enabled by setting the POLARITY_GEN_EN[1] to 1. The polarity generator has 2 modes: "toggle mode" and "normal polarity mode". The mode is defined according to POLARITY_GEN_EN[0].

- Normal polarity mode - In this mode the polarity is changed according to 2 other counters. The counter selected by POLARITY_TRIGGER_SEL define the sampling point. The counter selected by POLARITY_CLR_SEL defines the polarity value. At the sampling point the polarity value is defined. The current polarity value defines the current polarity of the waveform.
- Toggle mode - In this mode, the output of the timebase generator's output causes the polarity to toggle. Any tick of the timebase generator inverts the polarity. When this mode is enabled the ticks generated by the counter selected by POLARITY_TRIGGER_SEL initialize the polarity generator and the timebase generator causes the polarity to toggle.

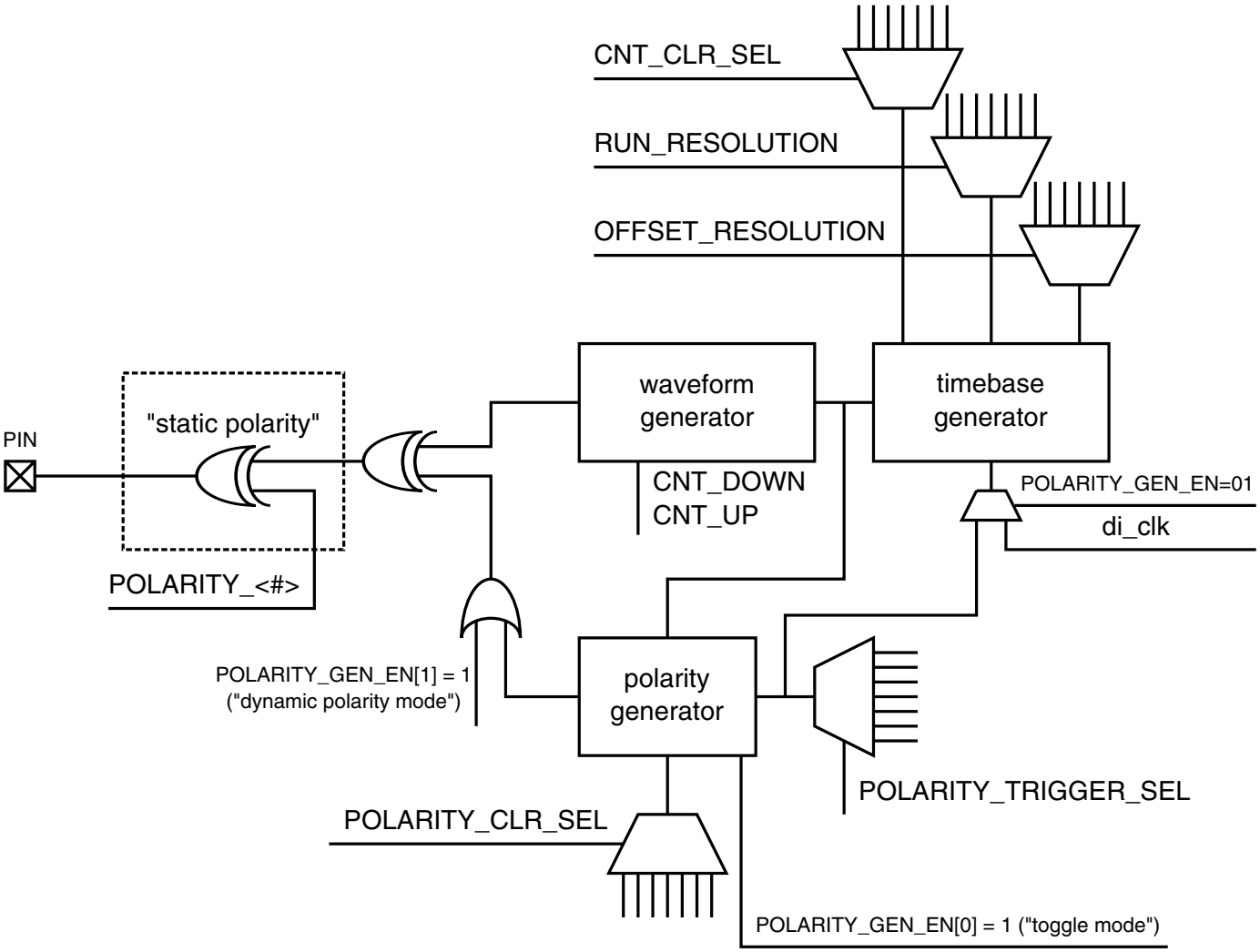


Figure 45-43. DI's counter's structure

The following figure provides an example for waveform generation using different trigger sources. The waveform is generated for PIN_D. PIN_D counter is cleared by the PIN_C's timebase. The offset period is calculated by counting cycles of PIN_B's timebase. The RUN period is calculated by counting cycles of PIN_A's timebase. The UP and DOWN periods of the waveform are calculated by counting cycles of PIN_D's timebase.

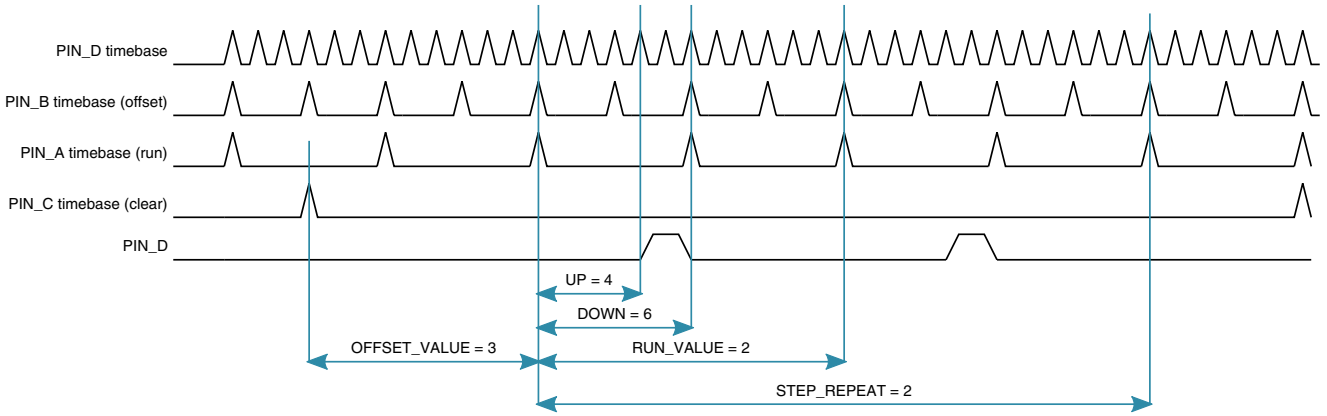


Figure 45-44. Clear, offset and run triggers and values - Example

45.4.10.3.3 Counter number 9

The last counter (counter #9) is an auxiliary counter and it is not used for generating a waveform for a specific pin. It can be used in order to attach another waveform to an existing one.

The user define the waveform for a specific pin (main pin). The user defines the auxiliary waveform using counter #9. The 2 waveforms are logically ORed. The combined waveform will be routed to the main pin. The main waveform that this counter is attached to is defined according to DI#_GENTIME_SEL_9.

The tag of counter #9 can be generated from counter #9 or from the main waveform. This is selected according to the DI#_TAG_SEL_9.

45.4.10.3.4 DI's active window

The DI provides an alternative way to define the synchronous display setting by using an active window and thus, needs to program less counters. The synchronous display's active window is a rectangle on the display where IPU sends data. It is set by programming the parameters defined on DI#_AW0 and DI#_AW1 registers.

The following figure illustrates the different parameters defining the active window. The display's vertical and horizontal position are defined according to counters. The DI#_AW_HCOUNT_SEL selects the counter which the horizontal position is calculated according to. DI#_AW_VCOUNT_SEL selects the counter which the vertical position is calculated according to. The Active data is sent according to a trigger. DI#_AW_TRIG_SEL selects the counter which calculates this trigger. This trigger usually functions as a data enable signal.

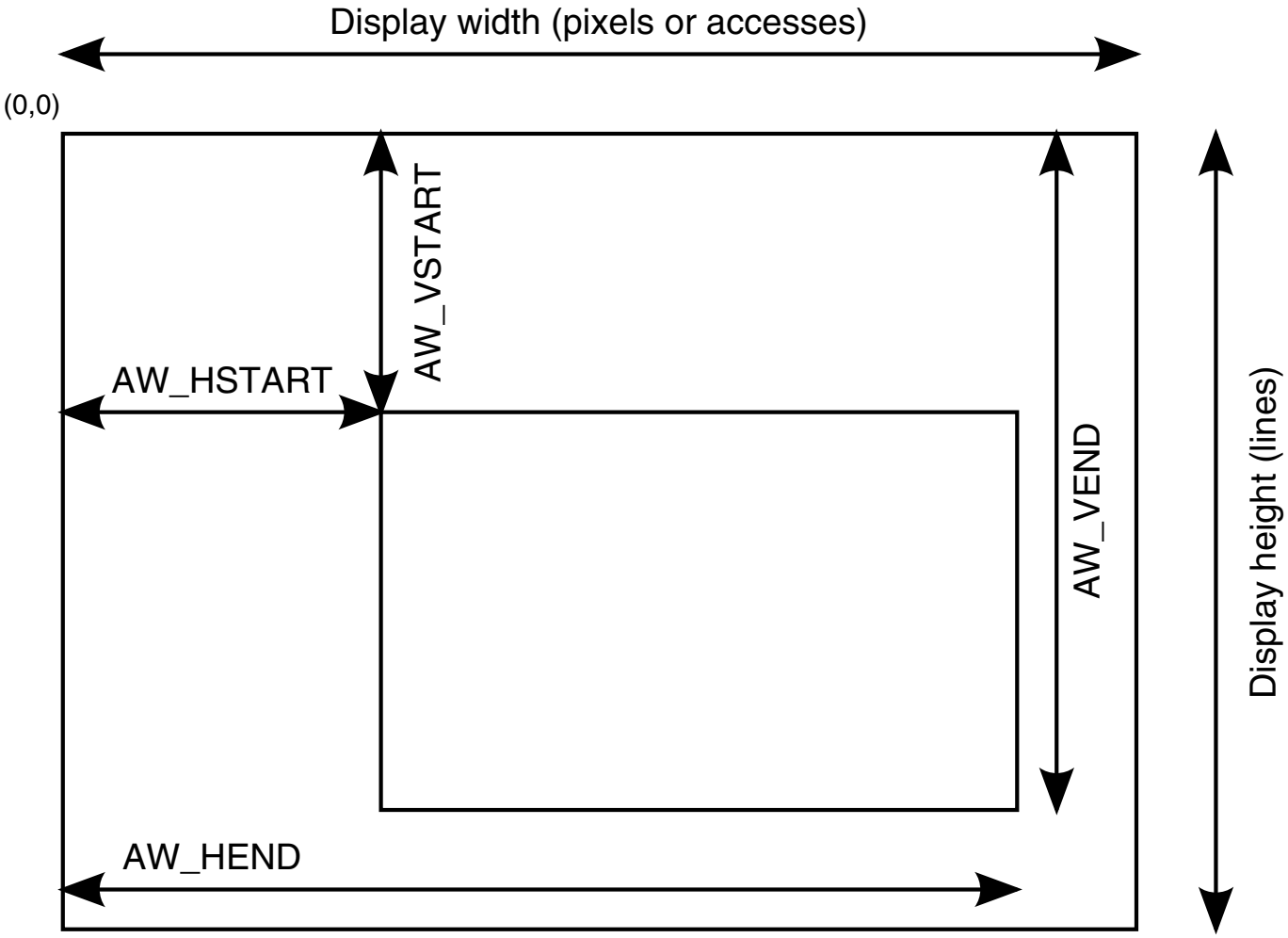


Figure 45-45. DI's Active Window

45.4.10.4 Waveform settings for asynchronous interface pins

The DI provides 8 signals that are used for asynchronous interface. These signals are PIN11 through PIN17 (ipp_di_#_pin_11 through ipp_di_0_pin_17) and the CS (ipp_di_0_do_dispb_d0_cs).

The DI holds 12 wave set quartets. Each quartet includes 4 registers (DI#_DW_SET<j>_<i>). Each DW_SET register holds the UP and DOWN values of the waveform. The DW_SET register is selected in the following way. The WAVEFORM field in the DC template is a pointer that points to one of the 12 quartets. In addition the WAVEFORM field points to one of 12 DI#_DW_GEN_<i> registers. The DI#_DW_GEN_<i> holds 9 pointers. The pointers are 2 bits field that points to one of the registers from the DI#_DW_SET<j>_<i> quartet. Each one of the 8 pointers in the

DI#_DW_GEN_<i> registers is related to a specific pin of the DI's asynchronous interface. The following figure illustrates the relations between the registers controlling the asynchronous interface's signals.

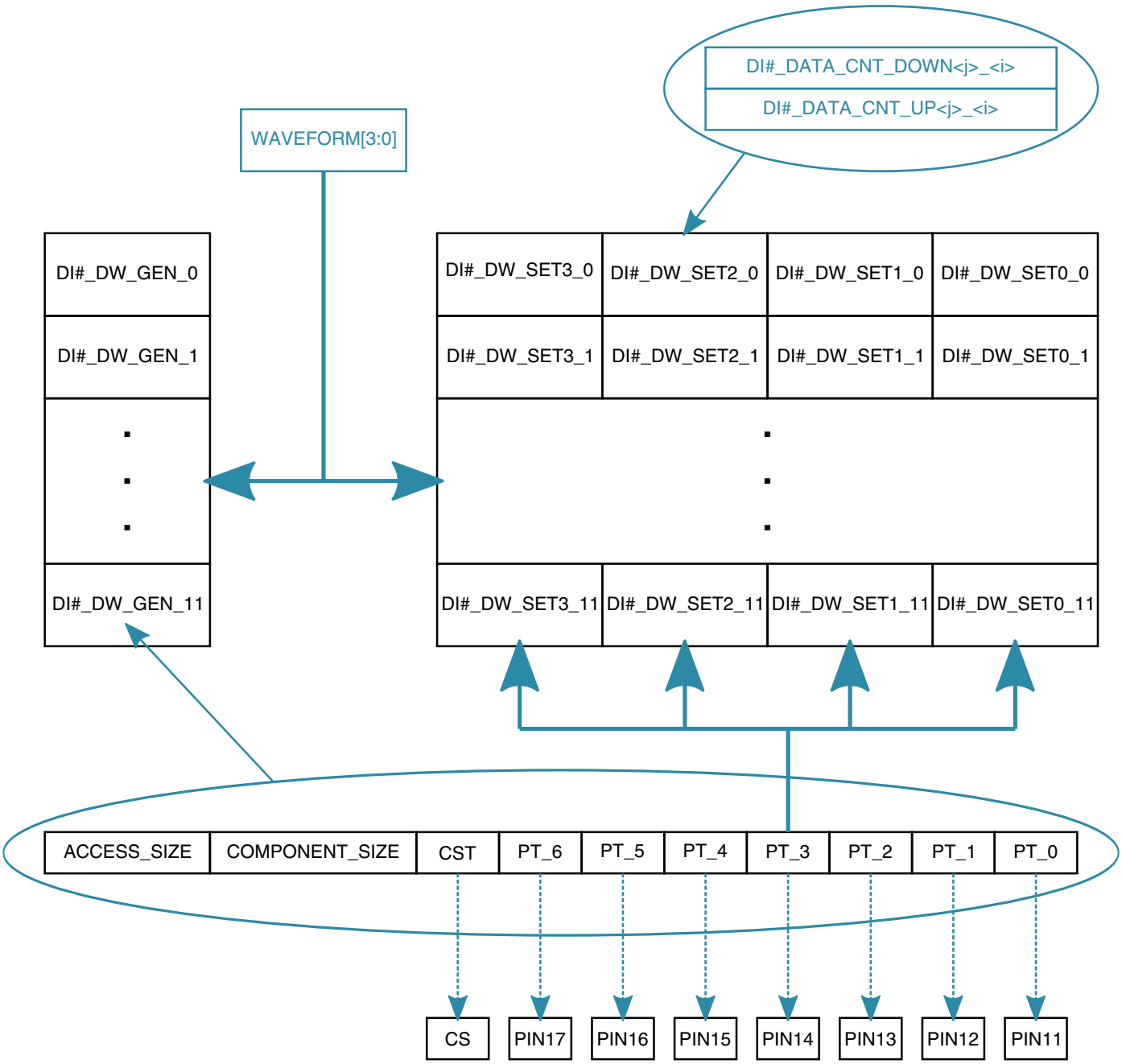


Figure 45-46. Waveform settings for asynchronous interface pins - parallel interface

The DI#_DW_GEN_<i> register includes the data's waveform settings as well. ACCESS_SIZE defines the amount of DI clock cycles that a pixel is valid on the bus. When generic data is sent, this field defines the amount of cycles that the generic data is

valid on the bus. A pixel may be broken into few components. The COMPONNENT_SIZE field defines the amount of cycles that each component is valid on the bus.

The COMPONNENT_SIZE is always smaller or equal to ACCESS_SIZE. For synchronous interface COMPONNENT_SIZE is always equal to ACCESS_SIZE. In case that there's a need for some gap between one type of accesses to another having the COMPONNENT_SIZE smaller than ACCESS_SIZE can be useful (for example read after write accesses that require some gap between them).

45.4.10.5 Serial display interface

The DI supports the following asynchronous serial interfaces:

1. 3-wire (with bidirectional data line).
2. 4-wire (with separate data input and output lines).
3. 5-wire type 1 (with sampling RS by the serial clock).
4. 5-wire type 2 (with sampling RS by the chip select signal).
    - For serial interfaces accessed in LLA mode, data and preamble lengths and other parameters are controlled via the DI0_SER_CONF and DI1_SER_CONF Registers.

The serial display interface includes generators for the CS, RS and CLK signals of the display interface and a bidirectional datashifter. The figure below provides a block diagram of the serial display interface.

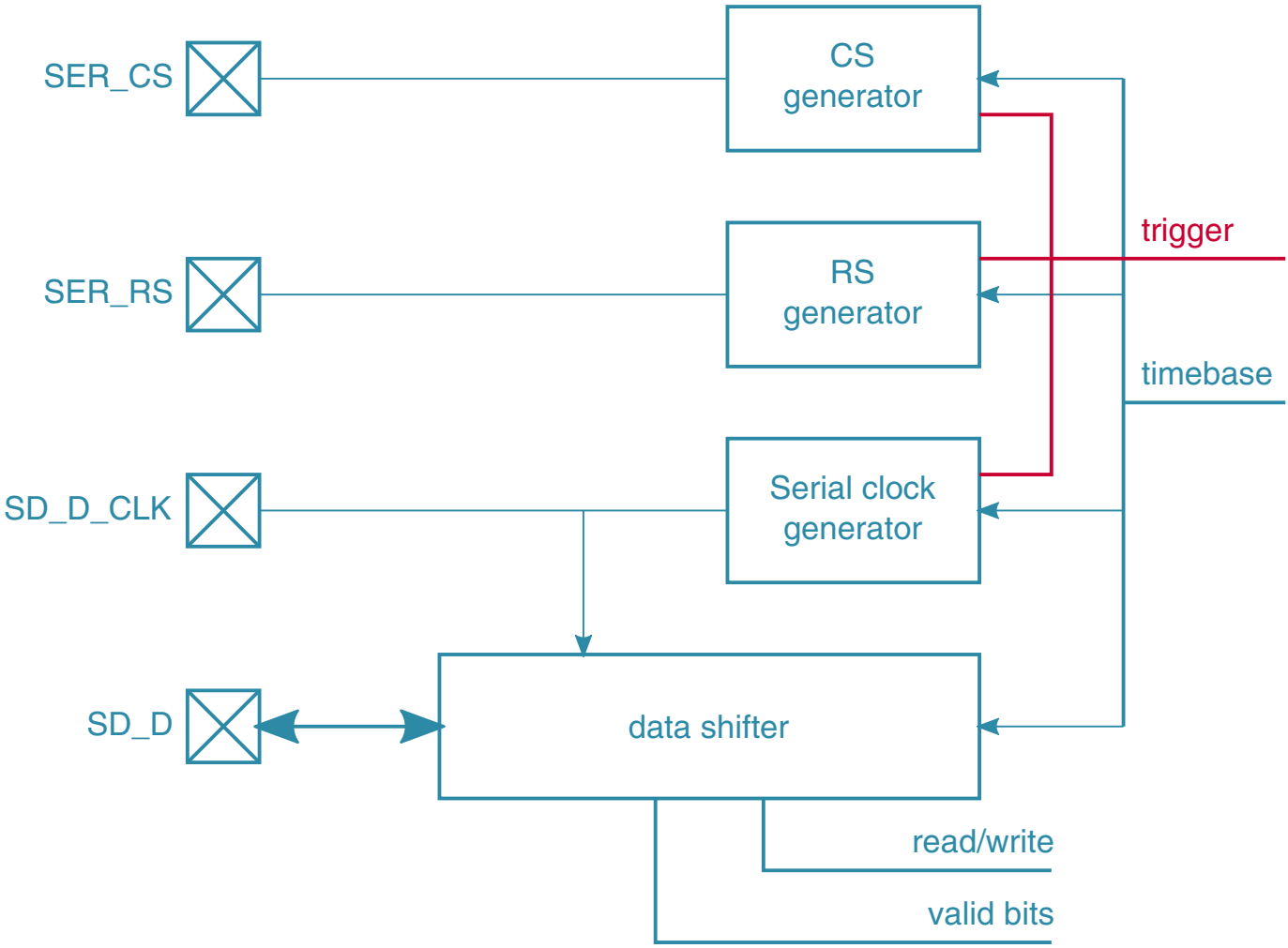


Figure 45-47. Serial display interface

45.4.10.5.1 Waveform settings for serial interface pins

When serial interface is used the fields in the DI#_DW_GEN_<i> register have different meaning than the parallel interface. The registers includes pointers to DW_SET for the CS, RS and CLK pins of the serial interface.

The following figure illustrates the relations between the registers controlling the serial interface's signals.

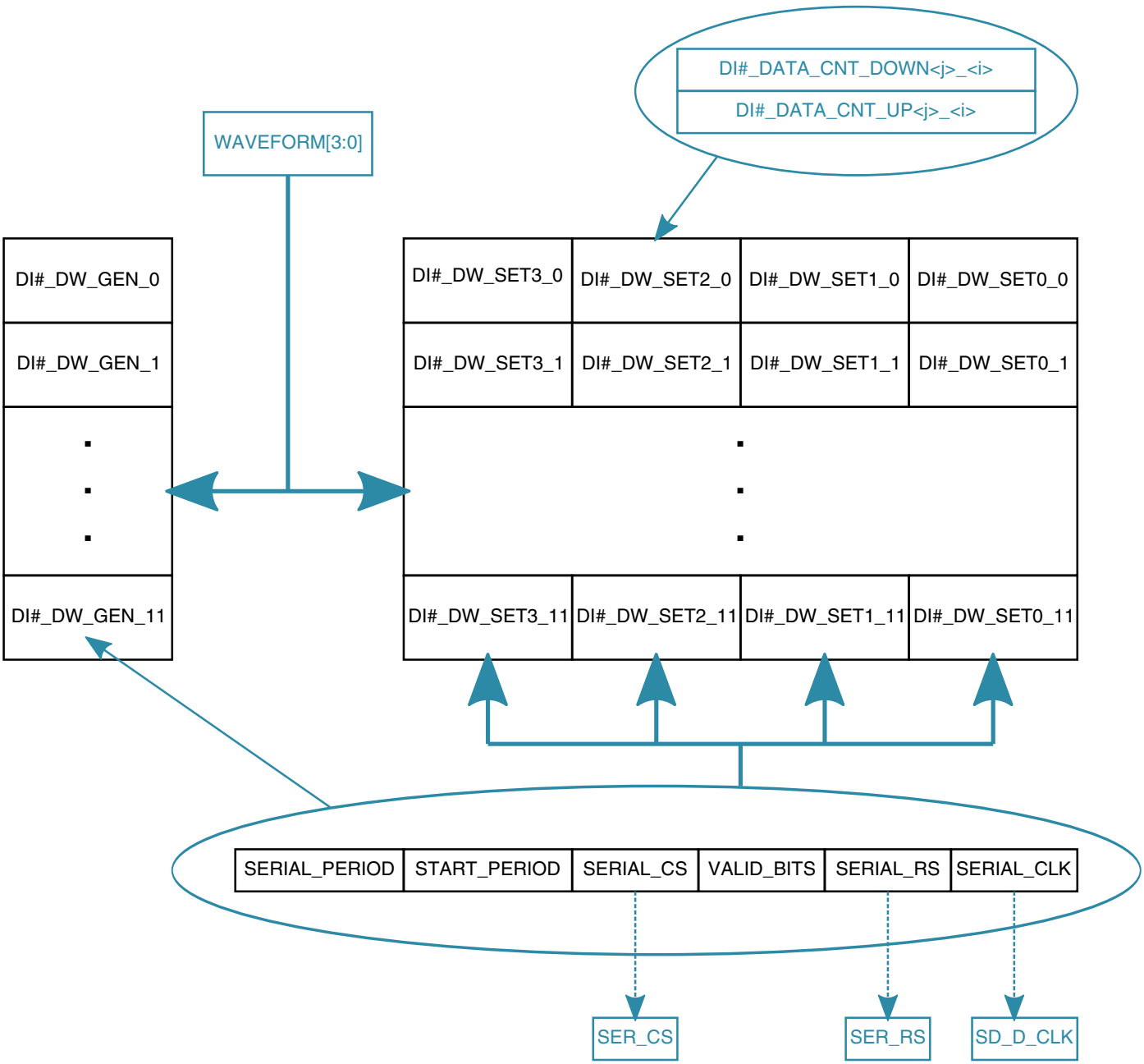


Figure 45-48. Waveform settings for asynchronous interface pins - serial interface

The following figure provides an example of the waveform's settings for the serial interface. A waveform generation starts following a trigger indicating that the serial interface has won the DI's arbitration. Therefore the next access will be via the serial interface. The waveform is based on timebase ticks. All the waveform's settings are defined by counting DI_CLK cycles. The timebase starts ticking after counting DI_CLK cycles defined on the START_PERIOD parameters. The number of timebase ticks is defined according to the VALID_BITS field. The VALID_BITS field defines how many bits will from the 32 bit word sent to the DI are valid and should be sent via the serial

interface. The valid bits are the least significant bits of the word coming from the DC. The amount of DI_CLK cycles between each timebase tick is defined according to the SERIAL_PERIOD parameter. Each one of the serial interfaces control signals may have a different waveform defined according to its corresponding UP and DOWN values.

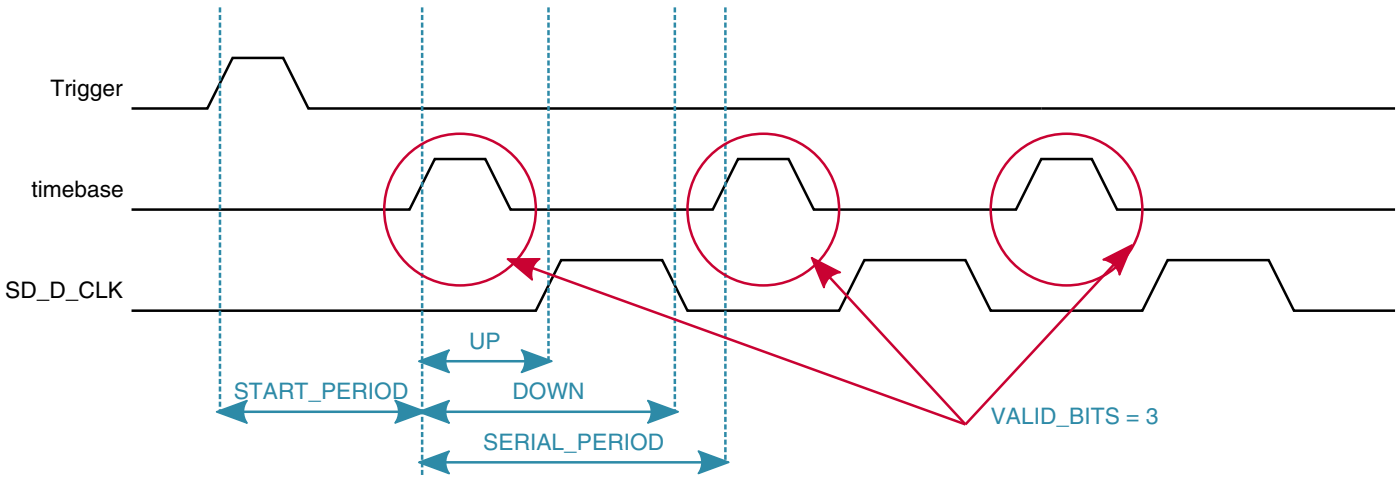


Figure 45-49. Serial interface waveform example

The CS and RS signals can have a single change (one rising and one falling edges) between 2 consecutive triggers.The UP value for RS and CS is measured from the first timebase tick. The DOWN value is measured from the last timebase tick. The figure below provides an example of the RS behavior, CS behaves in a similar way.

The data is sent according to the timebase ticks. The MSB bit of the data is transmitted first. When receiving serial data, the data is sampled every timebase tick. The user can program an offset from the timebase tick by programming the DI#_SERIAL_LATCH field. This value will move the serial data sampling point. The value is measured by counting DI_CLK cycles.

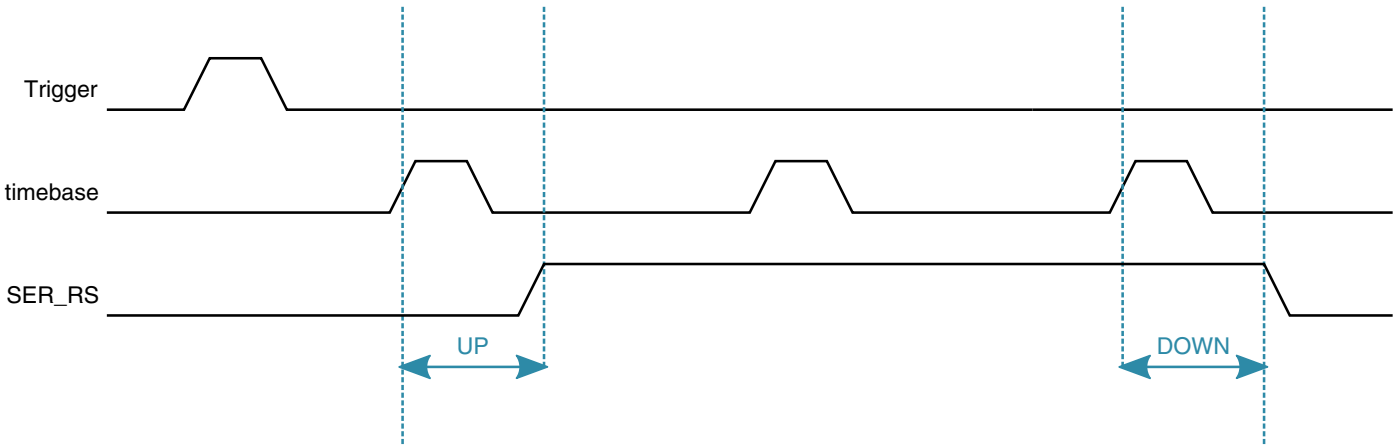


Figure 45-50. Serial interface waveform example of RS

45.4.10.6 Low Level Access - LLA

LLA is a ARM platform direct access to the display. For parallel displays the DI's behavior for LLA is the same as any other access to a parallel display. For serial displays the DI allows a DI arbitration bypass. This is done if the DI#_LLA_SER_ACCESS bit is set.

In that case there is a direct access from the DMFC to the DI which bypasses the DC allowing simultaneous access to both serial and parallel interfaces. When DI#_LLA_SER_ACCESS bit is set, the user must not do any other kind of accesses to the serial interface except LLA. When DI#_LLA_SER_ACCESS bit is set the corresponding DI#_WAIT4SERIAL must be clear.

45.4.10.7 Using a mask channel

The IPU is able to provide the windowing function on displays that have data enable control. This is achieved by masking of some screen regions according to a 1-bit/pixel mask read from the memory via IDMAC through channel #44.

When the mask value is zero, the pixel is not displayed. This feature can be used for dual-port smart displays.

45.4.11 Video De Interlacing or Combining Block (VDIC)

The Video De-Interlacer and Combiner has two operation modes:

- De-interlacing: converts an interlaced video stream to progressive order.
- Combining: combines two video/graphics planes and a background color.

The Video De-interlace block (VDIC) deinterlaces standard interlaced video to produce progressive video, that is used for upsizing to HD formats or for display on progressive displays. For VDIC operation three fields are necessary F(n-1),F(n),F(n+1). The F(n-1) field arrived through CSI interface in real time mode or through channel 1 and then stored in FIFO1. The F(n) arrived through channel 2. At least three lines of F(n) are stored in Line Store memory. The F(n+1) arrived through channel 3 and stored in FIFO3. FIFO controllers read data from FIFOs and then data aligned in pixels buffers. From the buffers the data arrived to Line Padding Controller (LPC). The LPC padding missing line at the beginning and end of the frame. The DeInterlacing sub-block (DI) perform the processing. Then data send to IC sub-block for processing or for transferring to external memory.

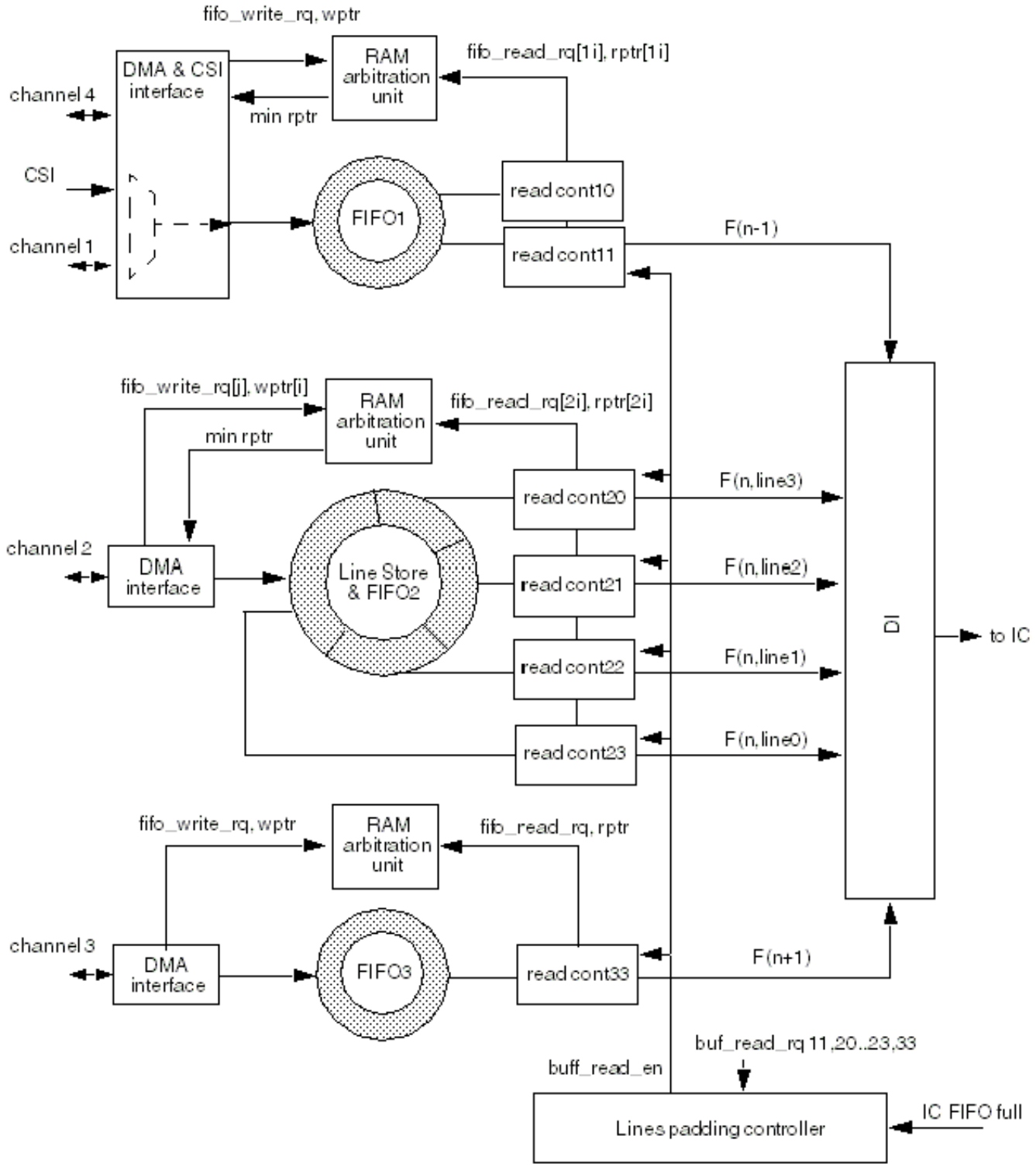


Figure 45-51. VDIC Block Diagram

45.4.11.1 VDIC FeaturesVDI Features

Key features of the VDIC block include:

Key features of the VDI block include:

- Deintarlacing
  - maximum horizontal resolution 968 pixels
  - maximum pixel rate 75MP
  - Support YUV422 and YUV420 formats
- CSI FIFO mode
- Combining

45.4.11.2 De interlacer (DI) sub-block

The block diagram of the DI block is shown in figure below.

Pipelining is inserted at all stages, so that the design may run at a fast clock speed, if needed.

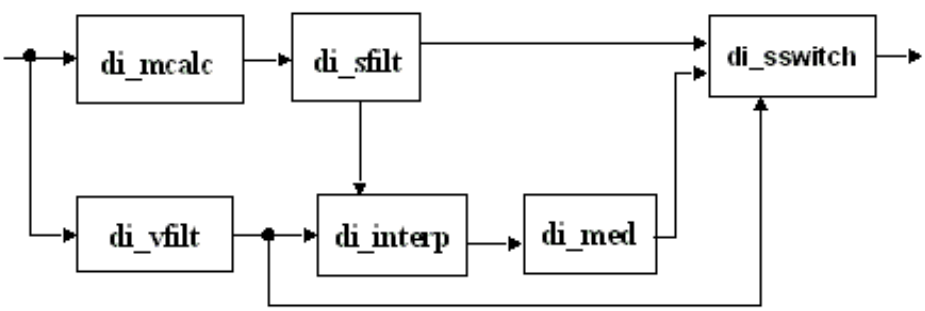


Figure 45-52. DI block diagram

45.4.11.2.1 Vertical Filter Block (di_vfilt)

The di_vfilt block performs spatial vertical filtering of pixels.

It is a four tap vertical filter:

vfilt_out = (-3.0*pix1 + 19.0*pix2 + 19.0*pix3 - 3.0*pix4)/32.0

Where pix1, pix2, pix3, pix4 are four pixels in same horizontal location on four consecutive lines of a field.

vfilt_out is pixel being predicted (between pix2 and pix3)

45.4.11.2.2 Motion Calculator Block (di_mcalc)

The mcalc block estimates the amount of motion for any given pixel by looking at pixel values in current, previous and next fields.

The generic formula used to calculate the estimated motion is:

m = SAT(Ks*(|e-w|/(|e-w|+|n-s|+SPA_DETAIL)))

- Where, n is the pixel above the pixel being predicted
- s is the pixel below the pixel being predicted
- e is the pixel in previous field at same spatial location as the pixel being predicted
- w is the pixel in next field at same spatial location as the pixel being predicted
- m is motion estimation for the pixel being predicted (range from 0 to 1)
- Ks is slope control and decides how quickly the algorithm switches from no motion (m=0) to full motion (m=1)
- SPA_DETAIL is a constant (50) that is added to |n-s|.
- SAT() is saturate at 1 function

The motion calculator block is simplified in a certain respect, by removing the need for a divider, while providing a degree of flexibility. The main motivators for this are the following observations:

1. the above equation defines a set of curves based on the value of |n-s|, but the curves are fairly closely spaced, so that using a granularity of 8 in |n-s| to define specific curves to use gives pretty much the same quality of picture as using all of the curves; and once |n-s| reaches about 120, the effect of an increased |n-s| value is hard to observe
2. once |e-w| gets to about 15, the motion is usually saturated to a value of 1

Based on the above observations, the motion calculator is now implemented with the use of two "ROM"s, using the 4 LSBs of |e-w| and bits 6:3 of |n-s|.

The motion calculator has 3 modes of operation. These modes are defined by the VDI_MOT_SEL field. In case that the user has an information about the motion (For example SW analyzing motion vectors provided by a video decoder) he can select one of the modes listed below. Changing the value of the VDI_MOT_SEL has an affect only on the next frame.

- When VDI_MOT_SEL==2'b01, m_calc is 0 (no motion - use weave)

- When VDI_MOT_SEL==2'b10, We assume high motion and use min(15,delta_t).
- When VDI_MOT_SEL==2'b00, We assume low motion and use sat([0,15],delta_t-8) (delta_t-8 is signed operation).

45.4.11.2.3 Spatial Motion Filter (di_sfilt)

The di_sfilt block spreads motion signal over five pixels:

Mspread = MAX(m3, (0.5*m1 + m2 + m3 + m4 + 0.5*m5)/4.0)

- Where, m3 is motion estimate for current pixel
- m2 is motion estimate for previous pixel
- m4 is motion estimate for next pixel
- m1 is motion estimate for pixel before previous pixel
- m5 is motion estimate for pixel after next pixel

45.4.11.2.4 Interpolated Pixel Calculator Block (di_interp)

The di_interp block uses the motion estimated by the di_sfilt block and computes an interpolated pixel that is weighted sum of the surrounding four pixels (n, s, e, w).

The block performs the following calculations:

```
          if (Mspread <= 0.5) {
i = (1-2*Mspread)*(e+w)/2 + 2*Mspread*vfilt_out
} else {
          i = vfilt_out
}
```

Where, i is the interpolated pixel

n, s, e, w are surrounding pixels as explained earlier

45.4.11.2.5 Median Filter Block (di_med)

The di_med block performs a 5-point median of n, s, e, w and i pixels.

It should be noted that the median required here is not a true 5-point, but can be implemented more efficiently as a 3-point median:

med = MEDIAN(min(max(n,s),max(e,w)), max(min(n,s),min(e,w)), i)

45.4.11.2.6 Soft Switch Block (di_sswitch)

The final output of the deinterlacer is a blend of the median value and the vertical filter, assuming that the pixel data n,s are uncorrelated with the pixel data e,w. By uncorrelated, we mean (max(n,s) < min(e,w)) or (min(n,s) > max(e,w)).

```
          if (Mspread <= 0.5 || not(F)) {
pix_out =    med
} else { /* Mspread > 0.5 */
pix_out = (1-2*(Mspread-0.5))*med + 2*(Mspread-0.5)*vfilt_out
}
```

45.4.11.3 DMA only Mode

In DMA only mode the data is coming from IDMAC only.

45.4.11.4 Real Time Mode

In Real Time Mode the F(n-1) are coming from CSI. The CSI write to FIFO1. The DI sub-block read F(n-1) from processing. In addition IDMAC read the field from FIFO1 and store in external memory. Then stored frames are used as F(n) and F(n+1).

45.4.11.5 CSI only Mode

In CSI only mode VDIC do not perform any processing and used as FIFO only. The data arrived from CSI written to FIFO1. The IDMAC read data from the FIFO1. The FIFO3 and Line store memory are not used. The ID sub-blocks are turned OFF. The CSI only mode can be used as alternative to SMFC, when appropriate.

45.4.11.6 Using Combining in the VDIC

As an alternate function to the de interacing function, the VDIC can perform combining of two planes.

- Both planes have to be at the same color space.
- Overlaying of a single plane over a unified background is supported.
- The planes can be at a different sizes - one plane can be smaller than the other.
- Combining requires a single cycle per output pixel.
- Alpha blending (global or local) is supported.
- Color keying is supported.

The plane's location and size is programmable as shown in the following figure.

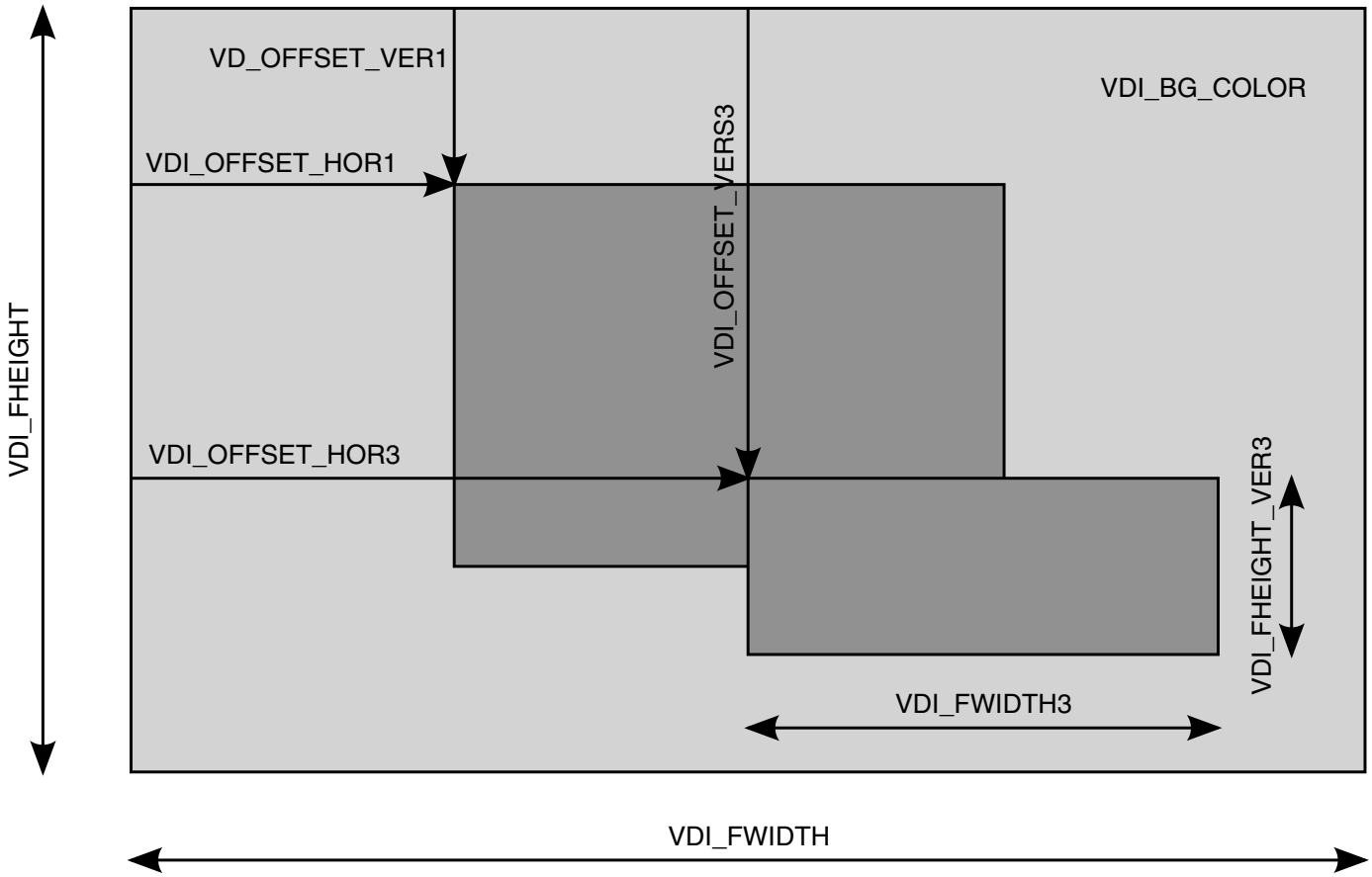


Figure 45-53. The relations between planes of the combining unit

When local alpha is used it should be arrived through channel3 of the VDIC.

The combining equation is:

OP = BG*(1 - alpha) + FG*alpha

Where "OP" is output pixel, "FG" is input pixel of foreground plan arrived thought channel3, BG is input pixel of background plan arrived thought channel1, "alpha" is global or local transparency.

45.4.11.7 VDIC Restrictions

- Maximum output pixel rate is 75 MP.
- The output pixel format (YUV422 or YUV420) are equal to input format.
- The VDIC can perform combining or de interlacing. It cannnot perform both functions at the same time.

45.4.12 Control Module (CM)

45.4.12.1 Block Diagram

The CM Block Diagram is shown in the figure below.

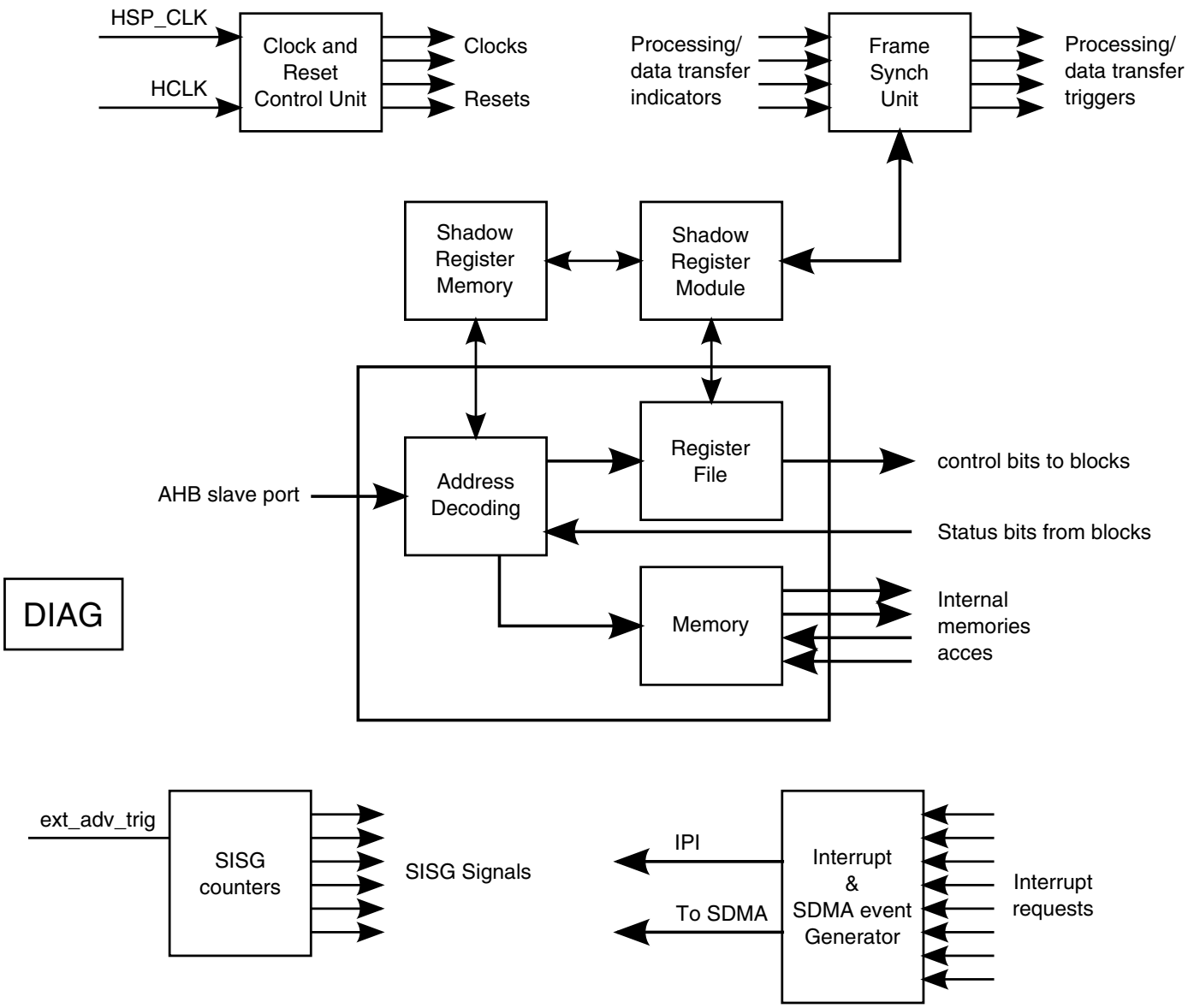


Figure 45-54. CM Block Diagram

The CM consists of the Frame Synchronization Unit (FSU), the Interrupt Generator (IG), the General Configuration Registers (GCR), the Clock and Reset Control Unit (CRCU) and the Shadow Registers Block (SRM).

45.4.12.2 Frame Synchronization Unit

This section details the frame synchronization unit.

45.4.12.2.1 General Description

The FSU provides synchronization of tasks performed by different IPU sub-blocks and ARM platform tasks. This allows to build complex processing flows which are performed automatically (without ARM platform involvement in synchronization of the IPU's tasks).

The FSU supports double buffering of image frames stored in the external memory and allows chaining IPU processing flows in automatic mode.

45.4.12.2.2 Frame Synchronization Flow

1. Initialization

   The ARM platform initializes all the parameters for a task by writing to the GCR and to the parameters memories of each IPU sub-blocks. The initialization must occur before the ARM platform enables the task.

2. Enabling

   After the initialization step has been completed, the ARM platform enables the task by setting its enable bit in an appropriate register.

3. Triggering

   After the task is enabled, the FSU waits for triggering signal. The triggering signals is a combination of the enable bit and the buffer ready signal which can be driven by the ARM platform (DMA_CH_BUF<0/1>_RDY_<#>) and/or by the preceding task (IDMAC_EOF_# or Frame Complete signal from the DC).

   The trigger causes the FSU to invoke the relevant unit to start by assertion of the NEW_FRM_RDY signal. In some cases triggering occurs at the enabling step (when the enable bit is the trigger for the task).

4. Operating

   The triggering step cause the task to move to active mode, this is the operating step. In this step, the FSU monitors the synchronization signals from ARM platform, IDMAC and the corresponding processing units, and controls the units operation. The FSU also controls the IDMAC buffer toggling when double buffer page flipping is used.

The FSU checks at end of each frame if the next frame can be served. If the answer is yes, the FSU stays in active mode with re-sending the <TASK>_NEW_FRM_RDY signal and updating the relevant flags (e.g. DMA<BL>_<#>_CUR_BUF and DMA<BL>_<#>_BUF_RDY). If the answer is no, the FSU moves to pause mode and pauses the task waiting until the next frame can be served.

5. Disabling

   When the task is disabled by the ARM platform (by negating the enable bit), it moves back to non-active mode.

45.4.12.2.3 FSU's fundamentals

Trigger source select

A flow is triggered by a trigger. The trigger may be asserted manually by the ARM platform or may be a result of the completion of the preceding task. Trigger's source select choose the source of the trigger. The trigger means that the data is ready to be processed by the sub-blocks. The trigger source select is defined by the corresponding SRC_SEL bits of the block or task.

Trigger destination select

A block or task that process data needs to know that the following task in the chain is ready to receive the processed data. The user needs to specify what is the destination of the processed data. This is done by setting the corresponding DEST_SEL bits of the block or task.

Double buffering

The IPU supports double buffering in the system's memory. When a flow is processed frame by frame the first frame will be read from one location (BUF0) in the memory, the next frame will be read from another location in the memory (BUF1). The location in the memory of the buffers is defined by the EBA0 and EBA1 parameters of the corresponding channel in the CPMEM. The IDMAC use the correct buffer according to the DMA_CH_CUR_BUF_# signal. The FSU automatically toggles the DMA_CH_CUR_BUF_# to point on the correct buffer to be used by the channel. In order to work in double buffer mode the corresponding DMA_CH_DB_MODE_SEL_# needs to be set.

Alternative flow

Some of the IPU sub-blocks can handle 2 flows via them one is the main flow and the other is the alt flow. In order to support an alternate flow via the same sub-block an alternative configuration should be used by the sub-block. This includes

- Alternate registers including an alternative sub-block's setup - this is handled by the SRM
- Alternate IDMAC settings: parameters in the CPMEM, separate alpha
- Alternate SRC_SEL as the source of the trigger may come from a different function.
- Alternate FSU settings (CUR_BUF, BUF_RDY, DB_MODE_SEL)
- Some of the display sub-blocks alternate settings are handled by programming an alternative set of registers.

The FSU handles the switch to the alternate flow, it controls the update of the alternate configuration and send the appropriate signals to other sub-blocks in the IPU indicating about a need to switch to the alternate configuration.

Once a frame is completed there is a chance that there are 2 buffers ready. One is the next buffer of the current flow and the other is the a buffer in an alternate flow. The FSU will automatically select between the next buffer to handle in a round-robin fashion.

IPU task chaining - Single flow

The diagram below illustrates task chaining in the In this example a single flow is handled

The Frames are coming from the same camera sub-block. The frames are handled frame by Frame. The first frame arriving Frame0 is stored in BUF0 of the "INPUT BUFFER". Once the Frame0 is ready in BUF0, the processing sub-block is triggered and data is read from BUF0 to the processing sub-block. In the meantime, Frame1 coming from the camera is written to BUF1 of the "INPUT BUFFER". The processing sub-block processes the data and stores the first frame on BUF0 of the "OUTPUT BUFFER". Once the processing is done and the frame is ready, the display sub-block is triggered to pick the data and send it to the display. The processing sub-block will start working on the next frame once the data is ready in the "INPUT_BUFFER" and there's a free buffer in the "OUTPUT_BUFFER".

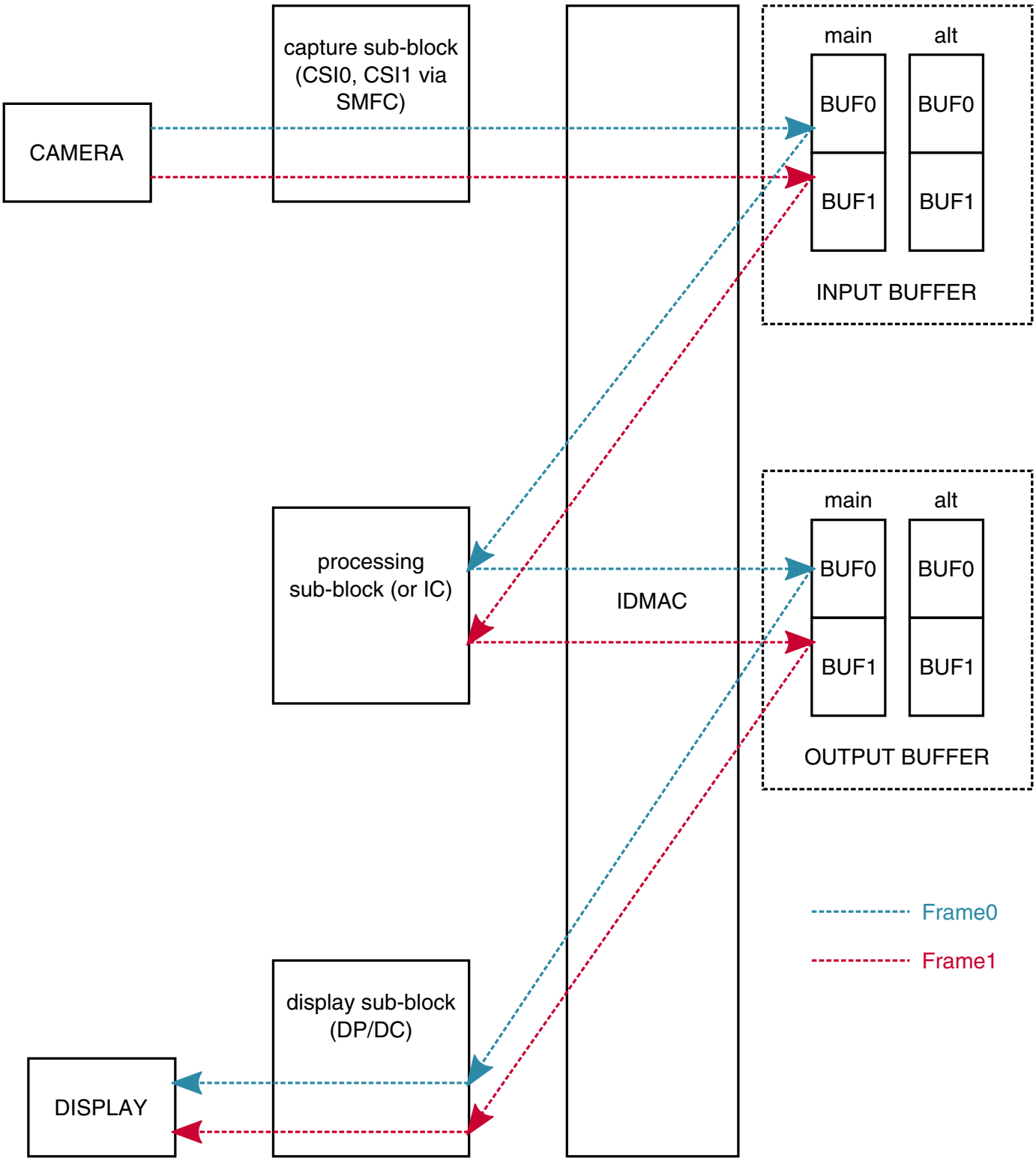


Figure 45-55. IPU tasks' chaining illustration - single flow

IPU task chaining - double flow

The diagram below illustrates task chaining in the In this example double flow is handled. In addition to the flow described on the previous example, another flow is handled via the display sub-blocks. In that case the DC will handle 2 flows. Once sending Frame0 to the display is complete, the FSU will decide in whether to handle Frame1 or Frame0_ALT. The decision is made according the readiness of the other buffers in the memory, in case of 2 ready buffers (main flow and alternate) the FSU will switch between them in a round robin fashion.

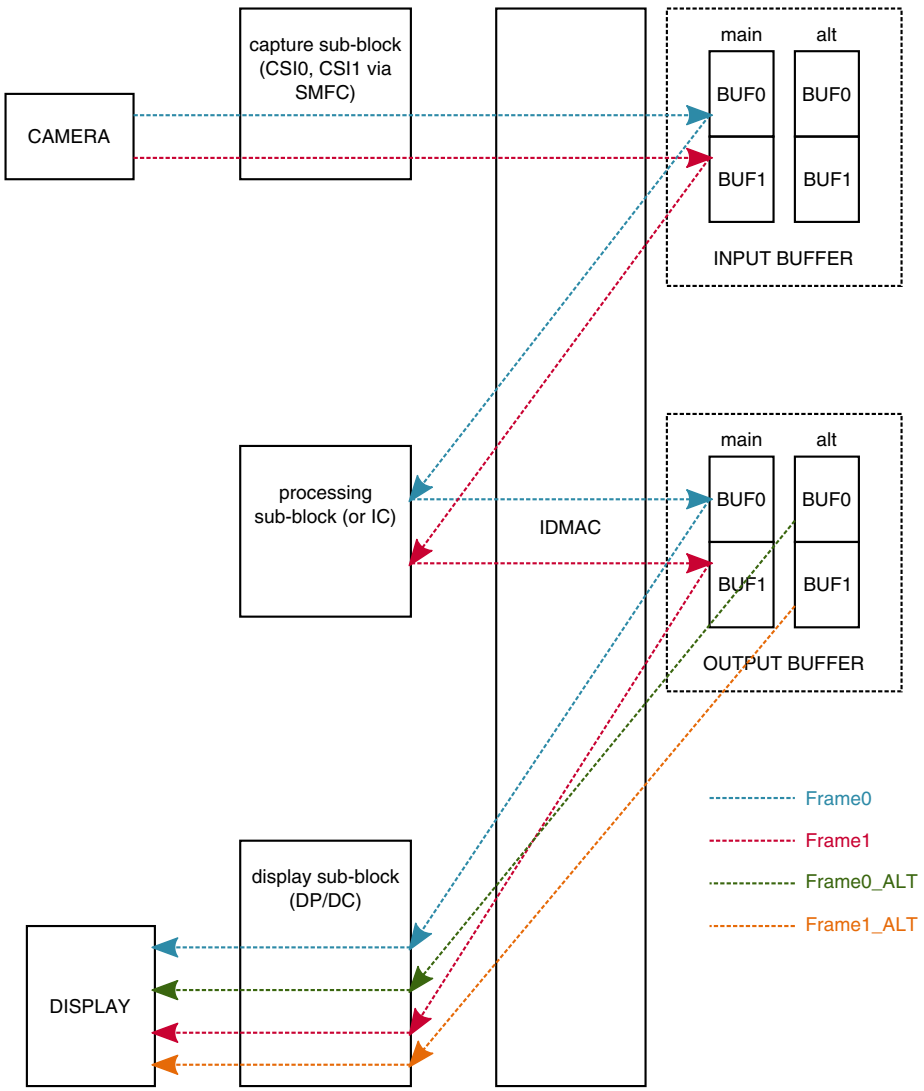


Figure 45-56. IPU tasks' chaining illustration - double flow

45.4.12.2.4 IPU main flows

IPU's flows can be partitioned into 5 groups.

- CF - capture flows

- PF - processing flows
- SF - synchronous display flows
- AF - Asynchronous display flows
- DF - Direct flow from IC to the display

Tasks from different groups can be chained as illustrated below.

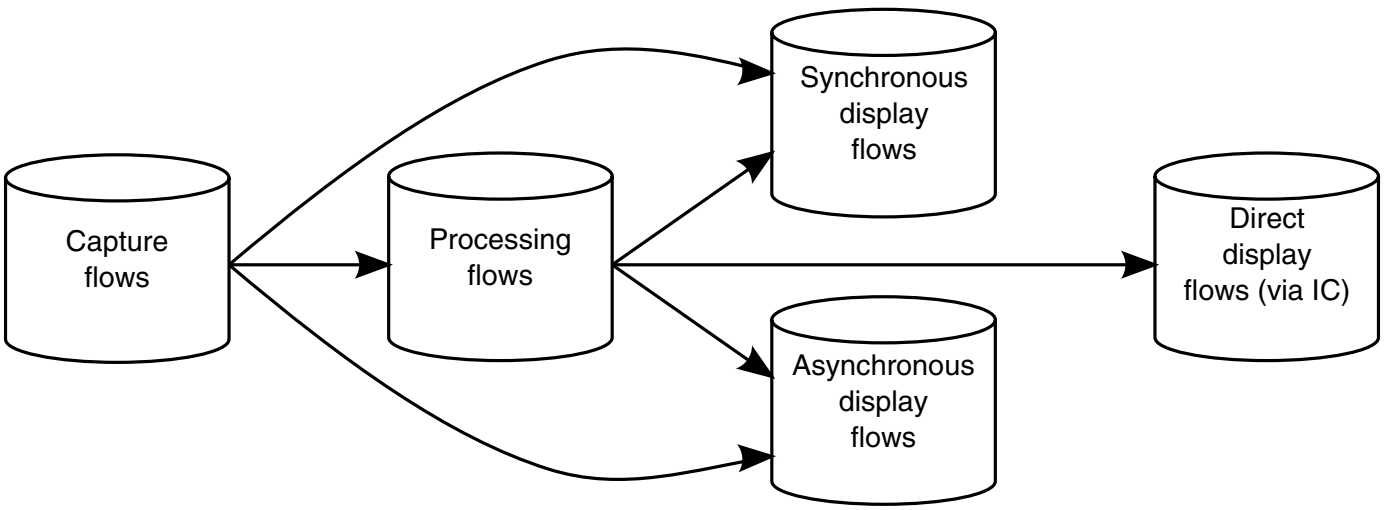


Figure 45-57. IPU task flows chaining

The tables below describe the most important use cases of chaining the IPU tasks. The tables show the main task for each group.

The physical DMA channel is the DMA channel that is used for the main flow - this is the IDMAC channel that is physically connected to the block - the CPMEM parameters should be configured according to the physical channel number. In case of an alternate flow then an alternate entry in the CPMEM should be configured as well

The following table describes capturing flows where data is captured from the sensor and sent to the memory without processing. This flows can be chained to the processing and display flows described on: Table 45-29, Table 45-30, Table 45-31 and on Table 45-32.

Table 45-28. IPU's capture flows

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Capturing image from sensor and storing it in the memory (via SMFC) without processing	CSI0 or CSI1 --> SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3

Table continues on the next page...

Table 45-28. IPU's capture flows (continued)

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Capturing image from sensor and storing it in the memory (via IC) without processing	CSI0 or CSI1 --> MEM	---	---	IDMAC_CH_5
	Capturing interlaced input and storing it in the memory (via VDIC) while performing video de-interlacing in the VDIC.	CSI0 or CSI1 -> VDIC --> MEM		IDMAC_CH_9 IDMAC_CH_10	IDMAC_CH_5 IDMAC_CH_13
		Interlaced input coming from one of the CSIs is sent to the memory without processing via channel #13. In addition 2 more inputs are read from the memory via channels 9 and 10. The processed image is written to the memory via ch 5 in progressive scan mode.			

The following table describe processing flows via the IC . This flows can start from the memory, can be chained to a capturing flow described on Table 45-28. The target of this flow can be chained to the display flows described on Table 45-30 and on Table 45-31.

Table 45-29. IPU's Processing flows

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Preprocessing image from sensor for encoding	CSI0 or CSI1 --> SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (PRP ENC) --> MEM	IDMAC_CH_12	---	IDMAC_CH_20
	Preprocessing image from memory for encoding	MEM --> IC (PRP ENC) --> MEM	IDMAC_CH_12	---	IDMAC_CH_20
	Preprocessing image from sensor for encoding	CSI0 or CSI1 --> IC (PRP ENC) --> MEM	---	---	IDMAC_CH_20
	Preprocessing + rotation of image from sensor for encoding	CSI0 or CSI1 -->SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (PRP ENC) --> MEM	IDMAC_CH_12	---	IDMAC_CH_20
		MEM --> IC (ROT ENC) --> MEM	IDMAC_CH_45	---	IDMAC_CH_48
	Preprocessing + rotation of image from memory for encoding	MEM --> IC (PRP ENC) --> MEM	IDMAC_CH_12	---	IDMAC_CH_20
		MEM --> IC (ROT ENC) --> MEM	IDMAC_CH_45	---	IDMAC_CH_48

Table continues on the next page...

Table 45-29. IPU's Processing flows (continued)

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Rotation and preprocessing of image from sensor for encoding	CSI0 or CSI1 --> SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (ROT ENC) --> MEM	IDMAC_CH_45	---	IDMAC_CH_48
		MEM --> IC (PRP ENC) --> MEM	IDMAC_CH_12	---	IDMAC_CH_20
	Preprocessing image from sensor for viewfinder	CSI0 or CSI1 --> SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (PRP VF) --> MEM	IDMAC_CH_12	IDMAC_CH_14	IDMAC_CH_21
	Preprocessing image from memory for viewfinder	MEM --> IC (PRP VF) --> MEM	IDMAC_CH_12	IDMAC_CH_14	IDMAC_CH_21
	Preprocessing and rotation of image from sensor for viewfinder	CSI0 or CSI1 --> SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (PRP VF) --> MEM	IDMAC_CH_12	IDMAC_CH_14	IDMAC_CH_21
		MEM --> IC (ROT VF) --> MEM	IDMAC_CH_46	---	IDMAC_CH_49
	Rotation and preprocessing of image from sensor for viewfinder	CSI0 or CSI1 -->SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (ROT VF) --> MEM	IDMAC_CH_46	---	IDMAC_CH_49
		MEM --> IC (PRP VF) --> MEM	IDMAC_CH_12	IDMAC_CH_14	IDMAC_CH_21
	Preprocessing and rotation of image from sensor for viewfinder	MEM --> IC (PRP VF) --> MEM	IDMAC_CH_12	IDMAC_CH_14	IDMAC_CH_21
		MEM --> IC (ROT VF) --> MEM	IDMAC_CH_46	---	IDMAC_CH_49
	Preprocessing image from sensor	CSI0 or CSI1 --> IC (PRP VF) --> MEM	---	IDMAC_CH_14	IDMAC_CH_21
	Preprocessing and rotation of image from sensor	CSI0 or CSI1 --> IC (PRP VF) --> MEM	---	IDMAC_CH_14	IDMAC_CH_21
		MEM --> IC (ROT VF) --> MEM	IDMAC_CH_46	---	IDMAC_CH_49
	Postprocessing image	MEM --> IC (PP) --> MEM	IDMAC_CH_11	IDMAC_CH_15	IDMAC_CH_22

Table continues on the next page...

Table 45-29. IPU's Processing flows (continued)

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Postprocessing and rotation of image	MEM --> IC (PP) --> MEM	IDMAC_CH_11	IDMAC_CH_14	IDMAC_CH_22
		MEM --> IC (ROT PP) -->MEM	IDMAC_CH_47	---	IDMAC_CH_50
	Rotation and postprocessing of image	MEM --> IC (ROT PP) -->MEM	IDMAC_CH_47	---	IDMAC_CH_50
		MEM --> IC (PP) --> MEM	IDMAC_CH_11	IDMAC_CH_14	IDMAC_CH_22
	Postprocessing image from sensor	CSI0 or CSI1 -->SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (PP) --> MEM	IDMAC_CH_11	IDMAC_CH_15	IDMAC_CH_22
	video de-interlacing in the VDIC.	MEM -> VDIC --> MEM	IDMAC_CH_8 IDMAC_CH_9 IDMAC_CH_10	--	IDMAC_CH_5
		Interlaced input coming from the memory via 3 channels 8, 9 and 10. The processed image is written to the memory via ch 5 in progressive scan mode.			
	video de-interlacing in the VDIC. Then processing in the IC.	MEM -> VDIC --> IC (PRP VF) --> MEM	IDMAC_CH_8 IDMAC_CH_9 IDMAC_CH_10	IDMAC_CH_14	IDMAC_CH_21
		Interlaced input coming from the memory via 3 channels 8, 9 and 10. The processed image is sent to the IC for further processing then it is sent to the memory via ch 21			
	video de-interlacing in the VDIC, then preprocessing and rotation of image coming from memory	MEM --> VDIC --> IC (PRP VF) --> MEM	IDMAC_CH_8 IDMAC_CH_9 IDMAC_CH_10	IDMAC_CH_14	IDMAC_CH_21
		MEM --> IC (ROT VF) --> MEM	IDMAC_CH_46	---	IDMAC_CH_49
	video de-interlacing in the VDIC, then preprocessing and rotation of image coming from CSI	CSI0 or CSI1 --> VDIC--> IC (PRP VF) --> MEM	IDMAC_CH_9 IDMAC_CH_10	IDMAC_CH_14	IDMAC_CH_13 IDMAC_CH_21
		MEM --> IC (ROT VF) --> MEM	IDMAC_CH_46	---	IDMAC_CH_49
	Combining in the VDIC.	MEM -> VDIC --> MEM	IDMAC_CH_25 IDMAC_CH_26	--	IDMAC_CH_5
		Plane3 is coming on IDMAC_CH_25; Additional plane (plane1) may come from IDMAC_CH_26.			
	Combining in the VDIC. Then processing in the IC.	MEM -> VDIC --> IC (PRP VF) --> MEM	IDMAC_CH_25 IDMAC_CH_26	IDMAC_CH_14	IDMAC_CH_21
		Plane3 is coming on IDMAC_CH_25; Additional plane (plane1) may come from IDMAC_CH_26.			

Table continues on the next page...

Table 45-29. IPU's Processing flows (continued)

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Combining in the VDIC, then preprocessing and rotation	MEM --> VDIC --> IC (PRP VF) --> MEM	IDMAC_CH_25 IDMAC_CH_26	IDMAC_CH_14	IDMAC_CH_21
		MEM --> IC (ROT VF) --> MEM		---	IDMAC_CH_49
		Plane3 is coming on IDMAC_CH_25; Additional plane (plane1) may come from IDMAC_CH_26.			
	Rotation of an image and Combining in the VDIC	MEM --> IC (ROT VF) -->MEM	IDMAC_CH_46	---	IDMAC_CH_49
		MEM -> VDIC --> MEM	IDMAC_CH_25 IDMAC_CH_26	--	IDMAC_CH_5
		Plane3 is rotated on the VF task. Then it sent to the VDIC IDMAC_CH_25; Additional plane (plane1) may come from IDMAC_CH_26.			
	Rotation of an image and Combining in the VDIC	MEM --> IC (ROT ENC) -->MEM	IDMAC_CH_45	---	IDMAC_CH_48
		MEM -> VDIC --> MEM	IDMAC_CH_25 IDMAC_CH_26	--	IDMAC_CH_5
		Plane1 is rotated on the ENC task. Then it sent to the VDIC IDMAC_CH_26; In additional plane3 is coming from IDMAC_CH_25			
	Rotation of two images and Combining in the VDIC	MEM --> IC (ROT VF) -->MEM	IDMAC_CH_46	---	IDMAC_CH_49
		MEM --> IC (ROT ENC) -->MEM	IDMAC_CH_45	---	IDMAC_CH_48
		MEM -> VDIC --> MEM	IDMAC_CH_25 IDMAC_CH_26	--	IDMAC_CH_5
		Plane3 is rotated on the VF task. Plane1 is rotated on the ENC task.			
	Rotation of two images and Combining in the VDIC. Then processing in the IC.	MEM --> IC (ROT VF) -->MEM	IDMAC_CH_46	---	IDMAC_CH_49
		MEM --> IC (ROT ENC) -->MEM	IDMAC_CH_45	---	IDMAC_CH_48
		MEM -> VDIC --> IC (PRP VF) --> MEM	IDMAC_CH_25 IDMAC_CH_26	IDMAC_CH_14	IDMAC_CH_21
		Plane3 is rotated on the VF task. Plane1 is rotated on the ENC task.			

The following table describes the synchronous display flows, this flows can be chained to the capture, processing and direct flows described on Table 45-28,Table 45-28and on Table 45-32

Table 45-30. IPU synchronous display flows

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Synchronous display refresh via DC	MEM --> DC	IDMAC_CH_28	---	---

Table continues on the next page...

Table 45-30. IPU synchronous display flows (continued)

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Synchronous display refresh via DP	MEM --> DP	IDMAC_CH_23	---	---
	Synchronous display refresh via DC	MEM --> DC	IDMAC_CH_28	IDMAC_CH_44	---
		Comment: IDMAC_CH_44 is an optional mask Channel			
	Synchronous display refresh via DP	MEM --> DP	IDMAC_CH_23	IDMAC_CH_44	---
		Comment: IDMAC_CH_44 is an optional mask Channel			
	Synchronous display refresh via DP + combining in the DP	MEM --> DP SYNC(BG/FG)	IDMAC_CH_23	IDMAC_CH_27	---
		Comment: IDMAC_CH_23 is the main channel; IDMAC_CH_27 is used for combining of another plane.			
	Synchronous display refresh via DP	MEM --> DP SYNC(BG/FG)	IDMAC_CH_23 / IDMAC_CH_27	IDMAC_CH_44 (mask)	---
		Comment: IDMAC_CH_23 is the main channel; IDMAC_CH_27 is optional and can be used for combining of another plane. IDMAC_CH_44 is an optional mask Channel			

The following table describes the asynchronous display flows, this flows can be chained to the capture, processing and direct flows described onTable 45-28,Table 45-28and on Table 45-32

Table 45-31. IPU Asynchronous display flows

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Asynchronous display refresh via DP	MEM --> DC	IDMAC_CH_24	---	---
	Asynchronous display refresh via DP	MEM --> DC	IDMAC_CH_24 / IDMAC_CH_29	---	---
		Comment: IDMAC_CH_29 is another input to the DP to be combined with data coming from IDMAC_CH_24			
	Asynchronous display refresh via DP in command buffer mode	MEM --> DC	IDMAC_CH_24 / IDMAC_CH_29	IDMAC_CH_42 (command)	---
		Comment: IDMAC_CH_29 is another input to the DP to be combined with data coming from IDMAC_CH_24			
	Asynchronous display refresh via DC	MEM --> DC	IDMAC_CH_28	---	---

Table continues on the next page...

Table 45-31. IPU Asynchronous display flows (continued)

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Asynchronous display refresh via DC	MEM --> DC	IDMAC_CH_41	---	---
	Asynchronous display refresh via DC	MEM --> DC	IDMAC_CH_28	IDMAC_CH_42(command)	---
		Comment: IDMAC_CH_42 can be optionally used as command channel associated with IDMAC_CH_28			
	Asynchronous display refresh via DC	MEM --> DC	IDMAC_CH_41	IDMAC_CH_43(command)	---
		Comment: IDMAC_CH_43 can be optionally used as command channel associated with IDMAC_CH_41			
	Asynchronous display refresh via DP	MEM --> DP ASYNC	IDMAC_CH_24	IDMAC_CH_42(command)	---
		Comment: IDMAC_CH_42 can be optionally used as command channel associated with IDMAC_CH_24			
	Asynchronous display refresh via DP	MEM --> DC	IDMAC_CH_24	IDMAC_CH_43 (command)	---
		Comment: IDMAC_CH_43 can be optionally used as command channel associated with IDMAC_CH_24			
	Reading data from asynchronous display	DC --> MEM	---	---	IDMAC_CH_50

The table below describes direct flows via the IC. These are processing flows via the IC where the output is sent directly to the DMFC. From that point the any of the display flows described on Table 45-30 and on Table 45-31 can be chained.

Table 45-32. IIPU direct flows to the display via the IC

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Rotation and preprocessing of image from sensor for viewfinder and displaying it on synchronous display via DP	CSI0 or CSI1 --> SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (ROT VF) --> MEM	IDMAC_CH_46	---	IDMAC_CH_49
		MEM --> IC (PRP VF) --> DMFC	IDMAC_CH_12	IDMAC_CH_14	Direct to the DMFC
		MEM --> DP SYNC (BG/FG)	IDMAC_CH_23 / IDMAC_CH_27	IDMAC_CH_44 (mask)	---

Table continues on the next page...

Table 45-32. IIPU direct flows to the display via the IC (continued)

	Flow	Tasks chain	Physical DMA Channels		
			Video Input	Other Input	Output
	Rotation and preprocessing of image from sensor for viewfinder and displaying it on asynchronous display via DP	CSI0 or CSI1 --> SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (ROT VF) --> MEM	IDMAC_CH_46	---	IDMAC_CH_49
		MEM --> IC (PRP VF) --> DMFC	IDMAC_CH_12	IDMAC_CH_14	Direct to the DMFC
		MEM --> DP SYNC (BG/FG)	IDMAC_CH_24 / IDMAC_CH_29	command channel	---
	Preprocessing image from sensor for viewfinder and direct displaying it on asynchronous display	CSI0 or CSI1 --> SMFC --> MEM	---	---	IDMAC_CH_0 IDMAC_CH_1 IDMAC_CH_2 IDMAC_CH_3
		MEM --> IC (PRP VF) --> DC	IDMAC_CH_12	IDMAC_CH_14	---
	Preprocessing image from sensor for viewfinder and direct displaying it on asynchronous display	CSI0 or CSI1 --> IC (PRP VF) --> DC	---	IDMAC_CH_14	IDMAC_CH_21
	Preprocessing image from sensor for viewfinder and direct displaying it on asynchronous display via DP	CSI0 or CSI1 --> IC (PRP VF) --> DP ASYNC	---	IDMAC_CH_14	IDMAC_CH_21
	Capturing interlaced input via CSI, then perform video de-interlacing in the VDIC. Then processing in the IC and direct displaying it on asynchronous display via DP	CSI0 or CSI1 -> VDIC --> IC (PRP VF) --> DP ASYNC	--	IDMAC_CH_9 IDMAC_CH_10 IDMAC_CH_14	IDMAC_CH_13 IDMAC_CH_21
		Interlaced input coming from one of the CSIs is sent to the memory without processing via channel #13. In addition 2 more inputs are read from the memory via channels 9 and 10. The processed image is sent to the IC for further processing then it is sent to the display			
	Capturing interlaced input via CSI, then perform video de-interlacing in the VDIC. Then processing in the IC and direct displaying it on asynchronous display	CSI0 or CSI1 -> VDIC --> IC (PRP VF) --> DC	--	IDMAC_CH_9 IDMAC_CH_10 IDMAC_CH_14	IDMAC_CH_13 IDMAC_CH_21
		Interlaced input coming from one of the CSIs is sent to the memory without processing via channel #13. In addition 2 more inputs are read from the memory via channels 9 and 10. The processed image is sent to the IC for further processing then it is sent to the display			

45.4.12.2.5 Sub-Frame Double-Buffering (Band Mode)

Page-flip double buffering is performed using full-frame buffers. In addition IPU supports also page-flip double buffering using smaller buffers, each containing 4/8/16/32/64/128/256 rows of pixels. This allows the use of internal memory for buffering.

This mode can be supported by the following modifications (relative to full-frame buffers):

The address in system memory is generated by the same formula, but inserting the full row number, only k LSB's (for a band of 2^k rows) are inserted (e.g. [2:0] for 8 rows).

Page flip is triggered at the end of each band (when the k LSB's are all 1) and not only at the end of the frame. This flip may be accompanied by an ARM/SDMA interrupt, if synchronization with other modules (e.g. VPU or ARM) is needed.

The channels that can work in this mode are controlled by the corresponding IDMAC_BNDM_EN_<i>

bit. The BNDM parameter of the corresponding IDMAC channel has to be set as well.

45.4.12.2.6 Snooping

This function allows to reduce the rate of write accesses from the system memory to a smart display. When snooping is enabled, writing to the display is performed only if the corresponding system memory buffers has been changed. Any change of data in the system memory buffers is registered by the External Memory Controller of the chip.

When writing to the system memory is performed, the External Memory Controller asserts the emi_snooping1 or emi_snooping2 pins of the After receiving these signals, the CM initiates transfer of the corresponding buffer to a display buffer, either in system's memory or in a smart display. Selecting the snooping mechanism as the trigger for the flow is done by setting the <>_SRC_SEL bits of the corresponding flow to snoop1 or snoop2 mode.

The emi_snooping1 and emi_snooping2 signals are synchronized to the IPU's clock. The synchronize on the emi_snooping2 signal can be bypassed by setting the bit SNOOP2_SYNC_BYP bit.

45.4.12.2.7 Automatic Window Refresh

By programming the <>_SRC_SEL bits of the corresponding flow to autoref mode, automatic refresh of a window on the smart display is enabled. This means that the flow is triggered any time the refresh counter completes its counting. The refresh period is defined via the AUTOREF_PER field (the time unit is 2^{17} periods of the HSP_CLK clock).

The actual value of refresh period is equal to:

THSP*2^17 *(AUTO_REF_PER+1).

45.4.12.2.8 Auto-refresh and snooping

Auto-refresh can be conditional. In this case, it is accomplished emi_snooping1 or emi_snooping2 pins of the IPU are asserted. This is defined by setting the <>_SRC_SEL bits to autoref+snoop1 or autoref_snoop2 mode.

45.4.12.2.9 Synchronization with A Video/Graphics source

The IPU supports direct synchronization with a video/graphics source (e.g. an integrated graphics accelerator, rendering graphics frames). Up to three frame buffers are supported for this synchronization. It is performed using the following signals

- The source instructs the i.MX53EX which frame should be transferred to the display by providing a 3 bit one-hot vector.
- The IPU notifies the source which frame is currently transferred to the display by providing a 3 bit one-hot vector.
- The IPU provides to the source an end-of-frame trigger (after which it switches to the frame indicated by the source).

This mechanism is provided for the main plane of the primary flow to the DP (ch 23). The SRC_SEL of this channel has to be defined as external source.

Triple buffer mode is enabled by setting the channel's corresponding bit on IPU_CH_TRB_MODE_SEL register. Setting this bit overrides the channel's corresponding settings on the IPU_CH_DB_MODE_SEL register. The CPMEM's settings of the corresponding channel can hold 2 pointers for 2 buffers (EBA0 and EBA1). The third pointer resides on an alternate entry in the CPMEM. The pointer to the alternate entry is defined by the channel's corresponding IDMAC_SUB_ADDR field. EBA0 of the alternate entry in the CPMEM is used as a third pointer. All the other settings of alternate entry in the CPMEM must be the same as the channel's standard entry in the CPMPM.

Once the channel reached end-of-frame event, the FSU automatically switch between pointers according to the signals coming from the source and notifies the source that the IPU reached the end-of-frame.

IPU has 2 identical bridges allowing direct synchronization with 2 sources. The 2 bridges support exactly the same protocol.

45.4.12.3 Interrupt Generator

The IG produces two interrupts to the ARM platform - the functional interrupt and the error interrupt. All of the interrupts are maskable.

The following table describes the functional interrupts.

Table 45-33. Functional Interrupts Summary

Location	Sub-blocks	Interrupt status bit name	Description
IPU_INT_STAT_1[0]	IDMAC	IDMAC_EOF_0	-
IPU_INT_STAT_1[1]	IDMAC	IDMAC_EOF_1	-
IPU_INT_STAT_1[2]	IDMAC	IDMAC_EOF_2	-
IPU_INT_STAT_1[3]	IDMAC	IDMAC_EOF_3	-
IPU_INT_STAT_1[5]	IDMAC	IDMAC_EOF_5	-
IPU_INT_STAT_1[11]	IDMAC	IDMAC_EOF_11	-
IPU_INT_STAT_1[12]	IDMAC	IDMAC_EOF_12	-
IPU_INT_STAT_1[14]	IDMAC	IDMAC_EOF_14	-
IPU_INT_STAT_1[15]	IDMAC	IDMAC_EOF_15	-
IPU_INT_STAT_1[17]	IDMAC	IDMAC_EOF_17	-
IPU_INT_STAT_1[18]	IDMAC	IDMAC_EOF_18	-
IPU_INT_STAT_1[20]	IDMAC	IDMAC_EOF_20	-
IPU_INT_STAT_1[21]	IDMAC	IDMAC_EOF_21	-
IPU_INT_STAT_1[22]	IDMAC	IDMAC_EOF_22	-
IPU_INT_STAT_1[23]	IDMAC	IDMAC_EOF_23	-
IPU_INT_STAT_1[24]	IDMAC	IDMAC_EOF_24	-
IPU_INT_STAT_1[27]	IDMAC	IDMAC_EOF_27	-
IPU_INT_STAT_1[28]	IDMAC	IDMAC_EOF_28	-
IPU_INT_STAT_1[29]	IDMAC	IDMAC_EOF_29	-
IPU_INT_STAT_1[31]	IDMAC	IDMAC_EOF_31	-
IPU_INT_STAT_2[1]	IDMAC	IDMAC_EOF_33	-
IPU_INT_STAT_2[8]	IDMAC	IDMAC_EOF_40	-

Table continues on the next page...

Table 45-33. Functional Interrupts Summary (continued)

Location	Sub-blocks	Interrupt status bit name	Description
IPU_INT_STAT_2[9]	IDMAC	IDMAC_EOF_41	-
IPU_INT_STAT_2[10]	IDMAC	IDMAC_EOF_42	-
IPU_INT_STAT_2[11]	IDMAC	IDMAC_EOF_43	-
IPU_INT_STAT_2[12]	IDMAC	IDMAC_EOF_44	-
IPU_INT_STAT_2[13]	IDMAC	IDMAC_EOF_45	-
IPU_INT_STAT_2[14]	IDMAC	IDMAC_EOF_46	-
IPU_INT_STAT_2[15]	IDMAC	IDMAC_EOF_47	-
IPU_INT_STAT_2[16]	IDMAC	IDMAC_EOF_48	-
IPU_INT_STAT_2[17]	IDMAC	IDMAC_EOF_49	-
IPU_INT_STAT_2[18]	IDMAC	IDMAC_EOF_50	-
IPU_INT_STAT_2[19]	IDMAC	IDMAC_EOF_51	-
IPU_INT_STAT_2[20]	IDMAC	IDMAC_EOF_52	-
IPU_INT_STAT_3[0]	IDMAC	IDMAC_NFACK_0	-
IPU_INT_STAT_3[1]	IDMAC	IDMAC_NFACK_1	-
IPU_INT_STAT_3[2]	IDMAC	IDMAC_NFACK_2	-
IPU_INT_STAT_3[3]	IDMAC	IDMAC_NFACK_3	-
IPU_INT_STAT_3[5]	IDMAC	IDMAC_NFACK_5	-
IPU_INT_STAT_3[8]	IDMAC	IDMAC_NFACK_8	-
IPU_INT_STAT_3[9]	IDMAC	IDMAC_NFACK_9	-
IPU_INT_STAT_3[10]	IDMAC	IDMAC_NFACK_10	-
IPU_INT_STAT_3[11]	IDMAC	IDMAC_NFACK_11	-
IPU_INT_STAT_3[12]	IDMAC	IDMAC_NFACK_12	-
IPU_INT_STAT_3[13]	IDMAC	IDMAC_NFACK_13	-
IPU_INT_STAT_3[14]	IDMAC	IDMAC_NFACK_14	-
IPU_INT_STAT_3[15]	IDMAC	IDMAC_NFACK_15	-
IPU_INT_STAT_3[17]	IDMAC	IDMAC_NFACK_17	-
IPU_INT_STAT_3[18]	IDMAC	IDMAC_NFACK_18	-
IPU_INT_STAT_3[20]	IDMAC	IDMAC_NFACK_20	-
IPU_INT_STAT_3[21]	IDMAC	IDMAC_NFACK_21	-
IPU_INT_STAT_3[22]	IDMAC	IDMAC_NFACK_22	-
IPU_INT_STAT_3[23]	IDMAC	IDMAC_NFACK_23	-
IPU_INT_STAT_3[24]	IDMAC	IDMAC_NFACK_24	-
IPU_INT_STAT_3[27]	IDMAC	IDMAC_NFACK_27	-
IPU_INT_STAT_3[28]	IDMAC	IDMAC_NFACK_28	-
IPU_INT_STAT_3[29]	IDMAC	IDMAC_NFACK_29	-

Table continues on the next page...

Table 45-33. Functional Interrupts Summary (continued)

Location	Sub-blocks	Interrupt status bit name	Description
IPU_INT_STAT_3[31]	IDMAC	IDMAC_NFACK_31	-
IPU_INT_STAT_4[1]	IDMAC	IDMAC_NFACK_33	-
IPU_INT_STAT_4[8]	IDMAC	IDMAC_NFACK_40	-
IPU_INT_STAT_4[9]	IDMAC	IDMAC_NFACK_41	-
IPU_INT_STAT_4[10]	IDMAC	IDMAC_NFACK_42	-
IPU_INT_STAT_4[11]	IDMAC	IDMAC_NFACK_43	-
IPU_INT_STAT_4[12]	IDMAC	IDMAC_NFACK_44	-
IPU_INT_STAT_4[13]	IDMAC	IDMAC_NFACK_45	-
IPU_INT_STAT_4[14]	IDMAC	IDMAC_NFACK_46	-
IPU_INT_STAT_4[15]	IDMAC	IDMAC_NFACK_47	-
IPU_INT_STAT_4[16]	IDMAC	IDMAC_NFACK_48	-
IPU_INT_STAT_4[17]	IDMAC	IDMAC_NFACK_49	-
IPU_INT_STAT_4[18]	IDMAC	IDMAC_NFACK_50	-
IPU_INT_STAT_4[19]	IDMAC	IDMAC_NFACK_51	-
IPU_INT_STAT_4[20]	IDMAC	IDMAC_NFACK_52	-
IPU_INT_STAT_7[23]	IDMAC	IDMAC_EOS_23	-
IPU_INT_STAT_7[24]	IDMAC	IDMAC_EOS_24	-
IPU_INT_STAT_7[27]	IDMAC	IDMAC_EOS_27	-
IPU_INT_STAT_7[28]	IDMAC	IDMAC_EOS_28	-
IPU_INT_STAT_7[29]	IDMAC	IDMAC_EOS_29	-
IPU_INT_STAT_7[31]	IDMAC	IDMAC_EOS_31	-
IPU_INT_STAT_8[2]	IDMAC	IDMAC_EOS_33	-
IPU_INT_STAT_8[9]	IDMAC	IDMAC_EOS_41	-
IPU_INT_STAT_8[10]	IDMAC	IDMAC_EOS_42	-
IPU_INT_STAT_8[11]	IDMAC	IDMAC_EOS_43	-
IPU_INT_STAT_8[12]	IDMAC	IDMAC_EOS_44	-
IPU_INT_STAT_8[19]	IDMAC	IDMAC_EOS_51	-
IPU_INT_STAT_8[20]	IDMAC	IDMAC_EOS_52	-
IPU_INT_STAT_11[0]	IDMAC	IDMAC_EOBND_0	-
IPU_INT_STAT_11[1]	IDMAC	IDMAC_EOBND_1	-
IPU_INT_STAT_11[2]	IDMAC	IDMAC_EOBND_2	-
IPU_INT_STAT_11[3]	IDMAC	IDMAC_EOBND_3	-
IPU_INT_STAT_11[5]	IDMAC	IDMAC_EOBND_5	-
IPU_INT_STAT_11[11]	IDMAC	IDMAC_EOBND_11	-
IPU_INT_STAT_11[12]	IDMAC	IDMAC_EOBND_12	-

Table continues on the next page...

Table 45-33. Functional Interrupts Summary (continued)

Location	Sub-blocks	Interrupt status bit name	Description
IPU_INT_STAT_11[20]	IDMAC	IDMAC_EOBND_20	-
IPU_INT_STAT_11[21]	IDMAC	IDMAC_EOBND_21	-
IPU_INT_STAT_11[22]	IDMAC	IDMAC_EOBND_22	-
IPU_INT_STAT_12[13]	IDMAC	IDMAC_EOBND_45	-
IPU_INT_STAT_12[14]	IDMAC	IDMAC_EOBND_46	-
IPU_INT_STAT_12[15]	IDMAC	IDMAC_EOBND_47	-
IPU_INT_STAT_12[16]	IDMAC	IDMAC_EOBND_48	-
IPU_INT_STAT_12[17]	IDMAC	IDMAC_EOBND_49	-
IPU_INT_STAT_12[18]	IDMAC	IDMAC_EOBND_50	-
IPU_INT_STAT_13[0]	IDMAC	IDMAC_TH_0	-
IPU_INT_STAT_13[1]	IDMAC	IDMAC_TH_1	-
IPU_INT_STAT_13[2]	IDMAC	IDMAC_TH_2	-
IPU_INT_STAT_13[3]	IDMAC	IDMAC_TH_3	-
IPU_INT_STAT_13[5]	IDMAC	IDMAC_TH_5	-
IPU_INT_STAT_13[8]	IDMAC	IDMAC_TH_8	-
IPU_INT_STAT_13[9]	IDMAC	IDMAC_TH_9	-
IPU_INT_STAT_13[10]	IDMAC	IDMAC_TH_10	-
IPU_INT_STAT_13[11]	IDMAC	IDMAC_TH_11	-
IPU_INT_STAT_13[12]	IDMAC	IDMAC_TH_12	-
IPU_INT_STAT_13[13]	IDMAC	IDMAC_TH_13	-
IPU_INT_STAT_13[14]	IDMAC	IDMAC_TH_14	-
IPU_INT_STAT_13[15]	IDMAC	IDMAC_TH_15	-
IPU_INT_STAT_13[17]	IDMAC	IDMAC_TH_17	-
IPU_INT_STAT_13[18]	IDMAC	IDMAC_TH_18	-
IPU_INT_STAT_13[20]	IDMAC	IDMAC_TH_20	-
IPU_INT_STAT_13[21]	IDMAC	IDMAC_TH_21	-
IPU_INT_STAT_13[22]	IDMAC	IDMAC_TH_22	-
IPU_INT_STAT_13[23]	IDMAC	IDMAC_TH_23	-
IPU_INT_STAT_13[24]	IDMAC	IDMAC_TH_24	-
IPU_INT_STAT_13[27]	IDMAC	IDMAC_TH_27	-
IPU_INT_STAT_13[28]	IDMAC	IDMAC_TH_28	-
IPU_INT_STAT_13[29]	IDMAC	IDMAC_TH_29	-
IPU_INT_STAT_13[31]	IDMAC	IDMAC_TH_31	-
IPU_INT_STAT_14[1]	IDMAC	IDMAC_TH_33	-
IPU_INT_STAT_14[8]	IDMAC	IDMAC_TH_40	-

Table continues on the next page...

Table 45-33. Functional Interrupts Summary (continued)

Location	Sub-blocks	Interrupt status bit name	Description
IPU_INT_STAT_14[9]	IDMAC	IDMAC_TH_41	-
IPU_INT_STAT_14[10]	IDMAC	IDMAC_TH_42	-
IPU_INT_STAT_14[11]	IDMAC	IDMAC_TH_43	-
IPU_INT_STAT_14[12]	IDMAC	IDMAC_TH_44	-
IPU_INT_STAT_14[13]	IDMAC	IDMAC_TH_45	-
IPU_INT_STAT_14[14]	IDMAC	IDMAC_TH_46	-
IPU_INT_STAT_14[15]	IDMAC	IDMAC_TH_47	-
IPU_INT_STAT_14[16]	IDMAC	IDMAC_TH_48	-
IPU_INT_STAT_14[17]	IDMAC	IDMAC_TH_49	-
IPU_INT_STAT_14[18]	IDMAC	IDMAC_TH_50	-
IPU_INT_STAT_14[19]	IDMAC	IDMAC_TH_51	-
IPU_INT_STAT_14[20]	IDMAC	IDMAC_TH_52	-
IPU_INT_STAT_15[0]	CM	IPU_SNOOPING1_INT	-
IPU_INT_STAT_15[1]	CM	IPU_SNOOPING2_INT	-
IPU_INT_STAT_15[2]	DP	DP_SF_START	-
IPU_INT_STAT_15[3]	DP	DP_SF_END	-
IPU_INT_STAT_15[4]	DP	DP_ASF_START	-
IPU_INT_STAT_15[5]	DP	DP_ASF_END	-
IPU_INT_STAT_15[6]	DP	DP_SF_BRAKE	-
IPU_INT_STAT_15[7]	DP	DP_ASF_BRAKE	-
IPU_INT_STAT_15[8]	DC	DC_FC_0	-
IPU_INT_STAT_15[9]	DC	DC_FC_1	-
IPU_INT_STAT_15[10]	DC	DC_FC_2	-
IPU_INT_STAT_15[11]	DC	DC_FC_3	-
IPU_INT_STAT_15[12]	DC	DC_FC_4	-
IPU_INT_STAT_15[13]	DC	DC_FC_6	-
IPU_INT_STAT_15[14]	DC	DI_VSYNC_PRE_0	-
IPU_INT_STAT_15[15]	DC	DI_VSYNC_PRE_1	-
IPU_INT_STAT_15[16]	DC	DC_DP_START	-
IPU_INT_STAT_15[17]	DC	DC_ASYNC_STOP	-
IPU_INT_STAT_15[18]	DI0	DI0_DISP_CLK_EN_PRE	-
IPU_INT_STAT_15[19]	DI0	DI0_CNT_EN_PRE_1	-
IPU_INT_STAT_15[20]	DI0	DI0_CNT_EN_PRE_2	-
IPU_INT_STAT_15[21]	DI0	DI0_CNT_EN_PRE_3	-
IPU_INT_STAT_15[22]	DI0	DI0_CNT_EN_PRE_4	-

Table continues on the next page...

Table 45-33. Functional Interrupts Summary (continued)

Location	Sub-blocks	Interrupt status bit name	Description
IPU_INT_STAT_15[23]	DI0	DI0_CNT_EN_PRE_5	-
IPU_INT_STAT_15[24]	DI0	DI0_CNT_EN_PRE_6	-
IPU_INT_STAT_15[25]	DI0	DI0_CNT_EN_PRE_7	-
IPU_INT_STAT_15[26]	DI0	DI0_CNT_EN_PRE_8	-
IPU_INT_STAT_15[27]	DI0	DI0_CNT_EN_PRE_9	-
IPU_INT_STAT_15[28]	DI0	DI0_CNT_EN_PRE_10	-
IPU_INT_STAT_15[29]	DI1	DI1_DISP_CLK_EN_PRE	-
IPU_INT_STAT_15[30]	DI1	DI1_CNT_EN_PRE_3	-
IPU_INT_STAT_15[31]	DI1	DI1_CNT_EN_PRE_8	-

The table below describes the error interrupts. The panic column indicates if this signal is part of the logic generating the ipu_panic signal. The ipu_panic signal can be used for indicating about errors that are result of data rate problems. Such problems may be a result of the IPU running in slower clock then required by the use case. This signal can be used in order to indicate the system that the IPU can't handle the desired data rate. In that case the system may need to increase the clock to the IPU or simplify the use case.

Table 45-34. Error Interrupts Summary (continued)

Location	Sub-blocks	Interrupt Status name	panic	Description
IPU_INT_STAT_5[0]	IDMAC	IDMAC_NFB4EOF_ERR_0	YES	-
IPU_INT_STAT_5[1]	IDMAC	IDMAC_NFB4EOF_ERR_1	YES	-
IPU_INT_STAT_5[2]	IDMAC	IDMAC_NFB4EOF_ERR_2	YES	-
IPU_INT_STAT_5[3]	IDMAC	IDMAC_NFB4EOF_ERR_3	YES	-
IPU_INT_STAT_5[5]	IDMAC	IDMAC_NFB4EOF_ERR_5	YES	-
IPU_INT_STAT_5[8]	IDMAC	IDMAC_NFB4EOF_ERR_8	YES	-
IPU_INT_STAT_5[9]	IDMAC	IDMAC_NFB4EOF_ERR_9	YES	-
IPU_INT_STAT_5[10]	IDMAC	IDMAC_NFB4EOF_ERR_10	YES	-
IPU_INT_STAT_5[11]	IDMAC	IDMAC_NFB4EOF_ERR_11	YES	-
IPU_INT_STAT_5[12]	IDMAC	IDMAC_NFB4EOF_ERR_12	YES	-
IPU_INT_STAT_5[13]	IDMAC	IDMAC_NFB4EOF_ERR_13	YES	-
IPU_INT_STAT_5[14]	IDMAC	IDMAC_NFB4EOF_ERR_14	YES	-
IPU_INT_STAT_5[15]	IDMAC	IDMAC_NFB4EOF_ERR_15	YES	-
IPU_INT_STAT_5[17]	IDMAC	IDMAC_NFB4EOF_ERR_17	YES	-
IPU_INT_STAT_5[18]	IDMAC	IDMAC_NFB4EOF_ERR_18	YES	-
IPU_INT_STAT_5[20]	IDMAC	IDMAC_NFB4EOF_ERR_20	YES	-

Table continues on the next page...

Table 45-34. Error Interrupts Summary (continued) (continued)

Location	Sub-blocks	Interrupt Status name	panic	Description
IPU_INT_STAT_5[21]	IDMAC	IDMAC_NFB4EOF_ERR_21	YES	-
IPU_INT_STAT_5[22]	IDMAC	IDMAC_NFB4EOF_ERR_22	YES	-
IPU_INT_STAT_5[23]	IDMAC	IDMAC_NFB4EOF_ERR_23	YES	-
IPU_INT_STAT_5[24]	IDMAC	IDMAC_NFB4EOF_ERR_24	YES	-
IPU_INT_STAT_5[27]	IDMAC	IDMAC_NFB4EOF_ERR_27	YES	-
IPU_INT_STAT_5[28]	IDMAC	IDMAC_NFB4EOF_ERR_28	YES	-
IPU_INT_STAT_5[29]	IDMAC	IDMAC_NFB4EOF_ERR_29	YES	-
IPU_INT_STAT_5[31]	IDMAC	IDMAC_NFB4EOF_ERR_31	YES	-
IPU_INT_STAT_6[1]	IDMAC	IDMAC_NFB4EOF_ERR_33	YES	-
IPU_INT_STAT_6[8]	IDMAC	IDMAC_NFB4EOF_ERR_40	YES	-
IPU_INT_STAT_6[9]	IDMAC	IDMAC_NFB4EOF_ERR_41	YES	-
IPU_INT_STAT_6[10]	IDMAC	IDMAC_NFB4EOF_ERR_42	YES	-
IPU_INT_STAT_6[11]	IDMAC	IDMAC_NFB4EOF_ERR_43	YES	-
IPU_INT_STAT_6[12]	IDMAC	IDMAC_NFB4EOF_ERR_44	YES	-
IPU_INT_STAT_6[13]	IDMAC	IDMAC_NFB4EOF_ERR_45	YES	-
IPU_INT_STAT_6[14]	IDMAC	IDMAC_NFB4EOF_ERR_46	YES	-
IPU_INT_STAT_6[15]	IDMAC	IDMAC_NFB4EOF_ERR_47	YES	-
IPU_INT_STAT_6[16]	IDMAC	IDMAC_NFB4EOF_ERR_48	YES	-
IPU_INT_STAT_6[17]	IDMAC	IDMAC_NFB4EOF_ERR_49	YES	-
IPU_INT_STAT_6[18]	IDMAC	IDMAC_NFB4EOF_ERR_50	YES	-
IPU_INT_STAT_6[19]	IDMAC	IDMAC_NFB4EOF_ERR_51	YES	-
IPU_INT_STAT_6[20]	IDMAC	IDMAC_NFB4EOF_ERR_52	YES	-
IPU_INT_STAT_9[0]	IC	VDI_FIFO1_OVF	YES	-
IPU_INT_STAT_9[26]	IC	IC_BAYER_BUF_OVF	YES	-
IPU_INT_STAT_9[27]	IC	IC_ENC_BUF_OVF	YES	-
IPU_INT_STAT_9[28]	IC	IC_VF_BUF_OVF	YES	-
IPU_INT_STAT_9[30]	CSI0	CSI0_PUPE	YES	-
IPU_INT_STAT_9[31]	CSI0	CSI1_PUPE	YES	-
IPU_INT_STAT_10[0]	SMFC	SMFC0_FRM_LOST	YES	-
IPU_INT_STAT_10[1]	SMFC	SMFC1_FRM_LOST	YES	-
IPU_INT_STAT_10[2]	SMFC	SMFC2_FRM_LOST	YES	-
IPU_INT_STAT_10[3]	SMFC	SMFC3_FRM_LOST	YES	-
IPU_INT_STAT_10[16]	DC	DC_TEARING_ERR_1	YES	-
IPU_INT_STAT_10[17]	DC	DC_TEARING_ERR_2	YES	-
IPU_INT_STAT_10[18]	DC	DC_TEARING_ERR_6	YES	-

Table continues on the next page...

Table 45-34. Error Interrupts Summary (continued) (continued)

Location	Sub-blocks	Interrupt Status name	panic	Description
IPU_INT_STAT_10[19]	DI0	DI0_SYNC_DISP_ERR	YES	-
IPU_INT_STAT_10[20]	DI1	DI1_SYNC_DISP_ERR	YES	-
IPU_INT_STAT_10[21]	DI0	DI0_TIME_OUT_ERR	YES	-
IPU_INT_STAT_10[22]	DI1	DI1_TIME_OUT_ERR	YES	-
IPU_INT_STAT_10[24]	IC	IC_VF_FRM_LOST_ERR	YES	-
IPU_INT_STAT_10[25]	IC	IC_ENC_FRM_LOST_ERR	YES	-
IPU_INT_STAT_10[26]	IC	IC_BAYER_FRM_LOST_ERR	YES	-
IPU_INT_STAT_10[28]	CM	NON_PRIVILEGED_ACC_ERR	NO	-
IPU_INT_STAT_10[29]	IDMAC	AXIW_ERR	NO	-
IPU_INT_STAT_10[30]	IDMAC	AXIR_ERR	NO	-

45.4.12.4 SDMA event generator

The IPU provides an SDMA event signal that can be used as trigger to the SoC's SDMA. IPU routes an internal event to the SDMA event signal. The internal event causing the assertion of the SDMA signal is enabled by setting the corresponding bit on the SDMA_EVENT_# registers.

The user is allowed to enable multiple events. When one of these events occurs the ipu_sdma_event signal will be asserted. Similar to interrupts, the ipu_sdma_event signal is cleared by writing one to the corresponding bit in the INT_STAT_# registers.

It is not recommended to use the same internal event for a simultaneous generation of an interrupt signal and an SDMA event. This will require special software care when clearing the corresponding bit in the INT_STAT_# registers.

45.4.12.5 General Configuration Registers

The GCR contains a set of control/status/data registers. It provides IPU interface to the AHB slave bus.

The HCLK rate is equal to the HSP_CLK rate. The detail description of the registers is found in the Programmable Registers section.

45.4.12.6 Shadow Registers Module (SRM)

IPU supports frame by frame task switching. This means that a sub-block can handle a frame with one configuration and handle the following frame with different configuration. Changing the configuration is done by updating the sub-block's parameters.

In order to allow automatic flow without a need of the SW to update all the parameters at the frame boundaries. A Register of a sub-block that has the shadowing capabilities has a shadow register file that resides in the Shadow Register Memory.

The sub-blocks supporting this function may use it in one of the following ways:

45.4.12.6.1 Switching between 2 flows

Upon request from the FSU the SRM switches between the registers and the content in the Shadow Register Memory. When a sub-block uses one of the configuration SW, it is allowed to update the parameters in the memory. This is normally used when 2 flows are supported via one module.

- The current flow's configuration is stored in the module's registers
- The alternate flow configuration is stored in the SRM.
- When switching between flows the current flow's configuration is stored in the SRM. The alternate flow's configuration is written to the module's registers. When the alternate flow ends the configurations are swapped again.

This process is fully controlled by the FSU

45.4.12.6.2 Updating parameters between frames

This mode is used when the user needs to update the parameters of the current task being processed by the module. The updates can't affect the frame that is currently being processed. The update is effective only on the next frame. The SRM performs the parameters update only when there's a specific request by the user.

- The current frame's configuration is stored in the sub-block's registers
- The next frame's configuration is stored in the SRM.
- On frame's boundary the SRM reads the new configuration from the SRM and writes it to the sub-block's registers. The old configuration is lost.

45.4.12.6.3 Updating the memory

In order to avoid data coherency problems, the user should set a flag indicating that he is currently updating the memory region of a specific sub-block. When the flag is set the SRM will not attempt to replace the parameters relevant to the specific sub-block currently being updated by the user.

The user should clear this flag when he completes the update, and the parameters are ready to be used.

The flag is the corresponding **SRM_MODE** field for each sub-block. This field controls the SRM logic that handles the sub-blocks registers

- 00 - ARM platform is allowed to access the sub-block's region in the RAM; The automatic swapping mechanism is disabled.
- 01 - The SRM logic is controlled by the FSU. The update will be done of the next frame.
- 10 - The SRM logic is controlled by the FSU. Registers are swapped continuously frame by frame
- 11 - Update now. The SRM is controlled by the ARM platform. The Register will be update now

Each sub-block uses the SRM mechanism according to the sub-block's behavior, the table below summarizes the SRM support for each sub-block

Table 45-35. SRM support per Sub-block

Sub-block	Switching between 2 flows support	Updating parameters between frames	Comment
CSI1, CSI0	NO	YES	SRM_MODE can be 00 or 01. The update will happen only once. When set to 01: after the update the state machine is automatically moved to 00 mode. It is recommended to make sure that the first frame has started by polling the corresponding NFACK bit before setting the SRM_MODE
DI1, DI0	NO	YES	DI0, DI1 parameters are needed when clock change is performed (Clock Change procedure). SRM_MODE can be 00 or 01. The update will happen only once. When set to 01: after the update the state machine is automatically moved to 00 mode
DC	YES	NO	SRM_MODE can be 00 or 10.
DP	YES	NO	SRM_MODE can be 00, 10 or 11. When set to 11: after the update the state machine is automatically moved to 10 mode 10 is not supported for SYNC flows

In order to update parameters the user should monitor the SRM_BUSY bit of the corresponding sub-block. When the SRM is not busy the user should set the **SRM_MODE to 00. The user will now update the register file in the memory. When done the user should switch the SRM_MODE field to the desired mode.**

45.4.12.6.4 SRM priority

The SRM updates the registers according to a pre defined priority. The priority is set according to the SRM_PRI bits of each sub-block. The user must set a unique value for each sub-block.

45.4.12.6.5 SRM entries mapping

The table below maps any IPU register to an address in the SRM. The registers marked as NONE do not have an SRM entry

PG column indicates if this register is saved during power gating mode

LPSR column indicates if this register is swapped during low power screen refresh mode (LPSR)

Table 45-36. IPU SRM entries mapping

Register's name	Register's address	SRM entry	PG	LPSR
IPU_CONF	0x0000_0000	NONE	YES	NO
IPU_SISG_CTRL0	0x00000004	NONE	YES	NO
IPU_SISG_CTRL1	0x00000008	NONE	YES	NO
IPU_SISG_SET_1	0x0000000C	NONE	YES	NO
IPU_SISG_SET_2	0x00000010	NONE	YES	NO
IPU_SISG_SET_3	0x00000014	NONE	YES	NO
IPU_SISG_SET_4	0x00000018	NONE	YES	NO
IPU_SISG_SET_5	0x0000001C	NONE	YES	NO
IPU_SISG_SET_6	0x00000020	NONE	YES	NO
IPU_SISG_CLR_1	0x00000024	NONE	YES	NO
IPU_SISG_CLR_2	0x00000028	NONE	YES	NO
IPU_SISG_CLR_3	0x0000002C	NONE	YES	NO
IPU_SISG_CLR_4	0x00000030	NONE	YES	NO
IPU_SISG_CLR_5	0x00000034	NONE	YES	NO
IPU_SISG_CLR_6	0x00000038	NONE	YES	NO
IPU_INT_CTRL_1	0x0000003C	NONE	YES	NO
IPU_INT_CTRL_2	0x00000040	NONE	YES	NO

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_INT_CTRL_3	0x00000044	NONE	YES	NO
IPU_INT_CTRL_4	0x00000048	NONE	YES	NO
IPU_INT_CTRL_5	0x0000004C	NONE	YES	NO
IPU_INT_CTRL_6	0x00000050	NONE	YES	NO
IPU_INT_CTRL_7	0x00000054	NONE	YES	NO
IPU_INT_CTRL_8	0x00000058	NONE	YES	NO
IPU_INT_CTRL_9	0x0000005C	NONE	YES	NO
IPU_INT_CTRL_10	0x00000060	NONE	YES	NO
IPU_INT_CTRL_11	0x00000064	NONE	YES	NO
IPU_INT_CTRL_12	0x00000068	NONE	YES	NO
IPU_INT_CTRL_13	0x0000006C	NONE	YES	NO
IPU_INT_CTRL_14	0x00000070	NONE	YES	NO
IPU_INT_CTRL_15	0x00000074	NONE	YES	NO
IPU_SDMA_EVENT_1	0x00000078	NONE	YES	NO
IPU_SDMA_EVENT_2	0x0000007C	NONE	YES	NO
IPU_SDMA_EVENT_3	0x00000080	NONE	YES	NO
IPU_SDMA_EVENT_4	0x00000084	NONE	YES	NO
IPU_SDMA_EVENT_7	0x00000088	NONE	YES	NO
IPU_SDMA_EVENT_8	0x0000008C	NONE	YES	NO
IPU_SDMA_EVENT_11	0x00000090	NONE	YES	NO
IPU_SDMA_EVENT_12	0x00000094	NONE	YES	NO
IPU_SDMA_EVENT_13	0x00000098	NONE	YES	NO
IPU_SDMA_EVENT_14	0x0000009C	NONE	YES	NO
IPU_SRM_PRI1	0x000000A0	NONE	YES	NO
IPU_SRM_PRI2	0x000000A4	NONE	YES	NO
IPU_FS_PROC_FLOW1	0x000000A8	NONE	YES	NO
IPU_FS_PROC_FLOW2	0x000000AC	NONE	YES	NO
IPU_FS_PROC_FLOW3	0x000000B0	NONE	YES	NO
IPU_FS_DISP_FLOW1	0x000000B4	NONE	YES	NO
IPU_FS_DISP_FLOW2	0x000000B8	NONE	YES	NO
IPU_SKIP	0x000000BC	NONE	YES	NO
IPU_DISP_ALT_CONF	0x000000C0	NONE	YES	NO
IPU_DISP_GEN	0x000000C4	NONE	YES	NO
IPU_DISP_ALT1	0x000000C8	NONE	YES	NO
IPU_DISP_ALT2	0x000000CC	NONE	YES	NO

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DISP_ALT3	0x000000D0	NONE	YES	NO
IPU_DISP_ALT4	0x000000D4	NONE	YES	NO
IPU_SNOOP	0x000000D8	NONE	YES	NO
IPU_MEM_RST	0x000000DC	NONE	YES	NO
IPU_PM	0x000000E0	NONE	YES	NO
IPU_GPR	0x000000E4	NONE	YES	NO
IPU_CH_DB_MODE_SEL_0	0x00000150	NONE	YES	NO
IPU_CH_DB_MODE_SEL_1	0x00000154	NONE	YES	NO
IPU_ALT_CH_DB_MODE_SEL_0	0x00000168	NONE	YES	NO
IPU_ALT_CH_DB_MODE_SEL_1	0x0000016C	NONE	YES	NO
IPU_CH_TRB_MODE_SEL_0	0x00000178	NONE	YES	NO
IPU_CH_TRB_MODE_SEL_1	0x0000017C	NONE	YES	NO
IPU_INT_STAT_1	0x00000200	NONE	NO	NO
IPU_INT_STAT_2	0x00000204	NONE	NO	NO
IPU_INT_STAT_3	0x00000208	NONE	NO	NO
IPU_INT_STAT_4	0x0000020C	NONE	NO	NO
IPU_INT_STAT_5	0x00000210	NONE	NO	NO
IPU_INT_STAT_6	0x00000214	NONE	NO	NO
IPU_INT_STAT_7	0x00000218	NONE	NO	NO
IPU_INT_STAT_8	0x0000021C	NONE	NO	NO
IPU_INT_STAT_9	0x00000220	NONE	NO	NO
IPU_INT_STAT_10	0x00000224	NONE	NO	NO
IPU_INT_STAT_11	0x00000228	NONE	NO	NO
IPU_INT_STAT_12	0x0000022C	NONE	NO	NO
IPU_INT_STAT_13	0x00000230	NONE	NO	NO
IPU_INT_STAT_14	0x00000234	NONE	NO	NO
IPU_INT_STAT_15	0x00000238	NONE	NO	NO
IPU_CUR_BUF_0	0x0000023C	NONE	NO	NO
IPU_CUR_BUF_1	0x00000240	NONE	NO	NO
IPU_ALT_CUR_BUF_0	0x00000244	NONE	NO	NO
IPU_ALT_CUR_BUF_1	0x00000248	NONE	NO	NO
IPU_SRM_STAT	0x0000024C	NONE	NO	NO
IPU_PROC_TASKS_STAT	0x00000250	NONE	NO	NO
IPU_DISP_TASKS_STAT	0x00000254	NONE	NO	NO
IPU_TRIPLE_CUR_BUF_0	0x00000258	NONE	NO	NO

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_TRIPLE_CUR_BUF_1	0x0000025C	NONE	NO	NO
IPU_TRIPLE_CUR_BUF_2	0x00000260	NONE	NO	NO
IPU_TRIPLE_CUR_BUF_3	0x00000264	NONE	NO	NO
IPU_CH_BUF0_RDY0	0x00000268	NONE	NO	NO
IPU_CH_BUF0_RDY1	0x0000026C	NONE	NO	NO
IPU_CH_BUF1_RDY0	0x00000270	NONE	NO	NO
IPU_CH_BUF1_RDY1	0x00000274	NONE	NO	NO
IPU_ALT_CH_BUF0_RDY0	0x00000278	NONE	NO	NO
IPU_ALT_CH_BUF0_RDY1	0x0000027C	NONE	NO	NO
IPU_ALT_CH_BUF1_RDY0	0x00000280	NONE	NO	NO
IPU_ALT_CH_BUF1_RDY1	0x00000284	NONE	NO	NO
IPU_CH_BUF2_RDY0	0x00000288	NONE	NO	NO
IPU_CH_BUF2_RDY1	0x0000028C	NONE	NO	NO
IPU_IDMAC_CONF	0x00008000	NONE	YES	NO
IPU_IDMAC_CH_EN_1	0x00008004	NONE	YES	NO
IPU_IDMAC_CH_EN_2	0x00008008	NONE	YES	NO
IPU_IDMAC_SEP_ALPHA	0x0000800C	NONE	YES	NO
IPU_IDMAC_ALT_SEP_ALPHA	0x00008010	NONE	YES	NO
IPU_IDMAC_CH_PRI_1	0x00008014	NONE	YES	NO
IPU_IDMAC_CH_PRI_2	0x00008018	NONE	YES	NO
IIPU_DMAC_WM_EN_1	0x0000801C	NONE	YES	NO
IPU_IDMAC_WM_EN_2	0x00008020	NONE	YES	NO
IPU_IDMAC_LOCK_EN_1	0x00008024	NONE	YES	NO
IPU_IDMAC_LOCK_EN_2	0x00008028	NONE	YES	NO
IPU_IDMAC_SUB_ADDR_0	0x0000802C	NONE	YES	NO
IPU_IDMAC_SUB_ADDR_1	0x00008030	NONE	YES	NO
IPU_IDMAC_SUB_ADDR_2	0x00008034	NONE	YES	NO
IPU_IDMAC_SUB_ADDR_3	0x00008038	NONE	YES	NO
IPU_IDMAC_SUB_ADDR_4	0x0000803C	NONE	YES	NO
IPU_IDMAC_BNDM_EN_1	0x00008040	NONE	YES	NO
IPU_IDMAC_BNDM_EN_2	0x00008044	NONE	YES	NO
IPU_IDMAC_SC_CORD	0x00008048	NONE	YES	NO
IPU_IDMAC_SC_CORD1	0x0000804C	NONE	YES	NO
IPU_IDMAC_CH_BUSY_1	0x00008100	NONE	NO	NO
IPU_IDMAC_CH_BUSY_2	0x00008104	NONE	NO	NO

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DP_COM_CONF_SYNC	0x1F040000	0x1F040000	YES	YES
IPU_DP_GRAPH_WIND_CTRL_SYNC	0x1F040004	0x1F040004	YES	YES
IPU_DP_FG_POS_SYNC	0x1F040008	0x1F040008	YES	YES
IPU_DP_CUR_POS_SYNC	0x1F04000C	0x1F04000C	YES	YES
IPU_DP_CUR_MAP_SYNC	0x1F040010	0x1F040010	YES	YES
IPU_DP_GAMMA_C_SYNC_0	0x1F040014	0x1F040014	YES	YES
IPU_DP_GAMMA_C_SYNC_1	0x1F040018	0x1F040018	YES	YES
IPU_DP_GAMMA_C_SYNC_2	0x1F04001C	0x1F04001C	YES	YES
IPU_DP_GAMMA_C_SYNC_3	0x1F040020	0x1F040020	YES	YES
IPU_DP_GAMMA_C_SYNC_4	0x1F040024	0x1F040024	YES	YES
IPU_DP_GAMMA_C_SYNC_5	0x1F040028	0x1F040028	YES	YES
IPU_DP_GAMMA_C_SYNC_6	0x1F04002C	0x1F04002C	YES	YES
IPU_DP_GAMMA_C_SYNC_7	0x1F040030	0x1F040030	YES	YES
IPU_DP_GAMMA_S_SYNC_0	0x1F040034	0x1F040034	YES	YES
IPU_DP_GAMMA_S_SYNC_1	0x1F040038	0x1F040038	YES	YES
IPU_DP_GAMMA_S_SYNC_2	0x1F04003C	0x1F04003C	YES	YES
IPU_DP_GAMMA_S_SYNC_3	0x1F040040	0x1F040040	YES	YES
IPU_DP_CSCA_SYNC_0	0x1F040044	0x1F040044	YES	YES
IPU_DP_CSCA_SYNC_1	0x1F040048	0x1F040048	YES	YES
IPU_DP_CSCA_SYNC_2	0x1F04004C	0x1F04004C	YES	YES
IPU_DP_CSCA_SYNC_3	0x1F040050	0x1F040050	YES	YES
IPU_DP_CSC_SYNC_0	0x1F040054	0x1F040054	YES	YES
IPU_DP_CSC_SYNC_1	0x1F040058	0x1F040058	YES	YES
IPU_DP_CUR_POS_ALT	0x1F04005C	0x1F04005C	YES	YES
IPU_DP_COM_CONF_ASYNC0	0x1F040060	0x1F040060	YES	YES
IPU_DP_GRAPH_WIND_CTRL_ASYNC0	0x1F040064	0x1F040064	YES	YES
IPU_DP_FG_POS_ASYNC0	0x1F040068	0x1F040068	YES	YES
IPU_DP_CUR_POS_ASYNC0	0x1F04006C	0x1F04006C	YES	YES
IPU_DP_CUR_MAP_ASYNC0	0x1F040070	0x1F040070	YES	YES
IPU_DP_GAMMA_C_ASYNC0_0	0x1F040074	0x1F040074	YES	YES
IPU_DP_GAMMA_C_ASYNC0_1	0x1F040078	0x1F040078	YES	YES
IPU_DP_GAMMA_C_ASYNC0_2	0x1F04007C	0x1F04007C	YES	YES
IPU_DP_GAMMA_C_ASYNC0_3	0x1F040080	0x1F040080	YES	YES
IPU_DP_GAMMA_C_ASYNC0_4	0x1F040084	0x1F040084	YES	YES
IPU_DP_GAMMA_C_ASYNC0_5	0x1F040088	0x1F040088	YES	YES

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DP_GAMMA_C_ASYNC0_6	0x1F04008C	0x1F04008C	YES	YES
IPU_DP_GAMMA_C_ASYNC0_7	0x1F040090	0x1F040090	YES	YES
IPU_DP_GAMMA_S_ASYNC0_0	0x1F040094	0x1F040094	YES	YES
IPU_DP_GAMMA_S_ASYNC0_1	0x1F040098	0x1F040098	YES	YES
IPU_DP_GAMMA_S_ASYNC0_2	0x1F04009C	0x1F04009C	YES	YES
IPU_DP_GAMMA_S_ASYNC0_3	0x1F0400A0	0x1F0400A0	YES	YES
IPU_DP_CSCA_ASYNC0_0	0x1F0400A4	0x1F0400A4	YES	YES
IPU_DP_CSCA_ASYNC0_1	0x1F0400A8	0x1F0400A8	YES	YES
IPU_DP_CSCA_ASYNC0_2	0x1F0400AC	0x1F0400AC	YES	YES
IPU_DP_CSCA_ASYNC0_3	0x1F0400B0	0x1F0400B0	YES	YES
IPU_DP_CSC_ASYNC0_0	0x1F0400B4	0x1F0400B4	YES	YES
IPU_DP_CSC_ASYNC0_1	0x1F0400B8	0x1F0400B8	YES	YES
IPU_DP_COM_CONF_ASYNC1	0x1F0400BC	0x1F0400BC	YES	YES
IPU_DP_GRAPH_WIND_CTRL_ASYNC1	0x1F0400C0	0x1F0400C0	YES	YES
IPU_DP_FG_POS_ASYNC1	0x1F0400C4	0x1F0400C4	YES	YES
IPU_DP_CUR_POS_ASYNC1	0x1F0400C8	0x1F0400C8	YES	YES
IPU_DP_CUR_MAP_ASYNC1	0x1F0400CC	0x1F0400CC	YES	YES
IPU_DP_GAMMA_C_ASYNC1_0	0x1F0400D0	0x1F0400D0	YES	YES
IPU_DP_GAMMA_C_ASYNC1_1	0x1F0400D4	0x1F0400D4	YES	YES
IPU_DP_GAMMA_C_ASYNC1_2	0x1F0400D8	0x1F0400D8	YES	YES
IPU_DP_GAMMA_C_ASYNC1_3	0x1F0400DC	0x1F0400DC	YES	YES
IPU_DP_GAMMA_C_ASYNC1_4	0x1F0400E0	0x1F0400E0	YES	YES
IPU_DP_GAMMA_C_ASYNC1_5	0x1F0400E4	0x1F0400E4	YES	YES
IPU_DP_GAMMA_C_ASYNC1_6	0x1F0400E8	0x1F0400E8	YES	YES
IPU_DP_GAMMA_C_ASYNC1_7	0x1F0400EC	0x1F0400EC	YES	YES
IPU_DP_GAMMA_S_ASYNC1_0	0x1F0400F0	0x1F0400F0	YES	YES
IPU_DP_GAMMA_S_ASYNC1_1	0x1F0400F4	0x1F0400F4	YES	YES
IPU_DP_GAMMA_S_ASYNC1_2	0x1F0400F8	0x1F0400F8	YES	YES
IPU_DP_GAMMA_S_ASYNC1_3	0x1F0400FC	0x1F0400FC	YES	YES
IPU_DP_CSCA_ASYNC1_0	0x1F040100	0x1F040100	YES	YES
IPU_DP_CSCA_ASYNC1_1	0x1F040104	0x1F040104	YES	YES
IPU_DP_CSCA_ASYNC1_2	0x1F040108	0x1F040108	YES	YES
IPU_DP_CSCA_ASYNC1_3	0x1F04010C	0x1F04010C	YES	YES
IPU_DP_CSC_ASYNC1_0	0x1F040110	0x1F040110	YES	YES
IPU_DP_CSC_ASYNC1_1	0x1F040114	0x1F040114	YES	YES

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DP_DEBUG_CNT	0x000180BC	NONE	NO	NO
IPU_DP_DEBUG_STAT	0x000180C0	NONE	NO	NO
IPU_IC_CONF	0x00020000	NONE	YES	NO
IPU_IC_PRP_ENC_RSC	0x00020004	NONE	YES	NO
IPU_IC_PRP_VF_RSC	0x00020008	NONE	YES	NO
IPU_IC_PP_RSC	0x0002000C	NONE	YES	NO
IPU_IC_CMBP_1	0x00020010	NONE	YES	NO
IPU_IC_CMBP_2	0x00020014	NONE	YES	NO
IPU_IC_IDMAC_1	0x00020018	NONE	YES	NO
IPU_IC_IDMAC_2	0x0002001C	NONE	YES	NO
IPU_IC_IDMAC_3	0x00020020	NONE	YES	NO
IPU_IC_IDMAC_4	0x00020024	NONE	YES	NO
IPU_CSI0_SENS_CONF	0x00030000	NONE	YES	NO
IPU_CSI0_SENS_FRM_SIZE	0x00030004	NONE	YES	NO
IPU_CSI0_ACT_FRM_SIZE	0x00030008	NONE	YES	NO
IPU_CSI0_OUT_FRM_CTRL	0x0003000C	NONE	YES	NO
IPU_CSI0_TST_CTRL	0x00030010	NONE	YES	NO
IPU_CSI0_CCIR_CODE_1	0x00030014	NONE	YES	NO
IPU_CSI0_CCIR_CODE_2	0x00030018	NONE	YES	NO
IPU_CSI0_CCIR_CODE_3	0x0003001C	NONE	YES	NO
IPU_CSI0_DI	0x00030020	NONE	YES	NO
IPU_CSI0_SKIP	0x00030024	NONE	YES	NO
IPU_CSI0_CPD_CTRL	0x00030028	0x1F040314	YES	NO
IPU_CSI0_CPD_RC_0	0x0003002C	0x1F040318	YES	NO
IPU_CSI0_CPD_RC_1	0x00030030	0x1F04031C	YES	NO
IPU_CSI0_CPD_RC_2	0x00030034	0x1F040320	YES	NO
IPU_CSI0_CPD_RC_3	0x00030038	0x1F040324	YES	NO
IPU_CSI0_CPD_RC_4	0x0003003C	0x1F040328	YES	NO
IPU_CSI0_CPD_RC_5	0x00030040	0x1F04032C	YES	NO
IPU_CSI0_CPD_RC_6	0x00030044	0x1F040330	YES	NO
IPU_CSI0_CPD_RC_7	0x00030048	0x1F040334	YES	NO
IPU_CSI0_CPD_RS_0	0x0003004C	0x1F040338	YES	NO
IPU_CSI0_CPD_RS_1	0x00030050	0x1F04033C	YES	NO
IPU_CSI0_CPD_RS_2	0x00030054	0x1F040340	YES	NO
IPU_CSI0_CPD_RS_3	0x00030058	0x1F040344	YES	NO

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_CSI0_CPD_GRC_0	0x0003005C	0x1F040348	YES	NO
IPU_CSI0_CPD_GRC_1	0x00030060	0x1F04034C	YES	NO
IPU_CSI0_CPD_GRC_2	0x00030064	0x1F040350	YES	NO
IPU_CSI0_CPD_GRC_3	0x00030068	0x1F040354	YES	NO
IPU_CSI0_CPD_GRC_4	0x0003006C	0x1F040358	YES	NO
IPU_CSI0_CPD_GRC_5	0x00030070	0x1F04035C	YES	NO
IPU_CSI0_CPD_GRC_6	0x00030074	0x1F040360	YES	NO
IPU_CSI0_CPD_GRC_7	0x00030078	0x1F040364	YES	NO
IPU_CSI0_CPD_GRS_0	0x0003007C	0x1F040368	YES	NO
IPU_CSI0_CPD_GRS_1	0x00030080	0x1F04036C	YES	NO
IPU_CSI0_CPD_GRS_2	0x00030084	0x1F040370	YES	NO
IPU_CSI0_CPD_GRS_3	0x00030088	0x1F040374	YES	NO
IPU_CSI0_CPD_GBC_0	0x0003008C	0x1F040378	YES	NO
IPU_CSI0_CPD_GBC_1	0x00030090	0x1F04037C	YES	NO
IPU_CSI0_CPD_GBC_2	0x00030094	0x1F040380	YES	NO
IPU_CSI0_CPD_GBC_3	0x00030098	0x1F040384	YES	NO
IPU_CSI0_CPD_GBC_4	0x0003009C	0x1F040388	YES	NO
IPU_CSI0_CPD_GBC_5	0x000300A0	0x1F04038C	YES	NO
IPU_CSI0_CPD_GBC_6	0x000300A4	0x1F040390	YES	NO
IPU_CSI0_CPD_GBC_7	0x000300A8	0x1F040394	YES	NO
IPU_CSI0_CPD_GBS_0	0x000300AC	0x1F040398	YES	NO
IPU_CSI0_CPD_GBS_1	0x000300B0	0x1F04039C	YES	NO
IPU_CSI0_CPD_GBS_2	0x000300B4	0x1F0403A0	YES	NO
IPU_CSI0_CPD_GBS_3	0x000300B8	0x1F0403A4	YES	NO
IPU_CSI0_CPD_BC_0	0x000300BC	0x1F0403A8	YES	NO
IPU_CSI0_CPD_BC_1	0x000300C0	0x1F0403AC	YES	NO
IPU_CSI0_CPD_BC_2	0x000300C4	0x1F0403B0	YES	NO
IPU_CSI0_CPD_BC_3	0x000300C8	0x1F0403B4	YES	NO
IPU_IPU_CSI0_CPD_BC_4	0x000300CC	0x1F0403B8	YES	NO
IPU_CSI0_CPD_BC_5	0x000300D0	0x1F0403BC	YES	NO
IPU_CSI0_CPD_BC_6	0x000300D4	0x1F0403C0	YES	NO
IPU_CSI0_CPD_BC_7	0x000300D8	0x1F0403C4	YES	NO
IPU_CSI0_CPD_BS_0	0x000300DC	0x1F0403C8	YES	NO
IPU_CSI0_CPD_BS_1	0x000300E0	0x1F0403CC	YES	NO
IPU_CSI0_CPD_BS_2	0x000300E4	0x1F0403D0	YES	NO

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_CSI0_CPD_BS_3	0x000300E8	0x1F0403D4	YES	NO
IPU_CSI0_CPD_OFFSET1	0x000300EC	0x1F0403D8	YES	NO
IPU_CSI0_CPD_OFFSET2	0x000300F0	0x1F0403DC	YES	NO
IPU_CSI1_SENS_CONF	0x00038000	NONE	YES	NO
IPU_CSI1_SENS_FRM_SIZE	0x00038004	NONE	YES	NO
IPU_CSI1_ACT_FRM_SIZE	0x00038008	NONE	YES	NO
IPU_CSI1_OUT_FRM_CTRL	0x0003800C	NONE	YES	NO
IPU_CSI1_TST_CTRL	0x00038010	NONE	YES	NO
IPU_CSI1_CCIR_CODE_1	0x00038014	NONE	YES	NO
IPU_CSI1_CCIR_CODE_2	0x00038018	NONE	YES	NO
IPU_CSI1_CCIR_CODE_3	0x0003801C	NONE	YES	NO
IPU_CSI1_DI	0x00038020	NONE	YES	NO
IPU_CSI1_SKIP	0x00038024	NONE	YES	NO
IPU_CSI1_CPD_CTRL	0x00038028	0x1F0403E0	YES	NO
IPU_CSI1_CPD_RC_0	0x0003802C	0x1F0403E4	YES	NO
IPU_CSI1_CPD_RC_1	0x00038030	0x1F0403E8	YES	NO
IPU_CSI1_CPD_RC_2	0x00038034	0x1F0403EC	YES	NO
IPU_CSI1_CPD_RC_3	0x00038038	0x1F0403F0	YES	NO
IPU_CSI1_CPD_RC_4	0x0003803C	0x1F0403F4	YES	NO
IPU_CSI1_CPD_RC_5	0x00038040	0x1F0403F8	YES	NO
IPU_CSI1_CPD_RC_6	0x00038044	0x1F0403FC	YES	NO
IPU_CSI1_CPD_RC_7	0x00038048	0x1F040400	YES	NO
IPU_CSI1_CPD_RS_0	0x0003804C	0x1F040404	YES	NO
IPU_CSI1_CPD_RS_1	0x00038050	0x1F040408	YES	NO
IPU_CSI1_CPD_RS_2	0x00038054	0x1F04040C	YES	NO
IPU_CSI1_CPD_RS_3	0x00038058	0x1F040410	YES	NO
IPU_CSI1_CPD_GRC_0	0x0003805C	0x1F040414	YES	NO
IPU_CSI1_CPD_GRC_1	0x00038060	0x1F040418	YES	NO
IPU_CSI1_CPD_GRC_2	0x00038064	0x1F04041C	YES	NO
IPU_CSI1_CPD_GRC_3	0x00038068	0x1F040420	YES	NO
IPU_CSI1_CPD_GRC_4	0x0003806C	0x1F040424	YES	NO
IPU_CSI1_CPD_GRC_5	0x00038070	0x1F040428	YES	NO
IPU_CSI1_CPD_GRC_6	0x00038074	0x1F04042C	YES	NO
IPU_CSI1_CPD_GRC_7	0x00038078	0x1F040430	YES	NO
IPU_CSI1_CPD_GRS_0	0x0003807C	0x1F040434	YES	NO

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_CSI1_CPD_GRS_1	0x00038080	0x1F040438	YES	NO
IPU_CSI1_CPD_GRS_2	0x00038084	0x1F04043C	YES	NO
IPU_CSI1_CPD_GRS_3	0x00038088	0x1F040440	YES	NO
IPU_CSI1_CPD_GBC_0	0x0003808C	0x1F040444	YES	NO
IPU_CSI1_CPD_GBC_1	0x00038090	0x1F040448	YES	NO
IPU_CSI1_CPD_GBC_2	0x00038094	0x1F04044C	YES	NO
IPU_CSI1_CPD_GBC_3	0x00038098	0x1F040450	YES	NO
IPU_CSI1_CPD_GBC_4	0x0003809C	0x1F040454	YES	NO
IPU_CSI1_CPD_GBC_5	0x000380A0	0x1F040458	YES	NO
IPU_CSI1_CPD_GBC_6	0x000380A4	0x1F04045C	YES	NO
IPU_CSI1_CPD_GBC_7	0x000380A8	0x1F040460	YES	NO
IPU_CSI1_CPD_GBS_0	0x000380AC	0x1F040464	YES	NO
IPU_CSI1_CPD_GBS_1	0x000380B0	0x1F040468	YES	NO
IPU_CSI1_CPD_GBS_2	0x000380B4	0x1F04046C	YES	NO
IPU_CSI1_CPD_GBS_3	0x000380B8	0x1F040470	YES	NO
IPU_CSI1_CPD_BC_0	0x000380BC	0x1F040474	YES	NO
IPU_CSI1_CPD_BC_1	0x000380C0	0x1F040478	YES	NO
IPU_CSI1_CPD_BC_2	0x000380C4	0x1F04047C	YES	NO
IPU_CSI1_CPD_BC_3	0x000380C8	0x1F040480	YES	NO
IPU_CSI1_CPD_BC_4	0x000380CC	0x1F040484	YES	NO
IPU_CSI1_CPD_BC_5	0x000380D0	0x1F040488	YES	NO
IPU_CSI1_CPD_BC_6	0x000380D4	0x1F04048C	YES	NO
IPU_CSI1_CPD_BC_7	0x000380D8	0x1F040490	YES	NO
IPU_CSI1_CPD_BS_0	0x000380DC	0x1F040494	YES	NO
IPU_CSI1_CPD_BS_1	0x000380E0	0x1F040498	YES	NO
IPU_CSI1_CPD_BS_2	0x000380E4	0x1F04049C	YES	NO
IPU_CSI1_CPD_BS_3	0x000380E8	0x1F0404A0	YES	NO
IPU_CSI1_CPD_OFFSET1	0x000380EC	0x1F0404A4	YES	NO
IPU_CSI1_CPD_OFFSET2	0x000380F0	0x1F0404A8	YES	NO
IPU_DI0_GENERAL	0x00040000	0x1F0404E4	YES	YES
IPU_DI0_BS_CLKGEN0	0x00040004	0x1F0404E8	YES	YES
IPU_DI0_BS_CLKGEN1	0x00040008	0x1F0404EC	YES	YES
IPU_DI0_SW_GEN0_1	0x0004000C	0x1F0404F0	YES	YES
IPU_DI0_SW_GEN0_2	0x00040010	0x1F0404F4	YES	YES
IPU_DI0_SW_GEN0_3	0x00040014	0x1F0404F8	YES	YES

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DI0_SW_GEN0_4	0x00040018	0x1F0404FC	YES	YES
IPU_DI0_SW_GEN0_5	0x0004001C	0x1F040500	YES	YES
IPU_DI0_SW_GEN0_6	0x00040020	0x1F040504	YES	YES
IPU_DI0_SW_GEN0_7	0x00040024	0x1F040508	YES	YES
IPU_DI0_SW_GEN0_8	0x00040028	0x1F04050C	YES	YES
IPU_DI0_SW_GEN0_9	0x0004002C	0x1F040510	YES	YES
IPU_DI0_SW_GEN1_1	0x00040030	0x1F040514	YES	YES
IPU_DI0_SW_GEN1_2	0x00040034	0x1F040518	YES	YES
IPU_DI0_SW_GEN1_3	0x00040038	0x1F04051C	YES	YES
IPU_DI0_SW_GEN1_4	0x0004003C	0x1F040520	YES	YES
IPU_DI0_SW_GEN1_5	0x00040040	0x1F040524	YES	YES
IPU_DI0_SW_GEN1_6	0x00040044	0x1F040528	YES	YES
IPU_DI0_SW_GEN1_7	0x00040048	0x1F04052C	YES	YES
IPU_DI0_SW_GEN1_8	0x0004004C	0x1F040530	YES	YES
IPU_DI0_SW_GEN1_9	0x00040050	0x1F040534	YES	YES
IPU_DI0_SYNC_AS_GEN	0x00040054	0x1F040538	YES	YES
IPU_DI0_DW_GEN_0	0x00040058	0x1F04053C	YES	YES
IPU_DI0_DW_GEN_1	0x0004005C	0x1F040540	YES	YES
IPU_DI0_DW_GEN_2	0x00040060	0x1F040544	YES	YES
IPU_DI0_DW_GEN_3	0x00040064	0x1F040548	YES	YES
IPU_DI0_DW_GEN_4	0x00040068	0x1F04054C	YES	YES
IPU_DI0_DW_GEN_5	0x0004006C	0x1F040550	YES	YES
IPU_DI0_DW_GEN_6	0x00040070	0x1F040554	YES	YES
IPU_DI0_DW_GEN_7	0x00040074	0x1F040558	YES	YES
IPU_DI0_DW_GEN_8	0x00040078	0x1F04055C	YES	YES
IPU_DI0_DW_GEN_9	0x0004007C	0x1F040560	YES	YES
IPU_DI0_DW_GEN_10	0x00040080	0x1F040564	YES	YES
IPU_DI0_DW_GEN_11	0x00040084	0x1F040568	YES	YES
IPU_DI0_DW_SET0_0	0x00040088	0x1F04056C	YES	YES
IPU_DI0_DW_SET0_1	0x0004008C	0x1F040570	YES	YES
IPU_DI0_DW_SET0_2	0x00040090	0x1F040574	YES	YES
IPU_DI0_DW_SET0_3	0x00040094	0x1F040578	YES	YES
IPU_DI0_DW_SET0_4	0x00040098	0x1F04057C	YES	YES
IPU_DI0_DW_SET0_5	0x0004009C	0x1F040580	YES	YES
IPU_DI0_DW_SET0_6	0x000400A0	0x1F040584	YES	YES

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DI0_DW_SET0_7	0x000400A4	0x1F040588	YES	YES
IPU_DI0_DW_SET0_8	0x000400A8	0x1F04058C	YES	YES
IPU_DI0_DW_SET0_9	0x000400AC	0x1F040590	YES	YES
IPU_DI0_DW_SET0_10	0x000400B0	0x1F040594	YES	YES
IPU_DI0_DW_SET0_11	0x000400B4	0x1F040598	YES	YES
IPU_DI0_DW_SET1_0	0x000400B8	0x1F04059C	YES	YES
IPU_DI0_DW_SET1_1	0x000400BC	0x1F0405A0	YES	YES
IPU_DI0_DW_SET1_2	0x000400C0	0x1F0405A4	YES	YES
IPU_DI0_DW_SET1_3	0x000400C4	0x1F0405A8	YES	YES
IPU_DI0_DW_SET1_4	0x000400C8	0x1F0405AC	YES	YES
IPU_DI0_DW_SET1_5	0x000400CC	0x1F0405B0	YES	YES
IPU_DI0_DW_SET1_6	0x000400D0	0x1F0405B4	YES	YES
IPU_DI0_DW_SET1_7	0x000400D4	0x1F0405B8	YES	YES
IPU_DI0_DW_SET1_8	0x000400D8	0x1F0405BC	YES	YES
IPU_DI0_DW_SET1_9	0x000400DC	0x1F0405C0	YES	YES
IPU_DI0_DW_SET1_10	0x000400E0	0x1F0405C4	YES	YES
IPU_DI0_DW_SET1_11	0x000400E4	0x1F0405C8	YES	YES
IPU_DI0_DW_SET2_0	0x000400E8	0x1F0405CC	YES	YES
IPU_DI0_DW_SET2_1	0x000400EC	0x1F0405D0	YES	YES
IPU_DI0_DW_SET2_2	0x000400F0	0x1F0405D4	YES	YES
IPU_DI0_DW_SET2_3	0x000400F4	0x1F0405D8	YES	YES
IPU_DI0_DW_SET2_4	0x000400F8	0x1F0405DC	YES	YES
IPU_DI0_DW_SET2_5	0x000400FC	0x1F0405E0	YES	YES
IPU_DI0_DW_SET2_6	0x00040100	0x1F0405E4	YES	YES
IPU_DI0_DW_SET2_7	0x00040104	0x1F0405E8	YES	YES
IPU_DI0_DW_SET2_8	0x00040108	0x1F0405EC	YES	YES
IPU_DI0_DW_SET2_9	0x0004010C	0x1F0405F0	YES	YES
IPU_DI0_DW_SET2_10	0x00040110	0x1F0405F4	YES	YES
IPU_DI0_DW_SET2_11	0x00040114	0x1F0405F8	YES	YES
IPU_DI0_DW_SET3_0	0x00040118	0x1F0405FC	YES	YES
IPU_DI0_DW_SET3_1	0x0004011C	0x1F040600	YES	YES
IPU_DI0_DW_SET3_2	0x00040120	0x1F040604	YES	YES
IPU_DI0_DW_SET3_3	0x00040124	0x1F040608	YES	YES
IPU_DI0_DW_SET3_4	0x00040128	0x1F04060C	YES	YES
IPU_DI0_DW_SET3_5	0x0004012C	0x1F040610	YES	YES

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DI0_DW_SET3_6	0x00040130	0x1F040614	YES	YES
IPU_DI0_DW_SET3_7	0x00040134	0x1F040618	YES	YES
IPU_DI0_DW_SET3_8	0x00040138	0x1F04061C	YES	YES
IPU_DI0_DW_SET3_9	0x0004013C	0x1F040620	YES	YES
IPU_DI0_DW_SET3_10	0x00040140	0x1F040624	YES	YES
IPU_DI0_DW_SET3_11	0x00040144	0x1F040628	YES	YES
IPU_DI0_STP_REP_1	0x00040148	0x1F04062C	YES	YES
IPU_DI0_STP_REP_2	0x0004014C	0x1F040630	YES	YES
IPU_DI0_STP_REP_3	0x00040150	0x1F040634	YES	YES
IPU_DI0_STP_REP_4	0x00040154	0x1F040638	YES	YES
IPU_DI0_STP_REP_9	0x00040158	0x1F04063C	YES	YES
IPU_DI0_SER_CONF	0x0004015C	0x1F040640	YES	YES
IPU_DI0_SSC	0x00040160	0x1F040644	YES	YES
IPU_DI0_POL	0x00040164	0x1F040648	YES	YES
IPU_DI0_AW0	0x00040168	0x1F04064C	YES	YES
IPU_DI0_AW1	0x0004016C	0x1F040650	YES	YES
IPU_DI0_SCR_CONF	0x00040170	0x1F040654	YES	YES
IPU_DI0_STAT	0x00040174	NONE	NO	NO
IPU_DI1_GENERAL	0x00048000	0x1F040658	YES	YES
IPU_DI1_BS_CLKGEN0	0x00048004	0x1F04065C	YES	YES
IPU_DI1_BS_CLKGEN1	0x00048008	0x1F040660	YES	YES
IPU_DI1_SW_GEN0_1	0x0004800C	0x1F040664	YES	YES
IPU_DI1_SW_GEN0_2	0x00048010	0x1F040668	YES	YES
IPU_DI1_SW_GEN0_3	0x00048014	0x1F04066C	YES	YES
IPU_DI1_SW_GEN0_4	0x00048018	0x1F040670	YES	YES
IPU_DI1_SW_GEN0_5	0x0004801C	0x1F040674	YES	YES
IPU_DI1_SW_GEN0_6	0x00048020	0x1F040678	YES	YES
IPU_DI1_SW_GEN0_7	0x00048024	0x1F04067C	YES	YES
IPU_DI1_SW_GEN0_8	0x00048028	0x1F040680	YES	YES
IPU_DI1_SW_GEN0_9	0x0004802C	0x1F040684	YES	YES
IPU_DI1_SW_GEN1_1	0x00048030	0x1F040688	YES	YES
IPU_DI1_SW_GEN1_2	0x00048034	0x1F04068C	YES	YES
IPU_DI1_SW_GEN1_3	0x00048038	0x1F040690	YES	YES
IPU_DI1_SW_GEN1_4	0x0004803C	0x1F040694	YES	YES
IPU_DI1_SW_GEN1_5	0x00048040	0x1F040698	YES	YES

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DI1_SW_GEN1_6	0x00048044	0x1F04069C	YES	YES
IPU_DI1_SW_GEN1_7	0x00048048	0x1F0406A0	YES	YES
IPU_DI1_SW_GEN1_8	0x0004804C	0x1F0406A4	YES	YES
IPU_DI1_SW_GEN1_9	0x00048050	0x1F0406A8	YES	YES
IPU_DI1_SYNC_AS_GEN	0x00048054	0x1F0406AC	YES	YES
IPU_DI1_DW_GEN_0	0x00048058	0x1F0406B0	YES	YES
IPU_DI1_DW_GEN_1	0x0004805C	0x1F0406B4	YES	YES
IPU_DI1_DW_GEN_2	0x00048060	0x1F0406B8	YES	YES
IPU_DI1_DW_GEN_3	0x00048064	0x1F0406BC	YES	YES
IPU_DI1_DW_GEN_4	0x00048068	0x1F0406C0	YES	YES
IPU_DI1_DW_GEN_5	0x0004806C	0x1F0406C4	YES	YES
IPU_DI1_DW_GEN_6	0x00048070	0x1F0406C8	YES	YES
IPU_DI1_DW_GEN_7	0x00048074	0x1F0406CC	YES	YES
IPU_DI1_DW_GEN_8	0x00048078	0x1F0406D0	YES	YES
IPU_DI1_DW_GEN_9	0x0004807C	0x1F0406D4	YES	YES
IPU_DI1_DW_GEN_10	0x00048080	0x1F0406D8	YES	YES
IPU_DI1_DW_GEN_11	0x00048084	0x1F0406DC	YES	YES
IPU_DI1_DW_SET0_0	0x00048088	0x1F0406E0	YES	YES
IPU_DI1_DW_SET0_1	0x0004808C	0x1F0406E4	YES	YES
IPU_DI1_DW_SET0_2	0x00048090	0x1F0406E8	YES	YES
IPU_DI1_DW_SET0_3	0x00048094	0x1F0406EC	YES	YES
IPU_DI1_DW_SET0_4	0x00048098	0x1F0406F0	YES	YES
IPU_DI1_DW_SET0_5	0x0004809C	0x1F0406F4	YES	YES
IPU_DI1_DW_SET0_6	0x000480A0	0x1F0406F8	YES	YES
IPU_DI1_DW_SET0_7	0x000480A4	0x1F0406FC	YES	YES
IPU_DI1_DW_SET0_8	0x000480A8	0x1F040700	YES	YES
IPU_DI1_DW_SET0_9	0x000480AC	0x1F040704	YES	YES
IPU_DI1_DW_SET0_10	0x000480B0	0x1F040708	YES	YES
IPU_DI1_DW_SET0_11	0x000480B4	0x1F04070C	YES	YES
IPU_DI1_DW_SET1_0	0x000480B8	0x1F040710	YES	YES
IPU_DI1_DW_SET1_1	0x000480BC	0x1F040714	YES	YES
IPU_DI1_DW_SET1_2	0x000480C0	0x1F040718	YES	YES
IPU_DI1_DW_SET1_3	0x000480C4	0x1F04071C	YES	YES
IPU_DI1_DW_SET1_4	0x000480C8	0x1F040720	YES	YES
IPU_DI1_DW_SET1_5	0x000480CC	0x1F040724	YES	YES

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DI1_DW_SET1_6	0x000480D0	0x1F040728	YES	YES
IPU_DI1_DW_SET1_7	0x000480D4	0x1F04072C	YES	YES
IPU_DI1_DW_SET1_8	0x000480D8	0x1F040730	YES	YES
IPU_DI1_DW_SET1_9	0x000480DC	0x1F040734	YES	YES
IPU_DI1_DW_SET1_10	0x000480E0	0x1F040738	YES	YES
IPU_DI1_DW_SET1_11	0x000480E4	0x1F04073C	YES	YES
IPU_DI1_DW_SET2_0	0x000480E8	0x1F040740	YES	YES
IPU_DI1_DW_SET2_1	0x000480EC	0x1F040744	YES	YES
IPU_DI1_DW_SET2_2	0x000480F0	0x1F040748	YES	YES
IPU_DI1_DW_SET2_3	0x000480F4	0x1F04074C	YES	YES
IPU_DI1_DW_SET2_4	0x000480F8	0x1F040750	YES	YES
IPU_DI1_DW_SET2_5	0x000480FC	0x1F040754	YES	YES
IPU_DI1_DW_SET2_6	0x00048100	0x1F040758	YES	YES
IPU_DI1_DW_SET2_7	0x00048104	0x1F04075C	YES	YES
IPU_DI1_DW_SET2_8	0x00048108	0x1F040760	YES	YES
IPU_DI1_DW_SET2_9	0x0004810C	0x1F040764	YES	YES
IPU_DI1_DW_SET2_10	0x00048110	0x1F040768	YES	YES
IPU_DI1_DW_SET2_11	0x00048114	0x1F04076C	YES	YES
IPU_DI1_DW_SET3_0	0x00048118	0x1F040770	YES	YES
IPU_DI1_DW_SET3_1	0x0004811C	0x1F040774	YES	YES
IPU_DI1_DW_SET3_2	0x00048120	0x1F040778	YES	YES
IPU_DI1_DW_SET3_3	0x00048124	0x1F04077C	YES	YES
IPU_DI1_DW_SET3_4	0x00048128	0x1F040780	YES	YES
IPU_DI1_DW_SET3_5	0x0004812C	0x1F040784	YES	YES
IPU_DI1_DW_SET3_6	0x00048130	0x1F040788	YES	YES
IPU_DI1_DW_SET3_7	0x00048134	0x1F04078C	YES	YES
IPU_DI1_DW_SET3_8	0x00048138	0x1F040790	YES	YES
IPU_DI1_DW_SET3_9	0x0004813C	0x1F040794	YES	YES
IPU_DI1_DW_SET3_10	0x00048140	0x1F040798	YES	YES
IPU_DI1_DW_SET3_11	0x00048144	0x1F04079C	YES	YES
IPU_DI1_STP_REP_1	0x00048148	0x1F0407A0	YES	YES
IPU_DI1_STP_REP_2	0x0004814C	0x1F0407A4	YES	YES
IPU_DI1_STP_REP_3	0x00048150	0x1F0407A8	YES	YES
IPU_DI1_STP_REP_4	0x00048154	0x1F0407AC	YES	YES
IPU_DI1_STP_REP_9	0x00048158	0x1F0407B0	YES	YES

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DI1_SER_CONF	0x0004815C	0x1F0407B4	YES	YES
IPU_DI1_SSC	0x00048160	0x1F0407B8	YES	YES
IPU_DI1_POL	0x00048164	0x1F0407BC	YES	YES
IPU_DI1_AW0	0x00048168	0x1F0407C0	YES	YES
IPU_DI1_AW1	0x0004816C	0x1F0407C4	YES	YES
IPU_DI1_SCR_CONF	0x00048170	0x1F0407C8	YES	YES
IPU_DI1_STAT	0x00048174	NONE	NO	NO
IPU_SMFC_MAP	0x00050000	NONE	YES	NO
IPU_SMFC_WMC	0x00050004	NONE	YES	NO
IPU_SMFC_BS	0x00050008	NONE	YES	NO
IPU_DC_READ_CH_CONF	0x00058000	NONE	YES	NO
IPU_DC_READ_CH_ADDR	0x00058004	NONE	YES	NO
IPU_DC_RL0_CH_0	0x00058008	NONE	YES	NO
IPU_DC_RL1_CH_0	0x0005800C	NONE	YES	NO
IPU_DC_RL2_CH_0	0x00058010	NONE	YES	NO
IPU_DC_RL3_CH_0	0x00058014	NONE	YES	NO
IPU_DC_RL4_CH_0	0x00058018	NONE	YES	NO
IPU_DC_WR_CH_CONF_1	0x0005801C	NONE	YES	NO
IPU_DC_WR_CH_ADDR_1	0x00058020	NONE	YES	NO
IPU_DC_RL0_CH_1	0x00058024	NONE	YES	NO
IPU_DC_RL1_CH_1	0x00058028	NONE	YES	NO
IPU_DC_RL2_CH_1	0x0005802C	NONE	YES	NO
IPU_DC_RL3_CH_1	0x00058030	NONE	YES	NO
IPU_DC_RL4_CH_1	0x00058034	NONE	YES	NO
IPU_DC_WR_CH_CONF_2	0x00058038	0x1F0404AC	YES	NO
IPU_DC_WR_CH_ADDR_2	0x0005803C	0x1F0404B0	YES	NO
IPU_DC_RL0_CH_2	0x00058040	0x1F0404B4	YES	NO
IPU_DC_RL1_CH_2	0x00058044	0x1F0404B8	YES	NO
IPU_DC_RL2_CH_2	0x00058048	0x1F0404BC	YES	NO
IPU_DC_RL3_CH_2	0x0005804C	0x1F0404C0	YES	NO
IPU_DC_RL4_CH_2	0x00058050	0x1F0404C4	YES	NO
IPU_DC_CMD_CH_CONF_3	0x00058054	NONE	YES	NO
IPU_DC_CMD_CH_CONF_4	0x00058058	NONE	YES	NO
IPU_DC_WR_CH_CONF_5	0x0005805C	NONE	YES	NO
IPU_DC_WR_CH_ADDR_5	0x00058060	NONE	YES	NO

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DC_RL0_CH_5	0x00058064	NONE	YES	NO
IPU_DC_RL1_CH_5	0x00058068	NONE	YES	NO
IPU_DC_RL2_CH_5	0x0005806C	NONE	YES	NO
IPU_DC_RL3_CH_5	0x00058070	NONE	YES	NO
IPU_DC_RL4_CH_5	0x00058074	NONE	YES	NO
IPU_DC_WR_CH_CONF_6	0x00058078	0x1F0404C8	YES	NO
IPU_DC_WR_CH_ADDR_6	0x0005807C	0x1F0404CC	YES	NO
IPU_DC_RL0_CH_6	0x00058080	0x1F0404D0	YES	NO
IPU_DC_RL1_CH_6	0x00058084	0x1F0404D4	YES	NO
IPU_DC_RL2_CH_6	0x00058088	0x1F0404D8	YES	NO
IPU_DC_RL3_CH_6	0x0005808C	0x1F0404DC	YES	NO
IPU_DC_RL4_CH_6	0x00058090	0x1F0404E0	YES	NO
IPU_DC_WR_CH_CONF1_8	0x00058094	NONE	YES	NO
IPU_DC_WR_CH_CONF2_8	0x00058098	NONE	YES	NO
IPU_DC_RL1_CH_8	0x0005809C	NONE	YES	NO
IPU_DC_RL2_CH_8	0x000580A0	NONE	YES	NO
IPU_DC_RL3_CH_8	0x000580A4	NONE	YES	NO
IPU_DC_RL4_CH_8	0x000580A8	NONE	YES	NO
IPU_DC_RL5_CH_8	0x000580AC	NONE	YES	NO
IPU_DC_RL6_CH_8	0x000580B0	NONE	YES	NO
IPU_DC_WR_CH_CONF1_9	0x000580B4	NONE	YES	NO
IPU_DC_WR_CH_CONF2_9	0x000580B8	NONE	YES	NO
IPU_DC_RL1_CH_9	0x000580BC	NONE	YES	NO
IPU_DC_RL2_CH_9	0x000580C0	NONE	YES	NO
IPU_DC_RL3_CH_9	0x000580C4	NONE	YES	NO
IPU_DC_RL4_CH_9	0x000580C8	NONE	YES	NO
IPU_DC_RL5_CH_9	0x000580CC	NONE	YES	NO
IPU_DC_RL6_CH_9	0x000580D0	NONE	YES	NO
IPU_DC_GEN	0x000580D4	NONE	YES	NO
IPU_DC_DISP_CONF1_0	0x000580D8	NONE	YES	NO
IPU_DC_DISP_CONF1_1	0x000580DC	NONE	YES	NO
IPU_DC_DISP_CONF1_2	0x000580E0	NONE	YES	NO
IPU_DC_DISP_CONF1_3	0x000580E4	NONE	YES	NO
IPU_DC_DISP_CONF2_0	0x000580E8	NONE	YES	NO
IPU_DC_DISP_CONF2_1	0x000580EC	NONE	YES	NO

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DC_DISP_CONF2_2	0x000580F0	NONE	YES	NO
IPU_DC_DISP_CONF2_3	0x000580F4	NONE	YES	NO
IPU_DC_DI0_CONF_1	0x000580F8	NONE	YES	NO
IPU_DC_DI0_CONF_2	0x000580FC	NONE	YES	NO
IPU_DC_DI1_CONF_1	0x00058100	NONE	YES	NO
IPU_DC_DI1_CONF_2	0x00058104	NONE	YES	NO
IPU_DC_MAP_CONF_0	0x00058108	NONE	YES	NO
IPU_DC_MAP_CONF_1	0x0005810C	NONE	YES	NO
IPU_DC_MAP_CONF_2	0x00058110	NONE	YES	NO
IPU_DC_MAP_CONF_3	0x00058114	NONE	YES	NO
IPU_DC_MAP_CONF_4	0x00058118	NONE	YES	NO
IPU_DC_MAP_CONF_5	0x0005811C	NONE	YES	NO
IPU_DC_MAP_CONF_6	0x00058120	NONE	YES	NO
IPU_DC_MAP_CONF_7	0x00058124	NONE	YES	NO
IPU_DC_MAP_CONF_8	0x00058128	NONE	YES	NO
IPU_DC_MAP_CONF_9	0x0005812C	NONE	YES	NO
IPU_DC_MAP_CONF_10	0x00058130	NONE	YES	NO
IPU_DC_MAP_CONF_11	0x00058134	NONE	YES	NO
IPU_DC_MAP_CONF_12	0x00058138	NONE	YES	NO
IPU_DC_MAP_CONF_13	0x0005813C	NONE	YES	NO
IPU_DC_MAP_CONF_14	0x00058140	NONE	YES	NO
IPU_DC_MAP_CONF_15	0x00058144	NONE	YES	NO
IPU_DC_MAP_CONF_16	0x00058148	NONE	YES	NO
IPU_DC_MAP_CONF_17	0x0005814C	NONE	YES	NO
IPU_DC_MAP_CONF_18	0x00058150	NONE	YES	NO
IPU_DC_MAP_CONF_19	0x00058154	NONE	YES	NO
IPU_DC_MAP_CONF_20	0x00058158	NONE	YES	NO
IPU_DC_MAP_CONF_21	0x0005815C	NONE	YES	NO
IPU_DC_MAP_CONF_22	0x00058160	NONE	YES	NO
IPU_DC_MAP_CONF_23	0x00058164	NONE	YES	NO
IPU_DC_MAP_CONF_24	0x00058168	NONE	YES	NO
IPU_DC_MAP_CONF_25	0x0005816C	NONE	YES	NO
IPU_DC_MAP_CONF_26	0x00058170	NONE	YES	NO
IPU_DC_UGDE0_0	0x00058174	NONE	YES	NO
IPU_DC_UGDE0_1	0x00058178	NONE	YES	NO

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_DC_UGDE0_2	0x0005817C	NONE	YES	NO
IPU_DC_UGDE0_3	0x00058180	NONE	YES	NO
IPU_DC_UGDE1_0	0x00058184	NONE	YES	NO
IPU_DC_UGDE1_1	0x00058188	NONE	YES	NO
IPU_DC_UGDE1_2	0x0005818C	NONE	YES	NO
IPU_DC_UGDE1_3	0x00058190	NONE	YES	NO
IPU_DC_UGDE2_0	0x00058194	NONE	YES	NO
IPU_DC_UGDE2_1	0x00058198	NONE	YES	NO
IPU_DC_UGDE2_2	0x0005819C	NONE	YES	NO
IPU_DC_UGDE2_3	0x000581A0	NONE	YES	NO
IPU_DC_UGDE3_0	0x000581A4	NONE	YES	NO
IPU_DC_UGDE3_1	0x000581A8	NONE	YES	NO
IPU_DC_UGDE3_2	0x000581AC	NONE	YES	NO
IPU_DC_UGDE3_3	0x000581B0	NONE	YES	NO
IPU_DC_LLA0	0x000581B4	NONE	YES	NO
IPU_DC_LLA1	0x000581B8	NONE	YES	NO
IPU_DC_R_LLA0	0x000581BC	NONE	YES	NO
IPU_DC_R_LLA1	0x000581C0	NONE	YES	NO
IPU_DC_WR_CH_ADDR_5_ALT	0x000581C4	NONE	YES	NO
IPU_DC_STAT	0x000581C8	NONE	NO	NO
IPU_DMFC_RD_CHAN	0x00060000	NONE	YES	NO
IPU_DMFC_WR_CHAN	0x00060004	NONE	YES	NO
IPU_DMFC_WR_CHAN_DEF	0x00060008	NONE	YES	NO
IPU_DMFC_DP_CHAN	0x0006000C	NONE	YES	NO
IPU_DMFC_DP_CHAN_DEF	0x00060010	NONE	YES	NO
IPU_DMFC_GENERAL1	0x00060014	NONE	YES	NO
IPU_DMFC_GENERAL2	0x00060018	NONE	YES	NO
IPU_DMFC_IC_CTRL	0x0006001C	NONE	YES	NO
IPU_DMFC_WR_CHAN_ALT	0x00060020	NONE	YES	NO
IPU_DMFC_WR_CHAN_DEF_ALT	0x00060024	NONE	YES	NO
IPU_DMFC_DP_CHAN_ALT	0x00060028	NONE	YES	NO
IPU_DMFC_DP_CHAN_DEF_ALT	0x0006002C	NONE	YES	NO
IPU_DMFC_GENERAL1_ALT	0x00060030	NONE	YES	NO
IPU_DMFC_STAT	0x00060034	NONE	NO	NO
IPU_VDI_FSIZE	0x00068000	NONE	YES	NO

Table continues on the next page...

Table 45-36. IPU SRM entries mapping (continued)

Register's name	Register's address	SRM entry	PG	LPSR
IPU_VDI_C	0x00068004	NONE	YES	NO
IPU_VDI_C2	0x00068008	NONE	YES	NO
IPU_VDI_CMBP_1	0x0006800C	NONE	YES	NO
IPU_VDI_CMBP_2	0x00068010	NONE	YES	NO
IPU_VDI_PS_1	0x00068014	NONE	YES	NO
IPU_VDI_PS_2	0x00068018	NONE	YES	NO
IPU_VDI_PS_3	0x0006801C	NONE	YES	NO
IPU_VDI_PS_4	0x00068020	NONE	YES	NO

45.4.12.7 Memory Access Unit

The Memory Access Unit (MA) supports ARM platform access to the IPU internal memories.

Some of the IPU internal memories are memory mapped. This unit handles accessing these memories. The table below describe the accessible memories and their limitations.

Table 45-39. Internal Memories Access Support and Limitations

Memory	Function	Support and Limitations
lut	IDMAC's look up table	Accessible only when All the IDMAC channels that use the LUT are disabled
cpmem	IDMAC's Channel parameter Memory	This memory can be accessed while tasks are enabled. Must be configured before enabling processing tasks. IDMAC channel parameters must not be changed in the CPM when the corresponding DMA channel is enabled excluding the base addresses (EBA0 and EBA1). One of these parameters can be changed during channel operation if it relates to the non-active double buffer.
tpm	IC's task parameter memory	Must be configured before enabling processing tasks. This memory can be accessed while tasks are enabled. IC task parameters must not be changed in the TPM when the corresponding task is active.
dc_template	DC's template memory	Can be configured before enabling tasks. Must not be accessed while operation. Both read and write access to the DC's template memory are forbidden when there is any enabled channel which can use the template

45.4.12.8 SISG - Still Image Synchronization Generator

The IPU incudes a "Still Image Synchronization Generator" (SISG), providing time-sensitive control signals synchronizing the image sensor with camera peripherals, such as a flash lamp and a mechanical shutter.

The SISG is implemented using a single time base counter, and six Time Compare Units - as described in the following figure.

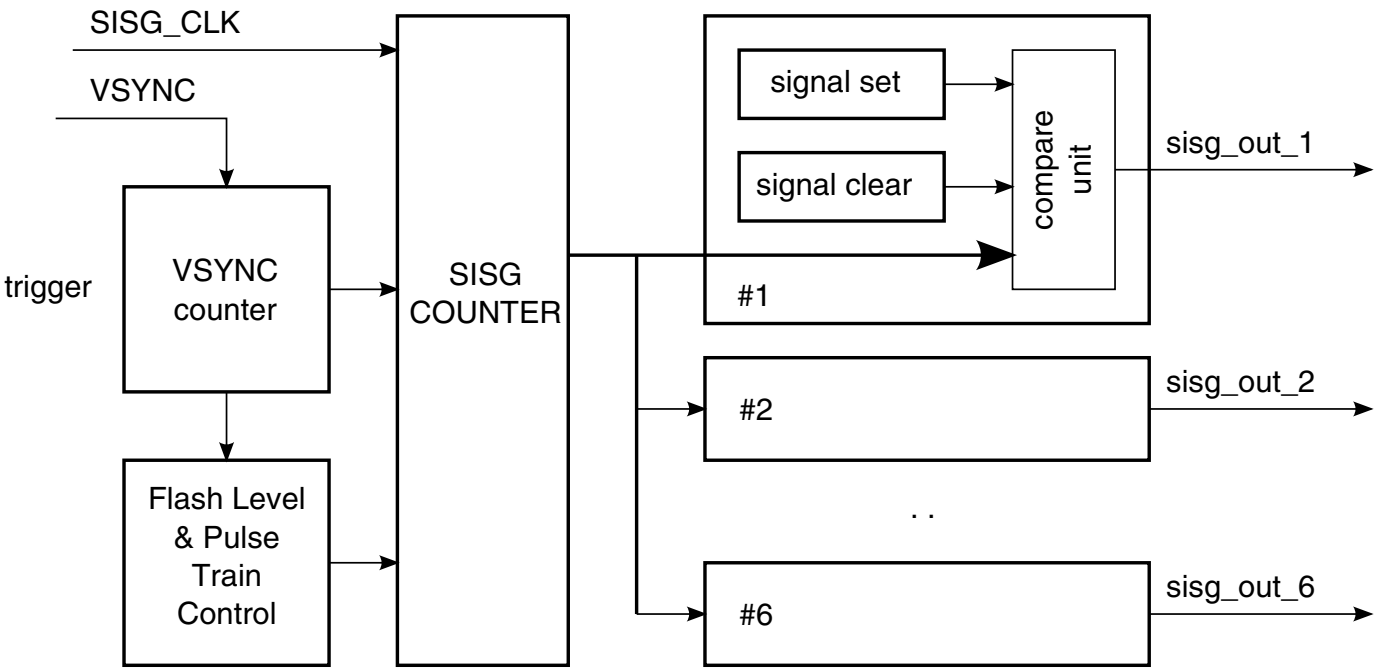


Figure 45-58. Still Image Signal Generator

The SISG inputs are:

- Activation trigger
- VSYNC: the frame-boundary signal from the sensor

The SISG is activated by one of the following triggers:

- The ARM platform by setting the MCU_ACTV_TRIG bit
- An external signal (GPIO pin)

Upon activation, the counter is reset and then there are the following two possibilities:

- Counting starts immediately
- Counting start is delayed until one of the next 7 VSYNC signals (programmable via the NO_OF_VSYNC bits)

During the counting period, the SISG can generate up to 6 output strobes:

- Each strobe can be individually enabled or disabled and has a programmable polarity
- The edges of the strobes are generated at specified counter values - to achieve pixel-level resolution - as specified by programmable SISG_SET& SISG_CLR time tag registers
- The clock has 25 bits, to allow strobe generation during a time period of up to two 12M pixel frames

The SISG can repeat the above sequence for up to 32 cycles (this is provided to generate a train of flash pulses, for anti-red-eye or for measurements in low-light conditions). The repetition is implemented by resetting the counter, which can be triggered by one of the following events:

- A VSYNC signal
- A pre-defined value reached by the counter

After the last sequence, when the counter reaches its maximal value, it stops counting and the SISG remains in idle mode until the next activation.

45.4.12.9 Clock Change procedure

The IPU supports dynamic clock rate changes.

Types of change:

- DVFS transitions: frequent, initiated by the SoC's power modes controller (GPC) and the SoC's clock controller module (CCM)
- Other: infrequent, initiated by SW.

IPU may have on-going activities at this stage.

The display interface clocks may either change or remain unchanged. During screen refresh, the display clock would typically not change. During asynchronous access, it may be appropriate to change also the display interface clock. The choice between these options would be made in advance by the user

If the IPU display interface uses the external clock (ipp_di_0_ext_clk or ipp_di_1_ext_clk) source, it remains unchanged. A change in the rate of this clock is performed fully by SW, without the special HW support described below. In particular, the SW may have to stop explicitly any interaction with the display (e.g. screen refresh) before performing the change.

The user is responsible to make sure that the lowest planned clock (in DVFS transitions) is still high enough to support the expected activities (e.g. data rate through the display bus)

The procedure below describes the IPU handshaking with the CCM.

1. The user prepares 2 sets of clock modes CLOCK_MODE_0 is the default clock mode, CLOCK_MODE_1 is the alternate clock mode. The IPU toggles between these two settings following the next assertion of ipg_clk_change_rq. If the user sets the SRM_CLOCK_CHANGE_MODE bit then he should also prepare the registers in the SRM for each of the DIs. These registers include all the DI settings adjusted to the new clock.
2. CCM asserts the ipg_clk_change_rq signal when a clock change is needed
3. The CM calculates the new clock frequency and send it to the DI (signals are di0_clk_freq, and di1_clk_freq). These signals should be sent to the DI only after getting the di_clk_change_ack signal from the DIs. The values of this field could be.
   - 00 - 1/4 of full frequency
   - 01 - 1/2 of full frequency
   - 10 - full frequency
   - 11 - illegal
4. The CM sends a cm_clk_change_rq signal to the DIs.
5. The DI stops the clock to the display (freeze mode) according to DI0_CLOCK_STOP_MODE & DI1_CLOCK_STOP_MODE bits. If the DI is disable, the ACK from the DI is not needed and the CM will assume that the DI sent an ACK.
6. Once the clock to the display is stopped, the DI sends a signal to the CM called di_clk_change_ack
7. The CM wait for the clock change signals from both of the DIs
8. If the SRM_CLOCK_CHANGE_MODE bit is set the CM should read the new DI settings from the SRM and override the previous DI settings. Then the CM clears the SRM_CLOCK_CHANGE_MODE bit
9. Once the above is complete the CM asserts the ipg_clk_change_ack signal to the CCM
10. The CM sends the signals di0_clk_freq, and di1_clk_freq to the DIs.
11. The CCM will negate the cm_clk_change_rq
12. The CCM will now change the clocks
13. When the new clock arrives the ipu_clk_changed signal will be asserted.
14. The state machine on each DI will now move out of freeze mode and continue working with the new clock.

45.4.12.10 Low Power Modes - Stop, PG and LPSR modes

IPU supports 3 low power modes.

- STOP: on this mode the clock to the IPU is stopped

- PG: power gating, where the clock to the IPU is stopped and the power is for the IPU is turned off. Some of the memories are kept powered, so the wake up from this mode will be faster.
- LPSR: low power screen refresh.The clock to the IPU is changed to a slower frequency, the IPU performs only screen refresh.

The user should not request both LPSR and PG. at the same time. This case is not supported and the results are not predictable.

The CCM may assert one of the 2 signals: stop_clk_at_stop_req OR stop_clk_at_wait_req

The IPU OR them internally as there's no difference between them with regards to IPU's behavior.

In all these modes the clock to the IPU is going to be stopped (assertion of stop_clk_at_stop_req).

IPU should complete all his tasks:

- CSIs complete transferring the last frame. Wait for csi_busy = 0
- IDMAC completes all the flows (all CH_BUSY = 0)
- All the flows in the FSU are complete. New ones do not start.
- CM sends stop request to the DIs
- Once the DI sent all the data to the display, it asserts an ACK signal

Only when all the above occurs, the CM can assert an internal signal called "IPU_IDLE"

IPU_IDLE is the starting point for any of the low power modes.

Note that if the VDIC was in use prior to entrance the low power mode the viewfinder task of the IC will be in WAIT_FOR_READY state (the task's status is reflected in the VF_TSTAT field). The user will need to manually switch that task to IDLE. This is done by performing the following steps:

- Wait for EOF of the viewfinder output channel (IDMAC_EOF_21)
- Set RSW_EN bit
- resume one frame via the viewfinder task.
- Disable the VDI and IC and the corresponding IDMAC channels

45.4.12.10.1 STOP Mode

In this mode the IPU sends the ipu_stby_ack after getting to IPU_IDLE state. The CCM will gate off the clock to the IPU.

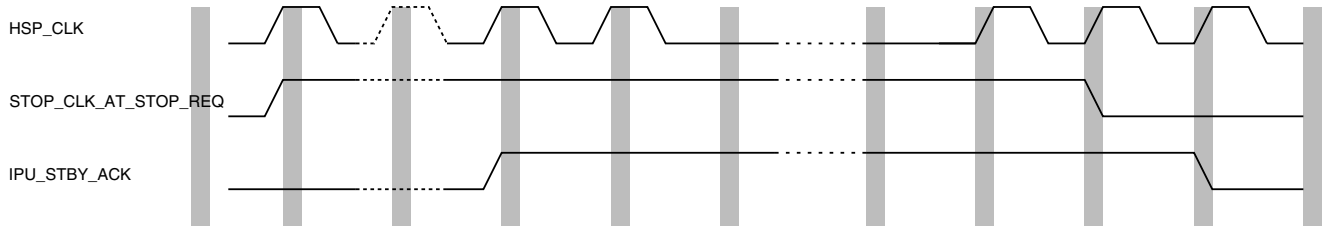


Figure 45-59. Entering and Exiting STOP Mode

Wake up from STOP mode

When the SoC decides to wake up from STOP mode it should:

- Resume the clock to the IPU.
- Negate the stop_clk_at_stop_req or stop_clk_at_wait_req signal.
- The IPU will then negate the ipu_stby_ack signal.
- The IPU will resume screen refresh.

45.4.12.10.2 Power Gating

In power gating mode the clock to the IPU is stopped and part of the IPU is powered down. Only the memories that are essential for a quick and smooth wake up are kept powered.

The procedure for entering PG mode is the similar to the procedure for entering STOP mode.

1. If the stop request is asserted and the "gpc_pg_ipu" signal is asserted then we are entering PG mode.
2. The IPU follows the same process as on STOP mode till getting into IPU_IDLE state.
3. The IPU saves in the SRM the content of the registers of the sub-blocks participating in screen refresh. The sub-blocks are:
   - DI0 & DI1
   - DC
   - DP
   - DMFC
   - IDMAC
   - IC
   - CM
4. After saving the registers the IPU will send the ipu_stby_ack signal to the CCM.
5. The CCM will gate off the clock to the IPU.

6. The GPC will gate off the power to the IPU. The memories listed below are needed for quick and smooth wake up, therefore the memories listed below are kept powered

Table 45-40. Memories that are Powered at PG mode

Sub-blocks	Instance name	Description
IDMAC	ipu_memories_lut0 ipu_memories_lut1	Look UP table memory
IDMAC	ipu_memories_cpmem	Channel parameter memory
CM shadow	ipu_memories_srm	Shadow registers memory
IC + IRT	ipu_memories_tpm	IC's task's parameters memory
DC	ipu_memories_dc_template	DC's template

Wake up from PG mode

1. The GPC resumes power to the IPU
2. The GPC drives the signal gpc_ipu_pwgtd, indicating that the IPU wakes up from PG mode.
3. The SoC's reset controller resets the IPU by asserting the ipg_hard_async_reset_b signal.
4. The CCM resumes the clock to the IPU.
5. The IPU reads the registers from the SRM and writes the content to the sub-block's registers.
6. The IPU sends the ipu_wakeup_ack to the CCM, indicating that the wake up procedure is complete. This signal is negated when gpc_ipu_pwgtd is negated.
7. The IPU resumes screen refresh.

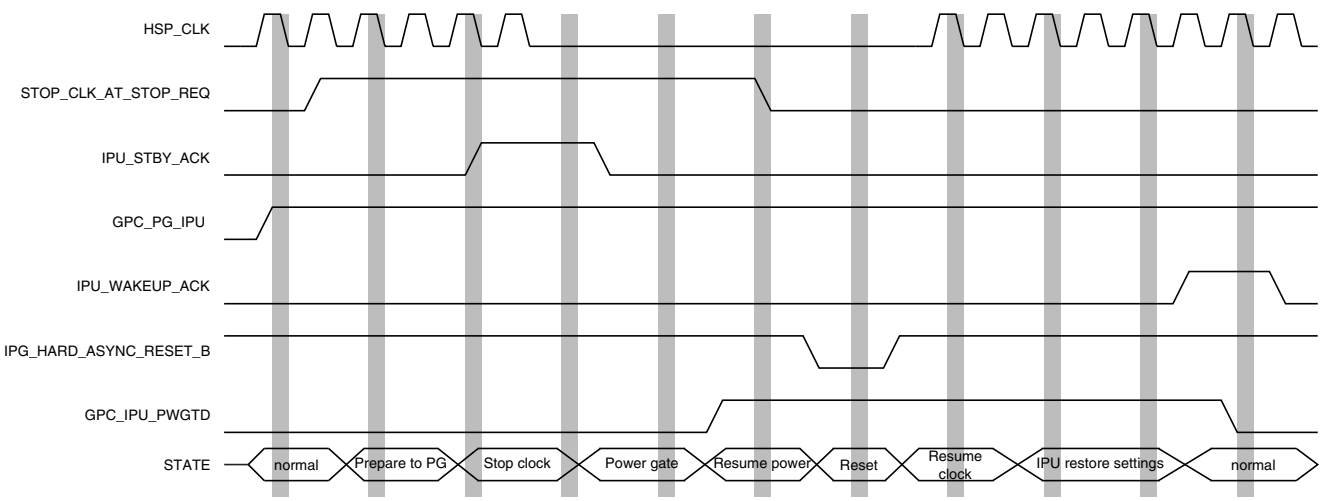


Figure 45-60. Entering and Exiting PG mode

45.4.12.10.3 Low Power Screen Refresh mode - LPSR

In Low Power Screen Refresh mode, the clock to the IPU is changed to a slower frequency, the IPU performs only screen refresh to a single display via the DP (channels 23 & 27).

Preparations

1. The user sets the LPSR_MODE bit indicating that the next assertion of stop_clk_at_stop_req OR stop_clk_at_wait_req activates the LPSR procedure.
2. The user moves the IPU to screen refresh flow. This means that if other tasks are active (flows via CSI or multiple flows to multiple displays) - this tasks needs to be complete and disabled. This is step is fully done by the user (SW). The only flow that remains active is screen refresh to a single display done via the DP (channels 23 & 27)
3. The user stores in the SRM the planned configuration for the sub-blocks involved in the LPSR flow. These configuration will e switched with the active registers' settings on later stage. The relevant sub-blocks are:
   - DI0 & DI1
   - DC
   - DP
   - DMFC
   - IDMAC
   - CM

Entering LPSR

The flow for entering LPSR is the same as stop mode.

1. The IPU follows the same procedure as on STOP mode till getting into IPU_IDLE state.
2. IPU swaps the registers of the relevant blocks with the pre stored content from the SRM. The content of the registers of the current flow is stored in the SRM. The IPU will switch back to this configuration after exiting from LPSR.
3. The IPU sends the ipu_stby_ack
4. The CCM will gate off the clock to the IPU.
5. The CCM will switch to the new clock
6. After changing the clock, the CCM asserts ccm_lpsr_ipu, this signal is synched inside the IPU to the hsp_clk
7. The IPU will resume screen refresh with the new settings.

Exit from LPSR

1. The CCM negates ccm_lpsr_ipu
2. The CM sends standby request to the DI

3. The DI should complete processing the current frame and stop the clock to the display and send an acknowledge signal to the CM.
4. The SRM swaps the registers' configuration. current configuration (LPSR) is saved in the SRM, the previous (original) configuration is stored in the blocks' registers.
5. IPU asserts the ipu_wakeup_ack signal indicating that it is now safe to leave LPSR mode.
6. CCM stops the clocks to IPU
7. CCM resumes the clock to the IPU - This clock is the same clock used prior to entering the LPSR mode.
8. CCM negates the stop_clk_at_stop_req (or stop_clk_at_wait_req)
9. IPU negates the ipu_stby_ack and the ipu_wakeup_ack signals
10. The IPU resumes screen refresh with the original settings.

The diagram below illustrates the procedure for entering and leaving LPSR mode

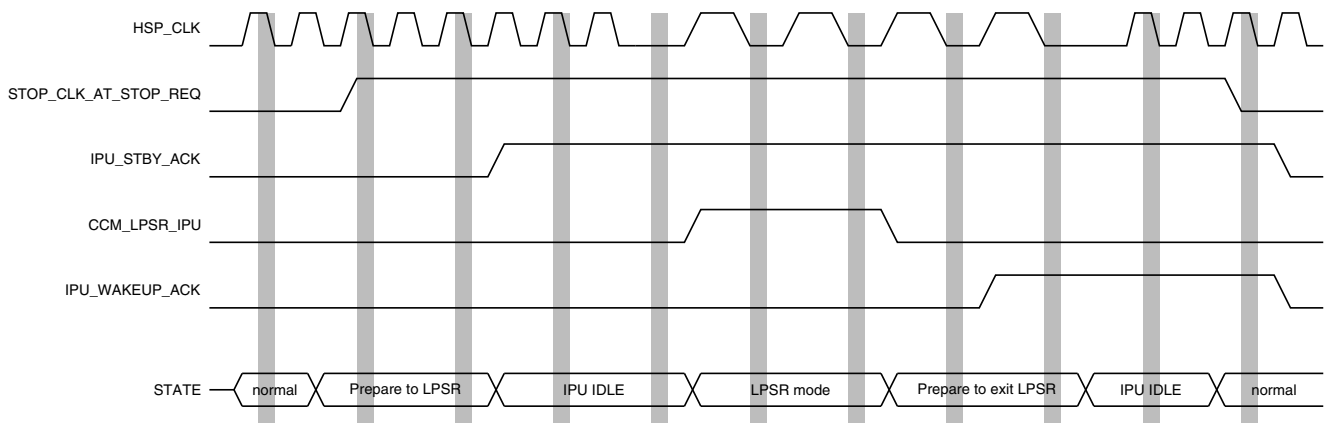


Figure 45-61. Entering and Exiting LPSR mode

45.4.13 IPU diagnostics unit

The IPU has a 16 bit diagnostic bus called ipu_diagbus. This bus can be connected at SoC level so these 16 outputs of the IPU can be routed to the SoC's pins and used for signals monitoring during debug.

The internal signals that can be monitored are internal status signals. There are 18 groups of signals. In order to route the selected group the user need to set the IPU_DIAGBUS_MODE and select the correct group. The sub-blocks are arranged in groups as described on the tables below.

Table 45-41. IPU diag bus groups 0-3

	group0	group1	group2	group3
0	IDMAC_NFACK_6	DC_TEARING_ERR_6	IDMAC_EOF_40	IDMAC_EOF_5
1	IDMAC_NFACK_4	DC_TEARING_ERR_2	IDMAC_NFACK_40	IDMAC_NFACK_5
2	CSI1_PUPE	DC_TEARING_ERR_1	DC_ASYNC_STOP	
3	CSI0_PUPE	DC_FC_6	DMA_CH_CUR_BUF_40	
4		DC_FC_4	DI0_READ_FIFO_FULL	DMA_CH_CUR_BUF_5
5	SMFC3_FRM_LOST	DC_FC_3	DI0_READ_FIFO_EMPTY	DMA_CH_ALT_CUR_BUF_5
6	SMFC2_FRM_LOST	DC_FC_2	DI1_READ_FIFO_FULL	
7	SMFC1_FRM_LOST	DC_FC_1	DI1_READ_FIFO_EMPTY	DMA_CH_ALT_BUF0_RDY_5
8	SMFC0_FRM_LOST	DC_FC_0	DC_TRIPLE_BUF_CNT_EMPTY_1	DMA_CH_ALT_BUF1_RDY_5
9	DMA_CH_ALT_CUR_BUF_52	DC_6_SRM_STAT	DC_TRIPLE_BUF_CNT_FULL_1	DMFC_FIFO_EMPTY_9
10	CSI1_SRM_STAT	DC_2_SRM_STAT	DC_TRIPLE_BUF_DATA_EMPTY_0	DMFC_FIFO_EMPTY_8
11	CSI0_SRM_STAT	DC_ASYNC1_CUR_FLOW	DC_TRIPLE_BUF_DATA_FULL_0	DMFC_FIFO_EMPTY_6
12	IDMAC_EOF_8	Reserved	DC_TRIPLE_BUF_CNT_EMPTY_0	DMFC_FIFO_EMPTY_1
13	IDMAC_EOF_9	Reserved	DC_TRIPLE_BUF_CNT_FULL_0	DMFC_FIFO_FULL_9
14	IDMAC_EOF_10	DC_TRIPLE_BUF_DATA_EMPTY_1	DMFC_FIFO_EMPTY_4	DMFC_FIFO_FULL_8
15	IDMAC_EOF_13	DC_TRIPLE_BUF_DATA_FULL_1	DMFC_FIFO_FULL_4	DMFC_FIFO_FULL_6

Table 45-42. IPU diag bus groups 4-7

	group4	group5	group6	group7
0	DI0_CNT_EN_PRE_10	IDMAC_EOF_52	IDMAC_EOF_28	IDMAC_EOF_31
1	DI0_CNT_EN_PRE_9	IDMAC_NFACK_52	IDMAC_EOF_27	IDMAC_EOF_51
2	DI0_CNT_EN_PRE_8	DI1_SYNC_DISP_ERR	IDMAC_EOF_24	IDMAC_NFACK_31
3	DI0_CNT_EN_PRE_6	DI1_CNT_EN_PRE_8	IDMAC_EOF_23	IDMAC_NFACK_28
4	DI0_CNT_EN_PRE_5	DI1_CNT_EN_PRE_3	DC_DP_START	IDMAC_NFACK_27
5	DI0_CNT_EN_PRE_4	DI1_DISP_CLK_EN_PRE	DP_ASF_BRAKE	IDMAC_NFACK_24
6	DI0_CNT_EN_PRE_3	DI_VSYNC_PRE_1	DP_SF_BRAKE	IDMAC_NFACK_23
7	DI0_CNT_EN_PRE_2	DMA_CH_CUR_BUF_52	DP_ASF_END	IDMAC_NFACK_51
8	DI0_CNT_EN_PRE_1	DI1_SRM_STAT	DP_ASF_START	DMA_CH_CUR_BUF_31

Table continues on the next page...

Table 45-42. IPU diag bus groups 4-7 (continued)

	group4	group5	group6	group7
9	DI0_DISP_CLK_EN_PRE	DMA_CH_ALT_BUF0_RDY_52	DP_SF_END	DMA_CH_CUR_BUF_28
10	DI_VSYNC_PRE_0	DMA_CH_ALT_BUF1_RDY_52	DP_SF_START	DMA_CH_CUR_BUF_27
11	DMA_CH_ALT_CUR_BUF_24	DI1_CNTR_FIFO_FULL	DP_A1_SRM_STAT	DMA_CH_CUR_BUF_24
12	DI0_SRM_STAT	DI1_CNTR_FIFO_EMPTY	DP_A0_SRM_STAT	DMA_CH_CUR_BUF_23
13	DMA_CH_ALT_BUF0_RDY_24	DMFC_FIFO_EMPTY_5	DP_S_SRM_STAT	DMA_CH_CUR_BUF_51
14	DI0_CNTR_FIFO_FULL	DMFC_FIFO_EMPTY_0	DP_ASYNC_CUR_FLOW	DMA_CH_CUR_BUF_47
15	DI0_CNTR_FIFO_EMPTY	DMFC_FIFO_FULL_5	Reserved	DMA_CH_ALT_BUF1_RDY_24

Table 45-43. IPU diag bus groups 8-11

	group8	group9	group10	group11
0	IDMAC_EOF_3	IDMAC_EOF_29	IDMAC_EOF_18	IDMAC_EOF_15
1	IDMAC_EOF_2	IDMAC_EOF_6	IDMAC_EOF_17	IDMAC_EOF_14
2	IDMAC_EOF_1	IDMAC_EOF_4	IDMAC_EOF_7	IDMAC_EOF_12
3	IDMAC_EOF_0	IDMAC_NFACK_29	IDMAC_NFACK_18	IDMAC_EOF_11
4	IDMAC_NFACK_3	DMA_CH_CUR_BUF_29	IDMAC_NFACK_17	IDMAC_NFACK_15
5	IDMAC_NFACK_2	DMA_CH_CUR_BUF_6	IDMAC_NFACK_7	IDMAC_NFACK_14
6	IDMAC_NFACK_1	DMA_CH_CUR_BUF_4	DI0_CNT_EN_PRE_7	IDMAC_NFACK_12
7	IDMAC_NFACK_0	DMA_CH_ALT_CUR_BUF_29	DMA_CH_CUR_BUF_18	IDMAC_NFACK_11
8	DMA_CH_CUR_BUF_3	DMA_CH_ALT_CUR_BUF_6	DMA_CH_CUR_BUF_17	DMA_CH_CUR_BUF_15
9	DMA_CH_CUR_BUF_2	DMA_CH_ALT_CUR_BUF_4	DMA_CH_CUR_BUF_7	DMA_CH_CUR_BUF_14
10	DMA_CH_CUR_BUF_1	DMA_CH_ALT_BUF0_RDY_29	DMA_CH_ALT_CUR_BUF_7	DMA_CH_CUR_BUF_12
11	DMA_CH_CUR_BUF_0	DMA_CH_ALT_BUF0_RDY_6	DMA_CH_ALT_BUF0_RDY_7	DMA_CH_CUR_BUF_11
12	DMFC_FIFO_EMPTY_3	DMA_CH_ALT_BUF0_RDY_4	DMA_CH_ALT_BUF1_RDY_7	DMFC_FIFO_EMPTY_11
13	DMFC_FIFO_EMPTY_2	DMA_CH_ALT_BUF1_RDY_29	DMFC_FIFO_EMPTY_7	DMFC_FIFO_EMPTY_10
14	DMFC_FIFO_FULL_3	DMA_CH_ALT_BUF1_RDY_6	DMFC_FIFO_FULL_7	DMFC_FIFO_FULL_11
15	DMFC_FIFO_FULL_2	DMA_CH_ALT_BUF1_RDY_4	DMFC_FIFO_FULL_0	DMFC_FIFO_FULL_10

Table 45-44. IPU diag bus groups 12-15

	group12	group13	group14	group15
0	IDMAC_EOF_50	IDMAC_EOF_44	IDMAC_EOF_22	IC_VF_BUF_OVF
1	IDMAC_EOF_49	IDMAC_EOF_43	IDMAC_EOF_21	IC_ENC_BUF_OVF
2	IDMAC_EOF_48	IDMAC_EOF_42	IDMAC_EOF_20	IC_BAYER_BUF_OVF
3	IDMAC_EOF_47	IDMAC_EOF_41	IDMAC_EOF_33	IC_BAYER_FRM_LOST_ERR
4	IDMAC_EOF_46	IDMAC_NFACK_44	IDMAC_NFACK_22	IC_ENC_FRM_LOST_ERR
5	IDMAC_EOF_45	IDMAC_NFACK_43	IDMAC_NFACK_21	IC_VF_FRM_LOST_ERR
6	IDMAC_NFACK_50	IDMAC_NFACK_42	IDMAC_NFACK_20	DMA_CH_CUR_BUF_50
7	IDMAC_NFACK_49	IDMAC_NFACK_41	IDMAC_NFACK_33	IDMAC_NFACK_8
8	IDMAC_NFACK_48	DMA_CH_CUR_BUF_44	DI0_SYNC_DISP_ERR	IDMAC_NFACK_9
9	IDMAC_NFACK_47	DMA_CH_CUR_BUF_43	DMA_CH_CUR_BUF_22	IDMAC_NFACK_10
10	IDMAC_NFACK_46	DMA_CH_CUR_BUF_42	DMA_CH_CUR_BUF_21	IDMAC_NFACK_13
11	IDMAC_NFACK_45	DMA_CH_CUR_BUF_41	DMA_CH_CUR_BUF_20	Reserved
12	DMA_CH_CUR_BUF_49	DMA_CH_ALT_CUR_BUF_41	DMA_CH_CUR_BUF_33	Reserved
13	DMA_CH_CUR_BUF_48	DMA_CH_ALT_BUF0_RDY_41	DMA_CH_ALT_CUR_BUF_33	VDI_FIFO1_OVF
14	DMA_CH_CUR_BUF_46	DMA_CH_ALT_BUF1_RDY_41	DMA_CH_ALT_BUF0_RDY_33	DMFC_IC_BUFFER_EMPTY
15	DMA_CH_CUR_BUF_45	DMFC_FIFO_FULL_1	DMA_CH_ALT_BUF1_RDY_33	DMFC_IC_BUFFER_FULL

Table 45-45. IPU diag bus group 16

	group16
0	CSI2MEM_SMFC3_TSTAT
1	
2	CSI2MEM_SMFC2_TSTAT
3	
4	CSI2MEM_SMFC1_TSTAT
5	
6	CSI2MEM_SMFC0_TSTAT
7	
8	DC_ASYNC1_TSTAT
9	
10	

Table continues on the next page...

Table 45-45. IPU diag bus group 16 (continued)

	group16
11	DC_ASYNC0_TSTAT
12	
13	DP_ASYNC_TSTAT
14	
15	

Table 45-46. IPU diag bus group 17

	group17
0	MEM2PRP_TSTAT
1	
2	
3	PP_ROT_TSTAT
4	
5	VF_ROT_TSTAT
6	
7	ENC_ROT_TSTAT
8	
9	PP_TSTAT
10	
11	VF_TSTAT
12	
13	ENC_TSTAT
14	
15	Reserved

45.5 Programmable Registers

45.5.1 IPU Memory Map/Register Definition

The address space for accesses through the AHB-lite slave port is 128 MB and it is split internally (with 32MB resolution) according to bits[28:25] of the address. Using the following notation

Address = (IPU_ID[31:29], MSB[28:25], LSB[24:0])

the address is used as follows ("T" is a configurable integer between12and 13 by configuring the MCU_T parameter):

1. MSB<T: access to an external device, with address = (MSB, LSB)
2. T<=MSB<14: access to an external device, with address (MSB-T, LSB)
3. MSB=14: Low-level access to an external device, with LSB[3:0] = (Lock, CS, RS[1:0])

- LSB[5:4] = RS[1:0] (the address on the display interface)
- LSB[6] = Choice of display's channel (0=channel 8, 1=channel 9)
- LSB[7] = Lock (Lock=1 prevents the use of the display port until the next ARM platform access)

1. MSB=15: access to internal IPU registers, with address LSB

The display port has 2 channels dedicated for ARM platform accesses via the AHB bus. Channel 8 is dedicated to the accesses where MSB<T. Channel 9 is dedicated to the accesses where MSB>T. When only a single channel is used, channel 8 should be used. The usage of channel 9 requires the usage of channel 8 as well.

The table below shows the register file of the block (addresses 0x1E000000 to 0x1FFFFFFF) and how the memory space allocation for each sub-block of the IPU.

NOTE

The addresses given in the table are relative to the IPU base address defined at SoC's level.

IPU memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E00_0000	Configuration Register (IPU_CONF)	32	R/W	0000_0000h	45.51.1/ 2855
1E00_0004	SISG Control 0 Register (IPU_SISG_CTRL0)	32	R/W	0000_0000h	45.51.2/ 2858
1E00_0008	SISG Control 1 Register (IPU_SISG_CTRL1)	32	R/W	0000_0000h	45.51.3/ 2859
1E00_000C	SISG Set<i> Register (IPU_SISG_SET_i)	32	R/W	0000_0000h	45.51.4/ 2859
1E00_0024	SISG Clear <i> Register (IPU_SISG_CLR_i)	32	R/W	0000_0000h	45.51.5/ 2860
1E00_003C	Interrupt Control Register 1 (IPU_INT_CTRL_1)	32	R/W	0000_0000h	45.51.6/ 2860

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E00_0040	Interrupt Control Register 2 (IPU_INT_CTRL_2)	32	R/W	0000_0000h	45.51.7/ 2864
1E00_0044	Interrupt Control Register 3 (IPU_INT_CTRL_3)	32	R/W	0000_0000h	45.51.8/ 2867
1E00_0048	Interrupt Control Register 4 (IPU_INT_CTRL_4)	32	R/W	0000_0000h	45.51.9/ 2871
1E00_004C	Interrupt Control Register 5 (IPU_INT_CTRL_5)	32	R/W	0000_0000h	45.51.10/ 2873
1E00_0050	Interrupt Control Register 6 (IPU_INT_CTRL_6)	32	R/W	0000_0000h	45.51.11/ 2878
1E00_0054	Interrupt Control Register 7 (IPU_INT_CTRL_7)	32	R/W	0000_0000h	45.51.12/ 2881
1E00_0058	Interrupt Control Register 8 (IPU_INT_CTRL_8)	32	R/W	0000_0000h	45.51.13/ 2883
1E00_005C	Interrupt Control Register 9 (IPU_INT_CTRL_9)	32	R/W	0000_0000h	45.51.14/ 2885
1E00_0060	Interrupt Control Register 10 (IPU_INT_CTRL_10)	32	R/W	0000_0000h	45.51.15/ 2886
1E00_0064	Interrupt Control Register 11 (IPU_INT_CTRL_11)	32	R/W	0000_0000h	45.51.16/ 2889
1E00_0068	Interrupt Control Register 12 (IPU_INT_CTRL_12)	32	R/W	0000_0000h	45.51.17/ 2891
1E00_006C	Interrupt Control Register 13 (IPU_INT_CTRL_13)	32	R/W	0000_0000h	45.51.18/ 2893
1E00_0070	Interrupt Control Register 14 (IPU_INT_CTRL_14)	32	R/W	0000_0000h	45.51.19/ 2897
1E00_0074	Interrupt Control Register15 (IPU_INT_CTRL_15)	32	R/W	0000_0000h	45.51.20/ 2900
1E00_0078	SDMA Event Control Register 1 (IPU_SDMA_EVENT_1)	32	R/W	0000_0000h	45.51.21/ 2904
1E00_007C	SDMA Event Control Register 2 (IPU_SDMA_EVENT_2)	32	R/W	0000_0000h	45.51.22/ 2908
1E00_0080	SDMA Event Control Register 3 (IPU_SDMA_EVENT_3)	32	R/W	0000_0000h	45.51.23/ 2910
1E00_0084	SDMA Event Control Register 4 (IPU_SDMA_EVENT_4)	32	R/W	0000_0000h	45.51.24/ 2915
1E00_0088	SDMA Event Control Register 7 (IPU_SDMA_EVENT_7)	32	R/W	0000_0000h	45.51.25/ 2918
1E00_008C	SDMA Event Control Register 8 (IPU_SDMA_EVENT_8)	32	R/W	0000_0000h	45.51.26/ 2920

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E00_0090	SDMA Event Control Register 11 (IPU_SDMA_EVENT_11)	32	R/W	0000_0000h	45.51.27/ 2921
1E00_0094	SDMA Event Control Register 12 (IPU_SDMA_EVENT_12)	32	R/W	0000_0000h	45.51.28/ 2924
1E00_0098	SDMA Event Control Register 13 (IPU_SDMA_EVENT_13)	32	R/W	0000_0000h	45.51.29/ 2926
1E00_009C	SDMA Event Control Register 14 (IPU_SDMA_EVENT_14)	32	R/W	0000_0000h	45.51.30/ 2930
1E00_00A0	Shadow Registers Memory Priority 1 Register (IPU_SRM_PRI1)	32	R/W	0000_0100h	45.51.31/ 2932
1E00_00A4	Shadow Registers Memory Priority 2 Register (IPU_SRM_PRI2)	32	R/W	0605_0803h	45.51.32/ 2933
1E00_00A8	FSU Processing Flow 1 Register (IPU_FS_PROC_FLOW1)	32	R/W	0000_0000h	45.51.33/ 2935
1E00_00AC	FSU Processing Flow 2 Register (IPU_FS_PROC_FLOW2)	32	R/W	0000_0000h	45.51.34/ 2939
1E00_00B0	FSU Processing Flow 3 Register (IPU_FS_PROC_FLOW3)	32	R/W	0000_0000h	45.51.35/ 2942
1E00_00B4	FSU Displaying Flow 1 Register (IPU_FS_DISP_FLOW1)	32	R/W	0000_0000h	45.51.36/ 2944
1E00_00B8	FSU Displaying Flow 2 Register (IPU_FS_DISP_FLOW2)	32	R/W	0000_0000h	45.51.37/ 2948
1E00_00BC	SKIP Register (IPU_SKIP)	32	R/W	0000_0000h	45.51.38/ 2950
1E00_00C4	Display General Control Register (IPU_DISP_GEN)	32	R/W	0040_0000h	45.51.39/ 2951
1E00_00C8	Display Alternate Flow Control Register 1 (IPU_DISP_ALT1)	32	R/W	0040_0000h	45.51.40/ 2954
1E00_00CC	Display Alternate Flow Control Register 2 (IPU_DISP_ALT2)	32	R/W	0000_0000h	45.51.41/ 2955
1E00_00D0	Display Alternate Flow Control Register 3 (IPU_DISP_ALT3)	32	R/W	0040_0000h	45.51.42/ 2956
1E00_00D4	Display Alternate Flow Control Register 4 (IPU_DISP_ALT4)	32	R/W	0000_0000h	45.51.43/ 2958
1E00_00D8	Autorefresh and Snooping Control Register (IPU_SNOOP)	32	R/W	0000_0000h	45.51.44/ 2959
1E00_00DC	Memory Reset Control Register (IPU_MEM_RST)	32	R/W	0000_0000h	45.51.45/ 2960
1E00_00E0	Power Modes Control Register (IPU_PM)	32	R/W	0810_0810h	45.51.46/ 2961

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E00_00E4	General Purpose Register (IPU_GPR)	32	R/W	0000_0000h	45.51.47/ 2964
1E00_0150	Channel Double Buffer Mode Select 0 Register (IPU_CH_DB_MODE_SEL0)	32	R/W	0000_0000h	45.51.48/ 2966
1E00_0154	Channel Double Buffer Mode Select 1 Register (IPU_CH_DB_MODE_SEL1)	32	R/W	0000_0000h	45.51.49/ 2970
1E00_0168	Alternate Channel Double Buffer Mode Select 0 Register (IPU_ALT_CH_DB_MODE_SEL0)	32	R/W	0000_0000h	45.51.50/ 2972
1E00_016C	Alternate Channel Double Buffer Mode Select1 Register (IPU_ALT_CH_DB_MODE_SEL1)	32	R/W	0000_0000h	45.51.51/ 2974
1E00_0178	Alternate Channel Triple Buffer Mode Select 0 Register (IPU_ALT_CH_TRB_MODE_SEL0)	32	R/W	0000_0000h	45.51.52/ 2975
1E00_0200	Interrupt Status Register 1 (IPU_INT_STAT_1)	32	w1c	0000_0000h	45.51.53/ 2978
1E00_0204	Interrupt Status Register2 (IPU_INT_STAT_2)	32	w1c	0000_0000h	45.51.54/ 2982
1E00_0208	Interrupt Status Register 3 (IPU_INT_STAT_3)	32	w1c	0000_0000h	45.51.55/ 2985
1E00_0210	Interrupt Status Register 5 (IPU_INT_STAT_5)	32	w1c	0000_0000h	45.51.56/ 2989
1E00_0214	Interrupt Status Register 6 (IPU_INT_STAT_6)	32	w1c	0000_0000h	45.51.57/ 2994
1E00_0218	Interrupt Status Register7 1 (IPU_INT_STAT_7)	32	w1c	0000_0000h	45.51.58/ 2997
1E00_021C	Interrupt Status Register 8 (IPU_INT_STAT_8)	32	w1c	0000_0000h	45.51.59/ 2999
1E00_0220	Interrupt Status Register 9 (IPU_INT_STAT_9)	32	w1c	0000_0000h	45.51.60/ 3001
1E00_0224	Interrupt Status Register 10 (IPU_INT_STAT_10)	32	w1c	0000_0000h	45.51.61/ 3003
1E00_0228	Interrupt Status Register 11 (IPU_INT_STAT_11)	32	w1c	0000_0000h	45.51.62/ 3005
1E00_022C	Interrupt Status Register 12 (IPU_INT_STAT_12)	32	w1c	0000_0000h	45.51.63/ 3009
1E00_0230	Interrupt Status Register 13 (IPU_INT_STAT_13)	32	w1c	0000_0000h	45.51.64/ 3011
1E00_0234	Interrupt Status Register 14 (IPU_INT_STAT_14)	32	w1c	0000_0000h	45.51.65/ 3016
1E00_0238	Interrupt Status Register 15 (IPU_INT_STAT_15)	32	w1c	0000_0000h	45.51.66/ 3019

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E00_023C	Current Buffer Register 0 (IPU_CUR_BUF_0)	32	R	0000_0000h	45.51.67/ 3023
1E00_0240	Current Buffer Register 1 (IPU_CUR_BUF_1)	32	R	0000_0000h	45.51.68/ 3027
1E00_0244	Alternate Current Buffer Register 0 (IPU_ALT_CUR_0)	32	R	0000_0000h	45.51.69/ 3029
1E00_0248	Alternate Current Buffer Register 1 (IPU_ALT_CUR_1)	32	R	0000_0000h	45.51.70/ 3031
1E00_024C	Shadow Registers Memory Status Register (IPU_SRM_STAT)	32	R	0000_0000h	45.51.71/ 3033
1E00_0250	Processing Status Tasks Register (IPU_PROC_TASKS_STAT)	32	R	0000_0000h	45.51.72/ 3035
1E00_0254	Display Tasks Status Register (IPU_DISP_TASKS_STAT)	32	R	0000_0000h	45.51.73/ 3036
1E00_0258	Triple Current Buffer Register 0 (IPU_TRIPLE_CUR_BUF_0)	32	R	0000_0000h	45.51.74/ 3037
1E00_025C	Triple Current Buffer Register 1 (IPU_TRIPLE_CUR_BUF_1)	32	R	0000_0000h	45.51.75/ 3039
1E00_0268	IPU Channels Buffer 0 Ready 0 Register (IPU_CH_BUF0_RDY0)	32	R/W	0000_0000h	45.51.76/ 3041
1E00_026C	IPU Channels Buffer 0 Ready 1 Register (IPU_CH_BUF0_RDY1)	32	R/W	0000_0000h	45.51.77/ 3044
1E00_0270	IPU Channels Buffer 1 Ready 0 Register (IPU_CH_BUF1_RDY0)	32	R/W	0000_0000h	45.51.78/ 3046
1E00_0274	IPU Channels Buffer 1 Ready 1Register (IPU_CH_BUF1_RDY1)	32	R/W	0000_0000h	45.51.79/ 3049
1E00_0278	IPU Alternate Channels Buffer 0 Ready 0 Register (IPU_ALT_CH_BUF0_RDY0)	32	R/W	0000_0000h	45.51.80/ 3052
1E00_027C	IPU Alternate Channels Buffer 0 Ready 1 Register (IPU_ALT_CH_BUF0_RDY1)	32	R/W	0000_0000h	45.51.81/ 3053
1E00_0280	IPU Alternate Channels Buffer1 Ready 0 Register (IPU_ALT_CH_BUF1_RDY0)	32	R/W	0000_0000h	45.51.82/ 3054
1E00_0284	IPU Alternate Channels Buffer 1 Ready 1 Register (IPU_ALT_CH_BUF1_RDY1)	32	R/W	0000_0000h	45.51.83/ 3055
1E00_0288	IPU Channels Buffer 2 Ready 0 Register (IPU_CH_BUF2_RDY0)	32	R/W	0000_0000h	45.51.84/ 3056
1E00_028C	IPU Channels Buffer 2 Ready 1 Register (IPU_CH_BUF2_RDY1)	32	R/W	0000_0000h	45.51.85/ 3058
1E00_02C0	Interrupt Status Register 4 (IPU_INT_STAT_4)	32	w1c	0000_0000h	45.51.86/ 3059

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E00_8000	IDMAC Configuration Register (IPU_IDMAC_CONF)	32	R/W	0000_002Fh	45.51.87/ 3062
1E00_8004	IDMAC Channel Enable 1 Register (IPU_IDMAC_CH_EN_1)	32	R/W	0000_0000h	45.51.88/ 3063
1E00_800C	IDMAC Separate Alpha Indication Register (IPU_IDMAC_SEP_ALPHA)	32	R/W	0000_0000h	45.51.89/ 3067
1E00_8010	IDMAC Alternate Separate Alpha Indication Register (IPU_IDMAC_ALT_SEP_ALPHA)	32	R/W	0000_0000h	45.51.90/ 3069
1E00_8014	IDMAC Channel Priority 1 Register (IPU_IDMAC_CH_PRI_1)	32	R/W	0000_0000h	45.51.91/ 3071
1E00_8018	IDMAC Channel Priority 2 Register (IPU_IDMAC_CH_PRI_2)	32	R/W	0000_0000h	45.51.92/ 3074
1E00_801C	IDMAC Channel Watermark Enable 1 Register (IPU_IDMAC_WM_EN_1)	32	R/W	0000_0000h	45.51.93/ 3076
1E00_8020	IDMAC Channel Watermark Enable 2 Register (IPU_IDMAC_WM_EN_2)	32	R/W	0000_0000h	45.51.94/ 3078
1E00_8024	IDMAC Channel Lock Enable 1Register (IPU_IDMAC_LOCK_EN_1)	32	R/W	0000_0000h	45.51.95/ 3079
1E00_804C	IDMAC Scroll Coordinations Register 1 (IPU_IDMAC_SC_CORD_1)	32	R/W	0000_0000h	45.51.96/ 3081
1E00_8100	IDMAC Channel Busy 1 Register (IPU_IDMAC_CH_BUSY_1)	32	R	0000_0000h	45.51.97/ 3082
1E00_8104	IDMAC Channel Busy 2 Register (IPU_IDMAC_CH_BUSY_2)	32	R	0000_0000h	45.51.98/ 3087
1E01_80BC	DP Debug Control Register (IPU_DP_DEBUG_CNT)	32	R/W	0000_0000h	45.51.99/ 3090
1E01_80C0	DP Debug Status Register (IPU_DP_DEBUG_STAT)	32	R	0000_0000h	45.51.100/ 3091
1E02_0000	IC Configuration Register (IPU_IC_CONF)	32	R/W	0000_0000h	45.51.101/ 3092
1E02_0004	IC Preprocessing Encoder Resizing Coefficients Register (IPU_IC_PRP_ENC_RSC)	32	R/W	2000_2000h	45.51.102/ 3095
1E02_0008	IC Preprocessing View-Finder Resizing Coefficients Register (IPU_IC_PRP_VF_RSC)	32	R/W	2000_2000h	45.51.103/ 3096
1E02_000C	IC Postprocessing Encoder Resizing Coefficients Register (IPU_IC_PP_RSC)	32	R/W	2000_2000h	45.51.104/ 3097
1E02_0010	IC Combining Parameters Register 1 (IPU_IC_CMBP_1)	32	R/W	0000_0000h	45.51.105/ 3098
1E02_0014	IC Combining Parameters Register 2 (IPU_IC_CMBP_2)	32	R/W	0000_0000h	45.51.106/ 3098

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E02_0018	IC IDMAC Parameters 1 Register (IPU_IC_IDMAC_1)	32	R/W	0000_0000h	45.51.107/ 3099
1E02_001C	IC IDMAC Parameters 2 Register (IPU_IC_IDMAC_2)	32	R/W	0000_0000h	45.51.108/ 3102
1E02_0020	IC IDMAC Parameters 3Register (IPU_IC_IDMAC_3)	32	R/W	0000_0000h	45.51.109/ 3103
1E02_0024	IC IDMAC Parameters 4 Register (IPU_IC_IDMAC_4)	32	R/W	0000_0000h	45.51.110/ 3103
1E03_0000	CSI0 Sensor Configuration Register (IPU_CSI0_SENS_CONF)	32	R/W	0000_0000h	45.51.111/ 3104
1E03_0004	CSI0 Sense Frame Size Register (IPU_CSI0_SENS_FRM_SIZE)	32	R/W	0000_0000h	45.51.112/ 3107
1E03_0008	CSI0 Actual Frame Size Register (IPU_CSI0_ACT_FRM_SIZE)	32	R/W	0000_0000h	45.51.113/ 3107
1E03_000C	CSI0 Output Control Register (IPU_CSI0_OUT_FRM_CTRL)	32	R/W	0000_0000h	45.51.114/ 3108
1E03_0010	CSIO Test Control Register (IPU_CSI0_TST_CTRL)	32	R/W	0000_0000h	45.51.115/ 3109
1E03_0014	CSIO CCIR Code Register 1 (IPU_CSI0_CCIR_CODE_1)	32	R/W	0000_0000h	45.51.116/ 3110
1E03_0018	CSIO CCIR Code Register 2 (IPU_CSI0_CCIR_CODE_2)	32	R/W	0000_0000h	45.51.117/ 3111
1E03_001C	CSIO CCIR Code Register 3 (IPU_CSI0_CCIR_CODE_3)	32	R/W	0000_0000h	45.51.118/ 3112
1E03_0020	CSI0 Data Identifier Register (IPU_CSI0_DI)	32	R/W	FFFF_ FFFFh	45.51.119/ 3112
1E03_0024	CSI0 SKIP Register (IPU_CSI0_SKIP)	32	R/W	0000_0000h	45.51.120/ 3113
1E03_0028	CSI0 Compander Control Register (IPU_CSIO_CPD_CTRL)	32	R/W	0000_0000h	45.51.121/ 3114
1E03_002C	CSI0 Red Component Compander Constants Register <i> (IPU_CSIO_CPD_RC_i)	32	R/W	0000_0000h	45.51.122/ 3115
1E03_004C	CSI0 Red Component Compander SLOPE Register <i> (IPU_CSIO_CPD_RS_i)	32	R/W	0000_0000h	45.51.123/ 3116
1E03_005C	CSI0 GR Component Compander Constants Register <i> (IPU_CSIO_CPD_GRC_i)	32	R/W	0000_0000h	45.51.124/ 3116
1E03_007C	CSI0 GR Component Compander SLOPE Register <i> (IPU_CSIO_CPD_GRS_i)	32	R/W	0000_0000h	45.51.125/ 3117
1E03_008C	CSI0 GB Component Compander Constants Register <i> (IPU_CSIO_CPD_GBC_i)	32	R/W	0000_0000h	45.51.126/ 3118

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E03_00AC	CSI0 GB Component Compander SLOPE Register <i> (IPU_CSIO_CPD_GBS_i)	32	R/W	0000_0000h	45.51.127/ 3118
1E03_00BC	CSI0 Blue Component Compander Constants Register <i> (IPU_CSIO_CPD_BC_i)	32	R/W	0000_0000h	45.51.128/ 3119
1E03_00DC	CSI0 Blue Component Compander SLOPE Register <i> (IPU_CSIO_CPD_BS_i)	32	R/W	0000_0000h	45.51.129/ 3120
1E03_00EC	CSI0 Compander Offset Register 1 (IPU_CSI0_CPD_OFFSET1)	32	R/W	0000_0000h	45.51.130/ 3120
1E03_00F0	CSI0 Compander Offset Register 2 (IPU_CSI0_CPD_OFFSET2)	32	R/W	0000_0000h	45.51.131/ 3121
1E03_8000	CSI1 Sensor Configuration Register (IPU_CSI1_SENS_CONF)	32	R/W	0000_0000h	45.51.132/ 3122
1E03_8004	CSI1 Sense Frame Size Register (IPU_CSI1_SENS_FRM_SIZE)	32	R/W	0000_0000h	45.51.133/ 3124
1E03_8008	CSI1 Actual Frame Size Register (IPU_CSI1_ACT_FRM_SIZE)	32	R/W	0000_0000h	45.51.134/ 3125
1E03_800C	CSI1 Output Control Register (IPU_CSI1_OUT_FRM_CTRL)	32	R/W	0000_0000h	45.51.135/ 3126
1E03_8010	CSI1 Test Control Register (IPU_CSI1_TST_CTRL)	32	R/W	0000_0000h	45.51.136/ 3127
1E03_8014	CSI1 CCIR Code Register 1 (IPU_CSI1_CCIR_CODE_1)	32	R/W	0000_0000h	45.51.137/ 3128
1E03_8018	CSI1 CCIR Code Register 2 (IPU_CSI1_CCIR_CODE_2)	32	R/W	0000_0000h	45.51.138/ 3129
1E03_801C	CSI1 CCIR Code Register 3 (IPU_CSI1_CCIR_CODE_3)	32	R/W	0000_0000h	45.51.139/ 3130
1E03_8020	CSI1 Data Identifier Register (IPU_CSI1_DI)	32	R/W	FFFF_ FFFFh	45.51.140/ 3130
1E03_8024	CSI1 SKIP Register (IPU_CSI1_SKIP)	32	R/W	0000_0000h	45.51.141/ 3131
1E03_8028	CSI1 Compander Control Register (IPU_CSI1_CPD_CTRL)	32	R/W	0000_0000h	45.51.142/ 3132
1E03_802C	CSI1 Red Component Compander Constants Register <i> (IPU_CSI1_CPD_RC_i)	32	R/W	0000_0000h	45.51.143/ 3133
1E03_804C	CSI1 Red Component Compander SLOPE Register <i> (IPU_CSI1_CPD_RS_i)	32	R/W	0000_0000h	45.51.144/ 3133
1E03_805C	CSI1 GR Component Compander Constants Register <i> (IPU_CSI1_CPD_GRC_i)	32	R/W	0000_0000h	45.51.145/ 3134
1E03_807C	CSI1 GR Component Compander SLOPE Register <i> (IPU_CSI1_CPD_GRS_i)	32	R/W	0000_0000h	45.51.146/ 3135

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E03_808C	CSI1 GB Component Compander Constants Register <i> (IPU_CSI1_CPD_GBC_i)	32	R/W	0000_0000h	45.51.147/ 3135
1E03_80AC	CSI1 GB Component Compander SLOPE Register <i> (IPU_CSI1_CPD_GBS_i)	32	R/W	0000_0000h	45.51.148/ 3136
1E03_80BC	CSI1 Blue Component Compander Constants Register <i> (IPU_CSI1_CPD_BC_i)	32	R/W	0000_0000h	45.51.149/ 3137
1E03_80DC	CSI1 Blue Component Compander SLOPE Register <i> (IPU_CSI1_CPD_BS_i)	32	R/W	0000_0000h	45.51.150/ 3137
1E03_80EC	CSI1 Compander Offset Register 1 (IPU_CSI1_CPD_OFFSET1)	32	R/W	0000_0000h	45.51.151/ 3138
1E03_80F0	CSI1 Compander Offset Register 2 (IPU_CSI1_CPD_OFFSET2)	32	R/W	0000_0000h	45.51.152/ 3139
1E04_0000	DI0 General Register (IPU_DI0_GENERAL)	32	R/W	0020_0000h	45.51.153/ 3140
1E04_0004	DI0 Base Sync Clock Gen 0 Register (IPU_DI0_BS_CLKGEN0)	32	R/W	0000_0000h	45.51.154/ 3142
1E04_0008	DI0 Base Sync Clock Gen 1 Register (IPU_DI0_BS_CLKGEN1)	32	R/W	0000_0000h	45.51.155/ 3143
1E04_000C	DI0 Sync Wave Gen 1 Register 0 (IPU_DI0_SW_GEN0_1)	32	R/W	0000_0000h	45.51.156/ 3143
1E04_0010	DI0 Sync Wave Gen 2 Register 0 (IPU_DI0_SW_GEN0_2)	32	R/W	0000_0000h	45.51.157/ 3145
1E04_0014	DI0 Sync Wave Gen 3 Register 0 (IPU_DI0_SW_GEN0_3)	32	R/W	0000_0000h	45.51.158/ 3146
1E04_0018	DI0 Sync Wave Gen 4 Register 0 (IPU_DI0_SW_GEN0_4)	32	R/W	0000_0000h	45.51.159/ 3147
1E04_001C	DI0 Sync Wave Gen 5 Register 0 (IPU_DI0_SW_GEN0_5)	32	R/W	0000_0000h	45.51.160/ 3149
1E04_0020	DI0 Sync Wave Gen 6 Register 0 (IPU_DI0_SW_GEN0_6)	32	R/W	0000_0000h	45.51.161/ 3150
1E04_0024	DI0 Sync Wave Gen 7 Register 0 (IPU_DI0_SW_GEN0_7)	32	R/W	0000_0000h	45.51.162/ 3151
1E04_0028	DI0 Sync Wave Gen 8 Register 0 (IPU_DI0_SW_GEN0_8)	32	R/W	0000_0000h	45.51.163/ 3152
1E04_002C	DI0 Sync Wave Gen 9 Register 0 (IPU_DI0_SW_GEN0_9)	32	R/W	0000_0000h	45.51.164/ 3154
1E04_0030	DI0 Sync Wave Gen 1 Register 1 (IPU_DI0_SW_GEN1_1)	32	R/W	0000_0000h	45.51.165/ 3155
1E04_0034	DI0 Sync Wave Gen 2 Register 1 (IPU_DI0_SW_GEN1_2)	32	R/W	0000_0000h	45.51.166/ 3157

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E04_0038	DI0 Sync Wave Gen 3 Register 1 (IPU_DI0_SW_GEN1_3)	32	R/W	0000_0000h	45.51.167/ 3159
1E04_003C	DI0 Sync Wave Gen 4 Register 1 (IPU_DI0_SW_GEN1_4)	32	R/W	0000_0000h	45.51.168/ 3161
1E04_0040	DI0 Sync Wave Gen 5 Register 1 (IPU_DI0_SW_GEN1_5)	32	R/W	0000_0000h	45.51.169/ 3163
1E04_0044	DI0 Sync Wave Gen 6 Register 1 (IPU_DI0_SW_GEN1_6)	32	R/W	0000_0000h	45.51.170/ 3165
1E04_0048	DI0 Sync Wave Gen 7 Register 1 (IPU_DI0_SW_GEN1_7)	32	R/W	0000_0000h	45.51.171/ 3167
1E04_004C	DI0 Sync Wave Gen 8 Register 1 (IPU_DI0_SW_GEN1_8)	32	R/W	0000_0000h	45.51.172/ 3169
1E04_0050	DI0 Sync Wave Gen 9 Register 1 (IPU_DI0_SW_GEN1_9)	32	R/W	0000_0000h	45.51.173/ 3171
1E04_0054	DI0 Sync Assistance Gen Register (IPU_DI0_SYNC_AS_GEN)	32	R/W	0000_0000h	45.51.174/ 3172
1E04_0058	DI0 Data Wave Gen <i> Register (IPU_DI0_DW_GEN_i)	32	R/W	0000_0000h	45.51.175/ 3173
1E04_0088	DI0 Data Wave Set 0 <i> Register (IPU_DI0_DW_SET0_i)	32	R/W	0000_0000h	45.51.176/ 3176
1E04_00B8	DI0 Data Wave Set 1 <i> Register (IPU_DI0_DW_SET1_i)	32	R/W	0000_0000h	45.51.177/ 3176
1E04_00E8	DI0 Data Wave Set 2 <i> Register (IPU_DI0_DW_SET2_i)	32	R/W	0000_0000h	45.51.178/ 3177
1E04_0118	DI0 Data Wave Set 3 <i> Register (IPU_DI0_DW_SET3_i)	32	R/W	0000_0000h	45.51.179/ 3178
1E04_0148	DI0 Step Repeat <i> Registers (IPU_DI0_STP_REP_i)	32	R/W	0000_0000h	45.51.180/ 3178
1E04_0158	DI0 Step Repeat 9 Registers (IPU_DI0_STP_REP_9)	32	R/W	0000_0000h	45.51.181/ 3179
1E04_015C	DI0 Serial Display Control Register (IPU_DI0_SER_CONF)	32	R/W	0000_0000h	45.51.182/ 3180
1E04_0160	DI0 Special Signals Control Register (IPU_DI0_SSC)	32	R/W	0000_0000h	45.51.183/ 3182
1E04_0164	DI0 Polarity Register (IPU_DI0_POL)	32	R/W	0000_0000h	45.51.184/ 3184
1E04_0168	DI0 Active Window 0 Register (IPU_DI0_AW0)	32	R/W	0000_0000h	45.51.185/ 3186
1E04_016C	DI0 Active Window 1 Register (IPU_DI0_AW1)	32	R/W	0000_0000h	45.51.186/ 3186

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E04_0170	DI0 Screen Configuration Register (IPU_DI0_SCR_CONF)	32	R/W	0000_0000h	45.51.187/ 3187
1E04_0174	DI0 Status Register (IPU_DI0_STAT)	32	R	0000_0005h	45.51.188/ 3188
1E04_8000	DI1General Register (IPU_DI1_GENERAL)	32	R/W	0020_0000h	45.51.189/ 3189
1E04_8004	DI1 Base Sync Clock Gen 0 Register (IPU_DI1_BS_CLKGEN0)	32	R/W	0000_0000h	45.51.190/ 3191
1E04_8008	DI1 Base Sync Clock Gen 1 Register (IPU_DI1_BS_CLKGEN1)	32	R/W	0000_0000h	45.51.191/ 3192
1E04_800C	DI1 Sync Wave Gen 1 Register 0 (IPU_DI1_SW_GEN0_1)	32	R/W	0000_0000h	45.51.192/ 3192
1E04_8010	DI1 Sync Wave Gen 2 Register 0 (IPU_DI1_SW_GEN0_2)	32	R/W	0000_0000h	45.51.193/ 3194
1E04_8014	DI1 Sync Wave Gen 3 Register 0 (IPU_DI1_SW_GEN0_3)	32	R/W	0000_0000h	45.51.194/ 3195
1E04_8018	DI1 Sync Wave Gen 4 Register 0 (IPU_DI1_SW_GEN0_4)	32	R/W	0000_0000h	45.51.195/ 3196
1E04_801C	DI1 Sync Wave Gen 5 Register 0 (IPU_DI1_SW_GEN0_5)	32	R/W	0000_0000h	45.51.196/ 3198
1E04_8020	DI1 Sync Wave Gen 6 Register 0 (IPU_DI1_SW_GEN0_6)	32	R/W	0000_0000h	45.51.197/ 3199
1E04_8024	DI1 Sync Wave Gen 7 Register 0 (IPU_DI1_SW_GEN0_7)	32	R/W	0000_0000h	45.51.198/ 3200
1E04_8028	DI1 Sync Wave Gen 8 Register 0 (IPU_DI1_SW_GEN0_8)	32	R/W	0000_0000h	45.51.199/ 3201
1E04_802C	DI1Sync Wave Gen 9 Register 0 (IPU_DI1_SW_GEN0_9)	32	R/W	0000_0000h	45.51.200/ 3203
1E04_8030	DI1 Sync Wave Gen 1 Register 1 (IPU_DI1_SW_GEN1_1)	32	R/W	0000_0000h	45.51.201/ 3204
1E04_8034	DI1 Sync Wave Gen 2 Register 1 (IPU_DI1_SW_GEN1_2)	32	R/W	0000_0000h	45.51.202/ 3206
1E04_8038	DI1 Sync Wave Gen 3 Register 1 (IPU_DI1_SW_GEN1_3)	32	R/W	0000_0000h	45.51.203/ 3208
1E04_803C	DI1 Sync Wave Gen 4 Register 1 (IPU_DI1_SW_GEN1_4)	32	R/W	0000_0000h	45.51.204/ 3210
1E04_8040	DI1 Sync Wave Gen 5 Register 1 (IPU_DI1_SW_GEN1_5)	32	R/W	0000_0000h	45.51.205/ 3212
1E04_8044	DI1 Sync Wave Gen 6 Register 1 (IPU_DI1_SW_GEN1_6)	32	R/W	0000_0000h	45.51.206/ 3214

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E04_8048	DI1Sync Wave Gen 7 Register 1 (IPU_DI1_SW_GEN1_7)	32	R/W	0000_0000h	45.51.207/ 3216
1E04_804C	DI1 Sync Wave Gen 8 Register 1 (IPU_DI1_SW_GEN1_8)	32	R/W	0000_0000h	45.51.208/ 3218
1E04_8050	DI1 Sync Wave Gen 9 Register 1 (IPU_DI1_SW_GEN1_9)	32	R/W	0000_0000h	45.51.209/ 3220
1E04_8054	DI1 Sync Assistance Gen Register (IPU_DI1_SYNC_AS_GEN)	32	R/W	0000_0000h	45.51.210/ 3221
1E04_8058	DI1 Data Wave Gen <i> Register (IPU_DI1_DW_GEN_i)	32	R/W	0000_0000h	45.51.211/ 3222
1E04_8088	DI1 Data Wave Set 0 <i> Register (IPU_DI1_DW_SET0_i)	32	R/W	0000_0000h	45.51.212/ 3225
1E04_80B8	DI1 Data Wave Set 1 <i> Register (IPU_DI1_DW_SET1_i)	32	R/W	0000_0000h	45.51.213/ 3225
1E04_80E8	DI1 Data Wave Set 2 <i> Register (IPU_DI1_DW_SET2_i)	32	R/W	0000_0000h	45.51.214/ 3226
1E04_8118	DI1 Data Wave Set 3 <i> Register (IPU_DI1_DW_SET3_i)	32	R/W	0000_0000h	45.51.215/ 3227
1E04_8148	DI1 Step Repeat <i> Registers (IPU_D1_STP_REP_i)	32	R/W	0000_0000h	45.51.216/ 3227
1E04_8158	DI1Step Repeat 9 Registers (IPU_DI1_STP_REP_9)	32	R/W	0000_0000h	45.51.217/ 3228
1E04_815C	DI1 Serial Display Control Register (IPU_DI1_SER_CONF)	32	R/W	0000_0000h	45.51.218/ 3229
1E04_8160	DI1 Special Signals Control Register (IPU_DI1_SSC)	32	R/W	0000_0000h	45.51.219/ 3231
1E04_8164	DI1 Polarity Register (IPU_DI1_POL)	32	R/W	0000_0000h	45.51.220/ 3233
1E04_8168	DI1Active Window 0 Register (IPU_DI1_AW0)	32	R/W	0000_0000h	45.51.221/ 3235
1E04_816C	DI1 Active Window 1 Register (IPU_DI1_AW1)	32	R/W	0000_0000h	45.51.222/ 3235
1E04_8170	DI1 Screen Configuration Register (IPU_DI1_SCR_CONF)	32	R/W	0000_0000h	45.51.223/ 3236
1E04_8174	DI1 Status Register (IPU_DI1_STAT)	32	R	0000_0005h	45.51.224/ 3237
1E05_0000	SMFC Mapping Register (IPU_SMFC_MAP)	32	R/W	0000_0000h	45.51.225/ 3238
1E05_0004	SMFC Watermark Control Register (IPU_SMFC_WMC)	32	R/W	0000_09A6h	45.51.226/ 3239

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E05_0008	SMFC Burst Size Register (IPU_SMFC_BS)	32	R/W	0000_0000h	45.51.227/ 3240
1E05_8000	DC Read Channel Configuration Register (IPU_DC_READ_CH_CONF)	32	R/W	FFFF_0000h	45.51.228/ 3242
1E05_8004	DC Read Channel Start Address Register (IPU_DC_READ_SH_ADDR)	32	R/W	0000_0000h	45.51.229/ 3243
1E05_8008	DC Routine Link Register 0 Channel 0 (IPU_DC_RL0_CH_0)	32	R/W	0000_0000h	45.51.230/ 3244
1E05_800C	DC Routine Link Register 1 Channel 0 (IPU_DC_RL1_CH_0)	32	R/W	0000_0000h	45.51.231/ 3245
1E05_8014	DC Routine Link Registe3 Channel 0 (IPU_DC_RL3_CH_0)	32	R/W	0000_0000h	45.51.232/ 3246
1E05_8018	DC Routine Link Register 4 Channel 0 (IPU_DC_RL4_CH_0)	32	R/W	0000_0000h	45.51.233/ 3247
1E05_801C	DC Write Channel 1 Configuration Register (IPU_DC_WR_CH_CONF_1)	32	R/W	0000_0000h	45.51.234/ 3248
1E05_8020	DC Routine Link Register2 Channel 0 (IPU_DC_RL2_CH_0)	32	R/W	0000_0000h	45.51.235/ 3249
1E05_8020	DC Write Channel 1 Address Configuration Register (IPU_DC_WR_CH_ADDR_1)	32	R/W	0000_0000h	45.51.236/ 3250
1E05_8024	DC Routine Link Register 0 Channel 1 (IPU_DC_RL0_CH_1)	32	R/W	0000_0000h	45.51.237/ 3251
1E05_8028	DC Routine Link Register 1 Channel 1 (IPU_DC_RL1_CH_1)	32	R/W	0000_0000h	45.51.238/ 3252
1E05_8028	DC Routine Link Register 2 Channel 2 (IPU_DC_RL2_CH_2)	32	R/W	0000_0000h	45.51.239/ 3253
1E05_8030	DC Routine Link Register 2 Channel 1 (IPU_DC_RL2_CH_1)	32	R/W	0000_0000h	45.51.240/ 3254
1E05_8032	DC Routine Link Register 3 Channel 1 (IPU_DC_RL3_CH_1)	32	R/W	0000_0000h	45.51.241/ 3255
1E05_8034	DC Routine Link Register 4 Channel 1 (IPU_DC_RL4_CH_1)	32	R/W	0000_0000h	45.51.242/ 3256
1E05_8038	DC Write Channel 2 Configuration Register (IPU_DC_WR_CH_CONF_2)	32	R/W	0000_0000h	45.51.243/ 3257
1E05_803C	DC Write Channel 2 Address Configuration Register (IPU_DC_WR_CH_ADDR_2)	32	R/W	0000_0000h	45.51.244/ 3258
1E05_8040	DC Routine Link Register 0 Channel 2 (IPU_DC_RL0_CH_2)	32	R/W	0000_0000h	45.51.245/ 3259
1E05_8044	DC Routine Link Register 1 Channel 2 (IPU_DC_RL1_CH_2)	32	R/W	0000_0000h	45.51.246/ 3260

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E05_804C	DC Routine Link Register 3 Channel 2 (IPU_DC_RL3_CH_2)	32	R/W	0000_0000h	45.51.247/ 3261
1E05_8050	DC Routine Link Register 4 Channel 2 (IPU_DC_RL4_CH_2)	32	R/W	0000_0000h	45.51.248/ 3262
1E05_8054	DC Command Channel 3 Configuration Register (IPU_DC_CMD_CH_CONF_3)	32	R/W	0000_0000h	45.51.249/ 3263
1E05_8058	DC Command Channel 4 Configuration Register (IPU_DC_CMD_CH_CONF_4)	32	R/W	0000_0000h	45.51.250/ 3264
1E05_805C	DC Write Channel 5Configuration Register (IPU_DC_WR_CH_CONF_5)	32	R/W	0000_0000h	45.51.251/ 3265
1E05_8060	DC Write Channel 5Address Configuration Register (IPU_DC_WR_CH_ADDR_5)	32	R/W	0000_0000h	45.51.252/ 3266
1E05_8064	DC Routine Link Register 0 Channel 5 (IPU_DC_RL0_CH_5)	32	R/W	0000_0000h	45.51.253/ 3267
1E05_8068	DC Routine Link Register 1 Channel 5 (IPU_DC_RL1_CH_5)	32	R/W	0000_0000h	45.51.254/ 3268
1E05_806C	DC Routine Link Register 2 Channel 5 (IPU_DC_RL2_CH_5)	32	R/W	0000_0000h	45.51.255/ 3269
1E05_8070	DC Routine Link Register3 Channel 5 (IPU_DC_RL3_CH_5)	32	R/W	0000_0000h	45.51.256/ 3270
1E05_8074	DC Routine Link Register 4 Channel 5 (IPU_DC_RL4_CH_5)	32	R/W	0000_0000h	45.51.257/ 3271
1E05_8078	DC Write Channel 6 Configuration Register (IPU_DC_WR_CH_CONF_6)	32	R/W	0000_0000h	45.51.258/ 3272
1E05_807C	DC Write Channel 6 Address Configuration Register (IPU_DC_WR_CH_ADDR_6)	32	R/W	0000_0000h	45.51.259/ 3273
1E05_8080	DC Routine Link Register 0Channel 6 (IPU_DC_RL0_CH_6)	32	R/W	0000_0000h	45.51.260/ 3273
1E05_8084	DC Routine Link Register 1 Channel 6 (IPU_DC_RL1_CH_6)	32	R/W	0000_0000h	45.51.261/ 3275
1E05_8088	DC Routine Link Register 2 Channel 6 (IPU_DC_RL2_CH_6)	32	R/W	0000_0000h	45.51.262/ 3276
1E05_808C	DC Routine Link Register 3 Channel 6 (IPU_DC_RL3_CH_6)	32	R/W	0000_0000h	45.51.263/ 3277
1E05_8090	DC Routine Link Register 4 Channel 6 (IPU_DC_RL4_CH_6)	32	R/W	0000_0000h	45.51.264/ 3278
1E05_8094	DC Write Channel 8 Configuration 1Register (IPU_DC_WR_CH_CONF1_8)	32	R/W	0000_0000h	45.51.265/ 3279
1E05_8098	DC Write Channel 8 Configuration 2 Register (IPU_DC_WR_CH_CONF2_8)	32	R/W	0000_0000h	45.51.266/ 3280

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E05_809C	DC Routine Link Register 1 Channel 8 (IPU_DC_RL1_CH_8)	32	R/W	0000_0000h	45.51.267/ 3280
1E05_80A0	DC Routine Link Register 2 Channel 8 (IPU_DC_RL2_CH_8)	32	R/W	0000_0000h	45.51.268/ 3281
1E05_80A4	DC Routine Link Register 3 Channel 8 (IPU_DC_RL3_CH_8)	32	R/W	0000_0000h	45.51.269/ 3282
1E05_80A8	DC Routine Link Register 4 Channel 8 (IPU_DC_RL4_CH_8)	32	R/W	0000_0000h	45.51.270/ 3283
1E05_80AC	DC Routine Link Register 5 Channel 8 (IPU_DC_RL5_CH_8)	32	R/W	0000_0000h	45.51.271/ 3283
1E05_80B0	DC Routine Link Register 6 Channel 8 (IPU_DC_RL6_CH_8)	32	R/W	0000_0000h	45.51.272/ 3284
1E05_80B4	DC Write Channel 9 Configuration 1Register (IPU_DC_WR_CH_CONF1_9)	32	R/W	0000_0000h	45.51.273/ 3285
1E05_80B8	DC Write Channel 9Configuration 2Register (IPU_DC_WR_CH_CONF2_9)	32	R/W	0000_0000h	45.51.274/ 3286
1E05_80BC	DC Routine Link Register 1 Channel 9 (IPU_DC_RL1_CH_9)	32	R/W	0000_0000h	45.51.275/ 3286
1E05_80C0	DC Routine Link Register 2 Channel 9 (IPU_DC_RL2_CH_9)	32	R/W	0000_0000h	45.51.276/ 3287
1E05_80C4	DC Routine Link Register 3Channel 9 (IPU_DC_RL3_CH_9)	32	R/W	0000_0000h	45.51.277/ 3288
1E05_80C8	DC Routine Link Register 4 Channel 9 (IPU_DC_RL4_CH_9)	32	R/W	0000_0000h	45.51.278/ 3289
1E05_80CC	DC Routine Link Register 5 Channel 9 (IPU_DC_RL5_CH_9)	32	R/W	0000_0000h	45.51.279/ 3289
1E05_80D0	DC Routine Link Register 6 Channel 9 (IPU_DC_RL6_CH_9)	32	R/W	0000_0000h	45.51.280/ 3290
1E05_80D4	DC General Register (IPU_DC_GEN)	32	R/W	0000_0060h	45.51.281/ 3291
1E05_80D8	DC Display Configuration 1 Register 0 (IPU_DC_DISP_CONF1_0)	32	R/W	0000_0042h	45.51.282/ 3292
1E05_80DC	DC Display Configuration 1 Register 1 (IPU_DC_DISP_CONF1_1)	32	R/W	0000_0042h	45.51.283/ 3294
1E05_80E0	DC Display Configuration 1 Register 2 (IPU_DC_DISP_CONF1_2)	32	R/W	0000_0042h	45.51.284/ 3295
1E05_80E4	DC Display Configuration 1 Register 3 (IPU_DC_DISP_CONF1_3)	32	R/W	0000_0042h	45.51.285/ 3296
1E05_80E8	DC Display Configuration 2 Register 0 (IPU_DC_DISP_CONF2_0)	32	R/W	0000_0000h	45.51.286/ 3297

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E05_80F0	DC Display Configuration 2 Register 2 (IPU_DC_DISP_CONF2_2)	32	R/W	0000_0000h	45.51.287/ 3298
1E05_80F4	DC Display Configuration 2 Register 3 (IPU_DC_DISP_CONF2_3)	32	R/W	0000_0000h	45.51.288/ 3298
1E05_80F8	DC DI0Configuration Register 1 (IPU_DC_DI0_CONF_1)	32	R/W	0000_0000h	45.51.289/ 3299
1E05_80FC	DC DI0Configuration Register 2 (IPU_DC_DI0_CONF_2)	32	R/W	0000_0000h	45.51.290/ 3299
1E05_8100	DC DI1Configuration Register 1 (IPU_DC_DI1_CONF_1)	32	R/W	0000_0000h	45.51.291/ 3299
1E05_8104	DC DI1Configuration Register 2 (IPU_DC_DI1_CONF_2)	32	R/W	0000_0000h	45.51.292/ 3300
1E05_8108	DC Mapping Configuration Register 0 (IPU_DC_MAP_CONF_0)	32	R/W	0000_0000h	45.51.293/ 3300
1E05_810C	DC Mapping Configuration Register 1 (IPU_DC_MAP_CONF_1)	32	R/W	0000_0000h	45.51.294/ 3301
1E05_8110	DC Mapping Configuration Register 2 (IPU_DC_MAP_CONF_2)	32	R/W	0000_0000h	45.51.295/ 3302
1E05_8114	DC Mapping Configuration Register 3 (IPU_DC_MAP_CONF_3)	32	R/W	0000_0000h	45.51.296/ 3303
1E05_8118	DC Mapping Configuration Register 4 (IPU_DC_MAP_CONF_4)	32	R/W	0000_0000h	45.51.297/ 3304
1E05_811C	DC Mapping Configuration Register 5 (IPU_DC_MAP_CONF_5)	32	R/W	0000_0000h	45.51.298/ 3304
1E05_8120	DC Mapping Configuration Register 6 (IPU_DC_MAP_CONF_6)	32	R/W	0000_0000h	45.51.299/ 3305
1E05_8124	DC Mapping Configuration Register 7 (IPU_DC_MAP_CONF_7)	32	R/W	0000_0000h	45.51.300/ 3306
1E05_8128	DC Mapping Configuration Register 8 (IPU_DC_MAP_CONF_8)	32	R/W	0000_0000h	45.51.301/ 3307
1E05_812C	DC Mapping Configuration Register 9 (IPU_DC_MAP_CONF_9)	32	R/W	0000_0000h	45.51.302/ 3308
1E05_8130	DC Mapping Configuration Register 10 (IPU_DC_MAP_CONF_10)	32	R/W	0000_0000h	45.51.303/ 3309
1E05_8134	DC Mapping Configuration Register 11 (IPU_DC_MAP_CONF_11)	32	R/W	0000_0000h	45.51.304/ 3310
1E05_8138	DC Mapping Configuration Register 12 (IPU_DC_MAP_CONF_12)	32	R/W	0000_0000h	45.51.305/ 3311
1E05_813C	DC Mapping Configuration Register 13 (IPU_DC_MAP_CONF_13)	32	R/W	0000_0000h	45.51.306/ 3312

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E05_8140	DC Mapping Configuration Register 14 (IPU_DC_MAP_CONF_14)	32	R/W	0000_0000h	45.51.307/ 3313
1E05_8144	DC Mapping Configuration Register 15 (IPU_DC_MAP_CONF_15)	32	R/W	0000_0000h	45.51.308/ 3314
1E05_8148	DC Mapping Configuration Register 16 (IPU_DC_MAP_CONF_16)	32	R/W	0000_0000h	45.51.309/ 3315
1E05_814C	DC Mapping Configuration Register 17 (IPU_DC_MAP_CONF_17)	32	R/W	0000_0000h	45.51.310/ 3316
1E05_8150	DC Mapping Configuration Register 18 (IPU_DC_MAP_CONF_18)	32	R/W	0000_0000h	45.51.311/ 3316
1E05_8154	DC Mapping Configuration Register 19 (IPU_DC_MAP_CONF_19)	32	R/W	0000_0000h	45.51.312/ 3317
1E05_8158	DC Mapping Configuration Register 20 (IPU_DC_MAP_CONF_20)	32	R/W	0000_0000h	45.51.313/ 3318
1E05_815C	DC Mapping Configuration Register 21 (IPU_DC_MAP_CONF_21)	32	R/W	0000_0000h	45.51.314/ 3318
1E05_8160	DC Mapping Configuration Register 22 (IPU_DC_MAP_CONF_22)	32	R/W	0000_0000h	45.51.315/ 3319
1E05_8164	DC Mapping Configuration Register 23 (IPU_DC_MAP_CONF_23)	32	R/W	0000_0000h	45.51.316/ 3320
1E05_8168	DC Mapping Configuration Register 24 (IPU_DC_MAP_CONF_24)	32	R/W	0000_0000h	45.51.317/ 3320
1E05_816C	DC Mapping Configuration Register 25 (IPU_DC_MAP_CONF_25)	32	R/W	0000_0000h	45.51.318/ 3321
1E05_8170	DC Mapping Configuration Register 26 (IPU_DC_MAP_CONF_26)	32	R/W	0000_0000h	45.51.319/ 3322
1E05_8174	DC User General Data Event 0 Register 0 (IPU_DC_UGDE0_0)	32	R/W	0000_0000h	45.51.320/ 3323
1E05_8178	DC User General Data Event 0 Register 1 (IPU_DC_UGDE0_1)	32	R/W	0000_0000h	45.51.321/ 3324
1E05_817C	DC User General Data Event 0 Register2 (IPU_DC_UGDE0_2)	32	R/W	0000_0000h	45.51.322/ 3325
1E05_8180	DC User General Data Event 0 Register 3 (IPU_DC_UGDE0_3)	32	R/W	0000_0000h	45.51.323/ 3325
1E05_8184	DC User General Data Event 1Register0 (IPU_DC_UGDE1_0)	32	R/W	0000_0000h	45.51.324/ 3326
1E05_8188	DC User General Data Event 1 Register 1 (IPU_DC_UGDE1_1)	32	R/W	0000_0000h	45.51.325/ 3327
1E05_818C	DC User General Data Event 1Register 2 (IPU_DC_UGDE1_2)	32	R/W	0000_0000h	45.51.326/ 3328

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E05_8190	DC User General Data Event 1Register 3 (IPU_DC_UGDE1_3)	32	R/W	0000_0000h	45.51.327/ 3328
1E05_8194	DC User General Data Event 2 Register 0 (IPU_DC_UGDE2_0)	32	R/W	0000_0000h	45.51.328/ 3329
1E05_8198	DC User General Data Event 2 Register 1 (IPU_DC_UGDE2_1)	32	R/W	0000_0000h	45.51.329/ 3330
1E05_819C	DC User General Data Event 2Register 2 (IPU_DC_UGDE2_2)	32	R/W	0000_0000h	45.51.330/ 3331
1E05_81A0	DC User General Data Event 2Register 3 (IPU_DC_UGDE2_3)	32	R/W	0000_0000h	45.51.331/ 3331
1E05_81A4	DC User General Data Event 3Register 0 (IPU_DC_UGDE3_0)	32	R/W	0000_0000h	45.51.332/ 3332
1E05_81A8	DC User General Data Event 3Register 1 (IPU_DC_UGDE3_1)	32	R/W	0000_0000h	45.51.333/ 3333
1E05_81AC	DC User General Data Event 3Register 2 (IPU_DC_UGDE3_2)	32	R/W	0000_0000h	45.51.334/ 3334
1E05_81B0	DC User General Data Event 3Register 2 (IPU_DC_UGDE3_3)	32	R/W	0000_0000h	45.51.335/ 3334
1E05_81B4	DC Low Level Access Control Register 0 (IPU_DC_LLA0)	32	R/W	0000_0000h	45.51.336/ 3335
1E05_81B8	DC Low Level Access Control Register 1 (IPU_DC_LLA1)	32	R/W	0000_0000h	45.51.337/ 3335
1E05_81BC	DC Read Low Level Read Access Control Register 0 (IPU_DC_R_LLA0)	32	R/W	0000_0000h	45.51.338/ 3336
1E05_81C0	DC Read Low Level Read Access Control Register1 (IPU_DC_R_LLA1)	32	R/W	0000_0000h	45.51.339/ 3336
1E05_81C4	DC Write Channel 5 Configuration Register (IPU_DC_WR_CH_ADDR_5_ALT)	32	R/W	0000_0000h	45.51.340/ 3337
1E05_81C8	DC Status Register (IPU_DC_STAT)	32	R	0000_00AAh	45.51.341/ 3338
1E05_90EC	DC Display Configuration 2 Register 1 (IPU_DC_DISP_CONF2_1)	32	R/W	0000_0000h	45.51.342/ 3339
1E06_0000	DMFC Read Channel Register (IPU_DMFC_RD_CHAN)	32	R/W	0000_0200h	45.51.343/ 3340
1E06_0004	DMFC Write Channel Register (IPU_DMFC_WR_CHAN)	32	R/W	0000_0000h	45.51.344/ 3341
1E06_0008	DMFC Write Channel Definition Register (IPU_DMFC_WR_CHAN_DEF)	32	R/W	2020_2020h	45.51.345/ 3344
1E06_000C	DMFC Display Processor Channel Register (IPU_DMFC_DP_CHAN)	32	R/W	0000_0000h	45.51.346/ 3346

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E06_0010	DMFC Display Processor Channel Definition Register (IPU_DMFC_DP_CHAN_DEF)	32	R/W	2020_2020h	45.51.347/ 3349
1E06_0014	DMFC General 1 Register (IPU_DMFC_GENERAL_1)	32	R/W	0000_0003h	45.51.348/ 3351
1E06_0018	DMFC General 2 Register (IPU_DMFC_GENERAL_2)	32	R/W	0000_0000h	45.51.349/ 3353
1E06_001C	DMFC IC Interface Control Register (IPU_DMFC_IC_CTRL)	32	R/W	0000_0002h	45.51.350/ 3354
1E06_0020	DMFC Write Channel Alternate Register (IPU_DMFC_WR_CHAN_ALT)	32	R/W	0000_0000h	45.51.351/ 3355
1E06_0024	DMFC Write Channel Definition Alternate Register (IPU_DMFC_WR_CHAN_DEF_ALT)	32	R/W	0000_2000h	45.51.352/ 3356
1E06_0028	DMFC MFC Display Processor Channel Alternate Register (IPU_DMFC_DP_CHAN_ALT)	32	R/W	0000_0000h	45.51.353/ 3357
1E06_002C	DMFC Display Channel Definition Alternate Register (IPU_DMFC_DP_CHAN_DEF_ALT)	32	R/W	2020_0020h	45.51.354/ 3360
1E06_0030	DMFC General 1 Alternate Register (IPU_DMFC_GENERAL1_ALT)	32	R/W	0000_0000h	45.51.355/ 3361
1E06_0034	DMFC Status Register (IPU_DMFC_STAT)	32	R	02FF_F000h	45.51.356/ 3363
1E06_8000	VDI Field Size Register (IPU_VDI_FSIZE)	32	R/W	0000_0000h	45.51.357/ 3364
1E06_8004	VDI Control Register (IPU_VDI_C)	32	R/W	0000_0000h	45.51.358/ 3365
1E06_8008	VDI Control Register 2 (IPU_VDI_C2_)	32	R/W	0000_0000h	45.51.359/ 3367
1E06_800C	VDI Combining Parameters Register 1 (IPU_VDI_CMDP_1)	32	R/W	0000_0000h	45.51.360/ 3368
1E06_8010	VDI Combining Parameters Register 2 (IPU_VDI_CMDP_2)	32	R/W	0000_0000h	45.51.361/ 3368
1E06_8014	VDI Plane Size Register 1 (IPU_VDI_PS_1)	32	R/W	0000_0000h	45.51.362/ 3369
1E06_8018	VDI Plane Size Register 2 (IPU_VDI_PS_2)	32	R/W	0000_0000h	45.51.363/ 3370
1E06_801C	VDI Plane Size Register 3 (IPU_VDI_PS_3)	32	R/W	0000_0000h	45.51.364/ 3370
1E06_8020	VDI Plane Size Register 4 (IPU_VDI_PS_4)	32	R/W	0000_0000h	45.51.365/ 3371
1E08_0008	IDMAC Channel Enable 2 Register (IPU_IDMAC_CH_EN_2)	32	R/W	0000_0000h	45.51.366/ 3372

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1E08_0248	IDMAC Channel Lock Enable 2Register (IPU_IDMAC_LOCK_EN_2)	32	R/W	0000_0000h	45.51.367/ 3374
1E08_028C	IDMAC Channel Alternate Address 0 Register (IPU_IDMAC_SUB_ADDR_0)	32	R/W	0000_0000h	45.51.368/ 3375
1E08_0304	IDMAC Channel Alternate Address 2 Register (IPU_IDMAC_SUB_ADDR_2)	32	R/W	0000_0000h	45.51.369/ 3376
1E08_0308	IDMAC Channel Alternate Address 3 Register (IPU_IDMAC_SUB_ADDR_3)	32	R/W	0000_0000h	45.51.370/ 3377
1E08_030C	IDMAC Channel Alternate Address 4 Register (IPU_IDMAC_SUB_ADDR_4)	32	R/W	0000_0000h	45.51.371/ 3378
1E08_0440	IDMAC Band Mode Enable 1 Register (IPU_IDMAC_BNDM_EN_1)	32	R/W	0000_0000h	45.51.372/ 3379
1E80_2C30	IDMAC Channel Alternate Address 1 Register (IPU_IDMAC_SUB_ADDR_1)	32	R/W	0000_0000h	45.51.373/ 3382
1F04_0000	DP Common Configuration Sync Flow Register (IPU_DP_COM_CONF_SYNC)	32	R/W	0000_0000h	45.51.374/ 3383
1F04_0004	DP Graphic Window Control Sync Flow Register (IPU_DP_Graph_Wind_CTRL_SYNC)	32	R/W	0000_0000h	45.51.375/ 3385
1F04_0008	DP Partial Plane Window Position Sync Flow Register (IPU_DP_FG_POS_SYNC)	32	R/W	0000_0000h	45.51.376/ 3386
1F04_000C	DP Cursor Position and Size Sync Flow Register (IPU_DP_CUR_POS_SYNC)	32	R/W	0000_0000h	45.51.377/ 3386
1F04_0010	DP Color Cursor Mapping Sync Flow Register (IPU_DP_CUR_MAP_SYNC)	32	R/W	0000_0000h	45.51.378/ 3387
1F04_0014	DP Gamma Constants Sync Flow Register i (IPU_DP_GAMMA_C_SYNC_i)	32	R/W	0000_0000h	45.51.379/ 3388
1F04_0034	DP Gamma Correction Slope Sync Flow Register i (IPU_DP_GAMMA_S_SYNC_i)	32	R/W	0000_0000h	45.51.380/ 3388
1F04_0044	DP Color Space Conversion Control Sync Flow Registers (IPU_DP_CSCA_SYNC_i)	32	R/W	0000_0000h	45.51.381/ 3389
1F04_0054	DP Color Conversion Control Sync Flow Register 0 (IPU_DP_SCS_SYNC_0)	32	R/W	0000_0000h	45.51.382/ 3390
1F04_0058	DP Color Conversion Control Sync Flow Register 1 (IPU_DP_SCS_SYNC_1)	32	R/W	0000_0000h	45.51.383/ 3390
1F04_005C	DP Cursor Position and Size Alternate Register (IPU_DP_CUR_POS_ALT)	32	R/W	0000_0000h	45.51.384/ 3391
1F04_0060	DP Common Configuration Async 0 Flow Register (IPU_DP_COM_CONF_ASYNC0)	32	R/W	0000_0000h	45.51.385/ 3392
1F04_0064	DP Graphic Window Control Async 0 Flow Register (IPU_DP_GRAPH_WIND_CTRL_ASYNC0)	32	R/W	0000_0000h	45.51.386/ 3394

Table continues on the next page...

IPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1F04_0068	DP Partial Plane Window Position Async 0 Flow Register (IPU_DP_FG_POS_ASYNC0)	32	R/W	0000_0000h	45.51.387/ 3395
1F04_006C	DP Cursor Position and Size Async 0 Flow Register (IPU_DP_CUR_POS_ASYNC0)	32	R/W	0000_0000h	45.51.388/ 3395
1F04_0070	DP Color Cursor Mapping Async 0 Flow Register (IPU_DP_CUR_MAP_ASYNC0)	32	R/W	0000_0000h	45.51.389/ 3396
1F04_0074	DP Gamma Constant Async 0 Flow Register i (IPU_DP_GAMMA_C_ASYNC0_i)	32	R/W	0000_0000h	45.51.390/ 3397
1F04_0094	DP Gamma Correction Slope Async 0 Flow Register i (IPU_DP_GAMMA_S_ASYNC0_i)	32	R/W	0000_0000h	45.51.391/ 3398
1F04_00A4	DP Color Space Conversion Control Async 0 Flow Register i (IPU_DP_CSCA_ASYNC0_i)	32	R/W	0000_0000h	45.51.392/ 3398
1F04_00B4	DP Color Conversion Control Async 0 Flow Register 0 (IPU_DP_CSC_ASYNC0_0)	32	R/W	0000_0000h	45.51.393/ 3399
1F04_00B8	DP Color Conversion Control Async 1 Flow Register (IPU_DP_CSC_ASYNC_1)	32	R/W	0000_0000h	45.51.394/ 3400
1F04_00BC	DP Common Configuration Async 1 Flow Register (IPU_DP_COM_CONF_ASYNC1)	32	R/W	0000_0000h	45.51.395/ 3401
1F04_00C0	DP Graphic Window Control Async 1 Flow Register (IPU_DP_GRAPH_WIND_CTRL_ASYNC1)	32	R/W	0000_0000h	45.51.396/ 3403
1F04_00C4	DP Partial Plane Window Position Async 1 Flow Register (IPU_DP_FG_POS_ASYNC1)	32	R/W	0000_0000h	45.51.397/ 3404
1F04_00C8	DP Cursor Postion and Size Async 1 Flow Register (IPU_DP_CUR_POS_ASYNC1)	32	R/W	0000_0000h	45.51.398/ 3404
1F04_00CC	DP Color Cursor Mapping Async 1 Flow Register (IPU_DP_CUR_MAP_ASYNC1)	32	R/W	0000_0000h	45.51.399/ 3405
1F04_00D0	DP Gamma Constants Async 1 Flow Register i (IPU_DP_GAMMA_C_ASYNC1_i)	32	R/W	0000_0000h	45.51.400/ 3406
1F04_00F0	DP Gamma Correction Slope Async 1 Flow Register i (IPU_DP_GAMMA_S_ASYN1_i)	32	R/W	0000_0000h	45.51.401/ 3407
1F04_0100	DP Color Space Converstion Control Async 1 Flow Register i (IPU_DP_CSCA_ASYNC1_i)	32	R/W	0000_0000h	45.51.402/ 3407
1F04_0110	DP Color Conversion Control Async 1 Flow Register 0 (IPU_DP_CSC_ASYNC1_0)	32	R/W	0000_0000h	45.51.403/ 3408
1F04_0114	DP Color Conversion Control Async 1 Flow Register 1 (IPU_DP_CSC_ASYNC1_1)	32	R/W	0000_0000h	45.51.404/ 3409
9E38_4444	IDMAC Band Mode Enable 2 Register (IPU_IDMAC_BNDM_EN_2)	32	R/W	0000_0000h	45.51.405/ 3410
9E3C_4848	IDMAC Scroll Coordinations Register (IPU_IDMAC_SC_CORD)	32	R/W	0000_0000h	45.51.406/ 3411

45.51.1 Configuration Register (IPU_CONF)

Address: IPU_CONF is 1E00_0000h base + 0h offset = 1E00_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CSI_SEL	IC_INPUT	0	0	VDI_DMFC_SYNC	IC_DMFC_SYNC	IC_DMFC_SEL	0	0	IDMAC_DISABLE	IPU_DIAGBUS_ON	IPU_DIAGBUS_MODE				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			VDI_EN	SISG_EN	DMFC_EN	DC_EN	SMFC_EN	DI1_EN	DI0_EN	DP_EN	0	IRT_EN	IC_EN	CSI1_EN	CSI0_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CONF field descriptions

Field	Description
31 CSI_SEL	CSI select bit; This bit selects manually between the 2 CSI's. This bit defines which CSI is the input to the IC. This bit is effective only if IC_INPUT is bit cleared 0 CSI0 is selected 1 CSI1 is selected
30 IC_INPUT	IC Input select bit. This bit selects manually between the 2 inputs to the IC 1 VDI
29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28 Reserved	This read-only field is reserved and always has the value zero. Reserved
27 VDI_DMFC_SYNC	This bit enables the direct path VDIC -> IC_VF -> DMFC for sync flow. If this bit is set IC_DMFC_SEL must be set. 0 the flow is disabled 1 the flow is enabled
26 IC_DMFC_SYNC	IC to DMFC Sync flow This bit defines if the direct flow between IC to DMFC is synchronous or asynchronous 0 async flow 1 Sync flow
25 IC_DMFC_SEL	IC to DMFC select Selects the DMAIC_1 (channel 21) channel's connectivity between the IC and the DMFC

Table continues on the next page...

IPU_CONF field descriptions (continued)

Field	Description
	0 DMAIC_1 (channel 21) is routed to the IDMAC 1 DMAIC_1 (channel 21) is routed to DMFC In case DMFC was selected the IDMAC_CH_EN[21] must be clear.
24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
22 IDMAC_ DISABLE	Image DMA controller (IDMAC) disable bit. This bit allows the user to turn off the clock of the IDMAC if the use case permits it. By default the IDMAC is enabled. 0 IDMAC is enabled 1 IDMAC is disabled
21 IPU_DIAGBUS_ ON	IPU Diagnostics bus on This bit is connected to the IPU's output. it can be used by the SoC in order to control the IOMUX and bring the IPU's diagnostics bus to the SoC's pins. 1 diagnostics bus is on 0 diagnostics bus is off
20–16 IPU_DIAGBUS_ MODE	IPU diagnostic bus mode. This 5 bits select one of 16 groups of signals to be routed to the IPU's diagnostics bus 00000 Route group 0 to the IPU's diagnostics bus 00001 Route group 1 to the IPU's diagnostics bus 01111 Route group 15 to the IPU's diagnostics bus 10000 Route group 16 to the IPU's diagnostics bus 10001 Route group 17 to the IPU's diagnostics bus 10010-11111 Reserved
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
12 VDI_EN	VDI enable bit. This bit must be cleared if the ISP_EN bit is set. 0 VDIC is disabled 1 VDIC is enabled
11 SISG_EN	Still Image Synchronization Generator (SISG) Enable bit 0 SISG is disabled 1 SISG is enabled
10 DMFC_EN	Display's Multi FIFO Controller sub-block (DMFC) Enable bit 0 DMFC is disabled 1 DMFC is enabled
9 DC_EN	Display Controller sub-block (DC) Enable bit 0 DC is disabled 1 DC is enabled

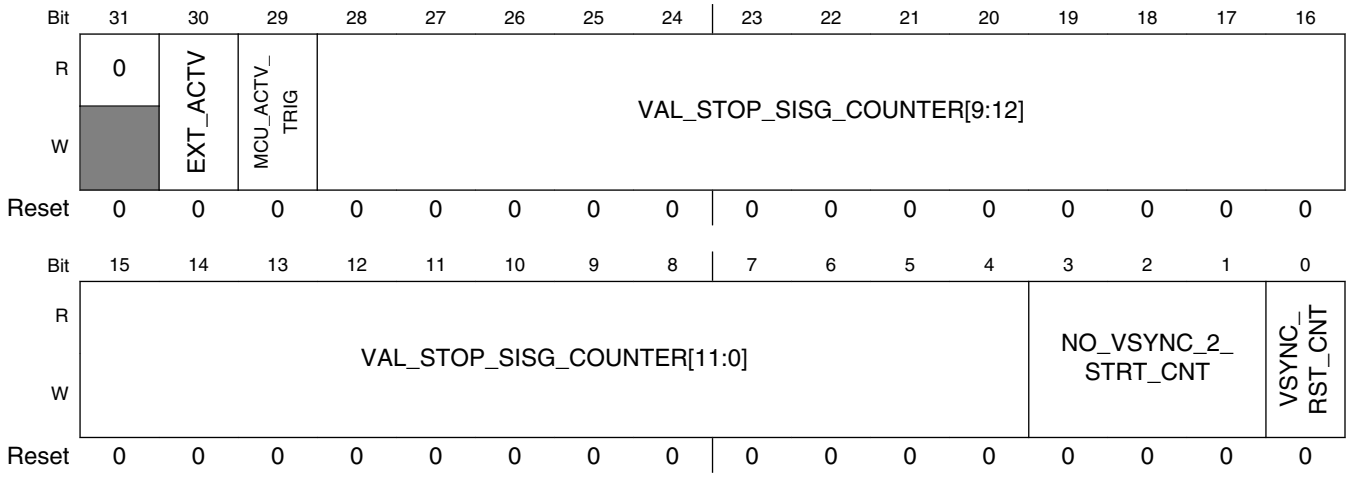
Table continues on the next page...

IPU_CONF field descriptions (continued)

Field	Description
8 SMFC_EN	Sensor's Multi FIFO Controller Sub-block (SMFC) Enable bit 0 SMFC is disabled 1 SMFC is enabled
7 DI1_EN	Display Interface Sub-block 1 Enable bit 0 DI1 is disabled 1 DI1 is enabled
6 DI0_EN	Display interface Sub-block 0 Enable bit 0 DI0 is disabled 1 DI0 is enabled
5 DP_EN	Display processor Sub-block Enable bit 0 DP is disabled 1 DP is enabled
4 Reserved	This read-only field is reserved and always has the value zero. Reserved
3 IRT_EN	Image Rotation Sub-Block Enable bit 0 IRT is disabled 1 IRT is enabled
2 IC_EN	Image Conversion Sub-Block Enable bit 0 IC is disabled 1 IC is enabled
1 CSI1_EN	Camera Sensor Interface 1 Enable bit 0 CSI1 is disabled 1 CSI1 is enabled
0 CSI0_EN	Camera Sensor Interface 0 Enable bit 0 CSI0 is disabled 1 CSI0 is enabled

45.51.2 SISG Control 0 Register (IPU_SISG_CTRL0)

Address: IPU_SISG_CTRL0 is 1E00_0000h base + 4h offset = 1E00_0004h

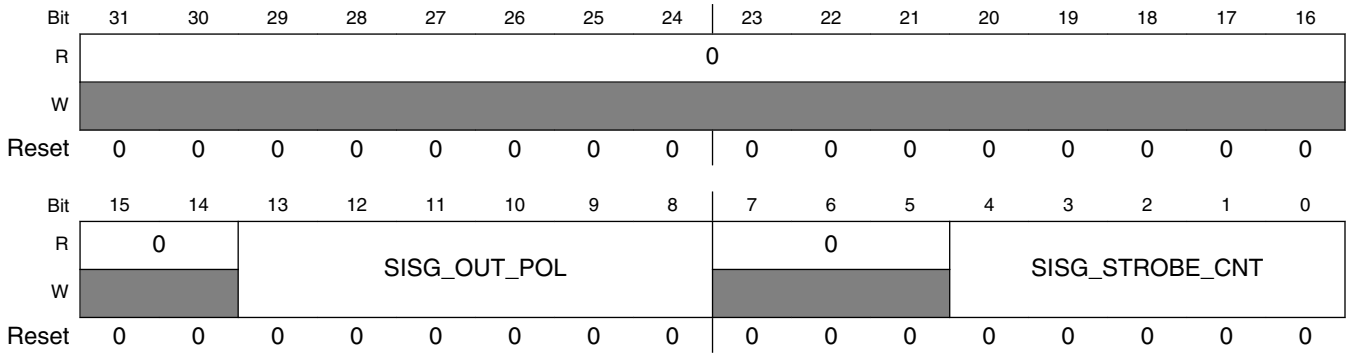


IPU_SISG_CTRL0 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30 EXT_ACTV	External Active Define if an external active trigger will start the counters. The external active trigger is an input signal to the IPU called ext_actv_trig
29 MCU_ACTV_TRIG	Reserved, should be cleared.
28–4 VAL_STOP_SISG_COUNTER	SISG Stop Counters value. This is a predefined value that stops the SISG counters. The user should write to this field the N-1 value of the desired value.
3–1 NO_VSYNC_2_STRT_CNT	VSYCs to Start Counter This bits define how many VSYNCs signals will be counter before activating the SISG counters. If set to 0 starts immediately. If set to N (1..7) starts after N VSYNCs.
0 VSYNC_RST_CNT	VSYNC Resets counters Defines if the counters are stooped following VSYNC or when the counters reach a pre defined value (VAL_STOP_SISG_COUNTER) 1 The counters are stopped at VSYNC 0 The counters are stooped when the counters reach the VAL_STOP_SISG_COUNTER value.

45.51.3 SISG Control 1 Register (IPU_SISG_CTRL1)

Address: IPU_SISG_CTRL1 is 1E00_0000h base + 8h offset = 1E00_0008h

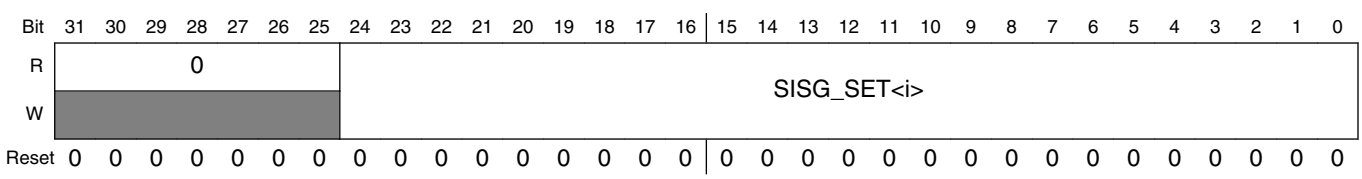


IPU_SISG_CTRL1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
13–8 SISG_OUT_POL	SISG_OUT_POL This bits defines the polarity of the SISG output signals 1 active high 0 active low
7–5 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
4–0 SISG_STROBE_CNT	SISG Strobe Count The SISG can repeat the sequence for up to 32 cycles; this is used for generating a train of pulses.

45.51.4 SISG Set<i> Register (IPU_SISG_SET_i)

Address: IPU_SISG_SET_i is 1E00_0000h base + Ch offset = 1E00_000Ch



IPU_SISG_SET_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.

Table continues on the next page...

IPU_SISG_SET_i field descriptions (continued)

Field	Description
24–0 SISG_SET<i>	SISG SET <i> value These bits define the set value of the SISG counter #<i>

45.51.5 SISG Clear <i> Register (IPU_SISG_CLR_i)

Address: IPU_SISG_CLR_i is 1E00_0000h base + 24h offset = 1E00_0024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								SISG_CLEAR<i>																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_SISG_CLR_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
24–0 SISG_CLEAR<i>	SISG CLR <i> value These bits define the clear value of the SISG counter #<i>

45.51.6 Interrupt Control Register 1 (IPU_INT_CTRL_1)

This register contains part of IPU interrupts controls. The controls of EOF (end of frame) of DMA Channels interrupts [31:0] can be found in this register.

Address: IPU_INT_CTRL_1 is 1E00_0000h base + 3Ch offset = 1E00_003Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_EOF_EN_31	0	IDMAC_EOF_EN_29	IDMAC_EOF_EN_28	IDMAC_EOF_EN_27	IDMAC_EOF_EN_26	IDMAC_EOF_EN_25	IDMAC_EOF_EN_24	IDMAC_EOF_EN_23	IDMAC_EOF_EN_22	IDMAC_EOF_EN_21	IDMAC_EOF_EN_20	IDMAC_EOF_EN_19	IDMAC_EOF_EN_18	IDMAC_EOF_EN_17	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_EOF_EN_15	IDMAC_EOF_EN_14	IDMAC_EOF_EN_13	IDMAC_EOF_EN_12	IDMAC_EOF_EN_11	IDMAC_EOF_EN_10	IDMAC_EOF_EN_9	IDMAC_EOF_EN_8	0		IDMAC_EOF_EN_5	0	IDMAC_EOF_EN_3	IDMAC_EOF_EN_2	IDMAC_EOF_EN_1	IDMAC_EOF_EN_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_CTRL_1 field descriptions

Field	Description
31 IDMAC_EOF_EN_31	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
29 IDMAC_EOF_EN_29	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
28 IDMAC_EOF_EN_28	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
27 IDMAC_EOF_EN_27	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
26 IDMAC_EOF_EN_26	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
25 IDMAC_EOF_EN_25	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
24 IDMAC_EOF_EN_24	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
23 IDMAC_EOF_EN_23	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_1 field descriptions (continued)

Field	Description
22 IDMAC_EOF_EN_22	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
21 IDMAC_EOF_EN_21	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_EOF_EN_20	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_EOF_EN_19	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
18 IDMAC_EOF_EN_18	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
17 IDMAC_EOF_EN_17	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
16 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 IDMAC_EOF_EN_15	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_EOF_EN_14	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_1 field descriptions (continued)

Field	Description
13 IDMAC_EOF_EN_13	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
12 IDMAC_EOF_EN_12	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_EOF_EN_11	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10 IDMAC_EOF_EN_10	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_EOF_EN_9	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
8 IDMAC_EOF_EN_8	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
5 IDMAC_EOF_EN_5	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
4 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_1 field descriptions (continued)

Field	Description
3 IDMAC_EOF_EN_3	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
2 IDMAC_EOF_EN_2	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_EOF_EN_1	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 IDMAC_EOF_EN_0	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

45.51.7 Interrupt Control Register 2 (IPU_INT_CTRL_2)

This register contains part of IPU interrupts controls. The controls of EOF (end of frame) of DMA Channels interrupts [63:32] can be found in this register.

Address: IPU_INT_CTRL_2 is 1E00_0000h base + 40h offset = 1E00_0040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											IDMAC_EOF_EN_52	IDMAC_EOF_EN_51	IDMAC_EOF_EN_50	IDMAC_EOF_EN_49	IDMAC_EOF_EN_48
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_EOF_EN_47	IDMAC_EOF_EN_46	IDMAC_EOF_EN_45	IDMAC_EOF_EN_44	IDMAC_EOF_EN_43	IDMAC_EOF_EN_42	IDMAC_EOF_EN_41	IDMAC_EOF_EN_40	0						IDMAC_EOF_EN_33	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_CTRL_2 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_EOF_EN_52	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_EOF_EN_51	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
18 IDMAC_EOF_EN_50	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
17 IDMAC_EOF_EN_49	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
16 IDMAC_EOF_EN_48	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 IDMAC_EOF_EN_47	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_EOF_EN_46	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
13 IDMAC_EOF_EN_45	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

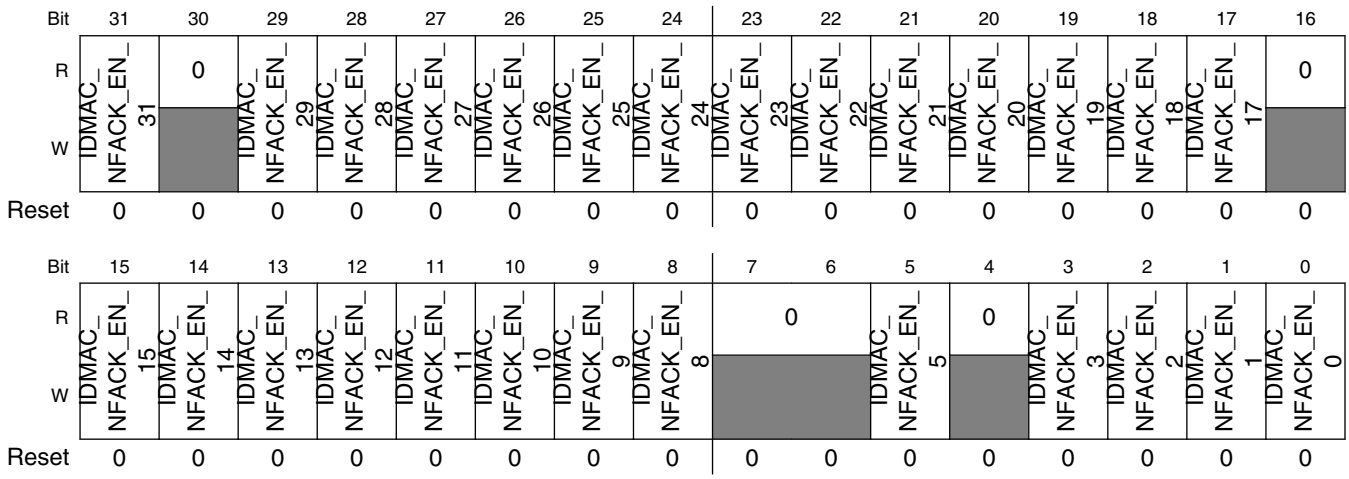
IPU_INT_CTRL_2 field descriptions (continued)

Field	Description
12 IDMAC_EOF_EN_44	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_EOF_EN_43	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10 IDMAC_EOF_EN_42	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_EOF_EN_41	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
8 IDMAC_EOF_EN_40	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
7–2 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_EOF_EN_33	Enable End of Frame of Channel interrupt. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.

45.51.8 Interrupt Control Register 3 (IPU_INT_CTRL_3)

This register contains part of IPU interrupts controls. The controls of NFACK (New Frame Ack) of DMA Channels interrupts [31:0] can be found in this register.

Address: IPU_INT_CTRL_3 is 1E00_0000h base + 44h offset = 1E00_0044h



IPU_INT_CTRL_3 field descriptions

Field	Description
31 IDMAC_ NFACK_EN_31	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
29 IDMAC_ NFACK_EN_29	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
28 IDMAC_ NFACK_EN_28	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
27 IDMAC_ NFACK_EN_27	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_CTRL_3 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
26 IDMAC_ NFACK_EN_26	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
25 IDMAC_ NFACK_EN_25	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
24 IDMAC_ NFACK_EN_24	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
23 IDMAC_ NFACK_EN_23	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
22 IDMAC_ NFACK_EN_22	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
21 IDMAC_ NFACK_EN_21	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_ NFACK_EN_20	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_ NFACK_EN_19	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
18 IDMAC_ NFACK_EN_18	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_CTRL_3 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
17 IDMAC_ NFACK_EN_17	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
16 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 IDMAC_ NFACK_EN_15	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_ NFACK_EN_14	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
13 IDMAC_ NFACK_EN_13	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
12 IDMAC_ NFACK_EN_12	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_ NFACK_EN_11	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10 IDMAC_ NFACK_EN_10	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_ NFACK_EN_9	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

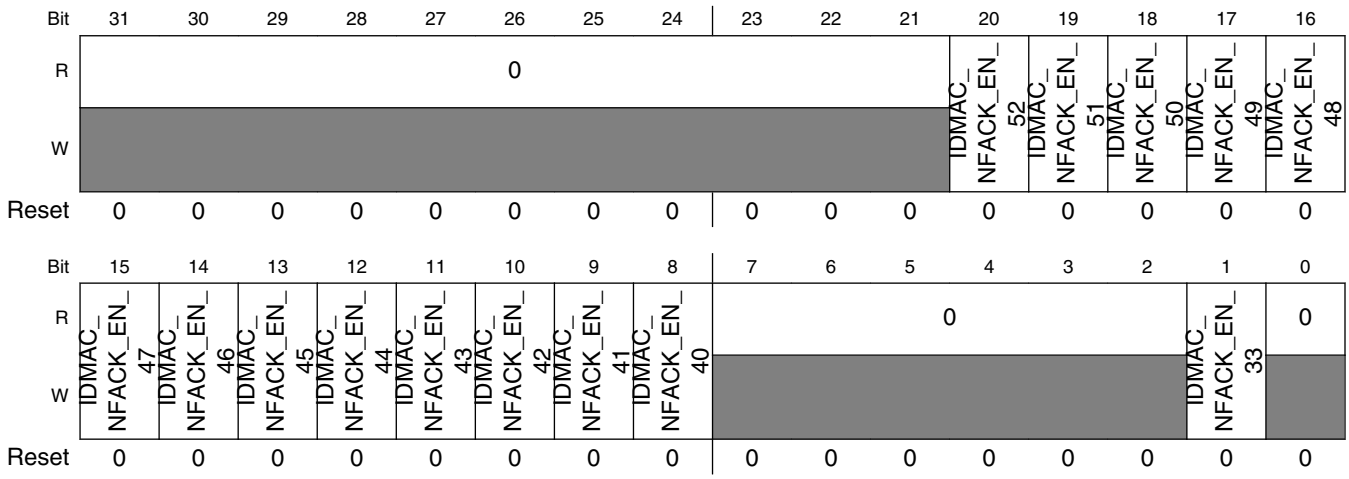
IPU_INT_CTRL_3 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
8 IDMAC_ NFACK_EN_8	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
5 IDMAC_ NFACK_EN_5	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
4 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
3 IDMAC_ NFACK_EN_3	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
2 IDMAC_ NFACK_EN_2	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_ NFACK_EN_1	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 IDMAC_ NFACK_EN_0	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

45.51.9 Interrupt Control Register 4 (IPU_INT_CTRL_4)

This register contains part of IPU interrupts controls. The controls of NFACK (New Frame Ack) of DMA Channels interrupts [63:32] can be found in this register.

Address: IPU_INT_CTRL_4 is 1E00_0000h base + 48h offset = 1E00_0048h



IPU_INT_CTRL_4 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_ NFACK_EN_52	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_ NFACK_EN_51	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
18 IDMAC_ NFACK_EN_50	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
17 IDMAC_ NFACK_EN_49	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_CTRL_4 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
16 IDMAC_ NFACK_EN_48	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 IDMAC_ NFACK_EN_47	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_ NFACK_EN_46	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
13 IDMAC_ NFACK_EN_45	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
12 IDMAC_ NFACK_EN_44	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_ NFACK_EN_43	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10 IDMAC_ NFACK_EN_42	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_ NFACK_EN_41	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
8 IDMAC_ NFACK_EN_40	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_CTRL_4 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
7–2 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_ NFACK_EN_33	Enable New Frame Ack of Channel interrupt. This bit is the control of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.

45.51.10 Interrupt Control Register 5 (IPU_INT_CTRL_5)

This register contains part of IPU interrupts controls. The controls of the New-frame before end-of-frame indication (NFB4EOF) of DMA Channels interrupts [31:0] can be found in this register.

Address: IPU_INT_CTRL_5 is 1E00_0000h base + 4Ch offset = 1E00_004Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_NFB4EOF_EN_31	0	IDMAC_NFB4EOF_EN_29	IDMAC_NFB4EOF_EN_28	IDMAC_NFB4EOF_EN_27	IDMAC_NFB4EOF_EN_26	IDMAC_NFB4EOF_EN_25	IDMAC_NFB4EOF_EN_24	IDMAC_NFB4EOF_EN_23	IDMAC_NFB4EOF_EN_22	IDMAC_NFB4EOF_EN_21	IDMAC_NFB4EOF_EN_20	IDMAC_NFB4EOF_EN_19	IDMAC_NFB4EOF_EN_18	IDMAC_NFB4EOF_EN_17	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_NFB4EOF_EN_15	IDMAC_NFB4EOF_EN_14	IDMAC_NFB4EOF_EN_13	IDMAC_NFB4EOF_EN_12	IDMAC_NFB4EOF_EN_11	IDMAC_NFB4EOF_EN_10	IDMAC_NFB4EOF_EN_9	IDMAC_NFB4EOF_EN_8	0		IDMAC_NFB4EOF_EN_5	0	IDMAC_NFB4EOF_EN_3	IDMAC_NFB4EOF_EN_2	IDMAC_NFB4EOF_EN_1	IDMAC_NFB4EOF_EN_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_CTRL_5 field descriptions

Field	Description
31 IDMAC_ NFB4EOF_EN_ 31	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
29 IDMAC_ NFB4EOF_EN_ 29	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
28 IDMAC_ NFB4EOF_EN_ 28	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
27 IDMAC_ NFB4EOF_EN_ 27	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
26 IDMAC_ NFB4EOF_EN_ 26	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
25 IDMAC_ NFB4EOF_EN_ 25	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
24 IDMAC_ NFB4EOF_EN_ 24	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_5 field descriptions (continued)

Field	Description
23 IDMAC_ NFB4EOF_EN_ 23	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
22 IDMAC_ NFB4EOF_EN_ 22	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
21 IDMAC_ NFB4EOF_EN_ 21	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_ NFB4EOF_EN_ 20	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_ NFB4EOF_EN_ 19	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
18 IDMAC_ NFB4EOF_EN_ 18	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
17 IDMAC_ NFB4EOF_EN_ 17	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
16 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_5 field descriptions (continued)

Field	Description
15 IDMAC_ NFB4EOF_EN_ 15	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_ NFB4EOF_EN_ 14	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
13 IDMAC_ NFB4EOF_EN_ 13	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
12 IDMAC_ NFB4EOF_EN_ 12	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_ NFB4EOF_EN_ 11	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10 IDMAC_ NFB4EOF_EN_ 10	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_ NFB4EOF_EN_9	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
8 IDMAC_ NFB4EOF_EN_8	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

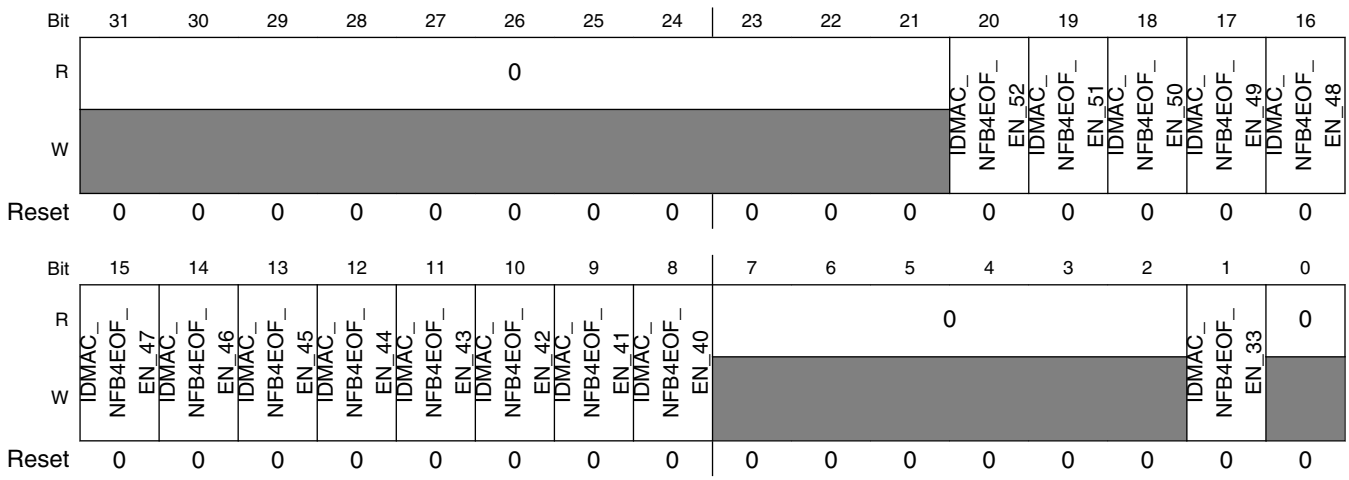
IPU_INT_CTRL_5 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
5 IDMAC_ NFB4EOF_EN_5	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
4 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
3 IDMAC_ NFB4EOF_EN_3	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
2 IDMAC_ NFB4EOF_EN_2	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_ NFB4EOF_EN_1	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 IDMAC_ NFB4EOF_EN_0	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

45.51.11 Interrupt Control Register 6 (IPU_INT_CTRL_6)

This register contains part of IPU interrupts controls. The controls of the New-frame before end-of-frame indication (NFB4EOF_ERR) of DMA Channels interrupts [63:32] can be found in this register.

Address: IPU_INT_CTRL_6 is 1E00_0000h base + 50h offset = 1E00_0050h



IPU_INT_CTRL_6 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_ NFB4EOF_EN_ 52	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_ NFB4EOF_EN_ 51	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
18 IDMAC_ NFB4EOF_EN_ 50	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_6 field descriptions (continued)

Field	Description
17 IDMAC_ NFB4EOF_EN_ 49	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
16 IDMAC_ NFB4EOF_EN_ 48	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 IDMAC_ NFB4EOF_EN_ 47	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_ NFB4EOF_EN_ 46	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
13 IDMAC_ NFB4EOF_EN_ 45	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
12 IDMAC_ NFB4EOF_EN_ 44	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_ NFB4EOF_EN_ 43	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10 IDMAC_ NFB4EOF_EN_ 42	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

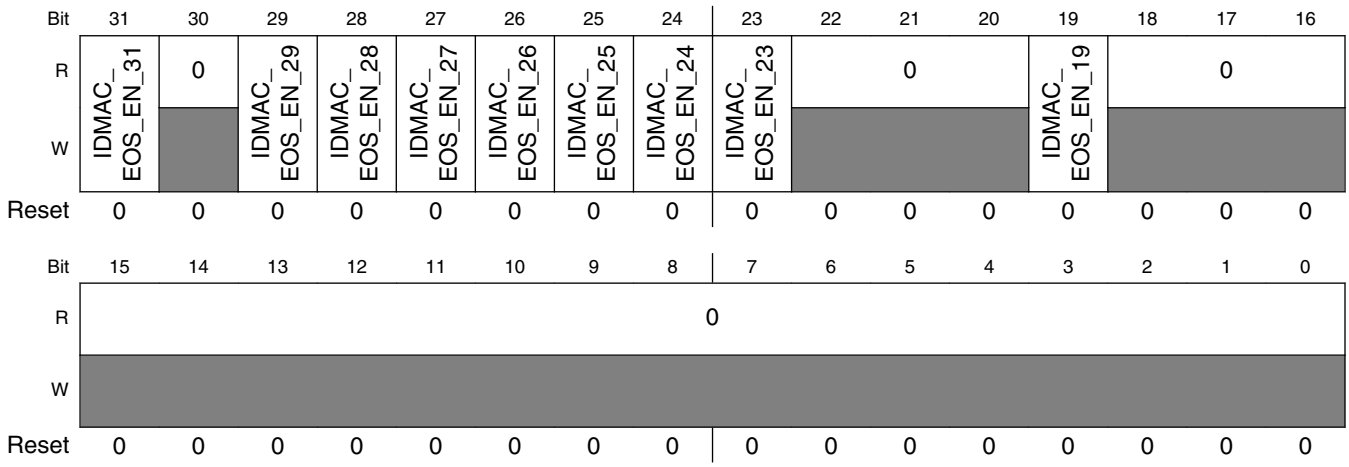
IPU_INT_CTRL_6 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_ NFB4EOF_EN_ 41	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
8 IDMAC_ NFB4EOF_EN_ 40	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
7–2 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_ NFB4EOF_EN_ 33	New Frame before end-of-frame error indication of Channel interrupt. This bit is the control of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.

45.51.12 Interrupt Control Register 7 (IPU_INT_CTRL_7)

This register contains part of IPUIPU interrupts controls. The controls of the End-of-Scroll indication (EOS) of DMA Channels interrupts [31:0] can be found in this register.

Address: IPU_INT_CTRL_7 is 1E00_0000h base + 54h offset = 1E00_0054h



IPU_INT_CTRL_7 field descriptions

Field	Description
31 IDMAC_EOS_EN_31	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
29 IDMAC_EOS_EN_29	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
28 IDMAC_EOS_EN_28	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
27 IDMAC_EOS_EN_27	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

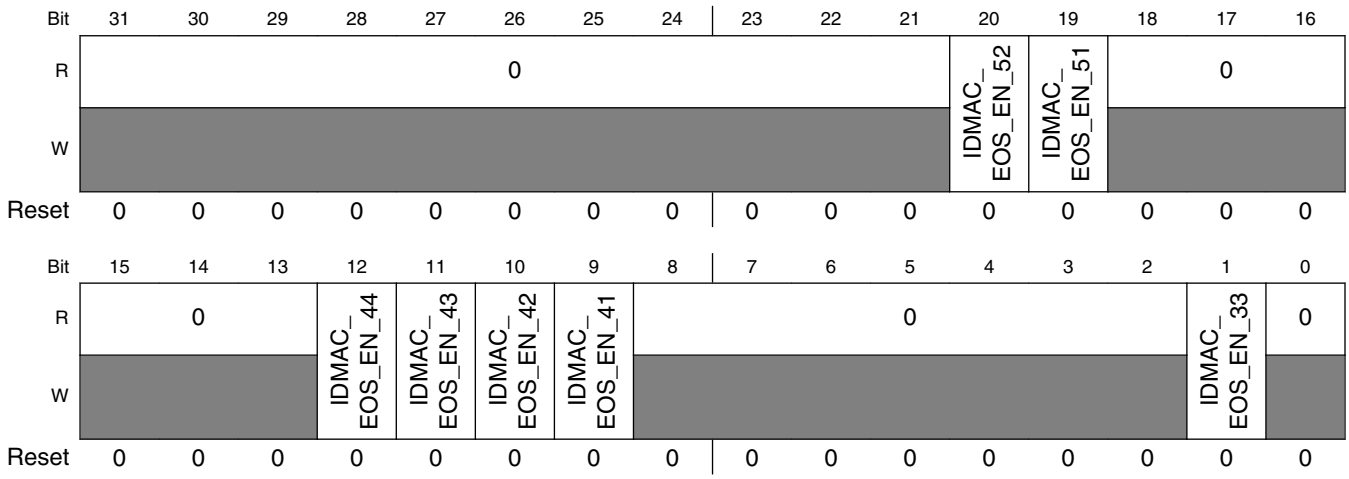
IPU_INT_CTRL_7 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
26 IDMAC_EOS_EN_26	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
25 IDMAC_EOS_EN_25	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
24 IDMAC_EOS_EN_24	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
23 IDMAC_EOS_EN_23	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
22–20 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_EOS_EN_19	End of Scroll indication of Channel interrupt. This bit is the control of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
18–0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.

45.51.13 Interrupt Control Register 8 (IPU_INT_CTRL_8)

This register contains part of IPU interrupts controls. The controls of the End of Scroll indication (EOS) of DMA Channels interrupts [63:32] can be found in this register.

Address: IPU_INT_CTRL_8 is 1E00_0000h base + 58h offset = 1E00_0058h



IPU_INT_CTRL_8 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_EOS_EN_52	End of Scroll of Channel interrupt. This bit is the control of End Of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_EOS_EN_51	End of Scroll of Channel interrupt. This bit is the control of End Of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
12 IDMAC_EOS_EN_44	End of Scroll of Channel interrupt. This bit is the control of End Of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

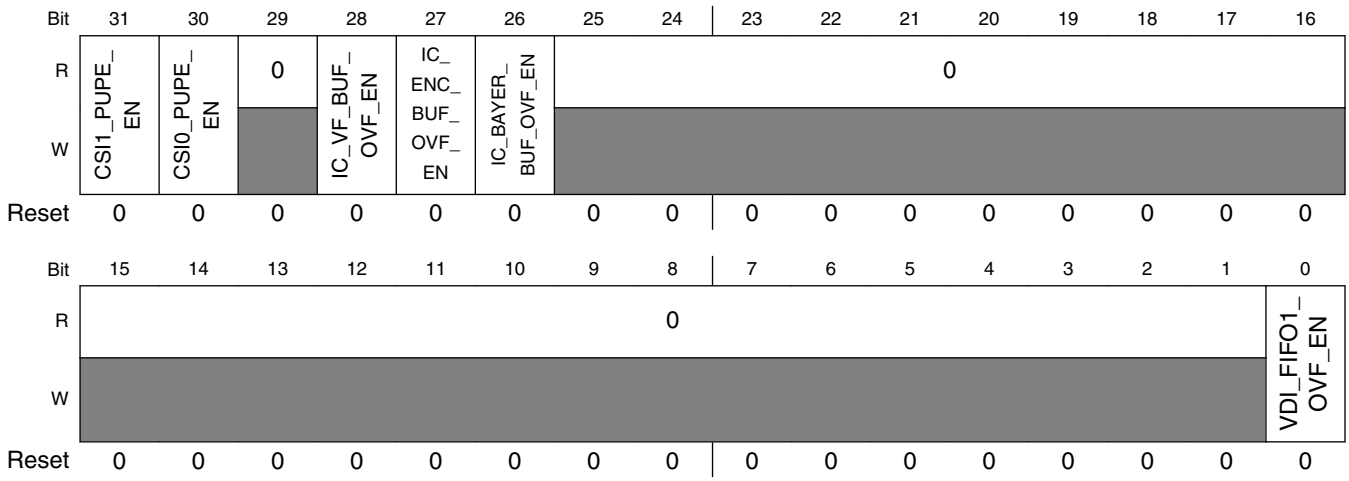
IPU_INT_CTRL_8 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_EOS_EN_43	End of Scroll of Channel interrupt. This bit is the control of End Of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10 IDMAC_EOS_EN_42	End of Scroll of Channel interrupt. This bit is the control of End Of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_EOS_EN_41	End of Scroll of Channel interrupt. This bit is the control of End Of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
8–2 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_EOS_EN_33	End of Scroll of Channel interrupt. This bit is the control of End Of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.

45.51.14 Interrupt Control Register 9 (IPU_INT_CTRL_9)

This register contains part of IPU interrupts controls. This register controls error interrupt signals coming from different sub-blocks within

Address: IPU_INT_CTRL_9 is 1E00_0000h base + 5Ch offset = 1E00_005Ch



IPU_INT_CTRL_9 field descriptions

Field	Description
31 CSI1_PUPE_EN	CSI1_PUPE_EN - CSI1 parameters update error interrupt enable. This bit enables an interrupt that is a result of an error generated by the CSI1. The error is generated in case where new frame arrived from the CSI1 before the completion of the CSI1's parameters update by the SRM 0 Interrupt is disabled. 1 Interrupt is enabled.
30 CSI0_PUPE_EN	CSI0_PUPE_EN - CSI0 parameters update error interrupt enable. This bit enables an interrupt that is a result of an error generated by the CSI0. The error is generated in case where new frame arrived from the CSI0 before the completion of the CSI0's parameters update by the SRM 0 Interrupt is disabled. 1 Interrupt is enabled.
29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28 IC_VF_BUF_OVF_EN	This bit enables an interrupt that is a result of the IC Buffer overflow for view finder coming from the IC. The user needs to write 1 to this bit in order to clear it. 0 Interrupt is disabled. 1 Interrupt is enabled.
27 IC_ENC_BUF_OVF_EN	This bit enables an interrupt that is a result of the IC Buffer overflow for encoding coming from the IC. The user needs to write 1 to this bit in order to clear it.

Table continues on the next page...

IPU_INT_CTRL_9 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
26 IC_BAYER_ BUF_OVF_EN	This bit enables an interrupt that is a result of the IC Buffer overflow for bayer coming from the IC. The user needs to write 1 to this bit in order to clear it. 0 Interrupt is disabled. 1 Interrupt is enabled.
25–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 VDI_FIFO1_ OVF_EN	FIFO1 overflow Interrupt1 Enable The VDIC generates FIFO1 overflow interrupt1 when the write pointer of FIFO1 overruns read pointer. 0 Interrupt is disabled. 1 Interrupt is enabled.

45.51.15 Interrupt Control Register 10 (IPU_INT_CTRL_10)

This register contains part of IPU interrupts controls. This register controls error interrupt signals coming from different modules within

Address: IPU_INT_CTRL_10 is 1E00_0000h base + 60h offset = 1E00_0060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	AXIR_ERR_EN	AXIW_ERR_EN	NON_PRIVILEGED_ACC_ERR_EN	0	IC_BAYER_FRM_LOST_ERR_EN	IC_ENC_FRM_LOST_ERR_EN	IC_VF_FRM_LOST_ERR_EN	0	DI1_TIME_OUT_ERR_EN	DI0_TIME_OUT_ERR_EN	DI1_SYNC_DISP_ERR_EN	DI0_SYNC_DISP_ERR_EN	DC_TEARING_ERR_6_EN	DC_TEARING_ERR_2_EN	DC_TEARING_ERR_1_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												SMFC3_FRM_LOST_EN	SMFC2_FRM_LOST_EN	SMFC1_FRM_LOST_EN	SMFC0_FRM_LOST_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_CTRL_10 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IPU_INT_CTRL_10 field descriptions (continued)

Field	Description
30 AXIR_ERR_EN	This bit enables an interrupt that is a result of AXI read access resulted with error response. 0 Interrupt is disabled. 1 Interrupt is enabled.
29 AXIW_ERR_EN	This bit enables an interrupt that is a result of AXI write access resulted with error response. 0 Interrupt is disabled. 1 Interrupt is enabled.
28 NON_ PRIVILEGED_ ACC_ERR_EN	Non Privileged Access Error interrupt enable. The CPMEM and the DP can be accessed by the ARM platform in privileged mode only HPROT[1] =1. An attempt to access these regions in user mode will issue an interrupt. This bit enables the interrupt. 0 Interrupt is disabled. 1 Interrupt is enabled.
27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26 IC_BAYER_ FRM_LOST_ ERR_EN	This bit enables an interrupt that is a result of IC's Bayer frame lost. 0 Interrupt is disabled. 1 Interrupt is enabled.
25 IC_ENC_FRM_ LOST_ERR_EN	This bit enables an interrupt that is a result of IC's encoding frame lost. 0 Interrupt is disabled. 1 Interrupt is enabled.
24 IC_VF_FRM_ LOST_ERR_EN	This bit enables an interrupt that is a result of IC's view finder frame lost. 0 Interrupt is disabled. 1 Interrupt is enabled.
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22 DI1_TIME_ OUT_ERR_EN	DI1 time out error interrupt enable This bit enables the interrupt that is a result of a time out error during a read access via DI1 0 Interrupt is disabled. 1 Interrupt is enabled.
21 DI0_TIME_ OUT_ERR_EN	DI0 time outwore interrupt enable This bit enables the interrupt that is a result of a time out error during a read access via DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
20 DI1_SYNC_ DISP_ERR_EN	DI1 Synchronous display error enable This bit enables the interrupt that is a result of an error during access to a synchronous display via DI1 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_10 field descriptions (continued)

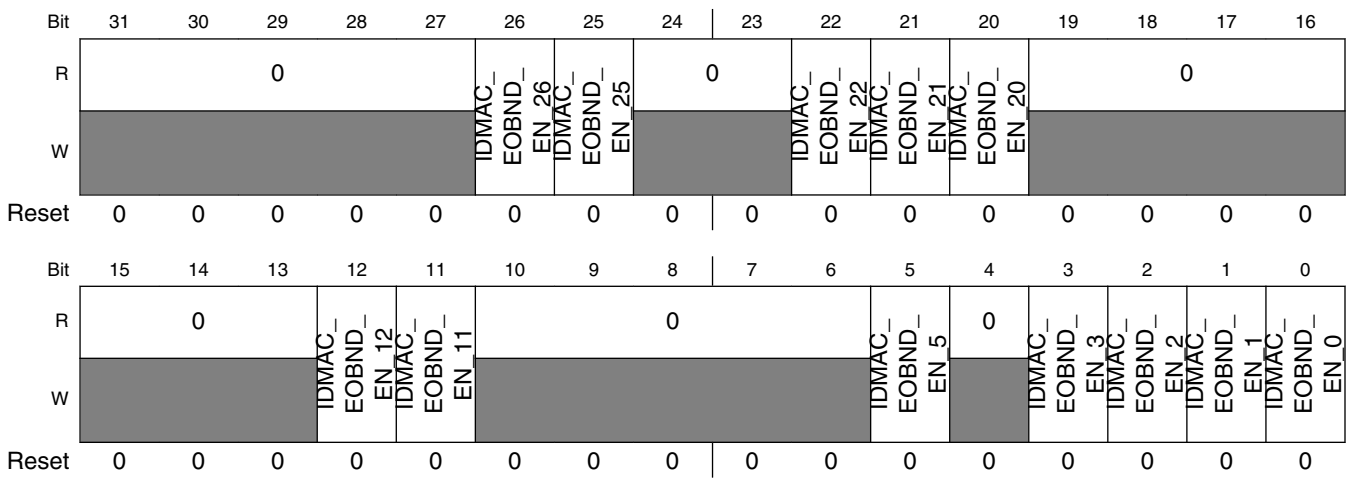
Field	Description
19 DI0_SYNC_ DISP_ERR_EN	DI0 Synchronous display error enable This bit enables the interrupt that is a result of an error during access to a synchronous display via DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
18 DC_TEARING_ ERR_6_EN	Tearing Error #6 enable This bit enables the interrupt that is a result of tearing error while the anti tearing mechanism is activated for DC channel 6 0 Interrupt is disabled. 1 Interrupt is enabled.
17 DC_TEARING_ ERR_2_EN	Tearing Error #2 enable This bit enables the interrupt that is a result of tearing error while the anti tearing mechanism is activated for DC channel 2 0 Interrupt is disabled. 1 Interrupt is enabled.
16 DC_TEARING_ ERR_1_EN	Tearing Error #1 enable This bit enables the interrupt that is a result of tearing error while the anti tearing mechanism is activated for DC channel 1 0 Interrupt is disabled. 1 Interrupt is enabled.
15–4 Reserved	This read-only field is reserved and always has the value zero. Reserved
3 SMFC3_FRM_ LOST_EN	Frame Lost of SMFC channel 3 interrupt enable bit This bit enables an interrupt that is a result of a Frame Lost of SMFC channel 3. 0 Interrupt is disabled. 1 Interrupt is enabled.
2 SMFC2_FRM_ LOST_EN	Frame Lost of SMFC channel 2 interrupt enable bit This bit enables an interrupt that is a result of a Frame Lost of SMFC channel 2. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 SMFC1_FRM_ LOST_EN	Frame Lost of SMFC channel 1 interrupt enable bit This bit enables an interrupt that is a result of a Frame Lost of SMFC channel 1. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 SMFC0_FRM_ LOST_EN	Frame Lost of SMFC channel 0 interrupt enable bit This bit enables an interrupt that is a result of a Frame Lost of SMFC channel 0. 0 Interrupt is disabled. 1 Interrupt is enabled.

45.51.16 Interrupt Control Register 11 (IPU_INT_CTRL_11)

This register contains part of IPU interrupts controls. The controls of the end-of-band indication (EOBND) of DMA Channels interrupts [31:0] can be found in this register.

- Hide VDOA_SYNC for all versions
- Show VDOA_SYNC for IPUv3H version.
- The table below tagged with other settings (like IPU3M_only) should be hidden in IPUv3H version.

Address: IPU_INT_CTRL_11 is 1E00_0000h base + 64h offset = 1E00_0064h



IPU_INT_CTRL_11 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
26 IDMAC_ EOBND_EN_26	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
25 IDMAC_ EOBND_EN_25	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
24–23 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_INT_CTRL_11 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
22 IDMAC_ EOBND_EN_22	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
21 IDMAC_ EOBND_EN_21	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_ EOBND_EN_20	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19–13 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
12 IDMAC_ EOBND_EN_12	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_ EOBND_EN_11	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10–6 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
5 IDMAC_ EOBND_EN_5	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
4 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

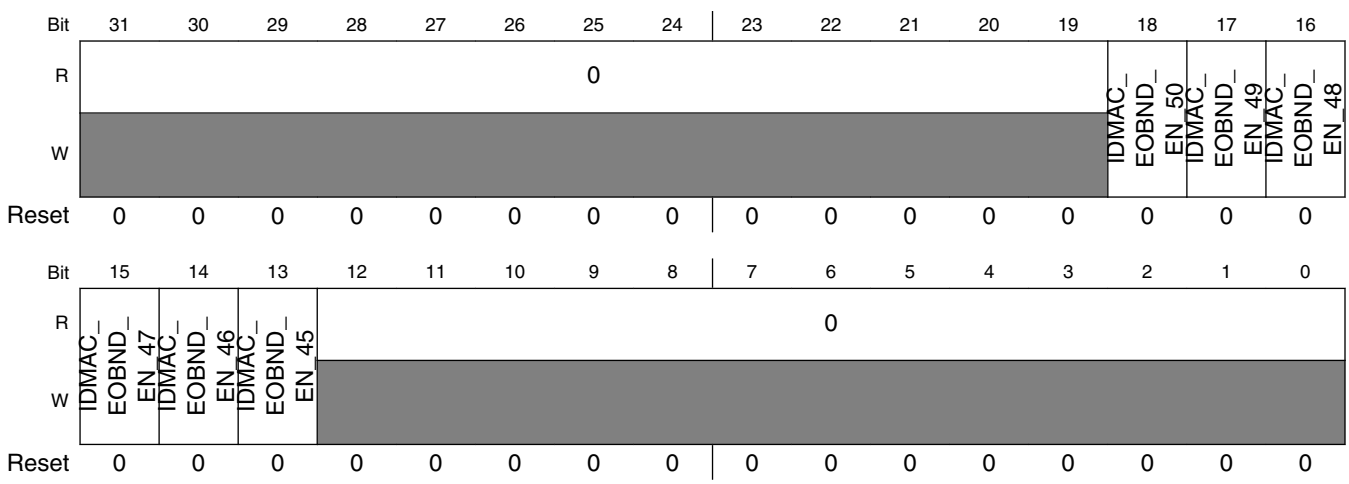
IPU_INT_CTRL_11 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
3 IDMAC_ EOBND_EN_3	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
2 IDMAC_ EOBND_EN_2	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_ EOBND_EN_1	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 IDMAC_ EOBND_EN_0	end-of-band indication of Channel interrupt. This bit is the control of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

45.51.17 Interrupt Control Register 12 (IPU_INT_CTRL_12)

This register contains part of IPU interrupts controls. The controls of the end-of-band indication (EOBND) of DMA Channels interrupts [63:32] can be found in this register.

Address: IPU_INT_CTRL_12 is 1E00_0000h base + 68h offset = 1E00_0068h



IPU_INT_CTRL_12 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
18 IDMAC_ EOBND_EN_50	end-of-band indication of Channel interrupt. This bit is the control end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
17 IDMAC_ EOBND_EN_49	end-of-band indication of Channel interrupt. This bit is the control end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
16 IDMAC_ EOBND_EN_48	end-of-band indication of Channel interrupt. This bit is the control end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 IDMAC_ EOBND_EN_47	end-of-band indication of Channel interrupt. This bit is the control end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_ EOBND_EN_46	end-of-band indication of Channel interrupt. This bit is the control end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
13 IDMAC_ EOBND_EN_45	end-of-band indication of Channel interrupt. This bit is the control end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
12–0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.

45.51.18 Interrupt Control Register 13 (IPU_INT_CTRL_13)

This register contains part of IPU interrupts controls. The controls of the threshold crossing indication (TH) of DMA Channels interrupts [31:0] can be found in this register.

Address: IPU_INT_CTRL_13 is 1E00_0000h base + 6Ch offset = 1E00_006Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_TH_EN_31	0	IDMAC_TH_EN_29	IDMAC_TH_EN_28	IDMAC_TH_EN_27	IDMAC_TH_EN_26	IDMAC_TH_EN_25	IDMAC_TH_EN_24	IDMAC_TH_EN_23	IDMAC_TH_EN_22	IDMAC_TH_EN_21	IDMAC_TH_EN_20	IDMAC_TH_EN_19	IDMAC_TH_EN_18	IDMAC_TH_EN_17	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_TH_EN_15	IDMAC_TH_EN_14	IDMAC_TH_EN_13	IDMAC_TH_EN_12	IDMAC_TH_EN_11	IDMAC_TH_EN_10	IDMAC_TH_EN_9	IDMAC_TH_EN_8	0		IDMAC_TH_EN_5	0	IDMAC_TH_EN_3	IDMAC_TH_EN_2	IDMAC_TH_EN_1	IDMAC_TH_EN_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_CTRL_13 field descriptions

Field	Description
31 IDMAC_TH_EN_31	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
29 IDMAC_TH_EN_29	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
28 IDMAC_TH_EN_28	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_13 field descriptions (continued)

Field	Description
27 IDMAC_TH_EN_27	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
26 IDMAC_TH_EN_26	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
25 IDMAC_TH_EN_25	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
24 IDMAC_TH_EN_24	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
23 IDMAC_TH_EN_23	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
22 IDMAC_TH_EN_22	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
21 IDMAC_TH_EN_21	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_TH_EN_20	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_CTRL_13 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_TH_EN_19	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
18 IDMAC_TH_EN_18	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
17 IDMAC_TH_EN_17	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
16 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 IDMAC_TH_EN_15	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_TH_EN_14	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
13 IDMAC_TH_EN_13	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
12 IDMAC_TH_EN_12	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_CTRL_13 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_TH_EN_11	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10 IDMAC_TH_EN_10	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_TH_EN_9	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
8 IDMAC_TH_EN_8	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
5 IDMAC_TH_EN_5	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
4 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
3 IDMAC_TH_EN_3	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_CTRL_13 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
2 IDMAC_TH_EN_2	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_TH_EN_1	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 IDMAC_TH_EN_0	Threshold crossing indication of Channel interrupt. This bit is the control of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

45.51.19 Interrupt Control Register 14 (IPU_INT_CTRL_14)

This register contains part of IPU interrupts controls. The controls of the Threshold crossing indication (TH) of DMA Channels interrupts [63:32] can be found in this register.

Address: IPU_INT_CTRL_14 is 1E00_0000h base + 70h offset = 1E00_0070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											IDMAC_TH_EN_52	IDMAC_TH_EN_51	IDMAC_TH_EN_50	IDMAC_TH_EN_49	IDMAC_TH_EN_48
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_TH_EN_47	IDMAC_TH_EN_46	IDMAC_TH_EN_45	IDMAC_TH_EN_44	IDMAC_TH_EN_43	IDMAC_TH_EN_42	IDMAC_TH_EN_41	IDMAC_TH_EN_40	0						IDMAC_TH_EN_33	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_CTRL_14 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
20 IDMAC_TH_EN_52	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
19 IDMAC_TH_EN_51	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
18 IDMAC_TH_EN_50	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
17 IDMAC_TH_EN_49	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
16 IDMAC_TH_EN_48	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 IDMAC_TH_EN_47	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
14 IDMAC_TH_EN_46	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_14 field descriptions (continued)

Field	Description
13 IDMAC_TH_EN_45	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
12 IDMAC_TH_EN_44	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
11 IDMAC_TH_EN_43	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
10 IDMAC_TH_EN_42	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
9 IDMAC_TH_EN_41	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
8 IDMAC_TH_EN_40	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.
7–2 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 IDMAC_TH_EN_33	Threshold crossing indication of Channel interrupt. This bit is the control Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_14 field descriptions (continued)

Field	Description
0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is disabled. 1 Interrupt is enabled.

45.51.20 Interrupt Control Register15 (IPU_INT_CTRL_15)

This register contains part of IPU interrupts controls. The controls of general purpose interrupts can be found in this register.

Address: IPU_INT_CTRL_15 is 1E00_0000h base + 74h offset = 1E00_0074h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DI1_CNT_EN_PRE_8_EN	DI1_CNT_EN_PRE_3_EN	DI1_DISP_CLK_EN_PRE_EN	DI0_CNT_EN_PRE_10_EN	DI0_CNT_EN_PRE_9_EN	DI0_CNT_EN_PRE_8_EN	DI0_CNT_EN_PRE_7_EN	DI0_CNT_EN_PRE_6_EN	DI0_CNT_EN_PRE_5_EN	DI0_CNT_EN_PRE_4_EN	DI0_CNT_EN_PRE_3_EN	DI0_CNT_EN_PRE_2_EN	DI0_CNT_EN_PRE_1_EN	DI0_CNT_EN_PRE_0_EN	DC_ASYNC_STOP_EN	DC_DP_START_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DI_VSYNC_PRE_1_EN	DI_VSYNC_PRE_0_EN	DC_FC_6_EN	DC_FC_4_EN	DC_FC_3_EN	DC_FC_2_EN	DC_FC_1_EN	DC_FC_0_EN	DP_ASF_BRAKE_EN	DP_SF_BRAKE_EN	DP_ASF_END_EN	DP_ASF_START_EN	DP_SF_END_EN	DP_SF_START_EN	SNOOPING2_INT_EN	SNOOPING1_INT_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_CTRL_15 field descriptions

Field	Description
31 DI1_CNT_EN_PRE_8_EN	This bit enables the interrupt that is a result of a trigger generated by counter #8 of DI1 0 Interrupt is disabled. 1 Interrupt is enabled.
30 DI1_CNT_EN_PRE_3_EN	This bit enables the interrupt that is a result of a trigger generated by counter #3 of DI1 0 Interrupt is disabled. 1 Interrupt is enabled.
29 DI1_DISP_CLK_EN_PRE_EN	DI1_DISP_CLK_EN_PRE_EN 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_15 field descriptions (continued)

Field	Description
28 DI0_CNT_EN_ PRE_10_EN	This bit enables the interrupt that is a result of a trigger generated by counter #10 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
27 DI0_CNT_EN_ PRE_9_EN	This bit enables the interrupt that is a result of a trigger generated by counter #9 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
26 DI0_CNT_EN_ PRE_8_EN	This bit enables the interrupt that is a result of a trigger generated by counter #8 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
25 DI0_CNT_EN_ PRE_7_EN	This bit enables the interrupt that is a result of a trigger generated by counter #7 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
24 DI0_CNT_EN_ PRE_6_EN	This bit enables the interrupt that is a result of a trigger generated by counter #6 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
23 DI0_CNT_EN_ PRE_5_EN	This bit enables the interrupt that is a result of a trigger generated by counter #5 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
22 DI0_CNT_EN_ PRE_4_EN	This bit enables the interrupt that is a result of a trigger generated by counter #4 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
21 DI0_CNT_EN_ PRE_3_EN	This bit enables the interrupt that is a result of a trigger generated by counter #3 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
20 DI0_CNT_EN_ PRE_2_EN	This bit enables the interrupt that is a result of a trigger generated by counter #2 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
19 DI0_CNT_EN_ PRE_1_EN	This bit enables the interrupt that is a result of a trigger generated by counter #1 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
18 DI0_CNT_EN_ PRE_0_EN	This bit enables the interrupt that is a result of a trigger generated by counter #0 of DI0 0 Interrupt is disabled. 1 Interrupt is enabled.
17 DC_ASYNC_ STOP_EN	This bit enables the interrupt asserted anytime the DP stops an async flow and moves to a sync flow

Table continues on the next page...

IPU_INT_CTRL_15 field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
16 DC_DP_ START_EN	This bit enables the interrupt asserted anytime the DP start a new sync or async flow or when an async flow is interrupted by a sync flow 0 Interrupt is disabled. 1 Interrupt is enabled.
15 DI_VSYNC_ PRE_1_EN	This bit enables the DI1 interrupt indicating of a VSYNC signal asserted 2 rows before the VSYNC sent to the display 0 Interrupt is disabled. 1 Interrupt is enabled.
14 DI_VSYNC_ PRE_0_EN	This bit enables the DI0 interrupt indicating of a VSYNC signal asserted 2 rows before the VSYNC sent to the display 0 Interrupt is disabled. 1 Interrupt is enabled.
13 DC_FC_6_EN	This bit enables the DC Frame Complete on channel #6 interrupt 0 Interrupt is disabled. 1 Interrupt is enabled.
12 DC_FC_4_EN	This bit enables the DC Frame Complete on channel #4 interrupt 0 Interrupt is disabled. 1 Interrupt is enabled.
11 DC_FC_3_EN	This bit enables the DC Frame Complete on channel #3 interrupt 0 Interrupt is disabled. 1 Interrupt is enabled.
10 DC_FC_2_EN	This bit enables the DC Frame Complete on channel #2 interrupt 0 Interrupt is disabled. 1 Interrupt is enabled.
9 DC_FC_1_EN	This bit enables they'd Frame Complete on channel #1 interrupt 0 Interrupt is disabled. 1 Interrupt is enabled.
8 DC_FC_0_EN	This bit enables they'd Frame Complete on channel #0 interrupt 0 Interrupt is disabled. 1 Interrupt is enabled.
7 DP_ASF_ BRAKE_EN	DP Async Flow Brake enable bit. This bit enables the interrupt that is a result of the async flow brake at the DP 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

IPU_INT_CTRL_15 field descriptions (continued)

Field	Description
6 DP_SF_ BRAKE_EN	DP Sync Flow Brake enable bit. This bit enables the interrupt that is a result of the Sync flow brake at the DP 0 Interrupt is disabled. 1 Interrupt is enabled.
5 DP_ASF_END_ EN	DP Async Flow End enable bit. This bit enables the interrupt that is a result of the Async flow end at the DP 0 Interrupt is disabled. 1 Interrupt is enabled.
4 DP_ASF_ START_EN	DP Async Flow Start enable bit. This bit enables the interrupt that is a result of the Async flow start at the DP 0 Interrupt is disabled. 1 Interrupt is enabled.
3 DP_SF_END_ EN	DP Sync Flow End enable bit. This bit enables the interrupt that is a result of the Sync flow end at the DP 0 Interrupt is disabled. 1 Interrupt is enabled.
2 DP_SF_START_ EN	DP Sync Flow Start enable bit. This bit enables the interrupt that is a result of the Sync flow start at the DP 0 Interrupt is disabled. 1 Interrupt is enabled.
1 SNOOPING2_ INT_EN	IPU snooping 2 interrupt enable bit. This bit enables the interrupt that is a result of the detection of a snooping 2 signal assertion coming to the IPU 0 Interrupt is disabled. 1 Interrupt is enabled.
0 SNOOPING1_ INT_EN	IPU snooping 1 interrupt enable bit. This bit enables the interrupt that is a result of the detection of a snooping 1 signal assertion coming to the IPU 0 Interrupt is disabled. 1 Interrupt is enabled.

45.51.21 SDMA Event Control Register 1 (IPU_SDMA_EVENT_1)

This register contains part of IPU SDMA events controls. The controls of EOF (end of frame) of DMA Channels SDMA events [31:0] can be found in this register.

Address: IPU_SDMA_EVENT_1 is 1E00_0000h base + 78h offset = 1E00_0078h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_EOF_SDMA_EN_31	0	IDMAC_EOF_SDMA_EN_29	IDMAC_EOF_SDMA_EN_28	IDMAC_EOF_SDMA_EN_27	IDMAC_EOF_SDMA_EN_26	IDMAC_EOF_SDMA_EN_25	IDMAC_EOF_SDMA_EN_24	IDMAC_EOF_SDMA_EN_23	IDMAC_EOF_SDMA_EN_22	IDMAC_EOF_SDMA_EN_21	IDMAC_EOF_SDMA_EN_20	IDMAC_EOF_SDMA_EN_19	IDMAC_EOF_SDMA_EN_18	IDMAC_EOF_SDMA_EN_17	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_EOF_SDMA_EN_15	IDMAC_EOF_SDMA_EN_14	IDMAC_EOF_SDMA_EN_13	IDMAC_EOF_SDMA_EN_12	IDMAC_EOF_SDMA_EN_11	IDMAC_EOF_SDMA_EN_10	IDMAC_EOF_SDMA_EN_9	IDMAC_EOF_SDMA_EN_8	0		IDMAC_EOF_SDMA_EN_5	0	IDMAC_EOF_SDMA_EN_3	IDMAC_EOF_SDMA_EN_2	IDMAC_EOF_SDMA_EN_1	IDMAC_EOF_SDMA_EN_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_SDMA_EVENT_1 field descriptions

Field	Description
31 IDMAC_EOF_SDMA_EN_31	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
29 IDMAC_EOF_SDMA_EN_29	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
28 IDMAC_EOF_SDMA_EN_28	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
27 IDMAC_EOF_SDMA_EN_27	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_SDMA_EVENT_1 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
26 IDMAC_EOF_ SDMA_EN_26	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
25 IDMAC_EOF_ SDMA_EN_25	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
24 IDMAC_EOF_ SDMA_EN_24	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
23 IDMAC_EOF_ SDMA_EN_23	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
22 IDMAC_EOF_ SDMA_EN_22	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
21 IDMAC_EOF_ SDMA_EN_21	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
20 IDMAC_EOF_ SDMA_EN_20	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
19 IDMAC_EOF_ SDMA_EN_19	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
18 IDMAC_EOF_ SDMA_EN_18	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_SDMA_EVENT_1 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
17 IDMAC_EOF_SDMA_EN_17	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
16 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
15 IDMAC_EOF_SDMA_EN_15	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
14 IDMAC_EOF_SDMA_EN_14	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
13 IDMAC_EOF_SDMA_EN_13	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
12 IDMAC_EOF_SDMA_EN_12	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
11 IDMAC_EOF_SDMA_EN_11	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
10 IDMAC_EOF_SDMA_EN_10	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
9 IDMAC_EOF_SDMA_EN_9	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

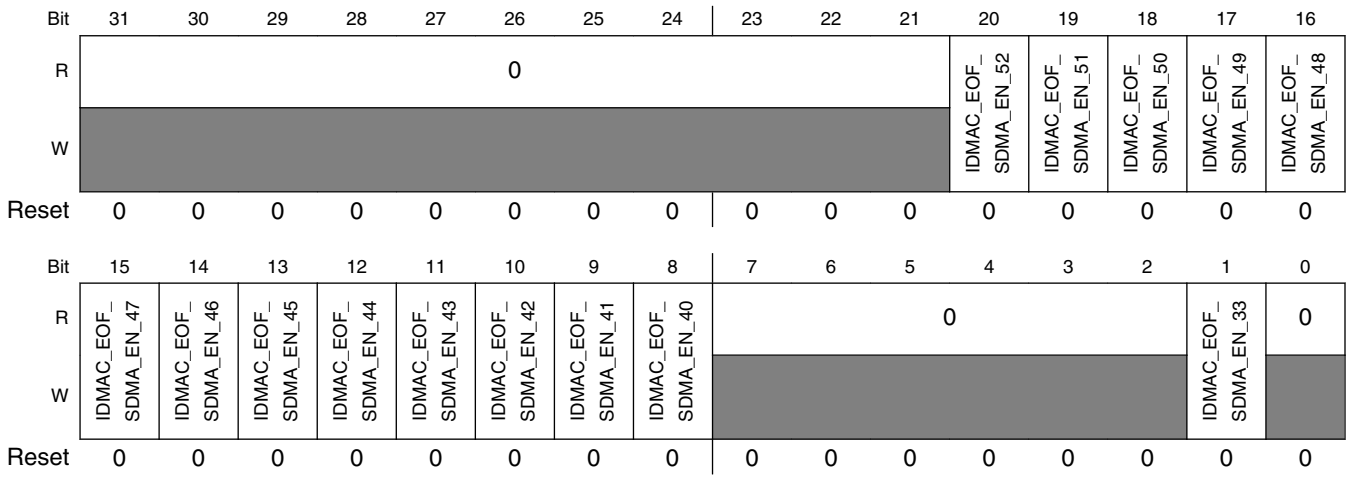
IPU_SDMA_EVENT_1 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
8 IDMAC_EOF_SDMA_EN_8	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
5 IDMAC_EOF_SDMA_EN_5	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
4 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
3 IDMAC_EOF_SDMA_EN_3	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
2 IDMAC_EOF_SDMA_EN_2	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
1 IDMAC_EOF_SDMA_EN_1	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
0 IDMAC_EOF_SDMA_EN_0	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

45.51.22 SDMA Event Control Register 2 (IPU_SDMA_EVENT_2)

This register contains part of IPU SDMA events controls. The controls of EOF (end of frame) of DMA Channels SDMA events [63:32] can be found in this register.

Address: IPU_SDMA_EVENT_2 is 1E00_0000h base + 7Ch offset = 1E00_007Ch



IPU_SDMA_EVENT_2 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
20 IDMAC_EOF_SDMA_EN_52	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
19 IDMAC_EOF_SDMA_EN_51	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
18 IDMAC_EOF_SDMA_EN_50	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
17 IDMAC_EOF_SDMA_EN_49	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_SDMA_EVENT_2 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
16 IDMAC_EOF_ SDMA_EN_48	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
15 IDMAC_EOF_ SDMA_EN_47	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
14 IDMAC_EOF_ SDMA_EN_46	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
13 IDMAC_EOF_ SDMA_EN_45	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
12 IDMAC_EOF_ SDMA_EN_44	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
11 IDMAC_EOF_ SDMA_EN_43	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
10 IDMAC_EOF_ SDMA_EN_42	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
9 IDMAC_EOF_ SDMA_EN_41	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
8 IDMAC_EOF_ SDMA_EN_40	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_SDMA_EVENT_2 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
7–2 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
1 IDMAC_EOF_ SDMA_EN_33	Enable End of Frame of Channel SDMA event. This bit is the control of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.

45.51.23 SDMA Event Control Register 3 (IPU_SDMA_EVENT_3)

This register contains part of IPU SDMA events controls. The controls of NFACK (New Frame Acknowledge) of DMA Channels SDMA events [31:0] can be found in this register.

Address: IPU_SDMA_EVENT_3 is 1E00_0000h base + 80h offset = 1E00_0080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_ NFACK_SDMA_ EN_31	0	IDMAC_ NFACK_SDMA_ EN_29	IDMAC_ NFACK_SDMA_ EN_28	IDMAC_ NFACK_SDMA_ EN_27	IDMAC_ NFACK_SDMA_ EN_26	IDMAC_ NFACK_SDMA_ EN_25	IDMAC_ NFACK_SDMA_ EN_24	IDMAC_ NFACK_SDMA_ EN_23	IDMAC_ NFACK_SDMA_ EN_22	IDMAC_ NFACK_SDMA_ EN_21	IDMAC_ NFACK_SDMA_ EN_20	IDMAC_ NFACK_SDMA_ EN_19	IDMAC_ NFACK_SDMA_ EN_18	IDMAC_ NFACK_SDMA_ EN_17	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_ NFACK_SDMA_ EN_15	IDMAC_ NFACK_SDMA_ EN_14	IDMAC_ NFACK_SDMA_ EN_13	IDMAC_ NFACK_SDMA_ EN_12	IDMAC_ NFACK_SDMA_ EN_11	IDMAC_ NFACK_SDMA_ EN_10	IDMAC_ NFACK_SDMA_ EN_9	IDMAC_ NFACK_SDMA_ EN_8	0		IDMAC_ NFACK_SDMA_ EN_5	0	IDMAC_ NFACK_SDMA_ EN_3	IDMAC_ NFACK_SDMA_ EN_2	IDMAC_ NFACK_SDMA_ EN_1	IDMAC_ NFACK_SDMA_ EN_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_SDMA_EVENT_3 field descriptions

Field	Description
31 IDMAC_ NFACK_SDMA_ EN_31	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
29 IDMAC_ NFACK_SDMA_ EN_29	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
28 IDMAC_ NFACK_SDMA_ EN_28	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
27 IDMAC_ NFACK_SDMA_ EN_27	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
26 IDMAC_ NFACK_SDMA_ EN_26	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
25 IDMAC_ NFACK_SDMA_ EN_25	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
24 IDMAC_ NFACK_SDMA_ EN_24	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

Table continues on the next page...

IPU_SDMA_EVENT_3 field descriptions (continued)

Field	Description
23 IDMAC_ NFACK_SDMA_ EN_23	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
22 IDMAC_ NFACK_SDMA_ EN_22	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
21 IDMAC_ NFACK_SDMA_ EN_21	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
20 IDMAC_ NFACK_SDMA_ EN_20	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
19 IDMAC_ NFACK_SDMA_ EN_19	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
18 IDMAC_ NFACK_SDMA_ EN_18	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
17 IDMAC_ NFACK_SDMA_ EN_17	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
16 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.

Table continues on the next page...

IPU_SDMA_EVENT_3 field descriptions (continued)

Field	Description
15 IDMAC_ NFACK_SDMA_ EN_15	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
14 IDMAC_ NFACK_SDMA_ EN_14	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
13 IDMAC_ NFACK_SDMA_ EN_13	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
12 IDMAC_ NFACK_SDMA_ EN_12	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
11 IDMAC_ NFACK_SDMA_ EN_11	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
10 IDMAC_ NFACK_SDMA_ EN_10	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
9 IDMAC_ NFACK_SDMA_ EN_9	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
8 IDMAC_ NFACK_SDMA_ EN_8	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

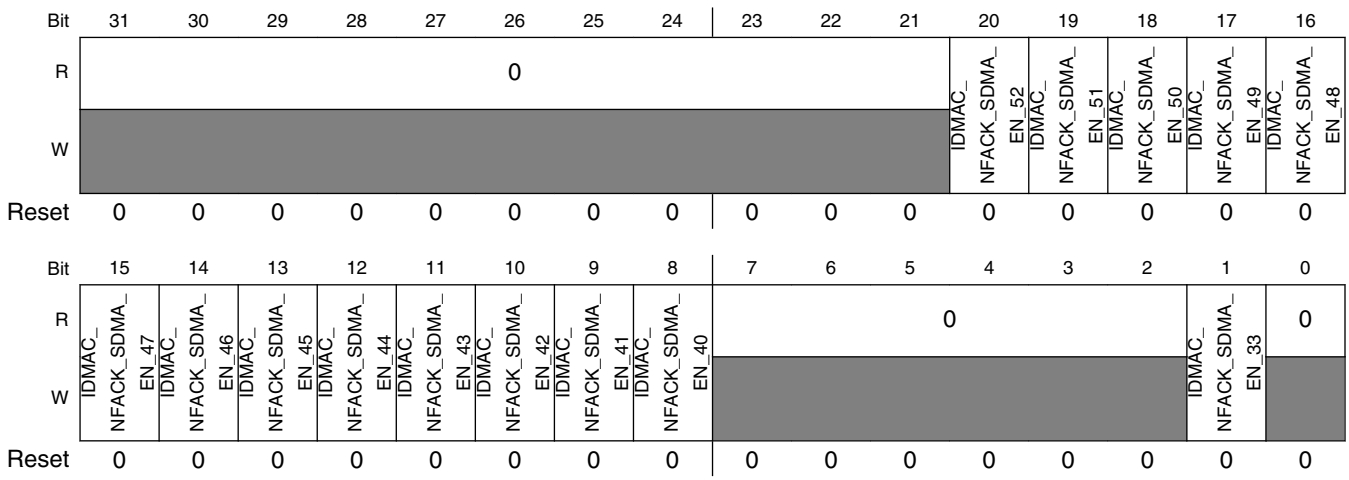
IPU_SDMA_EVENT_3 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
5 IDMAC_ NFACK_SDMA_ EN_5	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
4 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
3 IDMAC_ NFACK_SDMA_ EN_3	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
2 IDMAC_ NFACK_SDMA_ EN_2	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
1 IDMAC_ NFACK_SDMA_ EN_1	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
0 IDMAC_ NFACK_SDMA_ EN_0	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

45.51.24 SDMA Event Control Register 4 (IPU_SDMA_EVENT_4)

This register contains part of IPU SDMA events controls. The controls of NFACK (New Frame Acknowledge) of DMA Channels SDMA events [63:32] can be found in this register.

Address: IPU_SDMA_EVENT_4 is 1E00_0000h base + 84h offset = 1E00_0084h



IPU_SDMA_EVENT_4 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
20 IDMAC_ NFACK_SDMA_ EN_52	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
19 IDMAC_ NFACK_SDMA_ EN_51	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
18 IDMAC_ NFACK_SDMA_ EN_50	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

Table continues on the next page...

IPU_SDMA_EVENT_4 field descriptions (continued)

Field	Description
17 IDMAC_ NFACK_SDMA_ EN_49	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
16 IDMAC_ NFACK_SDMA_ EN_48	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
15 IDMAC_ NFACK_SDMA_ EN_47	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
14 IDMAC_ NFACK_SDMA_ EN_46	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
13 IDMAC_ NFACK_SDMA_ EN_45	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
12 IDMAC_ NFACK_SDMA_ EN_44	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
11 IDMAC_ NFACK_SDMA_ EN_43	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
10 IDMAC_ NFACK_SDMA_ EN_42	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

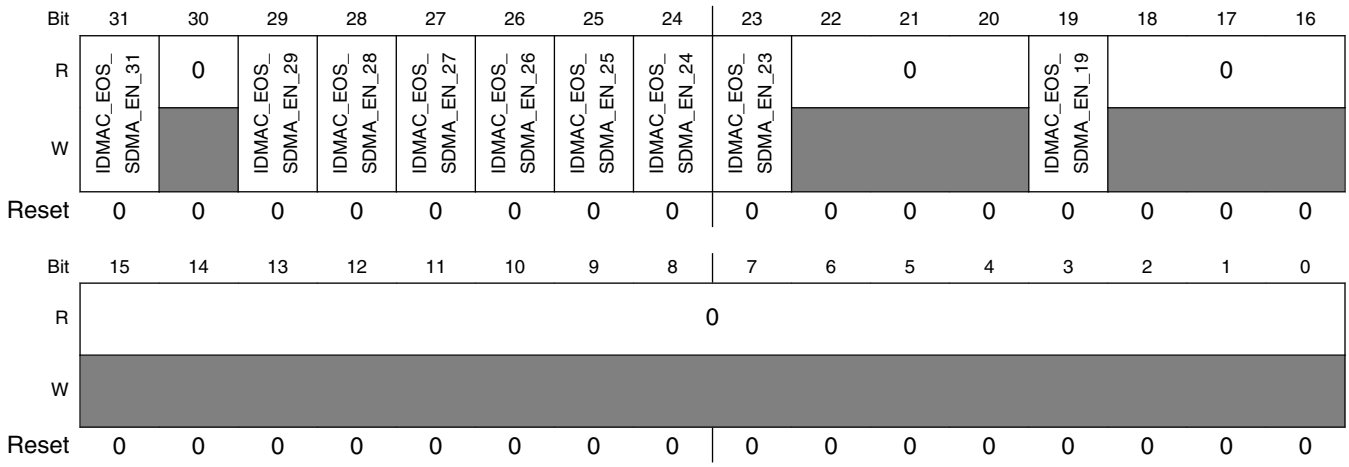
IPU_SDMA_EVENT_4 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
9 IDMAC_ NFACK_SDMA_ EN_41	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
8 IDMAC_ NFACK_SDMA_ EN_40	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
7–2 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
1 IDMAC_ NFACK_SDMA_ EN_33	Enable New Frame Acknowledge of Channel SDMA event. This bit is the control of New Frame Acknowledge of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.

45.51.25 SDMA Event Control Register 7 (IPU_SDMA_EVENT_7)

This register contains part of IPU SDMA events controls. The controls of EOS (End of Scroll) of DMA Channels SDMA events [31:0] can be found in this register.

Address: IPU_SDMA_EVENT_7 is 1E00_0000h base + 88h offset = 1E00_0088h



IPU_SDMA_EVENT_7 field descriptions

Field	Description
31 IDMAC_EOS_SDMA_EN_31	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
29 IDMAC_EOS_SDMA_EN_29	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
28 IDMAC_EOS_SDMA_EN_28	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
27 IDMAC_EOS_SDMA_EN_27	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

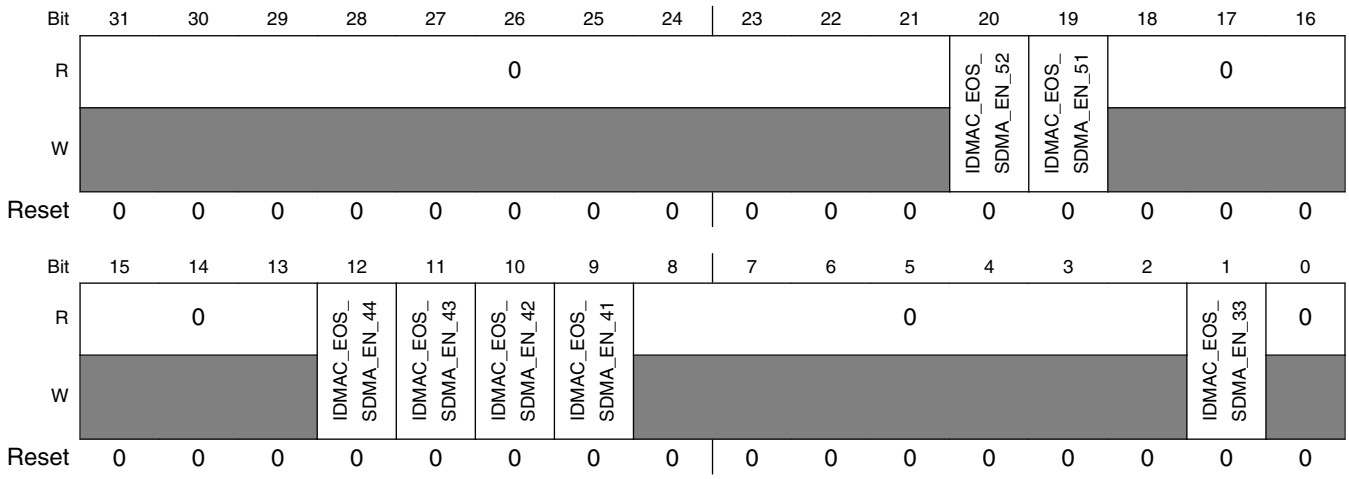
IPU_SDMA_EVENT_7 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
26 IDMAC_EOS_ SDMA_EN_26	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
25 IDMAC_EOS_ SDMA_EN_25	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
24 IDMAC_EOS_ SDMA_EN_24	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
23 IDMAC_EOS_ SDMA_EN_23	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
22–20 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
19 IDMAC_EOS_ SDMA_EN_19	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
18–0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.

45.51.26 SDMA Event Control Register 8 (IPU_SDMA_EVENT_8)

This register contains part of IPU SDMA events controls. The controls of EOS (End of Scroll) of DMA Channels SDMA events [63:32] can be found in this register.

Address: IPU_SDMA_EVENT_8 is 1E00_0000h base + 8Ch offset = 1E00_008Ch



IPU_SDMA_EVENT_8 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
20 IDMAC_EOS_SDMA_EN_52	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
19 IDMAC_EOS_SDMA_EN_51	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
12 IDMAC_EOS_SDMA_EN_44	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_SDMA_EVENT_8 field descriptions (continued)

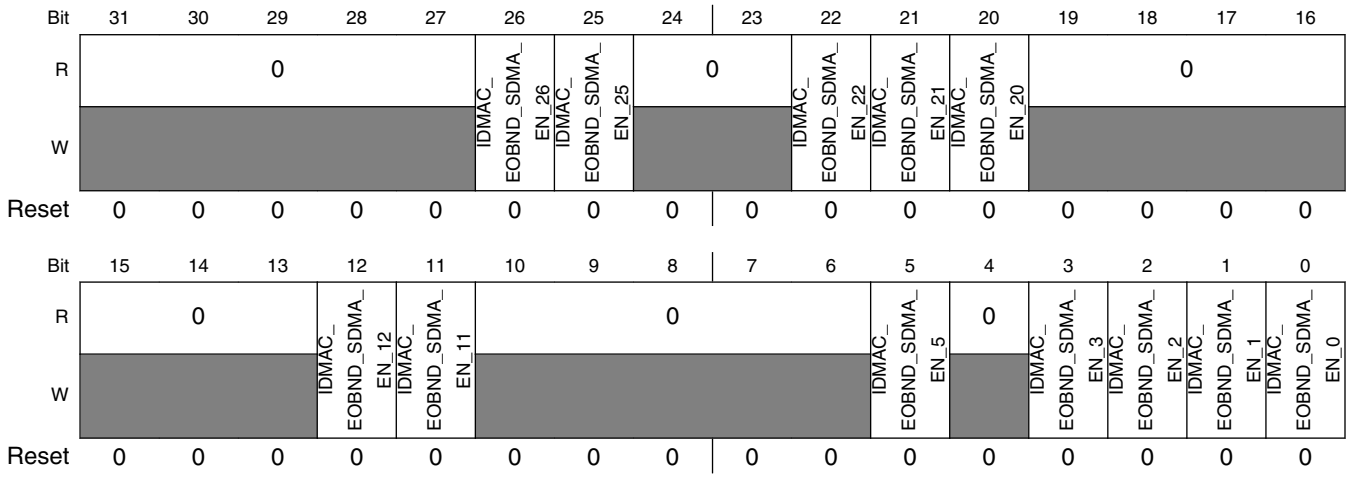
Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
11 IDMAC_EOS_SDMA_EN_43	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
10 IDMAC_EOS_SDMA_EN_42	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
9 IDMAC_EOS_SDMA_EN_41	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
8–2 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
1 IDMAC_EOS_SDMA_EN_33	Enable End of Scroll of Channel SDMA event. This bit is the control of End Of Scroll of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.

45.51.27 SDMA Event Control Register 11 (IPU_SDMA_EVENT_11)

This register contains part of IPU SDMA events controls. The controls of EOBND (End of Band) of DMA Channels SDMA events [31:0] can be found in this register.

- Hide VDOA_SYNC for all versions
- Show VDOA_SYNC for IPUv3H version.
- The table below tagged with other settings (like IPU3M_only) should be hidden in IPUv3H version.

Address: IPU_SDMA_EVENT_11 is 1E00_0000h base + 90h offset = 1E00_0090h



IPU_SDMA_EVENT_11 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
26 IDMAC_ EOBND_SDMA_ EN_26	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
25 IDMAC_ EOBND_SDMA_ EN_25	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
24–23 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
22 IDMAC_ EOBND_SDMA_ EN_22	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
21 IDMAC_ EOBND_SDMA_ EN_21	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

Table continues on the next page...

IPU_SDMA_EVENT_11 field descriptions (continued)

Field	Description
20 IDMAC_ EOBND_SDMA_ EN_20	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
19–13 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
12 IDMAC_ EOBND_SDMA_ EN_12	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
11 IDMAC_ EOBND_SDMA_ EN_11	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
10–6 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
5 IDMAC_ EOBND_SDMA_ EN_5	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
4 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
3 IDMAC_ EOBND_SDMA_ EN_3	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
2 IDMAC_ EOBND_SDMA_ EN_2	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

Table continues on the next page...

IPU_SDMA_EVENT_11 field descriptions (continued)

Field	Description
1 IDMAC_ EOBND_SDMA_ EN_1	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
0 IDMAC_ EOBND_SDMA_ EN_0	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

45.51.28 SDMA Event Control Register 12 (IPU_SDMA_EVENT_12)

This register contains part of IPU SDMA events controls. The controls of EOBND (End of Band) of DMA Channels SDMA events [63:32] can be found in this register.

Address: IPU_SDMA_EVENT_12 is 1E00_0000h base + 94h offset = 1E00_0094h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													IDMAC_ EOBND_SDMA_ EN_50	IDMAC_ EOBND_SDMA_ EN_49	IDMAC_ EOBND_SDMA_ EN_48
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_ EOBND_SDMA_ EN_47	IDMAC_ EOBND_SDMA_ EN_46	IDMAC_ EOBND_SDMA_ EN_45	0												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_SDMA_EVENT_12 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
18 IDMAC_ EOBND_SDMA_ EN_50	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

Table continues on the next page...

IPU_SDMA_EVENT_12 field descriptions (continued)

Field	Description
17 IDMAC_ EOBND_SDMA_ EN_49	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
16 IDMAC_ EOBND_SDMA_ EN_48	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
15 IDMAC_ EOBND_SDMA_ EN_47	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
14 IDMAC_ EOBND_SDMA_ EN_46	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
13 IDMAC_ EOBND_SDMA_ EN_45	Enable End of Band of Channel SDMA event. This bit is the control of End Of Band of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
12–0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.

45.51.29 SDMA Event Control Register 13 (IPU_SDMA_EVENT_13)

This register contains part of IPU SDMA events controls. The controls of TH (Threshold) of DMA Channels SDMA events [31:0] can be found in this register.

Address: IPU_SDMA_EVENT_13 is 1E00_0000h base + 98h offset = 1E00_0098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_TH_SDMA_EN_31	0	IDMAC_TH_SDMA_EN_29	IDMAC_TH_SDMA_EN_28	IDMAC_TH_SDMA_EN_27	IDMAC_TH_SDMA_EN_26	IDMAC_TH_SDMA_EN_25	IDMAC_TH_SDMA_EN_24	IDMAC_TH_SDMA_EN_23	IDMAC_TH_SDMA_EN_22	IDMAC_TH_SDMA_EN_21	IDMAC_TH_SDMA_EN_20	IDMAC_TH_SDMA_EN_19	IDMAC_TH_SDMA_EN_18	IDMAC_TH_SDMA_EN_17	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_TH_SDMA_EN_15	IDMAC_TH_SDMA_EN_14	IDMAC_TH_SDMA_EN_13	IDMAC_TH_SDMA_EN_12	IDMAC_TH_SDMA_EN_11	IDMAC_TH_SDMA_EN_10	IDMAC_TH_SDMA_EN_9	IDMAC_TH_SDMA_EN_8	0		IDMAC_TH_SDMA_EN_5	0	IDMAC_TH_SDMA_EN_3	IDMAC_TH_SDMA_EN_2	IDMAC_TH_SDMA_EN_1	IDMAC_TH_SDMA_EN_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_SDMA_EVENT_13 field descriptions

Field	Description
31 IDMAC_TH_SDMA_EN_31	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
29 IDMAC_TH_SDMA_EN_29	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
28 IDMAC_TH_SDMA_EN_28	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
27 IDMAC_TH_SDMA_EN_27	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_SDMA_EVENT_13 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
26 IDMAC_TH_ SDMA_EN_26	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
25 IDMAC_TH_ SDMA_EN_25	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
24 IDMAC_TH_ SDMA_EN_24	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
23 IDMAC_TH_ SDMA_EN_23	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
22 IDMAC_TH_ SDMA_EN_22	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
21 IDMAC_TH_ SDMA_EN_21	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
20 IDMAC_TH_ SDMA_EN_20	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
19 IDMAC_TH_ SDMA_EN_19	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
18 IDMAC_TH_ SDMA_EN_18	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_SDMA_EVENT_13 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
17 IDMAC_TH_ SDMA_EN_17	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
16 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
15 IDMAC_TH_ SDMA_EN_15	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
14 IDMAC_TH_ SDMA_EN_14	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
13 IDMAC_TH_ SDMA_EN_13	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
12 IDMAC_TH_ SDMA_EN_12	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
11 IDMAC_TH_ SDMA_EN_11	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
10 IDMAC_TH_ SDMA_EN_10	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
9 IDMAC_TH_ SDMA_EN_9	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_SDMA_EVENT_13 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
8 IDMAC_TH_ SDMA_EN_8	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
5 IDMAC_TH_ SDMA_EN_5	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
4 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
3 IDMAC_TH_ SDMA_EN_3	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
2 IDMAC_TH_ SDMA_EN_2	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
1 IDMAC_TH_ SDMA_EN_1	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
0 IDMAC_TH_ SDMA_EN_0	Enable Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

45.51.30 SDMA Event Control Register 14 (IPU_SDMA_EVENT_14)

This register contains part of IPU SDMA events controls. The controls of TH (Threshold) of DMA Channels SDMA events [63:32] can be found in this register.

Address: IPU_SDMA_EVENT_14 is 1E00_0000h base + 9Ch offset = 1E00_009Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											IDMAC_TH_SDMA_EN_52	IDMAC_TH_SDMA_EN_51	IDMAC_TH_SDMA_EN_50	IDMAC_TH_SDMA_EN_49	IDMAC_TH_SDMA_EN_48
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_TH_SDMA_EN_47	IDMAC_TH_SDMA_EN_46	IDMAC_TH_SDMA_EN_45	IDMAC_TH_SDMA_EN_44	IDMAC_TH_SDMA_EN_43	IDMAC_TH_SDMA_EN_42	IDMAC_TH_SDMA_EN_41	IDMAC_TH_SDMA_EN_40	0						IDMAC_TH_SDMA_EN_33	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_SDMA_EVENT_14 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
20 IDMAC_TH_SDMA_EN_52	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
19 IDMAC_TH_SDMA_EN_51	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
18 IDMAC_TH_SDMA_EN_50	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.

Table continues on the next page...

IPU_SDMA_EVENT_14 field descriptions (continued)

Field	Description
17 IDMAC_TH_ SDMA_EN_49	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
16 IDMAC_TH_ SDMA_EN_48	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
15 IDMAC_TH_ SDMA_EN_47	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
14 IDMAC_TH_ SDMA_EN_46	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
13 IDMAC_TH_ SDMA_EN_45	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
12 IDMAC_TH_ SDMA_EN_44	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
11 IDMAC_TH_ SDMA_EN_43	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
10 IDMAC_TH_ SDMA_EN_42	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

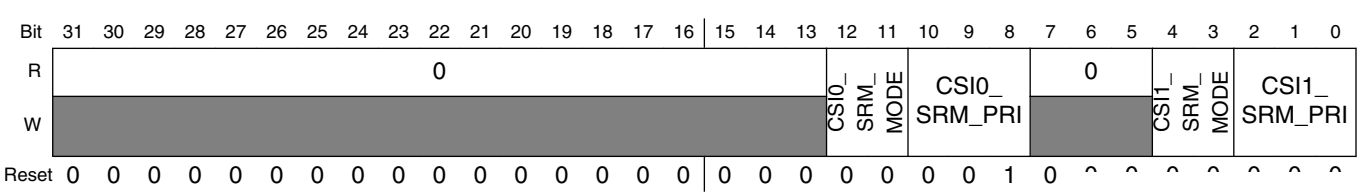
IPU_SDMA_EVENT_14 field descriptions (continued)

Field	Description
	0 SDMA event is disabled. 1 SDMA event is enabled.
9 IDMAC_TH_ SDMA_EN_41	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
8 IDMAC_TH_ SDMA_EN_40	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
7–2 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.
1 IDMAC_TH_ SDMA_EN_33	Threshold of Channel SDMA event. This bit is the control of Threshold of Channel #n. n Indicates the corresponding DMA channel number. *n* Indicates the corresponding DMA channel number. 0 SDMA event is disabled. 1 SDMA event is enabled.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 SDMA event is disabled. 1 SDMA event is enabled.

45.51.31 Shadow Registers Memory Priority 1 Register (IPU_SRM_PRI1)

The register controls the priority of SRM updates. The priority level for each block that has a shadow of its registers in the SRM should be unique. The priority level defines the order of SRM updates. A block with priority set to 010 will be updated before a block with priority set to 001.

Address: IPU_SRM_PRI1 is 1E00_0000h base + A0h offset = 1E00_00A0h



IPU_SRM_PRI1 field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12–11 CSI0_SRM_ MODE	CSI0 SRM Mode This field controls the SRM logic that handles the CSI0 registers 00 Automatic swapping is disabled; ARM platform is allowed to access the CSI1's region in the RAM 01 The SRM logic is controlled by the FSU. The update will be done of the next frame. 10 The SRM logic is controlled by the FSU. Registers are swapped continuously frame by frame 11 Update now. The SRM is controlled by the ARM Platform. The Register will be update now
10–8 CSI0_SRM_PRI	CSI0 SRM priority This bits define the priority of the CSI1 block
7–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4–3 CSI1_SRM_ MODE	CSI1 SRM Mode This field controls the SRM logic that handles the CSI1 registers 00 Automatic swapping is disabled; ARM platform is allowed to access the CSI0's region in the RAM 01 The SRM logic is controlled by the FSU. The update will be done of the next frame. 10 The SRM logic is controlled by the FSU. Registers are swapped continuously frame by frame 11 Update now. The SRM is controlled by the ARM platform. The Register will be update now
2–0 CSI1_SRM_PRI	CSI1 SRM priority This bits define the priority of the CSI0 module

45.51.32 Shadow Registers Memory Priority 2 Register (IPU_SRM_PRI2)

The register controls the priority of SRM updates. The priority level for each block that has a shadow of its registers in the SRM should be unique. The priority level defines the order of SRM updates. a block with priority set to 010 will be updated before a block with priority set to 001.

Address: IPU_SRM_PRI2 is 1E00_0000h base + A4h offset = 1E00_00A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			DI1_SRM_MODE		DI1_SRM_PRI			0			DI0_SRM_MCU_USE		DI0_SRM_PRI		
W																
Reset	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DC_6_SRM_MODE		DC_2_SRM_MODE		DC_SRM_PRI			DP_A1_SRM_MODE	DP_A0_SRM_MODE			DP_S_SRM_MODE		DP_SRM_PRI		
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1

IPU_SRM_PRI2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28–27 DI1_SRM_ MODE	DCI1 SRM Mode This field controls the SRM logic that handles the DI1 registers 00 Automatic swapping is disabled; ARM platform is allowed to access the DI1 region in the RAM 01 The SRM logic is controlled by the FSU. The update will be done of the next frame. 10 The SRM logic is controlled by the FSU. Registers are swapped continuously frame by frame 11 Update now. The SRM is controlled by the ARM platform. The Register will be update now
26–24 DI1_SRM_PRI	DI1 SRM priority This bits define the priority of the DI1 module
23–21 Reserved	This read-only field is reserved and always has the value zero. Reserved
20–19 DI0_SRM_ MCU_USE	DI0 SRM is used by ARM platform This bit indicates that the registers of the DI0 are currently being updated by the ARM platform. The ARM platform should set this bit before accessing the SRM part that is relevant to the DI0. The ARM platform should clear this bit when the update procedure is finished. When this bit is set the SRM mechanism will not update the DI0's registers to avoid data coherency problems. 1 DI0 SRM is currently updated by the ARM platform 0 DI0 SRM s currently not updated by the ARM platform
18–16 DI0_SRM_PRI	DI0 SRM priority This bits define the priority of the DI0 module
15–14 DC_6_SRM_ MODE	DC Group #6 SRM Mode This field controls the SRM logic that handles the DC Group #6 registers 00 Automatic swapping is disabled; ARM platform is allowed to access the DC Group #6's region in the RAM 01 The SRM logic is controlled by the FSU. The update will be done of the next frame. 10 The SRM logic is controlled by the FSU. Registers are swapped continuously frame by frame 11 Update now. The SRM is controlled by the ARM platform. The Register will be update now
13–12 DC_2_SRM_ MODE	DC Group #2 SRM Mode This field controls the SRM logic that handles the DC Group #2 registers 00 Automatic swapping is disabled; ARM platform is allowed to access the DC Group #2's region in the RAM 01 The SRM logic is controlled by the FSU. The update will be done of the next frame. 10 The SRM logic is controlled by the FSU. Registers are swapped continuously frame by frame 11 Update now. The SRM is controlled by the ARM platform. The Register will be update now
11–9 DC_SRM_PRI	DC SRM priority This bits define the priority of the DC module
8–7 DP_A1_SRM_ MODE	DP Async flow #1 SRM Mode This field controls the SRM logic that handles the DP Async flow #1 registers

Table continues on the next page...

IPU_SRM_PRI2 field descriptions (continued)

Field	Description
	00 Automatic swapping is disabled; ARM platform is allowed to access the DP Async flow #1 region in the RAM 01 The SRM logic is controlled by the FSU. The update will be done of the next frame. 10 The SRM logic is controlled by the FSU. Registers are swapped continuously frame by frame 11 Update now. The SRM is controlled by the ARM platform. The Register will be update now
6–5 DP_A0_SRM_MODE	DP Async flow #0 SRM Mode This field controls the SRM logic that handles the DP Async flow #0 registers 00 Automatic swapping is disabled; ARM platform is allowed to access the DP Async flow #0 region in the RAM 01 The SRM logic is controlled by the FSU. The update will be done of the next frame. 10 The SRM logic is controlled by the FSU. Registers are swapped continuously frame by frame 11 Update now. The SRM is controlled by the ARM platform. The Register will be update now
4–3 DP_S_SRM_MODE	DP sync flow SRM Mode This field controls the SRM logic that handles the DP sync flow registers 00 Automatic swapping is disabled; ARM platform is allowed to access the DP sync flow region in the RAM 01 The SRM logic is controlled by the FSU. The update will be done on the next frame. 10 Reserved 11 Update now. The SRM is controlled by the ARM platform. The Register will be update now
2–0 DP_SRM_PRI	DP SRM priority This bits define the priority of the DP module

45.51.33 FSU Processing Flow 1 Register (IPU_FS_PROC_FLOW1)

This register contain controls for IPU's tasks.

Address: IPU_FS_PROC_FLOW1 is 1E00_0000h base + A8h offset = 1E00_00A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VF_IN_VALID	ENC_IN_VALID	VDI_SRC_SEL		PRP_SRC_SEL				VDI3_SRC_SEL		VDI1_SRC_SEL		PP_ROT_SRC_SEL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PP_SRC_SEL				PRPVF_ROT_SRC_SEL				0				PRPENC_ROT_SRC_SEL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_FS_PROC_FLOW1 field descriptions

Field	Description
31 VF_IN_VALID	View-finder Input valid. Setting this bit indicates that the buffer in memory for viewfinder is validated by the ARM platform (valid only when RWS_EN is '1'). 0 View-finder should skip buffer in memory. 1 View-finder should use buffer in memory.
30 ENC_IN_VALID	Encoding Input valid. Setting this bit indicates that the buffer in memory for encoding is validated by the ARM platform (valid only when RWS_EN is '1'). 0 Encoding should skip buffer in memory. 1 Encoding should use buffer in memory.
29–28 VDI_SRC_SEL	Source select for the VDIC This field is relevant if the VDIC works in de-interlacing mode (when VDI_CMB_EN bit is clear) 00 ARM platform 01 CSI direct (cb7) 10 Reserved — 11 Reserved
27–24 PRP_SRC_SEL	Source select for the Pre Processing Task 0000 ARM platform 0001 capture0 (smfc0) — — — 0011 capture2 (smfc2) — — — 0101 IC direct (cb7) — 0110 IRT Encoding 0111 IRT viewfinder 1000 Reserved 1001 Reserved 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2
23–22 VDI3_SRC_SEL	Source select for the VDIC plane #3 (IDMAC's CH 25) 00 ARM platform This field is relevant only if the VDIC works in combining mode (VDI_CMB_EN bit is set) 01 IRT viewfinder (ch 49)

Table continues on the next page...

IPU_FS_PROC_FLOW1 field descriptions (continued)

Field	Description
	10 IRT playback (ch 50) 11 post-processing (ch 22)
21–20 VDI1_SRC_SEL	Source select for the VDIC plane #1 (IDMAC's CH26) This field is relevant only if the VDIC works in combining mode (VDI_CMB_EN bit is set) 00 ARM platform 01 IRT viewfinder 10 IRT playback 11 post-processing
19–16 PP_ROT_SRC_SEL	Source select for the pre processing task of the IRT (CH 50) 0000 ARM platform 0001 capture0 (smfc0) — 0010 Reserved 0011 capture2 (smfc2) — 0100 Reserved 0101 Post-processing 0110 Reserved 0111 Reserved — 1000 Reserved — 1001 Reserved 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2
15–12 PP_SRC_SEL	Source select for the pre processing task of the IC 0000 ARM platform 0001 capture0 (smfc0) — 0010 Reserved 0011 capture2 (smfc2) — 0100 Reserved 0101 Reserved 0110 Rotation for post-processing 0111 Reserved — 1000 Reserved — —

Table continues on the next page...

IPU_FS_PROC_FLOW1 field descriptions (continued)

Field	Description
	1001 Reserved 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2
11–8 PRPVF_ROT_SRC_SEL	Source select for the view finder task of the IRT 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture1 (smfc1) — 0011 capture2 (smfc2) — 0100 capture3 (smfc3) — 0101 IC direct (cb7) — 0110 Reserved 0111 Reserved 1000 View-finder 1001 Reserved 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved
3–0 PRPENC_ROT_SRC_SEL	Source select for the encoding task of the IRT 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture1 (smfc1) — 0011 capture2 (smfc2) — 0100 capture3 (smfc3) — 0101 IC direct (cb7) — 0110 Reserved

Table continues on the next page...

IPU_FS_PROC_FLOW1 field descriptions (continued)

Field	Description
	0111 encoding 1000 Reserved 1001 Reserved 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2

45.51.34 FSU Processing Flow 2 Register (IPU_FS_PROC_FLOW2)

This register contains controls for IPU's tasks.

Address: IPU_FS_PROC_FLOW2 is 1E00_0000h base + ACh offset = 1E00_00ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				PRP_DEST_SEL				PRPENC_ROT_DEST_SEL				PP_ROT_DEST_SEL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PP_DEST_SEL				PRPVF_ROT_DEST_SEL				PRPVF_DEST_SEL				PRP_ENC_DEST_SEL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_FS_PROC_FLOW2 field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. **Reserved**
27–24 PRP_DEST_SEL	**Pre processing destination select (for channel DMAIC_7)** 0000 ARM platform 0001 IC input buffer (ch12) 0010 PP (ch11) 0011 PP_ROT (ch47) 0100 DC1 (ch28) 0101 DC2 (ch41) 0110 DP_ASYNC1 (ch24) 0111 DP_ASYNC0 (ch29) 1000 DP_SYNC1 (ch27) 1001 DP_SYNC0 (ch23) 1010 Alt DC2 (ch41)

Table continues on the next page...

IPU_FS_PROC_FLOW2 field descriptions (continued)

Field	Description
	1011 Alt DP_ASYNC1 (ch24) 1100 Alt DP_ASYNC0 (ch29) 1111 Reserved
23–20 PRPENC_ROT_ DEST_SEL	Destination select for Rotation task coming from the Encoding input 0000 ARM platform 0001 Reserved 0010 Reserved — 0011 Reserved — 0100 Reserved 0101 IC Pre Processing 0110 Reserved 0111 DC1 (ch28) 1000 DC2 (ch41) 1001 **DP_SYNC0 (ch23)** 1010 **DP_SYNC1 (ch27)** 1011 **DP_ASYNC1 (ch24)** 1100 **DP_ASYNC0 (ch29)** 1101 Alt DC2 (ch41) 1110 Alt **DP_ASYNC1 (ch24)** 1111 Alt **DP_ASYNC0 (ch29)**
19–16 PP_ROT_ DEST_SEL	Destination select for Rotation task coming from the Post Processing input 0000 ARM platform 0001 Reserved 0010 Reserved 0011 Reserved 0100 IC Playback (Post Processing) — — 0101 VDI_PLANE3 (Ch 25) — — 0110 VDI_PLANE1 (Ch 26) 0111 DC1 (ch28) 1000 DC2 (ch41) 1001 **DP_SYNC0 (ch23)** 1010 **DP_SYNC1 (ch27)** 1011 **DP_ASYNC1 (ch24)** 1100 **DP_ASYNC0 (ch29)** 1101 Alt DC2 (ch41) 1110 Alt **DP_ASYNC1 (ch24)** 1111 Alt **DP_ASYNC0 (ch29)**
15–12 PP_DEST_SEL	Destination select for post processing task

Table continues on the next page...

IPU_FS_PROC_FLOW2 field descriptions (continued)

Field	Description
	0000 ARM platform 0001 Reserved 0010 Reserved 0011 IRT playback — 0100 VDI_PLANE3 (Ch 25) — 0101 VDI_PLANE1 (Ch 26) 0110 Reserved 0111 DC1 (ch28) 1000 DC2 (ch41) 1001 **DP_SYNC0 (ch23)** 1010 **DP_SYNC1 (ch27)** 1011 **DP_ASYNC1 (ch24)** 1100 **DP_ASYNC0 (ch29)** 1101 Alt DC2 (ch41) 1110 Alt **DP_ASYNC1 (ch24)** 1111 Alt **DP_ASYNC0 (ch29)**
11–8 PRPVF_ROT_DEST_SEL	Destination select for Rotation task coming from the View finder input 0000 ARM platform 0001 Reserved 0010 Reserved — — 0011 VDI_PLANE3 (Ch 25) — — 0100 VDI_PLANE1 (Ch 26) 0101 IC Pre Processing 0110 Reserved 0111 DC1 (ch28) 1000 DC2 (ch41) 1001 **DP_SYNC0 (ch23)** 1010 **DP_SYNC1 (ch27)** 1011 **DP_ASYNC1 (ch24)** 1100 **DP_ASYNC0 (ch29)** 1101 Alt DC2 (ch41) 1110 Alt **DP_ASYNC1 (ch24)** 1111 Alt **DP_ASYNC0 (ch29)**
7–4 PRPVF_DEST_SEL	Destination select for View finder task 0000 ARM platform 0001 IRT viewfinder 0010 Reserved 0011 Reserved 0100 Reserved

Table continues on the next page...

IPU_FS_PROC_FLOW2 field descriptions (continued)

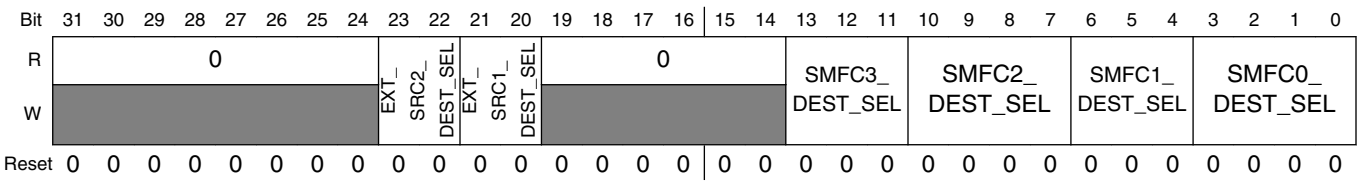
Field	Description
	0101 Reserved 0110 Reserved 0111 DC1 (ch28) 1000 DC2 (ch41) 1001 **DP_SYNC0 (ch23)** 1010 **DP_SYNC1 (ch27)** 1011 **DP_ASYNC1 (ch24)** 1100 **DP_ASYNC0 (ch29)** 1101 Alt DC2 (ch41) 1110 Alt **DP_ASYNC1 (ch24)** 1111 Alt **DP_ASYNC0 (ch29)**
3–0 PRP_ENC_ DEST_SEL	Destination select for Encoding task 0000 ARM platform 0001 IRT Encoding 0010 Reserved 0011 Reserved 0100 Reserved 0101 Reserved 0110 Reserved 0111 DC1 (ch28) 1000 DC2 (ch41) 1001 **DP_SYNC0 (ch23)** 1010 **DP_SYNC1 (ch27)** 1011 **DP_ASYNC1 (ch24)** 1100 **DP_ASYNC0 (ch29)** 1101 Alt DC2 (ch41) 1110 Alt **DP_ASYNC1 (ch24)** 1111 Alt **DP_ASYNC0 (ch29)**

45.51.35 FSU Processing Flow 3 Register (IPU_FS_PROC_FLOW3)

This register contains controls for IPU's tasks.

- Hide VPU_SUB_FRAME_SYNC for all versions
- Show VPU_SUB_FRAME_SYNC for IPUv3H version.
- The table below tagged with other settings (like IPU3M_only) should be hidden in IPUv3H version.
- This requires some sophisticated conditional tag settings

Address: IPU_FS_PROC_FLOW3 is 1E00_0000h base + B0h offset = 1E00_00B0h



IPU_FS_PROC_FLOW3 field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. **Reserved**
23–22 EXT_SRC2_DEST_SEL	Destination select for External Source 2 00 disabled 01 **DP_SYNC0 (ch23)** 10 **DP_SYNC1 (ch27)** 11 DC1 (ch28)
21–20 EXT_SRC1_DEST_SEL	Destination select for External Source 1 00 disabled 01 **DP_SYNC0 (ch23)** 10 **DP_SYNC1 (ch27)** 11 DC1 (ch28)
19–14 Reserved	This read-only field is reserved and always has the value zero. **Reserved**
13–11 SMFC3_DEST_SEL	Destination select for SMFC3 000 ARM platform 001 IRT Encoding 010 IRT viewfinder 011 IRT playback 100 IC Playback (Post Processing) 101 IC Pre Processing —
	111 Reserved
10–7 SMFC2_DEST_SEL	Destination select for SMFC2 0000 ARM platform 0001 IRT Encoding 0010 IRT viewfinder 0011 IRT playback 0100 IC Playback (Post Processing) 0101 IC Pre Processing —
	0111 DC1 (ch28) 1000 DC2 (ch41) 1001 **DP_SYNC0 (ch23)** 1010 **DP_SYNC1 (ch27)**

Table continues on the next page...

IPU_FS_PROC_FLOW3 field descriptions (continued)

Field	Description
	1011 **DP_ASYNC1 (ch24)** 1100 **DP_ASYNC0 (ch29)** 1101 Alt DC2 (ch41) 1110 Alt **DP_ASYNC1 (ch24)** 1111 Alt **DP_ASYNC0 (ch29)**
6–4 SMFC1_DEST_SEL	Destination select for SMFC1 000 ARM platform 001 IRT Encoding 010 IRT viewfinder 011 IRT playback 100 IC Playback (Post Processing) 101 IC Pre Processing — 111 Reserved
3–0 SMFC0_DEST_SEL	Destination select for SMFC0 0000 ARM platform 0001 IRT Encoding 0010 IRT viewfinder 0011 IRT playback 0100 IC Playback (Post Processing) 0101 IC Pre Processing — 0111 DC1 (ch28) 1000 DC2 (ch41) 1001 **DP_SYNC0 (ch23)** 1010 **DP_SYNC1 (ch27)** 1011 **DP_ASYNC1 (ch24)** 1100 **DP_ASYNC0 (ch29)** 1101 Alt DC2 (ch41) 1110 Alt **DP_ASYNC1 (ch24)** 1111 Alt **DP_ASYNC0 (ch29)**

45.51.36 FSU Displaying Flow 1 Register (IPU_FS_DISP_FLOW1)

This register contains controls for IPU's tasks.

Address: IPU_FS_DISP_FLOW1 is 1E00_0000h base + B4h offset = 1E00_00B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DC1_SRC_SEL				DC2_SRC_SEL				DP_ASYNC1_SRC_SEL				DP_ASYNC0_SRC_SEL				DP_SYNC1_SRC_SEL				DP_SYNC0_SRC_SEL			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_FS_DISP_FLOW1 field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23–20 DC1_SRC_SEL	Source select for DS1/DS2 - MG (graphics) plane (ch28) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2) — 0011 IC encoding 0100 IC viewfinder 0101 IC playback 0110 IRT Encoding 0111 IRT viewfinder 1000 IRT playback — 1001 Reserved — 1010 Reserved 1011 autoref 1100 autoref+snoop1 — 1101 External source #1 (e.g. an external block like GPU) 1110 snoop1 — 1111 External source #2 (e.g. an external block like GPU)
19–16 DC2_SRC_SEL	Source select for DS3 (ch41) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2) — 0011 IC encoding 0100 IC viewfinder 0101 IC playback 0110 IRT Encoding 0111 IRT viewfinder 1000 IRT playback — 1001 Reserved — 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2

Table continues on the next page...

IPU_FS_DISP_FLOW1 field descriptions (continued)

Field	Description
	1110 snoop1 1111 snoop2
15–12 DP_ASYNC1_ SRC_SEL	Source select for DS1/DS2 - Vx (video) plane (ch24) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2) — 0011 IC encoding 0100 IC viewfinder 0101 IC playback 0110 IRT Encoding 0111 IRT viewfinder 1000 IRT playback — 1001 Reserved — 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2
11–8 DP_ASYNC0_ SRC_SEL	Source select for DS2 - MG (graphics) plane (ch29) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2) — 0011 IC encoding 0100 IC viewfinder 0101 IC playback 0110 IRT Encoding 0111 IRT viewfinder 1000 IRT playback — 1001 Reserved — 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2

Table continues on the next page...

IPU_FS_DISP_FLOW1 field descriptions (continued)

Field	Description
7–4 DP_SYNC1_ SRC_SEL	Source select for DS1/DS2 - Vx (video) plane (ch27) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2) — 0011 IC encoding 0100 IC viewfinder 0101 IC playback 0110 IRT Encoding 0111 IRT viewfinder 1000 IRT playback — 1001 Reserved — 1010 Reserved — — — — — — 1110 snoop1 1111 snoop2
3–0 DP_SYNC0_ SRC_SEL	Source select for DS2 - MG (graphics) plane (ch23) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2) — 0011 IC encoding 0100 IC viewfinder 0101 IC playback 0110 IRT Encoding 0111 IRT viewfinder 1000 IRT playback — 1001 Reserved — 1010 Reserved — — — — —

Table continues on the next page...

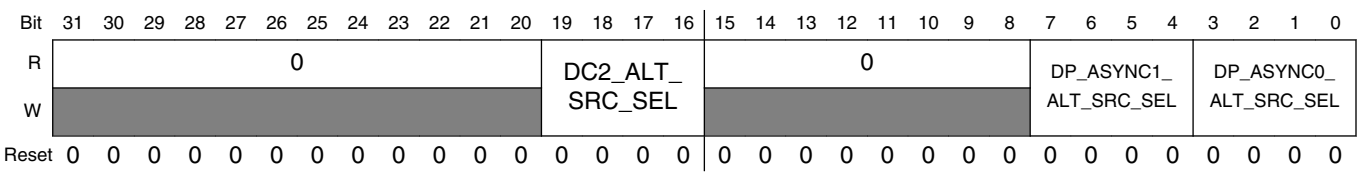
IPU_FS_DISP_FLOW1 field descriptions (continued)

Field	Description
	— 1110 snoop1 1111 snoop2

45.51.37 FSU Displaying Flow 2 Register (IPU_FS_DISP_FLOW2)

This register contains controls for IPU's tasks.

Address: IPU_FS_DISP_FLOW2 is 1E00_0000h base + B8h offset = 1E00_00B8h



IPU_FS_DISP_FLOW2 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value zero. Reserved
19–16 DC2_ALT_SRC_ SEL	Source select for Alternate DS3 (ch41) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2) — 0011 IC encoding 0100 IC viewfinder 0101 IC playback 0110 IRT Encoding 0111 IRT viewfinder 1000 IRT playback — 1001 Reserved — 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2
15–8 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IPU_FS_DISP_FLOW2 field descriptions (continued)

Field	Description
7–4 DP_ASYNC1_ ALT_SRC_SEL	Source select for alternate DS1/DS2 - Vx (video) plane (ch24) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2) — 0011 IC encoding 0100 IC viewfinder 0101 IC playback 0110 IRT Encoding 0111 IRT viewfinder 1000 IRT playback — 1001 Reserved — 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2
3–0 DP_ASYNC0_ ALT_SRC_SEL	Source select for alternate DS2 - MG (graphics) plane (ch29) 0000 ARM platform 0001 capture0 (smfc0) — 0010 capture2 (smfc2) — 0011 IC encoding 0100 IC viewfinder 0101 IC playback 0110 IRT Encoding 0111 IRT viewfinder 1000 IRT playback — 1001 Reserved — 1010 Reserved 1011 autoref 1100 autoref+snoop1 1101 autoref+snoop2 1110 snoop1 1111 snoop2

45.51.38 SKIP Register (IPU_SKIP)

This register controls the different frame skipping supported by the IPU.

Address: IPU_SKIP is 1E00_0000h base + BCh offset = 1E00_00BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VDI_SKIP												VDI_MAX_RATIO_SKIP			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSI_SKIP_IC_VF					CSI_MAX_RATIO_SKIP_IC_VF			CSI_SKIP_IC_ENC					CSI_MAX_RATIO_SKIP_IC_ENC		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_SKIP field descriptions

Field	Description
31–20 VDI_SKIP	VDI_SKIP These 12 bits define the skipping pattern of the frames send from the VDIC. The VDIC avoids reading fields from the memory if the output frame is skipped. Skipping is relevant only if the source to the VDIC is coming from the CSI. Skipping is done for a set of frames. The number of frames in a set is defined at VDI_MAX_RATIO_SKIP. when VDI_MAX_RATIO_SKIP = 1 => VDI_SKIP[1:0] is used; other bits are ignored when VDI_MAX_RATIO_SKIP = 2 => VDI_SKIP[2:0] are used; other bits are ignored .. .. when VDI_MAX_RATIO_SKIP = 11 => VDI_SKIP[11:0] are used;
19–16 VDI_MAX_ RATIO_SKIP	Maximum Ratio Skip for VDIC These bits define the number of frames in a skipping set. The maximum value of this bits is 11. When set to 0 the skipping is disabled.
15–11 CSI_SKIP_IC_ VF	CSI SKIP IC_VF These 5 bits define the skipping pattern of the frames send to the IC for view finder task from one of the CSIs as defined on the CSI_SEL and IC_INPUT bits Skipping is done for a set of frames. The number of frames in a set is defined at CSI_MAX_RATIO_SKIP_IC_VF. when CSI_MAX_RATIO_SKIP_IC_VF = 1 => CSI_SKIP_IC_VF[1:0] are used; other bits are ignored when CSI_MAX_RATIO_SKIP_IC_VF = 2 => CSI_SKIP_IC_VF[2:0] are used; other bits are ignored when CSI_MAX_RATIO_SKIP_IC_VF =3 => CSI_SKIP_IC_VF[3:0] are used; other bits are ignored when CSI_MAX_RATIO_SKIP_IC_VF = 4 => CSI_SKIP_IC_VF[4:0] are used; Setting bit #n of CSI_SKIP_IC_VF means that the #n frame in the set is skipped. For example: if CSI_MAX_RATIO_SKIP_IC_VF = 4 and CSI_SKIP_IC_VF = 11010 Frames #0 & Frame #2 will not be skipped as bit0 and bit2 are cleared Frames #1 & Frame #3 will be skipped as bit1 and bit3 are set

Table continues on the next page...

IPU_SKIP field descriptions (continued)

Field	Description
	bit #4 is ignored as CSI_MAX_RATIO_SKIP_IC_VF is set to 4
10–8 CSI_MAX_ RATIO_SKIP_ IC_VF	CSI Maximum Ratio Skip for IC (view finder task) These bits define the number of frames in a skipping set. The maximum value of this bits is 4. When set to 0 the skipping is disabled.
7–3 CSI_SKIP_IC_ ENC	CSI SKIP IC_ENC These 5 bits define the skipping pattern of the frames send to the IC for encoding task from one of the CSIs as defined on the CSI_SEL and IC_INPUT bits Skipping is done for a set of frames. The number of frames in a set is defined at CSI_MAX_RATIO_SKIP_IC_ENC. when CSI_MAX_RATIO_SKIP_IC_ENC = 1 => CSI_SKIP_IC_ENC[1:0] are used; other bits are ignored when CSI_MAX_RATIO_SKIP_IC_ENC = 2 => CSI_SKIP_IC_ENC[2:0] are used; other bits are ignored when CSI_MAX_RATIO_SKIP_IC_ENC = 3 => CSI_SKIP_IC_ENC[3:0] are used; other bits are ignored when CSI_MAX_RATIO_SKIP_IC_ENC = 4 => CSI_SKIP_IC_ENC[4:0] are used; Setting bit #n of CSI_SKIP_IC_ENC means that the #n frame in the set is skipped. For example: if CSI_MAX_RATIO_SKIP_IC_ENC = 4 and CSI_SKIP_IC_ENC = 11010 Frames #0 & Frame #2 will not be skipped as bit0 and bit2 are cleared Frames #1 & Frame #3 will be skipped as bit1 and bit3 are set bit #4 is ignored as CSI_MAX_RATIO_SKIP_IC_ENC is set to 4
2–0 CSI_MAX_ RATIO_SKIP_ IC_ENC	CSI Maximum Ratio Skip for IC (encoding task) These bits define the number of frames in a skipping set. The maximum value of this bits is 4. When set to 0 the skipping is disabled.

45.51.39 Display General Control Register (IPU_DISP_GEN)

This register controls various aspects of the display port.

Address: IPU_DISP_GEN is 1E00_0000h base + C4h offset = 1E00_00C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						DI1_COUNTER_RELEASE	DI0_COUNTER_RELEASE	CSI_VSYNC_DEST	MCU_MAX_BURST_STOP	MCU_T				MCU_DI_ID_9	MCU_DI_ID_8
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									0	DP_FG_EN_ASYNC1	DP_FG_EN_ASYNC0	DP_ASYNC_DOUBLE_FLOW_DC2_	DOUBLE_FLOW	DI1_DUAL_MODE	DI0_DUAL_MODE
W										DP_PIPE_CLR						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DISP_GEN field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value zero. Reserved
25 DI1_COUNTER_RELEASE	DI1 Counter release By default the DI0 counters responsible for waveform generation for sync flow are frozen. For the first attempt to use the DI in sync flow the user should set this bit 1 counter is released and running 0 counter is cleared and stopped
24 DI0_COUNTER_RELEASE	DI1 Counter release By default the DI0 counters responsible for waveform generation for sync flow are frozen. For the first attempt to use the DI in sync flow the user should set this bit 1 counter is released and running 0 counter is cleared and stopped
23 CSI_VSYNC_DEST	CSI_VSYNC destination This bit defines the destination of the VSYNC coming from the CSI's 1 csi1_vsync is connected to DI0; csi0_vsync is connected to DI1 0 csi0_vsync is connected to DI0; csi1_vsync is connected to DI1
22 MCU_MAX_BURST_STOP	ARM platform Maximal burst This bit limit the maximal unspecified length burst. 1 The maximum unspecified burst length is 8-beat 0 The unspecified burst length is unlimited
21–18 MCU_T	The address space for accesses through the AHB-lite slave port is 128MB and it is split internally (with 32MB resolution) according to bits [28:25] of the address. Using the following notation: Address = (ID[31:29], MSB[28:25], LSB[24:0]) The address is used as follows ("T" is a configurable integer between 0 and 13): MSB<T: access to an external device, with address = (MSB, LSB) T<=MSB<14: access to an external device, with address (MSB-T, LSB)
17 MCU_DI_ID_9	MCU_DI_ID_9 - DI ID via DC channel 9. This bit defines the DI that the ARM platform DC's access via channel #9 1 ARM platform accesses DC's channel #9 via DI1. 0 ARM platform accesses DC's channel #9 via DI0.
16 MCU_DI_ID_8	MCU_DI_ID_8 - DI ID via DC channel 8. This bit defines the DI that the ARM platform DC's access via channel #8 1 ARM platform accesses DC's channel #8 via DI1. 0 ARM platform accesses DC's channel #8 via DI0.
15–7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6 DP_PIPE_CLR	DP Pipe Clear This bit clears the internal pipe of the DP. The user may use this bit in case of an error condition

Table continues on the next page...

IPU_DISP_GEN field descriptions (continued)

Field	Description
	This is a self clear bit 1 Clear the internal pipe of the DP 0 Idle - does nothing
5 DP_FG_EN_ASYNC1	FG_EN - partial plane Enable for async flow 1. This bit enables the partial plane channel. 1 partial plane channel is enabled. 0 partial plane channel is disabled.
4 DP_FG_EN_ASYNC0	FG_EN - partial plane Enable for async flow 0. This bit enables the partial plane channel. 1 partial plane channel is enabled. 0 partial plane channel is disabled.
3 DP_ASYNC_DOUBLE_FLOW	DP Async Double Flow. This bit define how many async flows are currently handles via DP channel (ch24+29) 1 2 flows are handled via DP 0 single flow is handled via DP
2 DC2_DOUBLE_FLOW	DC2 Double Flow. This bit define how many flows are currently handles via DC2 channel (ch41) 1 2 flows are handled via DC2 0 single flow is handled via DC2
1 DI1_DUAL_MODE	DI1 dual mode control 1 DI1 operates in dual mode 0 DI1 is not in dual mode
0 DI0_DUAL_MODE	DI0 dual mode control 1 DI0 operates in dual mode 0 DI0 is not in dual mode

45.51.40 Display Alternate Flow Control Register 1 (IPU_DISP_ALT1)

This register controls various aspects of the display port.

Address: IPU_DISP_ALT1 is 1E00_0000h base + C8h offset = 1E00_00C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	sel_alt_0				step_repeat_alt_0											
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cnt_auto_reload_alt_0	cnt_clr_sel_alt_0			run_value_m1_alt_0											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DISP_ALT1 field descriptions

Field	Description
31–28 sel_alt_0	Select alternative parameters instead of DI Sync Wave Gen counter#. The DI is selected according to DP's synchronous channel destination 0000-disable 0001 instead of counter 1 0010 instead of counter 2 1000 instead of counter 8
27–16 step_repeat_alt_0	This fields defines the amount of repetitions that will be performed by the counter
15 cnt_auto_reload_alt_0	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the step_repeat_alt_0 field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the step_repeat_alt_0 field
14–12 cnt_clr_sel_alt_0	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 Reserved 011 Reserved 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. —

Table continues on the next page...

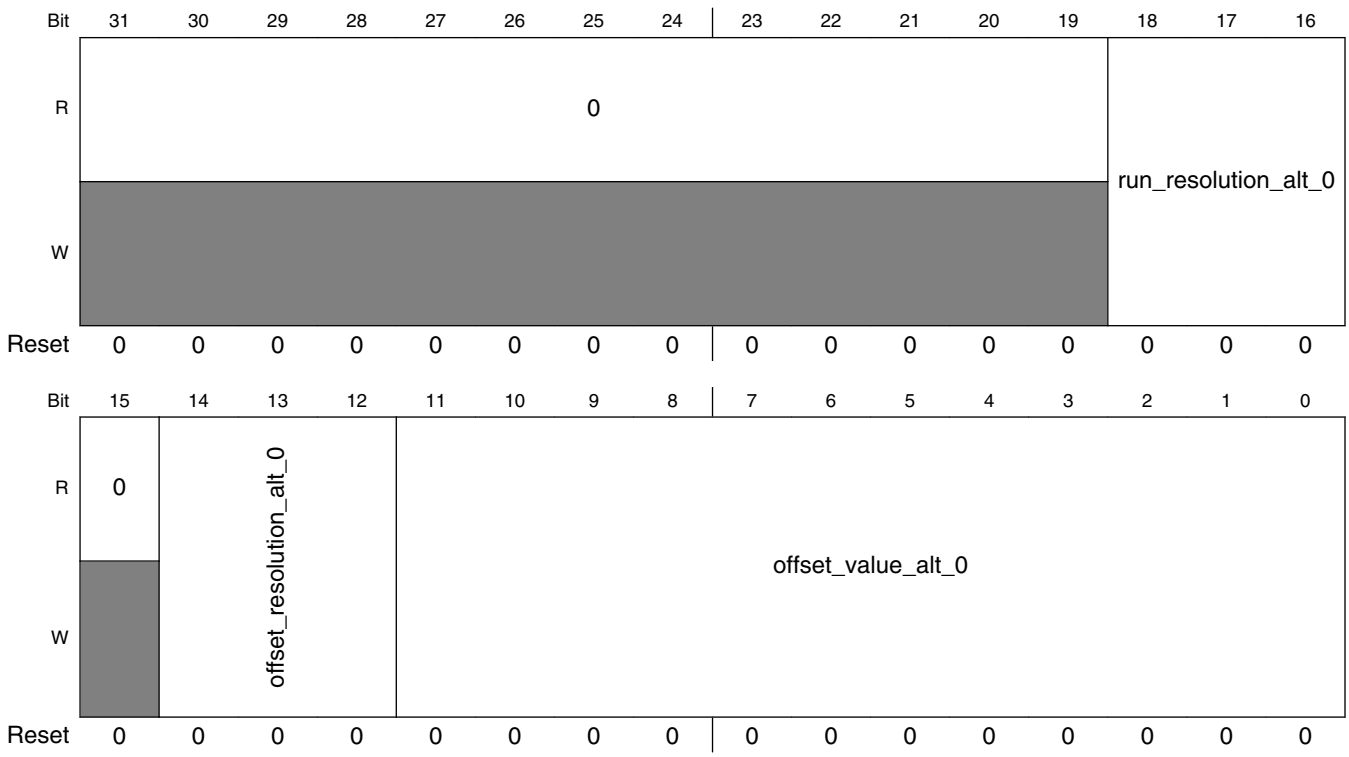
IPU_DISP_ALT1 field descriptions (continued)

Field	Description
	110 External VSYNC 111 Counter is always on.
11–0 run_value_m1_alt_0	Counter pre defined value This fields defines the counter pre defines value. real value- 1

45.51.41 Display Alternate Flow Control Register 2 (IPU_DISP_ALT2)

This register controls various aspects of the display port.

Address: IPU_DISP_ALT2 is 1E00_0000h base + CCh offset = 1E00_00CCh



IPU_DISP_ALT2 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18–16 run_resolution_alt_0	Counter Run Resolution This field defines the trigger causing the counter to increment. The counter run resolution should be defined in the same way as in original DI's counter#
15 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IPU_DISP_ALT2 field descriptions (continued)

Field	Description
14–12 offset_ resolution_alt_0	Counter offset Resolution This field defines the trigger causing the offset counter to increment The counter offset resolution should be defined in the same way as in original DI's counter#
11–0 offset_value_alt_ 0	Counter offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by

45.51.42 Display Alternate Flow Control Register 3 (IPU_DISP_ALT3)

This register controls various aspects of the display port.

Address: IPU_DISP_ALT3 is 1E00_0000h base + D0h offset = 1E00_00D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	sel_alt_1				step_repeat_alt_1											
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	cnt_auto_ reload_alt_1	cnt_clr_sel_alt_1			run_value_m1_alt_1											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DISP_ALT3 field descriptions

Field	Description
31–28 sel_alt_1	Select alternative parameters instead of DI Sync Wave Gen counter#. The DI is selected according to DP's synchronous channel destination 0000 disable 0001 instead of counter 1 0010 instead of counter 2 1000 instead of counter 8
27–16 step_repeat_alt_ 1	This fields defines the amount of repetitions that will be performed by the counter
15 cnt_auto_ reload_alt_1	Counter auto reload mode

Table continues on the next page...

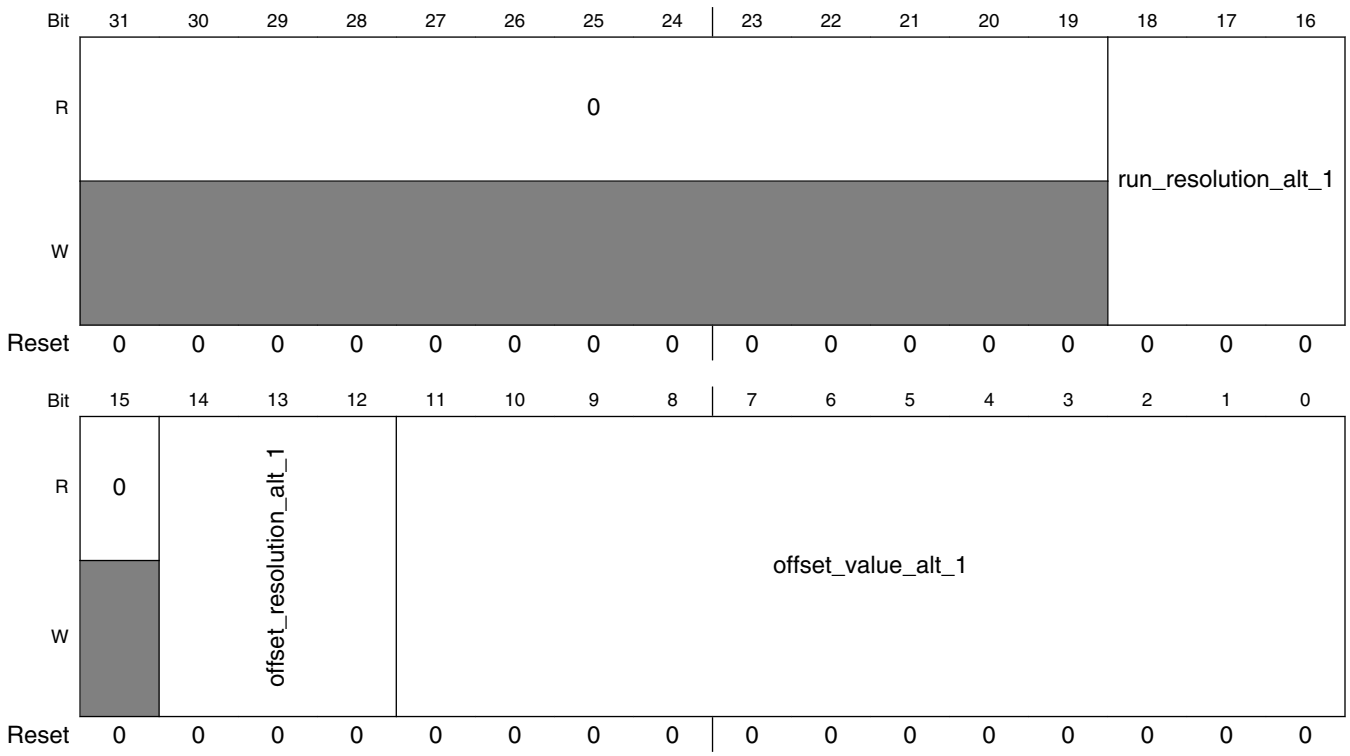
IPU_DISP_ALT3 field descriptions (continued)

Field	Description
	1 The counter will automatically be reloaded forever, ignoring the value of the step_repeat_alt_0 field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the step_repeat_alt_0 field
14–12 cnt_clr_sel_alt_1	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 Reserved 011 Reserved 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
11–0 run_value_m1_alt_1	Counter pre defined value This fields defines the counter pre defines value. real value- 1

45.51.43 Display Alternate Flow Control Register 4 (IPU_DISP_ALT4)

This register controls various aspects of the display port.

Address: IPU_DISP_ALT4 is 1E00_0000h base + D4h offset = 1E00_00D4h



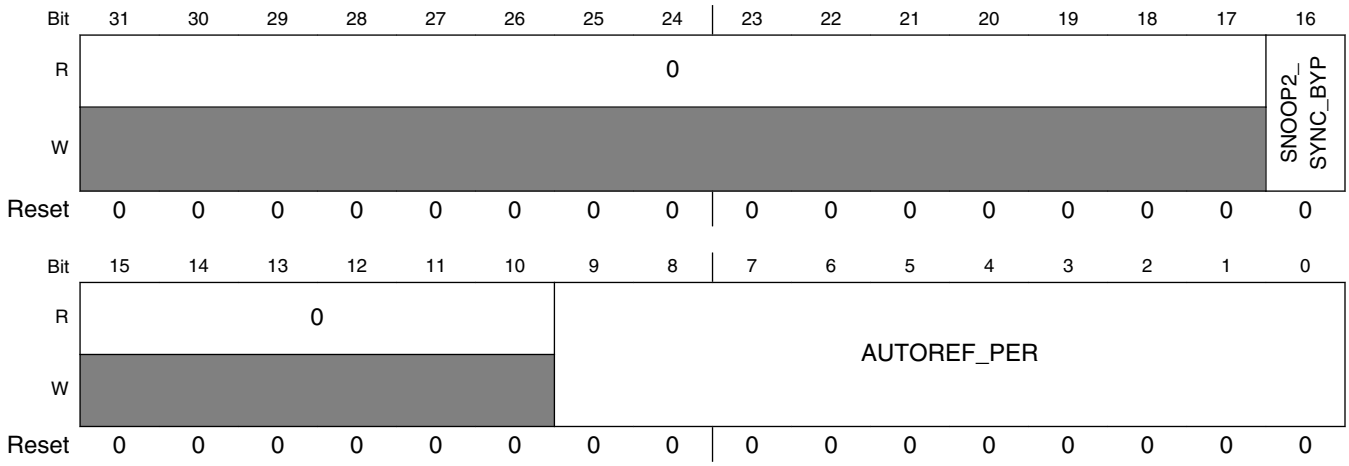
IPU_DISP_ALT4 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18–16 run_resolution_alt_1	Counter Run Resolution This field defines the trigger causing the counter to increment. The counter run resolution should be defined in the same way as in original DI's counter#
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 offset_resolution_alt_1	Counter offset Resolution This field defines the trigger causing the offset counter to increment The counter offset resolution should be defined in the same way as in original DI's counter#
11–0 offset_value_alt_1	Counter offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by

45.51.44 Autorefresh and Snooping Control Register (IPU_SNOOP)

This register controls the snooping mechanism

Address: IPU_SNOOP is 1E00_0000h base + D8h offset = 1E00_00D8h



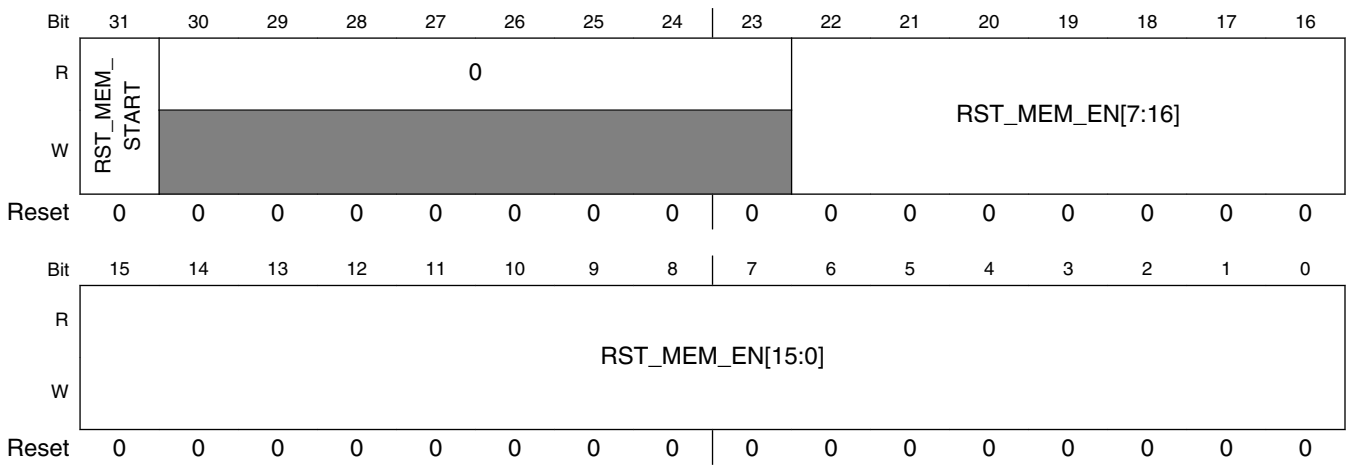
IPU_SNOOP field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value zero. Reserved
16 SNOOP2_ SYNC_BYP	This bits control the bypass of the synchronizer on emi_snooping2 signal. This bit is for test purposes only. The user should not set this bit 1 bypass the emi_snooping2 synchronizer 0 normal mode emi_snooping2 is internally synchronized
15–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–0 AUTOREF_PER	Autorefresh period minus 1. The actual value of autorefresh period is equal to the high speed clock (HSP_CLK) period multiplied by 2^17*(AUTO_REF_PER+1).

45.51.45 Memory Reset Control Register (IPU_MEM_RST)

This register controls the memory reset mechanism. IPU has a hardware mechanism for clearing the content of the internal memories. This allows the user to clear the content of or more of the internal memories without the need to perform write accesses to the memories.

Address: IPU_MEM_RST is 1E00_0000h base + DCh offset = 1E00_00DCh



IPU_MEM_RST field descriptions

Field	Description
31 RST_MEM_START	Memory Reset Start Writing one to this bit activate the memory reset mechanism. The memories that their corresponding RST_MEM_EN bit is set will be cleared. When the memory reset mechanism completes the memory clearing procedure this bit will be automatically cleared. 1 The memory reset mechanism is activated and busy 0 Idle, the memory reset mechanism is not working.
30–23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–0 RST_MEM_EN	Reset Memory Enable Each bit on this field enables the memory reset mechanism for a specific memory. The user should set the relevant bits for the memories that need to be cleared. Below is the list of memories and their corresponding bit. srm = rst_mem_en[0] alpha = rst_mem_en[1] cpmem = rst_mem_en[2] tpm = rst_mem_en[3] mpm = rst_mem_en[4] bm = rst_mem_en[5] rm = rst_mem_en[6]

Table continues on the next page...

IPU_MEM_RST field descriptions (continued)

Field	Description
	dstm = rst_mem_en[7] dsom = rst_mem_en[8] lut0 = rst_mem_en[9] lut1 = rst_mem_en[10] ram_smfc = rst_mem_en[11] vdi_fifo2 = rst_mem_en[12] vdi_fifo3 = rst_mem_en[13] icb = rst_mem_en[14] vdi_fifo1 = rst_mem_en[15] dc_template = rst_mem_en[20] dmfc_rd = rst_mem_en[21] dmfc_wr = rst_mem_en[22]

45.51.46 Power Modes Control Register (IPU_PM)

This register controls the automatic transitions of the IPU between different power modes of the SoC and handles the clock change modes.

Address: IPU_PM is 1E00_0000h base + E0h offset = 1E00_00E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LPSR_MODE	DI1_SRM_CLOCK_CHANGE_MODE	DI1_CLK_PERIOD_1						DI1_CLK_PERIOD_0							
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CLCOK_MODE_STAT	DI0_SRM_CLOCK_CHANGE_MODE	DI0_CLK_PERIOD_1						DI0_CLK_PERIOD_0							
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0

IPU_PM field descriptions

Field	Description
31 LPSR_MODE	LPSR Mode This bit indicates that the next attempt for entering low power mode is an attempt to move to LPST mode. Setting this bit by the user is essential in order to assure proper response of the IPU to the assertion of the stop request from the CCM. 1 Next low power mode will be LPSR 0 Next low power mode is not LPSR
30 DI1_SRM_ CLOCK_ CHANGE_ MODE	SRM clock change mode When the clock is going to be changed to any new ratio other then 1:1, 1:2. 1:4. The user needs to prepare an alternate set of DI setting in the SRM. This bit enable this mode. This bit is self cleared. 1 SRM clock change mode is enabled; the next clock change will be done by updating the DI settings from the SRM 0 SRM clock change mode is disabled.
29–23 DI1_CLK_ PERIOD_1	DI1_CLK period option 1. This parameter defines the period of the clock that the DI1 works with. This parameter contains integer part (bits [6:4]) and fractional part (bits [3:0]). Setting this value to 1.0 (default) means that the DI1 works on the fastest possible clock. Setting a value smaller than 1.0 is not allowed. The value to be programmed to the DI1_CLK_PERIOD_1 field is equal to: Fast_freq/Target_freq Where: Target_freq = The frequency that the DI clock works with Fast_freq = fastest possible clock that the DI can work with
22–16 DI1_CLK_ PERIOD_0	DI1_CLK period option 0. This parameter defines the period of the clock that the DI1 works with. This parameter contains integer part (bits [6:4]) and fractional part (bits [3:0]). Setting this value to 1.0 (default) means that the DI1 works on the fastest possible clock. Setting a value smaller than 1.0 is not allowed. The value to be programmed to the DI1_CLK_PERIOD_1 field is equal to: Fast_freq/Target_freq Where: Target_freq = The frequency that the DI clock works with Fast_freq = fastest possible clock that the DI can work with
15 CLCOK_MODE_ STAT	Clock mode status This is a read only bit indicating what is the current clock mode 1 current clock mode is 1 0 current clock mode is 0

Table continues on the next page...

IPU_PM field descriptions (continued)

Field	Description
14 DI0_SRM_ CLOCK_ CHANGE_ MODE	SRM clock change mode When the clock is going to be changed to any new ratio other then 1:1, 1:2. 1:4. The user needs to prepare an alternate set of DI setting in the SRM. This bit enable this mode. This bit is self cleared. 1 SRM clock change mode is enabled; the next clock change will be done by updating the DI settings from the SRM 0 SRM clock change mode is disabled.
13–7 DI0_CLK_ PERIOD_1	DI0_CLK period option 1. This parameter defines the period of the clock that the DI0 works with. This parameter contains integer part (bits [6:4]) and fractional part (bits [3:0]). Setting this value to 1.0 (default) means that the DI0 works on the fastest possible clock. Setting a value smaller than 1.0 is not allowed. The value to be programmed to the DI0_CLK_PERIOD_1 field is equal to: Fast_freq/Target_freq Where: Target_freq = The frequency that the DI clock works with Fast_freq = fastest possible clock that the DI can work with
6–0 DI0_CLK_ PERIOD_0	DI0_CLK period option 0. This parameter defines the period of the clock that the DI0 works with. This parameter contains integer part (bits [6:4]) and fractional part (bits [3:0]). Setting this value to 1.0 (default) means that the DI0 works on the fastest possible clock. Setting a value smaller than 1.0 is not allowed. The value to be programmed to the DI0_CLK_PERIOD_1 field is equal to: Fast_freq/Target_freq Where: Target_freq = The frequency that the DI clock works with Fast_freq = fastest possible clock that the DI can work with

45.51.47 General Purpose Register (IPU_GPR)

The register contains general purpose bits.

Address: IPU_GPR is 1E00_0000h base + E4h offset = 1E00_00E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IPU_CH_BUF1_RDY1_CLR	IPU_CH_BUF1_RDY0_CLR	IPU_CH_BUF0_RDY1_CLR	IPU_CH_BUF0_RDY0_CLR	IPU_ALT_CH_BUF1_RDY1_CLR	IPU_ALT_CH_BUF1_RDY0_CLR	IPU_ALT_CH_BUF0_RDY1_CLR	IPU_ALT_CH_BUF0_RDY0_CLR	IPU_DI1_CLK_CHANGE_ACK_DIS	IPU_DI0_CLK_CHANGE_ACK_DIS	IPU_CH_BUF2_RDY1_CLR	IPU_CH_BUF2_RDY0_CLR	IPU_GPn[4:16]			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IPU_GPn[15:0]															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_GPR field descriptions

Field	Description
31 IPU_CH_BUF1_RDY1_CLR	This bit defines the IPU_CH_BUF1_RDY1 properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_CH_BUF1_RDY1** is w1c register 0 writing one to a bit of this register sets this bit **IPU_CH_BUF1_RDY1** is w1s register
30 IPU_CH_BUF1_RDY0_CLR	This bit defines the **IPU_CH_BUF1_RDY0** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_CH_BUF1_RDY0** is w1c register 0 writing one to a bit of this register sets this bit **IPU_CH_BUF1_RDY0** is w1s register
29 IPU_CH_BUF0_RDY1_CLR	This bit defines the **IPU_CH_BUF0_RDY1** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_CH_BUF0_RDY1** is w1c register 0 writing one to a bit of this register sets this bit **IPU_CH_BUF0_RDY1** is w1s register
28 IPU_CH_BUF0_RDY0_CLR	This bit defines the **IPU_CH_BUF0_RDY0** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_CH_BUF0_RDY0** is w1c register 0 writing one to a bit of this register sets this bit **IPU_CH_BUF0_RDY0** is w1s register
27 IPU_ALT_CH_BUF1_RDY1_CLR	This bit defines the **IPU_ALT_CH_BUF1_RDY1** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_ALT_CH_BUF1_RDY1** is w1c register 0 writing one to a bit of this register sets this bit **IPU_ALT_CH_BUF1_RDY1** is w1s register

Table continues on the next page...

IPU_GPR field descriptions (continued)

Field	Description
26 IPU_ALT_CH_BUF1_RDY0_CLR	This bit defines the **IPU_ALT_CH_BUF1_RDY0** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_ALT_CH_BUF1_RDY0** is w1c register 0 writing one to a bit of this register sets this bit **IPU_ALT_CH_BUF1_RDY0** is w1s register
25 IPU_ALT_CH_BUF0_RDY1_CLR	This bit defines the **IPU_ALT_CH_BUF0_RDY1** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_ALT_CH_BUF0_RDY1** is w1c register 0 writing one to a bit of this register sets this bit **IPU_ALT_CH_BUF0_RDY1** is w1s register
24 IPU_ALT_CH_BUF0_RDY0_CLR	This bit defines the **IPU_ALT_CH_BUF0_RDY0** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_ALT_CH_BUF0_RDY0** is w1c register 0 writing one to a bit of this register sets this bit **IPU_ALT_CH_BUF0_RDY0** is w1s register
23 IPU_DI1_CLK_CHANGE_ACK_DIS	Disable DI1's clock change mechanism. 1 clock change mechanism is disabled. DI automatically acknowledges a clock change request 0 clock change mechanism is disabled. DI performs the clock change procedure
22 IPU_DI0_CLK_CHANGE_ACK_DIS	Disable DI0's clock change mechanism. 1 clock change mechanism is disabled. DI automatically acknowledges a clock change request 0 clock change mechanism is disabled. DI performs the clock change procedure
21 IPU_CH_BUF2_RDY1_CLR	This bit defines the **IPU_CH_BUF2_RDY1** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_CH_BUF2_RDY1** is w1c register 0 writing one to a bit of this register sets this bit **IPU_CH_BUF2_RDY1** is w1s register
20 IPU_CH_BUF2_RDY0_CLR	This bit defines the **IPU_CH_BUF2_RDY0** properties. This register can be a write one to clear OR write one to set. 1 writing one to a bit of this register clears this bit; **IPU_CH_BUF2_RDY0** is w1c register 0 writing one to a bit of this register sets this bit **IPU_CH_BUF2_RDY0** is w1s register
19–0 IPU_GPn	IPU General Purpose bit. *n* Indicates the corresponding DMA channel number. This bits are general Read/Write bits, reserved for future use

45.51.48 Channel Double Buffer Mode Select 0 Register (IPU_CH_DB_MODE_SEL0)

The register contains double buffer mode select control information for 32 IPU's DMA channels.

Address: IPU_CH_DB_MODE_SEL0 is 1E00_0000h base + 150h offset = 1E00_0150h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DMA_CH_DB_MODE_SEL_31	0	DMA_CH_DB_MODE_SEL_29	DMA_CH_DB_MODE_SEL_28	DMA_CH_DB_MODE_SEL_27	DMA_CH_DB_MODE_SEL_26	DMA_CH_DB_MODE_SEL_25	DMA_CH_DB_MODE_SEL_24	DMA_CH_DB_MODE_SEL_23	DMA_CH_DB_MODE_SEL_22	DMA_CH_DB_MODE_SEL_21	DMA_CH_DB_MODE_SEL_20	DMA_CH_DB_MODE_SEL_19	DMA_CH_DB_MODE_SEL_18	DMA_CH_DB_MODE_SEL_17	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DMA_CH_DB_MODE_SEL_15	DMA_CH_DB_MODE_SEL_14	DMA_CH_DB_MODE_SEL_13	DMA_CH_DB_MODE_SEL_12	DMA_CH_DB_MODE_SEL_11	DMA_CH_DB_MODE_SEL_10	DMA_CH_DB_MODE_SEL_9	DMA_CH_DB_MODE_SEL_8	0		DMA_CH_DB_MODE_SEL_5	0	DMA_CH_DB_MODE_SEL_3	DMA_CH_DB_MODE_SEL_2	DMA_CH_DB_MODE_SEL_1	DMA_CH_DB_MODE_SEL_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CH_DB_MODE_SEL0 field descriptions

Field	Description
31 DMA_CH_DB_MODE_SEL_31	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
29 DMA_CH_DB_MODE_SEL_29	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
28 DMA_CH_DB_MODE_SEL_28	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
27 DMA_CH_DB_MODE_SEL_27	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_CH_DB_MODE_SEL0 field descriptions (continued)

Field	Description
	0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
26 DMA_CH_DB_ MODE_SEL_26	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
25 DMA_CH_DB_ MODE_SEL_25	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
24 DMA_CH_DB_ MODE_SEL_24	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
23 DMA_CH_DB_ MODE_SEL_23	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
22 DMA_CH_DB_ MODE_SEL_22	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
21 DMA_CH_DB_ MODE_SEL_21	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
20 DMA_CH_DB_ MODE_SEL_20	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
19 DMA_CH_DB_ MODE_SEL_19	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
18 DMA_CH_DB_ MODE_SEL_18	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_CH_DB_MODE_SEL0 field descriptions (continued)

Field	Description
	0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
17 DMA_CH_DB_ MODE_SEL_17	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
16 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
15 DMA_CH_DB_ MODE_SEL_15	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
14 DMA_CH_DB_ MODE_SEL_14	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
13 DMA_CH_DB_ MODE_SEL_13	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
12 DMA_CH_DB_ MODE_SEL_12	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
11 DMA_CH_DB_ MODE_SEL_11	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
10 DMA_CH_DB_ MODE_SEL_10	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
9 DMA_CH_DB_ MODE_SEL_9	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

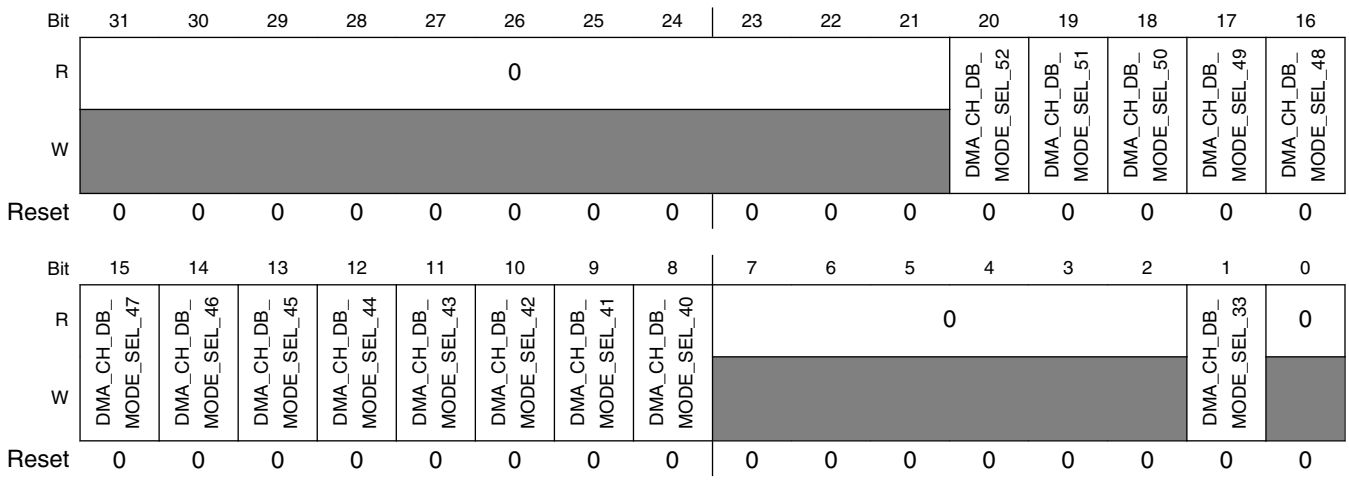
IPU_CH_DB_MODE_SEL0 field descriptions (continued)

Field	Description
	0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
8 DMA_CH_DB_ MODE_SEL_8	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
5 DMA_CH_DB_ MODE_SEL_5	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
4 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
3 DMA_CH_DB_ MODE_SEL_3	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
2 DMA_CH_DB_ MODE_SEL_2	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
1 DMA_CH_DB_ MODE_SEL_1	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
0 DMA_CH_DB_ MODE_SEL_0	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.

45.51.49 Channel Double Buffer Mode Select 1 Register (IPU_CH_DB_MODE_SEL1)

The register contains double buffer mode select control information for 32 IPU's DMA channels.

Address: IPU_CH_DB_MODE_SEL1 is 1E00_0000h base + 154h offset = 1E00_0154h



IPU_CH_DB_MODE_SEL1 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
20 DMA_CH_DB_MODE_SEL_52	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
19 DMA_CH_DB_MODE_SEL_51	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
18 DMA_CH_DB_MODE_SEL_50	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
17 DMA_CH_DB_MODE_SEL_49	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_CH_DB_MODE_SEL1 field descriptions (continued)

Field	Description
	0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
16 DMA_CH_DB_ MODE_SEL_48	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
15 DMA_CH_DB_ MODE_SEL_47	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
14 DMA_CH_DB_ MODE_SEL_46	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
13 DMA_CH_DB_ MODE_SEL_45	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
12 DMA_CH_DB_ MODE_SEL_44	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
11 DMA_CH_DB_ MODE_SEL_43	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
10 DMA_CH_DB_ MODE_SEL_42	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
9 DMA_CH_DB_ MODE_SEL_41	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
8 DMA_CH_DB_ MODE_SEL_40	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

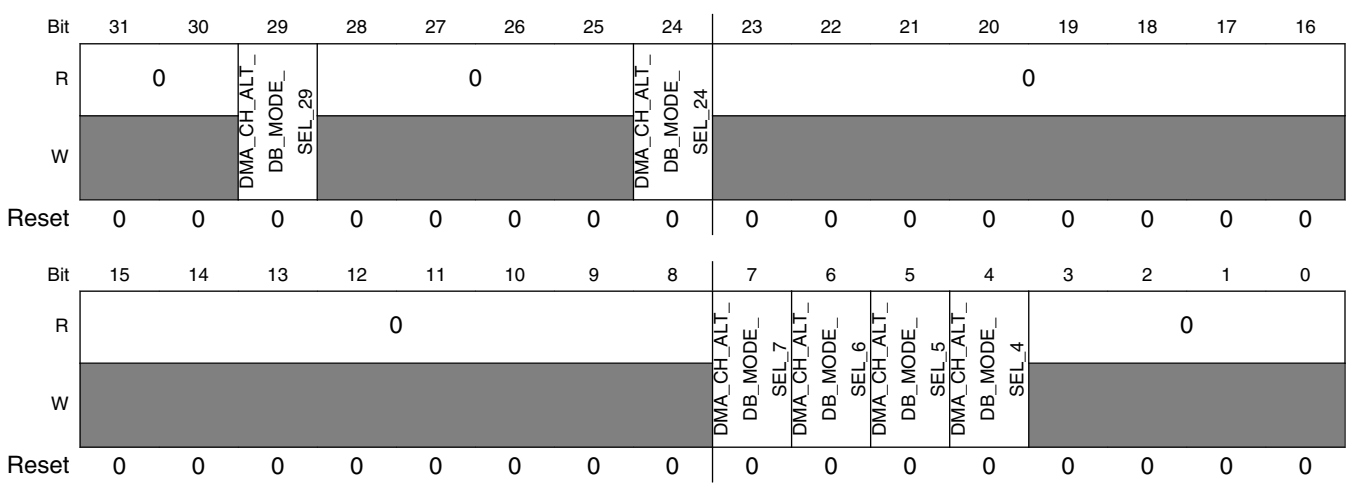
IPU_CH_DB_MODE_SEL1 field descriptions (continued)

Field	Description
	0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
7–2 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
1 DMA_CH_DB_ MODE_SEL_33	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.

45.51.50 Alternate Channel Double Buffer Mode Select 0 Register (IPU_ALT_CH_DB_MODE_SEL0)

The register contains double buffer mode select control information for 32 IPU's DMA channels.

Address: IPU_ALT_CH_DB_MODE_SEL0 is 1E00_0000h base + 168h offset = 1E00_0168h



IPU_ALT_CH_DB_MODE_SEL0 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_ALT_CH_DB_MODE_SEL0 field descriptions (continued)

Field	Description
	0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
29 DMA_CH_ALT_ DB_MODE_ SEL_29	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
28–25 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
24 DMA_CH_ALT_ DB_MODE_ SEL_24	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
23–8 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
7 DMA_CH_ALT_ DB_MODE_ SEL_7	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
6 DMA_CH_ALT_ DB_MODE_ SEL_6	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
5 DMA_CH_ALT_ DB_MODE_ SEL_5	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
4 DMA_CH_ALT_ DB_MODE_ SEL_4	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
3–0 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

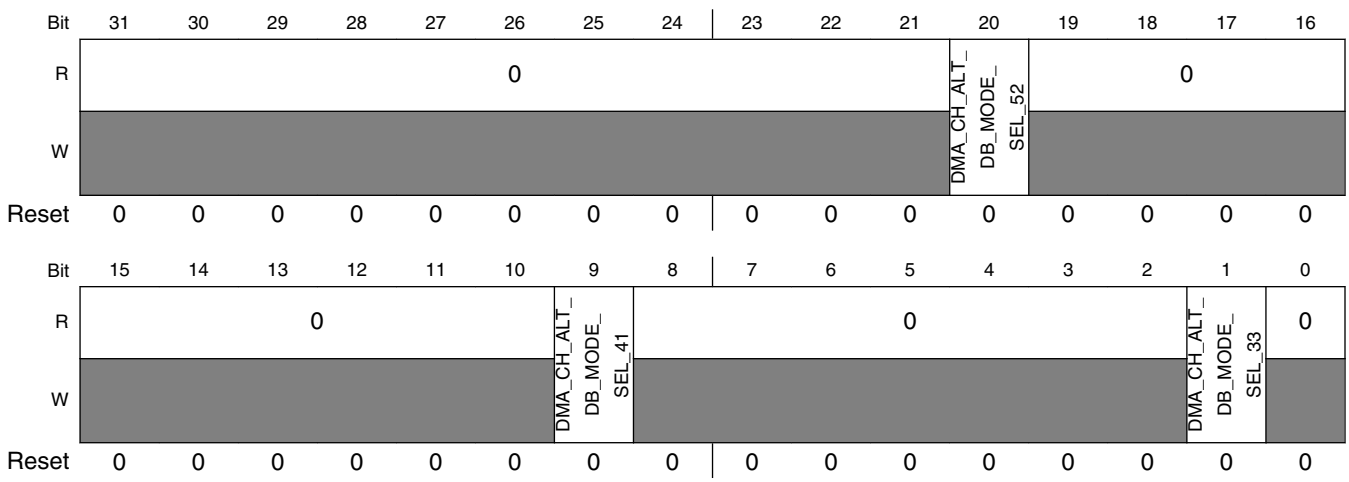
IPU_ALT_CH_DB_MODE_SEL0 field descriptions (continued)

Field	Description
	0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.

45.51.51 Alternate Channel Double Buffer Mode Select1 Register (IPU_ALT_CH_DB_MODE_SEL1)

The register contains double buffer mode select control information for 32 IPU's DMA channels.

Address: IPU_ALT_CH_DB_MODE_SEL1 is 1E00_0000h base + 16Ch offset = 1E00_016Ch



IPU_ALT_CH_DB_MODE_SEL1 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
20 DMA_CH_ALT_DB_MODE_SEL_52	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
19–10 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.

Table continues on the next page...

IPU_ALT_CH_DB_MODE_SEL1 field descriptions (continued)

Field	Description
9 DMA_CH_ALT_DB_MODE_SEL_41	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
8–2 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
1 DMA_CH_ALT_DB_MODE_SEL_33	Double Buffer Mode Select. This bit indicates if a double buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Double buffer is not used for this channel. 1 Double buffer is used for this channel.

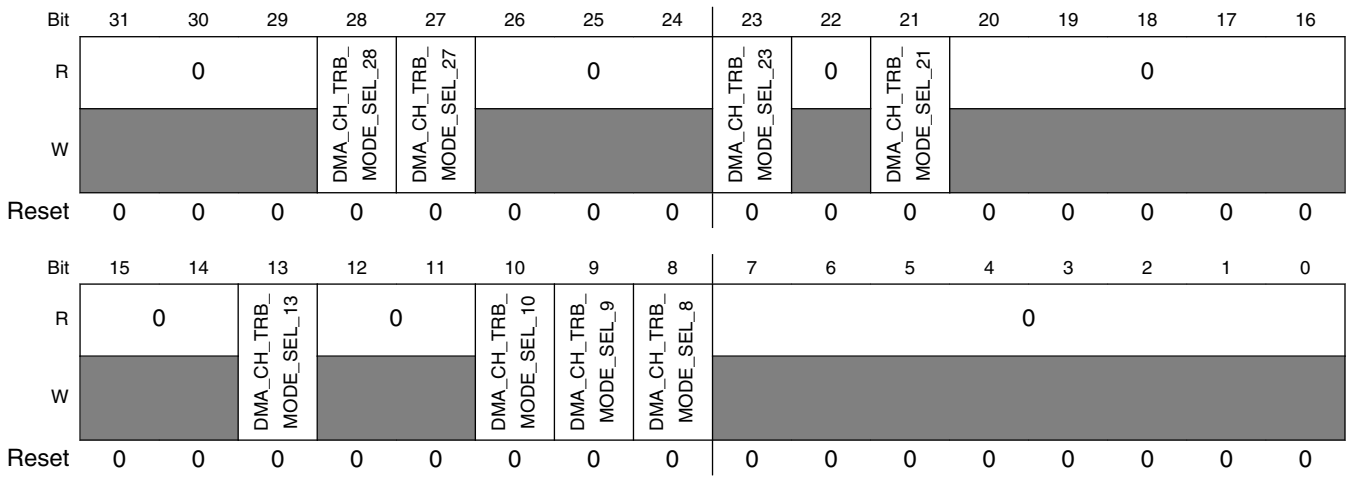
45.51.52 Alternate Channel Triple Buffer Mode Select 0 Register (IPU_ALT_CH_TRB_MODE_SEL0)

The register contains triple buffer mode select control information for 32 IPU's DMA channels.

When the channel is configured for triple buffer mode. The double buffer mode settings configured on the corresponding DB_MODE_SEL bit are overridden.

- Hide VPU_SUB_FRAME_SYNC for all versions
- Show VPU_SUB_FRAME_SYNC for IPUv3H version.
- The table below tagged with other settings (like IPU3M_only) should be hidden in IPUv3H version.
- This requires some sophisticated conditional tag settings

Address: IPU_ALT_CH_TRB_MODE_SEL0 is 1E00_0000h base + 178h offset = 1E00_0178h



IPU_ALT_CH_TRB_MODE_SEL0 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
28 DMA_CH_TRB_MODE_SEL_28	Triple Buffer Mode Select. This bit indicates if a triple buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
27 DMA_CH_TRB_MODE_SEL_27	Triple Buffer Mode Select. This bit indicates if a triple buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
26–24 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
23 DMA_CH_TRB_MODE_SEL_23	Triple Buffer Mode Select. This bit indicates if a triple buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
22 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.

Table continues on the next page...

IPU_ALT_CH_TRB_MODE_SEL0 field descriptions (continued)

Field	Description
21 DMA_CH_TRB_MODE_SEL_21	Triple Buffer Mode Select. This bit indicates if a triple buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
20–14 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
13 DMA_CH_TRB_MODE_SEL_13	Triple Buffer Mode Select. This bit indicates if a triple buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
12–11 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
10 DMA_CH_TRB_MODE_SEL_10	Triple Buffer Mode Select. This bit indicates if a triple buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
9 DMA_CH_TRB_MODE_SEL_9	Triple Buffer Mode Select. This bit indicates if a triple buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
8 DMA_CH_TRB_MODE_SEL_8	Triple Buffer Mode Select. This bit indicates if a triple buffer is used for this channel. *n* Indicates the corresponding DMA channel number. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.
7–0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Triple buffer is not used for this channel. 1 Triple buffer is used for this channel.

45.51.53 Interrupt Status Register 1 (IPU_INT_STAT_1)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of EOF (end of frame) of DMA Channels interrupts [31:0] can be found in this register.

Address: IPU_INT_STAT_1 is 1E00_0000h base + 200h offset = 1E00_0200h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_EOF_31	0	IDMAC_EOF_29	IDMAC_EOF_28	IDMAC_EOF_27	IDMAC_EOF_26	IDMAC_EOF_25	IDMAC_EOF_24	IDMAC_EOF_23	IDMAC_EOF_22	IDMAC_EOF_21	IDMAC_EOF_20	IDMAC_EOF_19	IDMAC_EOF_18	IDMAC_EOF_17	0
W	w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_EOF_15	IDMAC_EOF_14	IDMAC_EOF_13	IDMAC_EOF_12	IDMAC_EOF_11	IDMAC_EOF_10	IDMAC_EOF_9	IDMAC_EOF_8	0		IDMAC_EOF_5	0	IDMAC_EOF_3	IDMAC_EOF_2	IDMAC_EOF_1	IDMAC_EOF_0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c			w1c		w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_1 field descriptions

Field	Description
31 IDMAC_EOF_31	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
29 IDMAC_EOF_29	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
28 IDMAC_EOF_28	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_1 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
27 IDMAC_EOF_27	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
26 IDMAC_EOF_26	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
25 IDMAC_EOF_25	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
24 IDMAC_EOF_24	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
23 IDMAC_EOF_23	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
22 IDMAC_EOF_22	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
21 IDMAC_EOF_21	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
20 IDMAC_EOF_20	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
19 IDMAC_EOF_19	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_1 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
18 IDMAC_EOF_18	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_EOF_17	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
16 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_EOF_15	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_EOF_14	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_EOF_13	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_EOF_12	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_EOF_11	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
10 IDMAC_EOF_10	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_1 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_EOF_9	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
8 IDMAC_EOF_8	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
5 IDMAC_EOF_5	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
4 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
3 IDMAC_EOF_3	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
2 IDMAC_EOF_2	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_EOF_1	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
0 IDMAC_EOF_0	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

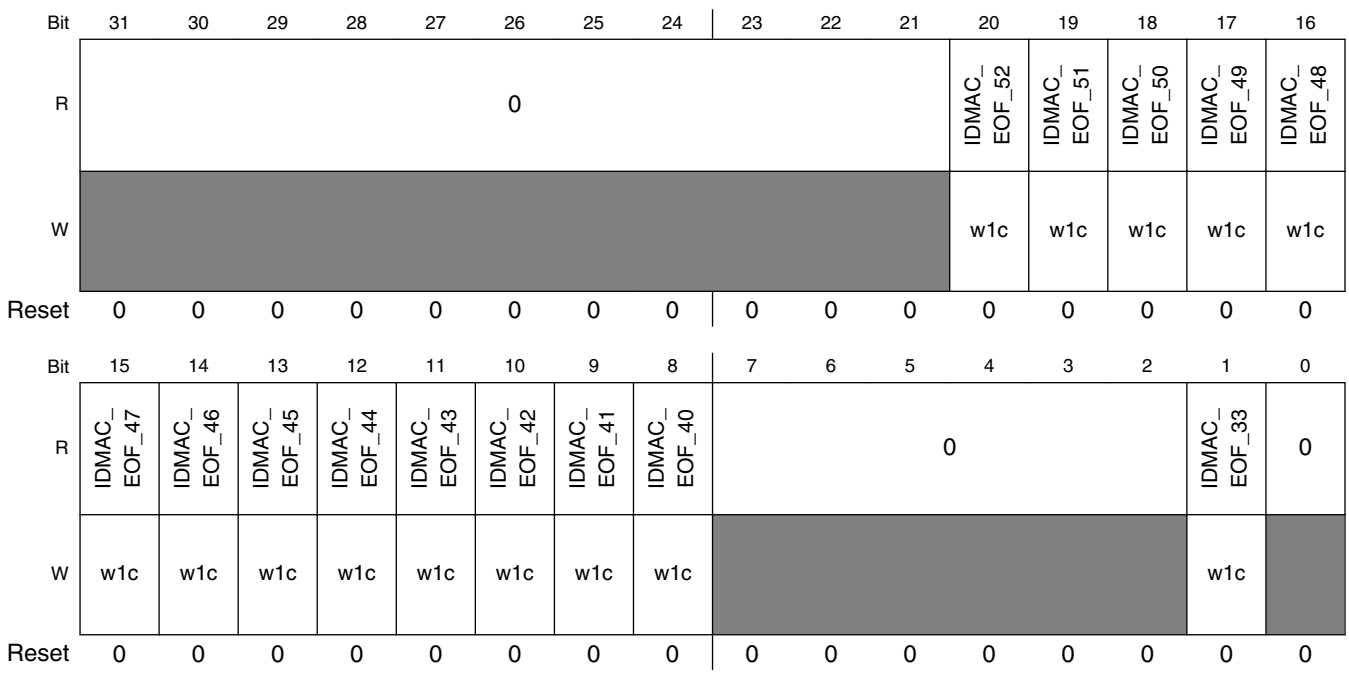
IPU_INT_STAT_1 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.

45.51.54 Interrupt Status Register2 (IPU_INT_STAT_2)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of EOF (end of frame) of DMA Channels interrupts [63:32] can be found in this register.

Address: IPU_INT_STAT_2 is 1E00_0000h base + 204h offset = 1E00_0204h



IPU_INT_STAT_2 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
20 IDMAC_EOF_52	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_2 field descriptions (continued)

Field	Description
19 IDMAC_EOF_51	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
18 IDMAC_EOF_50	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_EOF_49	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
16 IDMAC_EOF_48	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_EOF_47	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_EOF_46	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_EOF_45	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_EOF_44	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_EOF_43	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_2 field descriptions (continued)

Field	Description
10 IDMAC_EOF_42	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_EOF_41	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
8 IDMAC_EOF_40	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
7–2 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_EOF_33	End of Frame of Channel interrupt. This bit is the status bit of End Of Frame of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.

45.51.55 Interrupt Status Register 3 (IPU_INT_STAT_3)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of NFACK (New Frame Ack) of DMA Channels interrupts [31:0] can be found in this register.

Address: IPU_INT_STAT_3 is 1E00_0000h base + 208h offset = 1E00_0208h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_NFACK_31	0	IDMAC_NFACK_29	IDMAC_NFACK_28	IDMAC_NFACK_27	IDMAC_NFACK_26	IDMAC_NFACK_25	IDMAC_NFACK_24	IDMAC_NFACK_23	IDMAC_NFACK_22	IDMAC_NFACK_21	IDMAC_NFACK_20	IDMAC_NFACK_19	IDMAC_NFACK_18	IDMAC_NFACK_17	0
W	w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_NFACK_15	IDMAC_NFACK_14	IDMAC_NFACK_13	IDMAC_NFACK_12	IDMAC_NFACK_11	IDMAC_NFACK_10	IDMAC_NFACK_9	IDMAC_NFACK_8	0		IDMAC_NFACK_5	0	IDMAC_NFACK_3	IDMAC_NFACK_2	IDMAC_NFACK_1	IDMAC_NFACK_0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c			w1c		w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_3 field descriptions

Field	Description
31 IDMAC_ NFACK_31	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
29 IDMAC_ NFACK_29	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_3 field descriptions (continued)

Field	Description
28 IDMAC_ NFACK_28	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
27 IDMAC_ NFACK_27	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
26 IDMAC_ NFACK_26	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
25 IDMAC_ NFACK_25	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
24 IDMAC_ NFACK_24	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
23 IDMAC_ NFACK_23	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
22 IDMAC_ NFACK_22	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
21 IDMAC_ NFACK_21	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
20 IDMAC_ NFACK_20	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_3 field descriptions (continued)

Field	Description
19 IDMAC_ NFACK_19	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
18 IDMAC_ NFACK_18	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_ NFACK_17	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
16 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_ NFACK_15	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_ NFACK_14	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_ NFACK_13	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_ NFACK_12	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_ NFACK_11	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_3 field descriptions (continued)

Field	Description
10 IDMAC_ NFACK_10	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_ NFACK_9	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
8 IDMAC_ NFACK_8	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
5 IDMAC_ NFACK_5	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
4 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
3 IDMAC_ NFACK_3	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
2 IDMAC_ NFACK_2	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_ NFACK_1	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_3 field descriptions (continued)

Field	Description
0 IDMAC_ NFACK_0	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

45.51.56 Interrupt Status Register 5 (IPU_INT_STAT_5)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of the New-frame before end-of-frame indication (NFB4EOF_ERR) of DMA Channels interrupts [31:0] can be found in this register.

Address: IPU_INT_STAT_5 is 1E00_0000h base + 210h offset = 1E00_0210h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_NFB4EOF_ERR_31	0	IDMAC_NFB4EOF_ERR_29	IDMAC_NFB4EOF_ERR_28	IDMAC_NFB4EOF_ERR_27	IDMAC_NFB4EOF_ERR_26	IDMAC_NFB4EOF_ERR_25	IDMAC_NFB4EOF_ERR_24	IDMAC_NFB4EOF_ERR_23	IDMAC_NFB4EOF_ERR_22	IDMAC_NFB4EOF_ERR_21	IDMAC_NFB4EOF_ERR_20	IDMAC_NFB4EOF_ERR_19	IDMAC_NFB4EOF_ERR_18	IDMAC_NFB4EOF_ERR_17	0
W	w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_NFB4EOF_ERR_15	IDMAC_NFB4EOF_ERR_14	IDMAC_NFB4EOF_ERR_13	IDMAC_NFB4EOF_ERR_12	IDMAC_NFB4EOF_ERR_11	IDMAC_NFB4EOF_ERR_10	IDMAC_NFB4EOF_ERR_9	IDMAC_NFB4EOF_ERR_8	0		IDMAC_NFB4EOF_ERR_5	0	IDMAC_NFB4EOF_ERR_3	IDMAC_NFB4EOF_ERR_2	IDMAC_NFB4EOF_ERR_1	IDMAC_NFB4EOF_ERR_0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c			w1c		w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_5 field descriptions

Field	Description
31 IDMAC_ NFB4EOF_ ERR_31	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_5 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
29 IDMAC_ NFB4EOF_ ERR_29	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
28 IDMAC_ NFB4EOF_ ERR_28	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
27 IDMAC_ NFB4EOF_ ERR_27	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
26 IDMAC_ NFB4EOF_ ERR_26	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
25 IDMAC_ NFB4EOF_ ERR_25	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
24 IDMAC_ NFB4EOF_ ERR_24	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
23 IDMAC_ NFB4EOF_ ERR_23	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_5 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
22 IDMAC_ NFB4EOF_ ERR_22	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
21 IDMAC_ NFB4EOF_ ERR_21	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
20 IDMAC_ NFB4EOF_ ERR_20	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
19 IDMAC_ NFB4EOF_ ERR_19	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
18 IDMAC_ NFB4EOF_ ERR_18	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_ NFB4EOF_ ERR_17	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
16 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_ NFB4EOF_ ERR_15	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_5 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_ NFB4EOF_ ERR_14	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_ NFB4EOF_ ERR_13	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_ NFB4EOF_ ERR_12	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_ NFB4EOF_ ERR_11	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
10 IDMAC_ NFB4EOF_ ERR_10	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_ NFB4EOF_ ERR_9	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
8 IDMAC_ NFB4EOF_ ERR_8	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

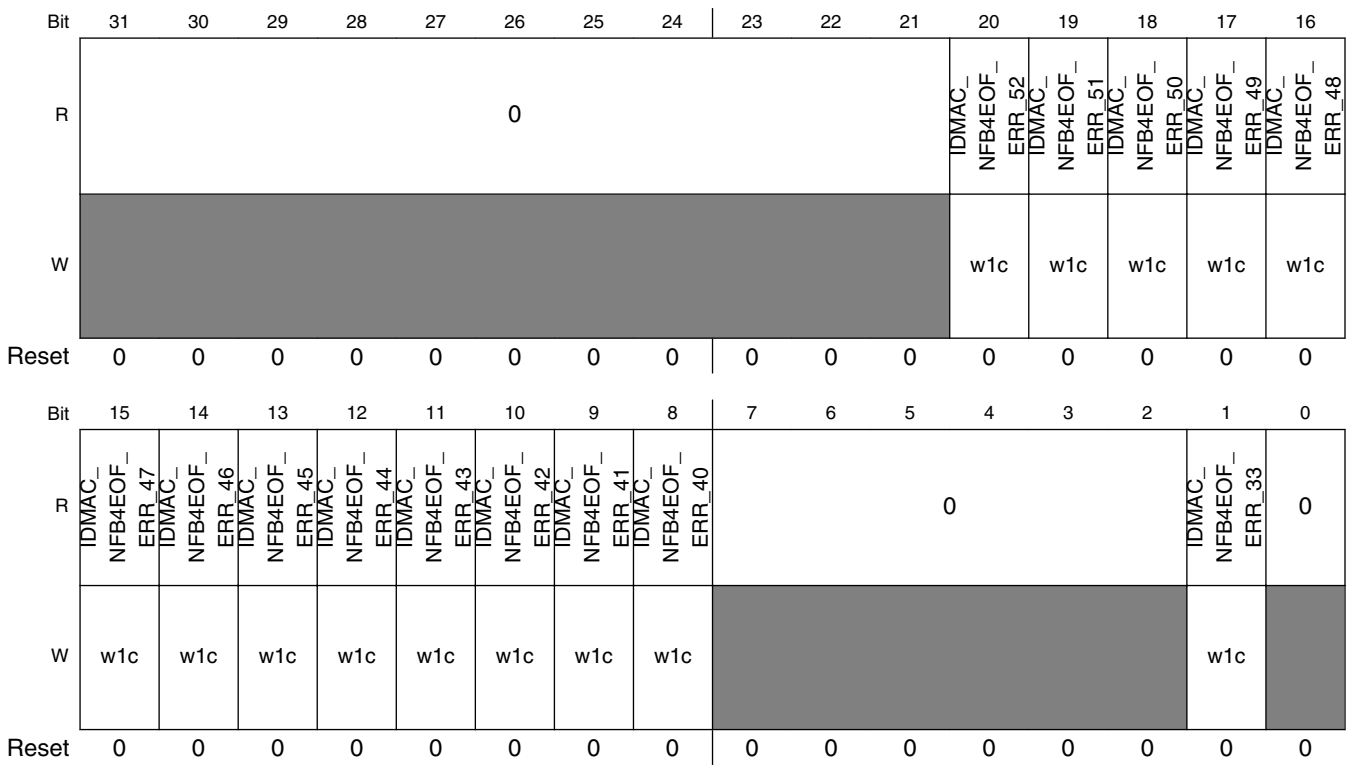
IPU_INT_STAT_5 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
5 IDMAC_ NFB4EOF_ ERR_5	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
4 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
3 IDMAC_ NFB4EOF_ ERR_3	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
2 IDMAC_ NFB4EOF_ ERR_2	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_ NFB4EOF_ ERR_1	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
0 IDMAC_ NFB4EOF_ ERR_0	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

45.51.57 Interrupt Status Register 6 (IPU_INT_STAT_6)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of the New-frame before end-of-frame indication (NFB4EOF_ERR) of DMA Channels interrupts [63:32] can be found in this register.

Address: IPU_INT_STAT_6 is 1E00_0000h base + 214h offset = 1E00_0214h



IPU_INT_STAT_6 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
20 IDMAC_ NFB4EOF_ ERR_52	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_6 field descriptions (continued)

Field	Description
19 IDMAC_ NFB4EOF_ ERR_51	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
18 IDMAC_ NFB4EOF_ ERR_50	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_ NFB4EOF_ ERR_49	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
16 IDMAC_ NFB4EOF_ ERR_48	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_ NFB4EOF_ ERR_47	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_ NFB4EOF_ ERR_46	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_ NFB4EOF_ ERR_45	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_ NFB4EOF_ ERR_44	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_6 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_ NFB4EOF_ ERR_43	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
10 IDMAC_ NFB4EOF_ ERR_42	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_ NFB4EOF_ ERR_41	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
8 IDMAC_ NFB4EOF_ ERR_40	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
7–2 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_ NFB4EOF_ ERR_33	New Frame before end-of-frame error indication of Channel interrupt. This bit is the status bit of New Frame before end-of-frame error interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.

45.51.58 Interrupt Status Register7 1 (IPU_INT_STAT_7)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of the End-of-Scroll indication (EOS) of DMA Channels interrupts [31:0] can be found in this register.

Address: IPU_INT_STAT_7 is 1E00_0000h base + 218h offset = 1E00_0218h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_EOS_EN_31	0	IDMAC_EOS_EN_29	IDMAC_EOS_EN_28	IDMAC_EOS_EN_27	IDMAC_EOS_EN_26	IDMAC_EOS_EN_25	IDMAC_EOS_EN_24	IDMAC_EOS_EN_23	0			IDMAC_EOS_EN_19	0		
W	w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c				w1c			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_7 field descriptions

Field	Description
31 IDMAC_EOS_EN_31	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
29 IDMAC_EOS_EN_29	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_7 field descriptions (continued)

Field	Description
28 IDMAC_EOS_EN_28	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
27 IDMAC_EOS_EN_27	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
26 IDMAC_EOS_EN_26	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
25 IDMAC_EOS_EN_25	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
24 IDMAC_EOS_EN_24	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
23 IDMAC_EOS_EN_23	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
22–20 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
19 IDMAC_EOS_EN_19	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_7 field descriptions (continued)

Field	Description
18–0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.

45.51.59 Interrupt Status Register 8 (IPU_INT_STAT_8)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. All the status bits of the End of Scroll indication (EOS) of DMA Channels interrupts [63:32] can be found in this register.

Address: IPU_INT_STAT_8 is 1E00_0000h base + 21Ch offset = 1E00_021Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											IDMAC_EOS_EN_52	IDMAC_EOS_EN_51	0		
W												w1c	w1c			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			IDMAC_EOS_EN_44	IDMAC_EOS_EN_43	IDMAC_EOS_EN_42	IDMAC_EOS_EN_41		0						IDMAC_EOS_EN_33	0
W				w1c	w1c	w1c	w1c								w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_8 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
20 IDMAC_EOS_EN_52	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_8 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
19 IDMAC_EOS_EN_51	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
18–13 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_EOS_EN_44	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_EOS_EN_43	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
10 IDMAC_EOS_EN_42	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_EOS_EN_41	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
8–2 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_EOS_EN_33	End of Scroll indication of Channel interrupt. This bit is the status bit of End of Scroll interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

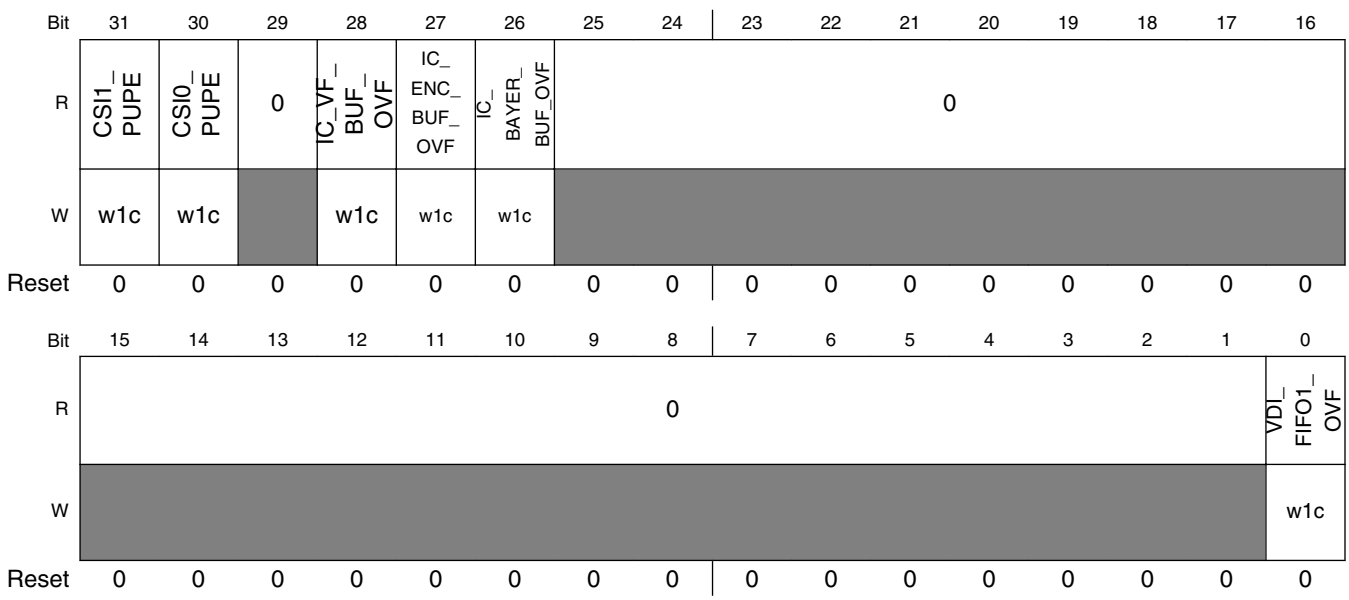
IPU_INT_STAT_8 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.

45.51.60 Interrupt Status Register 9 (IPU_INT_STAT_9)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. This register holds the error interrupt indications coming from different modules within All the bits in this register are write one to clear.

Address: IPU_INT_STAT_9 is 1E00_0000h base + 220h offset = 1E00_0220h



IPU_INT_STAT_9 field descriptions

Field	Description
31 CSI1_PUPE	CSI1_PUPE - CSI1 parameters update error interrupt. This bit indicates on an interrupt that is a result of an error generated by the CSI1. The error is generated in case where new frame arrived from the CSI1 before the completion of the CSI1's parameters update by the SRM 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

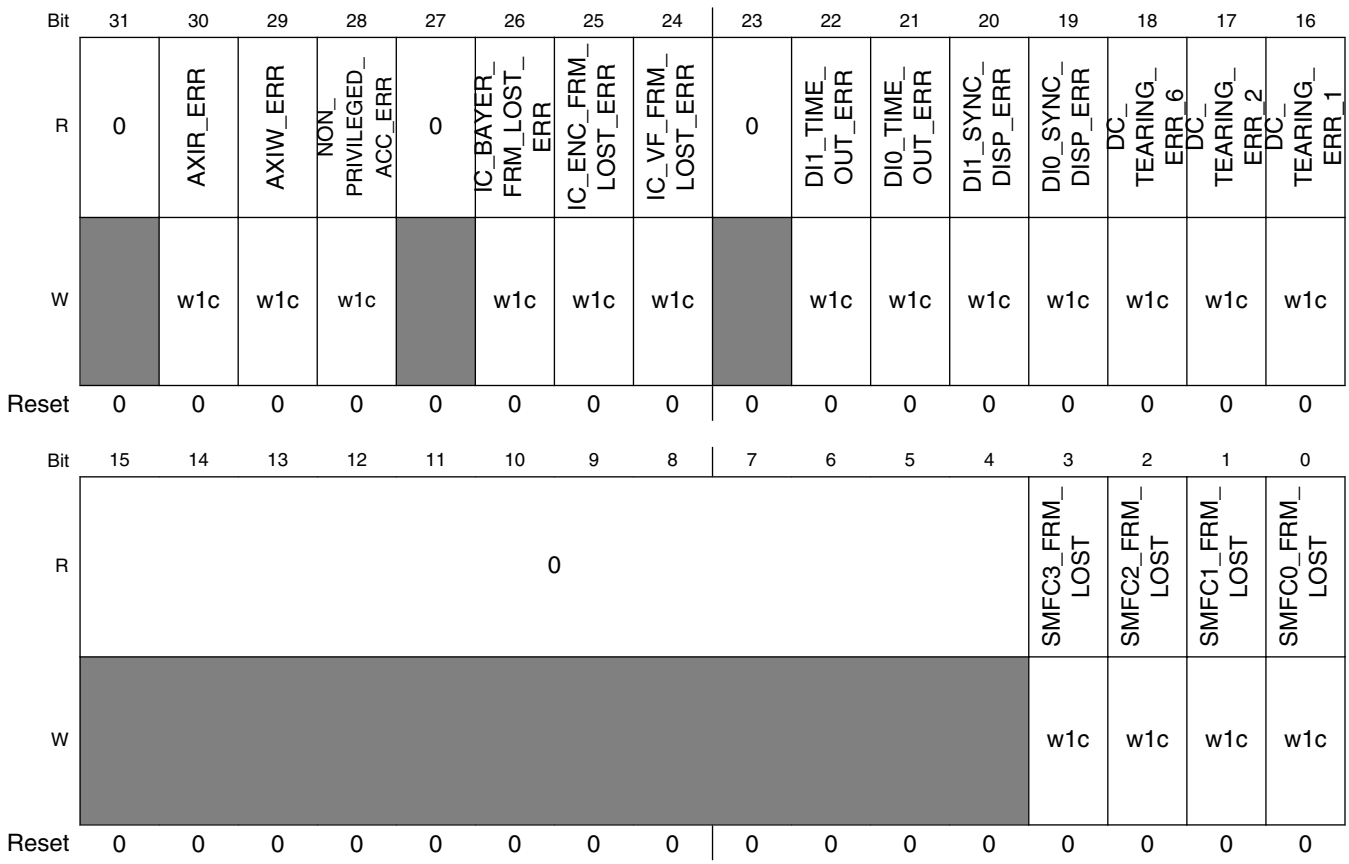
IPU_INT_STAT_9 field descriptions (continued)

Field	Description
30 CSI0_PUPE	CSI0_PUPE - CSI0 parameters update error interrupt. This bit indicates on an interrupt that is a result of an error generated by the CSI0. The error is generated in case where new frame arrived from the CSI0 before the completion of the CSI0's parameters update by the SRM 0 Interrupt is cleared. 1 Interrupt is requested.
29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28 IC_VF_BUF_OVF	This bit indicates on an interrupt that is a result of the IC Buffer overflow for view finder coming from the IC. The user needs to write 1 to this bit in order to clear it. 0 Interrupt is cleared. 1 Interrupt is requested.
27 IC_ENC_BUF_OVF	This bit indicates on an interrupt that is a result of the IC Buffer overflow for encoding coming from the IC. The user needs to write 1 to this bit in order to clear it. 0 Interrupt is cleared. 1 Interrupt is requested.
26 IC_BAYER_BUF_OVF	This bit indicates on an interrupt that is a result of the IC Buffer overflow for Bayer coming from the IC. The user needs to write 1 to this bit in order to clear it. 0 Interrupt is cleared. 1 Interrupt is requested.
25–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 VDI_FIFO1_OVF	FIFO1 overflow Interrupt1 The VDIC generate FIFO1 overflow interrupt1 when write pointer of FIFO1 overrun read pointer. 0 Interrupt is cleared. 1 Interrupt is requested.

45.51.61 Interrupt Status Register 10 (IPU_INT_STAT_10)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. This register holds error interrupt indications coming from different modules within All the bits in this register are write one to clear.

Address: IPU_INT_STAT_10 is 1E00_0000h base + 224h offset = 1E00_0224h



IPU_INT_STAT_10 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30 AXIR_ERR	This bit indicates on an interrupt that is a result of AXI read access resulted with error response. The user needs to write 1 to this bit in order to clear it. 0 Interrupt is cleared. 1 Interrupt is requested.
29 AXIW_ERR	This bit indicates on an interrupt that is a result of AXI write access resulted with error response. The user needs to write 1 to this bit in order to clear it. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_10 field descriptions (continued)

Field	Description
28 NON_ PRIVILEGED_ ACC_ERR	Non Privileged Access Error interrupt. This bit indicates on an interrupt that is a result of access the CPMEM or the DP memory in user mode 0 Interrupt is cleared. 1 Interrupt is requested.
27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26 IC_BAYER_ FRM_LOST_ ERR	This bit indicates on an interrupt that is a result of IC's Bayer frame lost. 0 Interrupt is disabled. 1 Interrupt is enabled.
25 IC_ENC_FRM_ LOST_ERR	This bit indicates on an interrupt that is a result of IC's encoding frame lost. 0 Interrupt is disabled. 1 Interrupt is enabled.
24 IC_VF_FRM_ LOST_ERR	This bit indicates on an interrupt that is a result of IC's view finder frame lost. 0 Interrupt is disabled. 1 Interrupt is enabled.
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22 DI1_TIME_ OUT_ERR	DI1 time out error interrupt This bit indicates on the interrupt that is a result of a time out error during a read access via DI1
21 DI0_TIME_ OUT_ERR	DI0 time out error interrupt This bit indicates on the interrupt that is a result of a time out error during a read access via DI0
20 DI1_SYNC_ DISP_ERR	DI1 Synchronous display error interrupt This bit indicates on the interrupt that is a result of an error during access to a synchronous display via DI1
19 DI0_SYNC_ DISP_ERR	DI0 Synchronous display error interrupt This bit indicates on the interrupt that is a result of an error during access to a synchronous display via DI0
18 DC_TEARING_ ERR_6	Tearing Error #6 interrupt This bit indicates on the interrupt that is a result of tearing error while the anti tearing mechanism is activated for DC channel 6
17 DC_TEARING_ ERR_2	Tearing Error #2 interrupt This bit indicates on the interrupt that is a result of tearing error while the anti tearing mechanism is activated for DC channel 2
16 DC_TEARING_ ERR_1	Tearing Error #1 interrupt This bit indicates on the interrupt that is a result of tearing error while the anti tearing mechanism is activated for DC channel 1
15–4 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IPU_INT_STAT_10 field descriptions (continued)

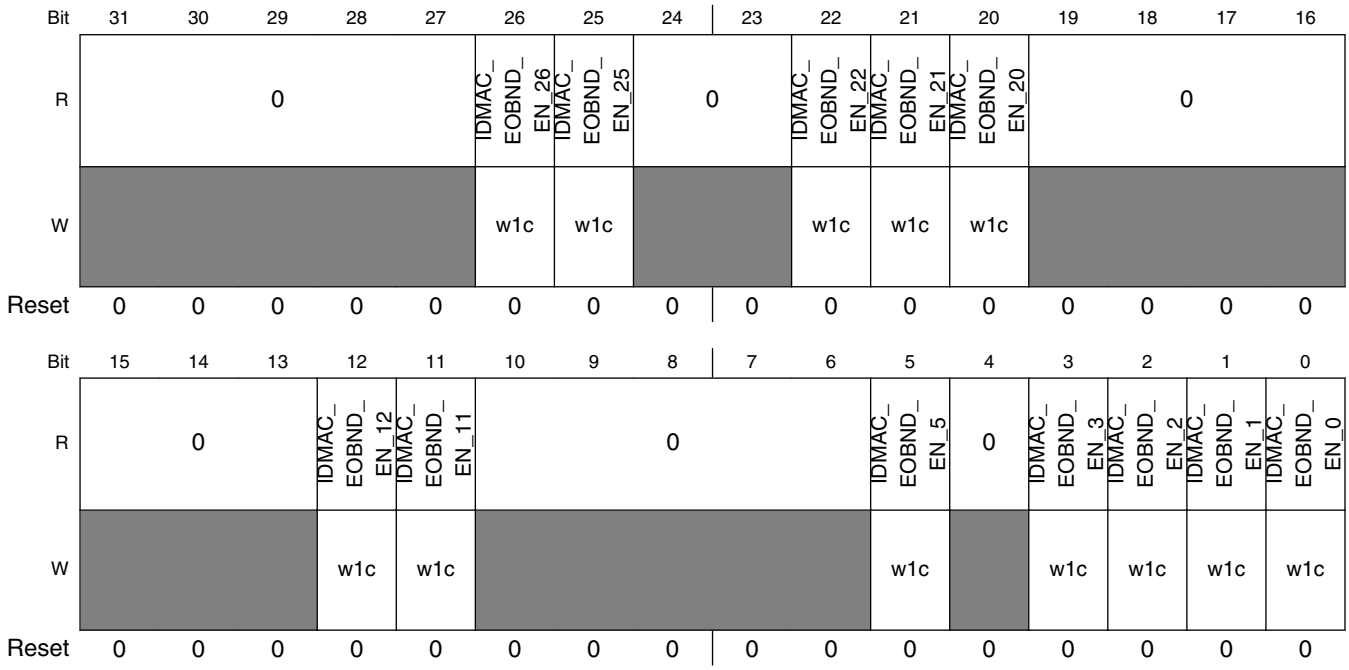
Field	Description
3 SMFC3_FRM_LOST	Frame Lost of SMFC channel 3 interrupt. This bit indicates on an interrupt that is a result of a result of a Frame Lost of SMFC channel 3 0 Interrupt is cleared. 1 Interrupt is requested.
2 SMFC2_FRM_LOST	Frame Lost of SMFC channel 2 interrupt. This bit indicates on an interrupt that is a result of a result of a Frame Lost of SMFC channel 2 0 Interrupt is cleared. 1 Interrupt is requested.
1 SMFC1_FRM_LOST	Frame Lost of SMFC channel 1 interrupt. This bit indicates on an interrupt that is a result of a result of a Frame Lost of SMFC channel 1 0 Interrupt is cleared. 1 Interrupt is requested.
0 SMFC0_FRM_LOST	Frame Lost of SMFC channel 0 interrupt. This bit indicates on an interrupt that is a result of a result of a Frame Lost of SMFC channel 0 0 Interrupt is cleared. 1 Interrupt is requested.

45.51.62 Interrupt Status Register 11 (IPU_INT_STAT_11)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of the end-of-band indication (EOBND) of DMA Channels interrupts [31:0] can be found in this register.

- Hide VDOA_SYNC for all versions
- Show VDOA_SYNC for IPUv3H version.
- The table below tagged with other settings (like IPU3M_only) should be hidden in IPUv3H version.

Address: IPU_INT_STAT_11 is 1E00_0000h base + 228h offset = 1E00_0228h



IPU_INT_STAT_11 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
26 IDMAC_ EOBND_EN_26	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
25 IDMAC_ EOBND_EN_25	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
24–23 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
22 IDMAC_ EOBND_EN_22	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_11 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
21 IDMAC_ EOBND_EN_21	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
20 IDMAC_ EOBND_EN_20	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
19–13 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_ EOBND_EN_12	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_ EOBND_EN_11	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
10–6 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
5 IDMAC_ EOBND_EN_5	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
4 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

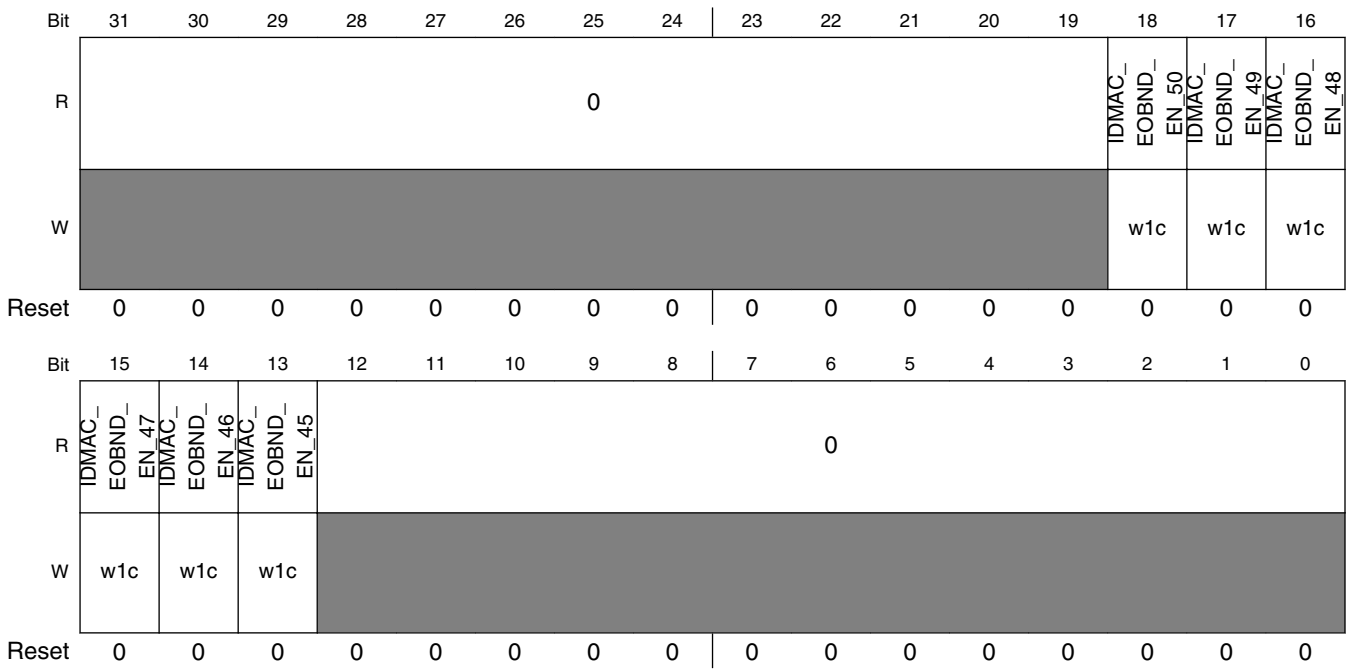
IPU_INT_STAT_11 field descriptions (continued)

Field	Description
3 IDMAC_ EOBND_EN_3	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
2 IDMAC_ EOBND_EN_2	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_ EOBND_EN_1	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
0 IDMAC_ EOBND_EN_0	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

45.51.63 Interrupt Status Register 12 (IPU_INT_STAT_12)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of the end-of-band indication (EOBND) of DMA Channels interrupts [63:32] can be found in this register.

Address: IPU_INT_STAT_12 is 1E00_0000h base + 22Ch offset = 1E00_022Ch



IPU_INT_STAT_12 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
18 IDMAC_ EOBND_EN_50	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_ EOBND_EN_49	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_12 field descriptions (continued)

Field	Description
16 IDMAC_ EOBND_EN_48	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_ EOBND_EN_47	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_ EOBND_EN_46	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_ EOBND_EN_45	end-of-band indication of Channel interrupt. This bit is the status bit of end-of-band interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
12–0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.

45.51.64 Interrupt Status Register 13 (IPU_INT_STAT_13)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of the Threshold crossing indication (TH) of DMA Channels interrupts [31:0] can be found in this register.

Address: IPU_INT_STAT_13 is 1E00_0000h base + 230h offset = 1E00_0230h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_TH_31	0	IDMAC_TH_29	IDMAC_TH_28	IDMAC_TH_27	IDMAC_TH_26	IDMAC_TH_25	IDMAC_TH_24	IDMAC_TH_23	IDMAC_TH_22	IDMAC_TH_21	IDMAC_TH_20	IDMAC_TH_19	IDMAC_TH_18	IDMAC_TH_17	0
W	w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_TH_15	IDMAC_TH_14	IDMAC_TH_13	IDMAC_TH_12	IDMAC_TH_11	IDMAC_TH_10	IDMAC_TH_9	IDMAC_TH_8	0		IDMAC_TH_5	0	IDMAC_TH_3	IDMAC_TH_2	IDMAC_TH_1	IDMAC_TH_0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c			w1c		w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_13 field descriptions

Field	Description
31 IDMAC_TH_31	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
29 IDMAC_TH_29	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_13 field descriptions (continued)

Field	Description
28 IDMAC_TH_28	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
27 IDMAC_TH_27	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
26 IDMAC_TH_26	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
25 IDMAC_TH_25	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
24 IDMAC_TH_24	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
23 IDMAC_TH_23	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
22 IDMAC_TH_22	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
21 IDMAC_TH_21	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_13 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
20 IDMAC_TH_20	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
19 IDMAC_TH_19	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
18 IDMAC_TH_18	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_TH_17	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
16 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_TH_15	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_TH_14	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_TH_13	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_13 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_TH_12	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_TH_11	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
10 IDMAC_TH_10	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_TH_9	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
8 IDMAC_TH_8	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
5 IDMAC_TH_5	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
4 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

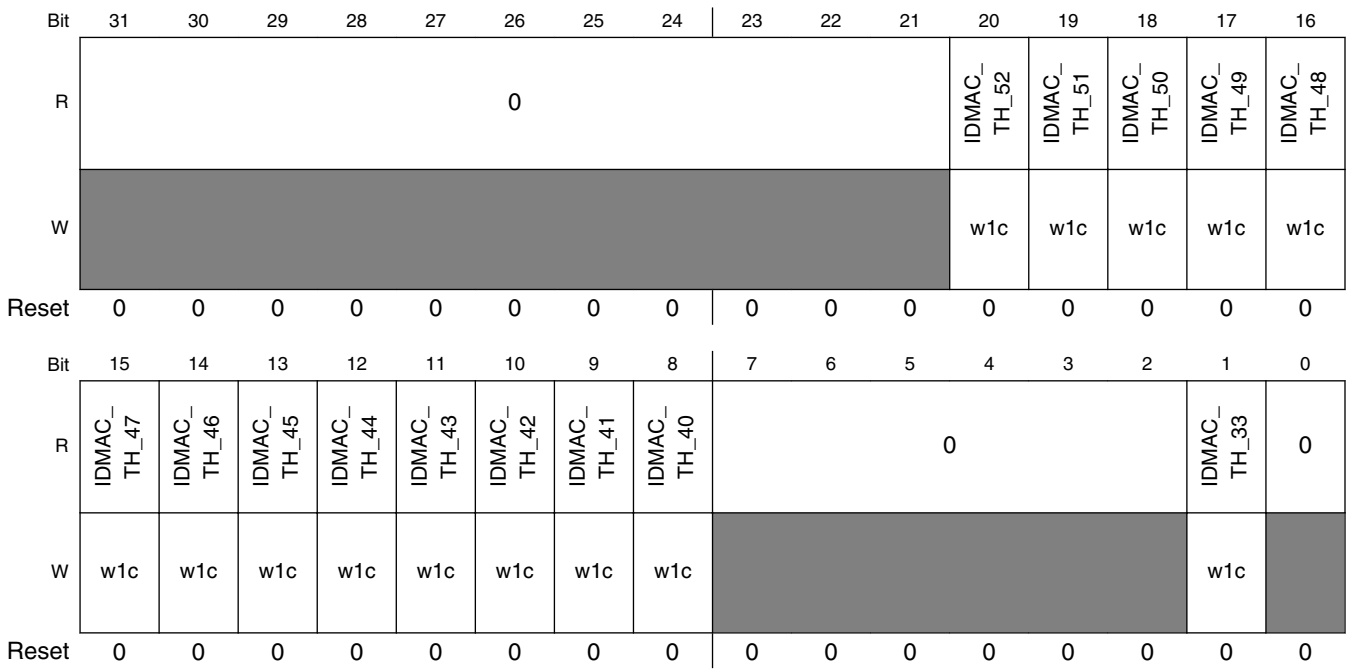
IPU_INT_STAT_13 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
3 IDMAC_TH_3	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
2 IDMAC_TH_2	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_TH_1	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
0 IDMAC_TH_0	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

45.51.65 Interrupt Status Register 14 (IPU_INT_STAT_14)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of the Threshold crossing indication (TH) of DMA Channels interrupts [63:32] can be found in this register.

Address: IPU_INT_STAT_14 is 1E00_0000h base + 234h offset = 1E00_0234h



IPU_INT_STAT_14 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
20 IDMAC_TH_52	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
19 IDMAC_TH_51	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_14 field descriptions (continued)

Field	Description
18 IDMAC_TH_50	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_TH_49	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
16 IDMAC_TH_48	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_TH_47	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_TH_46	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_TH_45	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_TH_44	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_TH_43	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number.

Table continues on the next page...

IPU_INT_STAT_14 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
10 IDMAC_TH_42	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_TH_41	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
8 IDMAC_TH_40	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
7–2 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_TH_33	Threshold crossing indication of Channel interrupt. This bit is the status bit of Threshold crossing interrupt of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.

45.51.66 Interrupt Status Register 15 (IPU_INT_STAT_15)

IPU status registers are not stored in the SRM during power gating mode.IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of general purpose interrupts can be found in this register.

Address: IPU_INT_STAT_15 is 1E00_0000h base + 238h offset = 1E00_0238h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DI1_CNT_EN_PRE_8	DI1_CNT_EN_PRE_3	DI1_DISP_CLK_EN_PRE	DI0_CNT_EN_PRE_10	DI0_CNT_EN_PRE_9	DI0_CNT_EN_PRE_8	DI0_CNT_EN_PRE_7	DI0_CNT_EN_PRE_6	DI0_CNT_EN_PRE_5	DI0_CNT_EN_PRE_4	DI0_CNT_EN_PRE_3	DI0_CNT_EN_PRE_2	DI0_CNT_EN_PRE_1	DI0_CNT_EN_PRE_0	DC_ASYNC_STOP	DC_DP_START
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DI_VSYNC_PRE_1	DI_VSYNC_PRE_0	DC_FC_6	DC_FC_4	DC_FC_3	DC_FC_2	DC_FC_1	DC_FC_0	DP_ASF_BRAKE	DP_SF_BRAKE	DP_ASF_END	DP_ASF_START	DP_SF_END	DP_SF_START	SNOOPING2_INT	SNOOPING1_INT
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_15 field descriptions

Field	Description
31 DI1_CNT_EN_PRE_8	This bit indicates on the interrupt that is a result of a trigger generated by counter #8 of DI1 0 Interrupt is cleared. 1 Interrupt is requested.
30 DI1_CNT_EN_PRE_3	This bit indicates on the interrupt that is a result of a trigger generated by counter #3 of DI1 0 Interrupt is cleared. 1 Interrupt is requested.
29 DI1_DISP_CLK_EN_PRE	**DI1_DISP_CLK_EN_PRE**

Table continues on the next page...

IPU_INT_STAT_15 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
28 DI0_CNT_EN_PRE_10	This bit indicates on the interrupt that is a result of a trigger generated by counter #10 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
27 DI0_CNT_EN_PRE_9	This bit indicates on the interrupt that is a result of a trigger generated by counter #9 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
26 DI0_CNT_EN_PRE_8	This bit indicates on the interrupt that is a result of a trigger generated by counter #8 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
25 DI0_CNT_EN_PRE_7	This bit indicates on the interrupt that is a result of a trigger generated by counter #7 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
24 DI0_CNT_EN_PRE_6	This bit indicates on the interrupt that is a result of a trigger generated by counter #6 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
23 DI0_CNT_EN_PRE_5	This bit indicates on the interrupt that is a result of a trigger generated by counter #5 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
22 DI0_CNT_EN_PRE_4	This bit indicates on the interrupt that is a result of a trigger generated by counter #4 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
21 DI0_CNT_EN_PRE_3	This bit indicates on the interrupt that is a result of a trigger generated by counter #3 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
20 DI0_CNT_EN_PRE_2	This bit indicates on the interrupt that is a result of a trigger generated by counter #2 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
19 DI0_CNT_EN_PRE_1	This bit indicates on the interrupt that is a result of a trigger generated by counter #1 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.
18 DI0_CNT_EN_PRE_0	This bit indicates on the interrupt that is a result of a trigger generated by counter #0 of DI0 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_15 field descriptions (continued)

Field	Description
17 DC_ASYNC_ STOP	This bit indicates on an interrupt asserted anytime the DP stops an async flow and moves to a sync flow 0 Interrupt is cleared. 1 Interrupt is requested.
16 DC_DP_START	This bit indicates on an interrupt asserted anytime the DP start a new sync or async flow or when an async flow is interrupted by a sync flow 0 Interrupt is cleared. 1 Interrupt is requested.
15 DI_VSYNC_ PRE_1	DI1 interrupt indicating of a VSYNC signal asserted 2 rows before the VSYNC sent to the display 0 Interrupt is cleared. 1 Interrupt is requested.
14 DI_VSYNC_ PRE_0	DI0 interrupt indicating of a VSYNC signal asserted 2 rows before the VSYNC sent to the display 0 Interrupt is cleared. 1 Interrupt is requested.
13 DC_FC_6	DC Frame Complete on channel #6 interrupt indication 0 Interrupt is cleared. 1 Interrupt is requested.
12 DC_FC_4	DC Frame Complete on channel #4 interrupt indication 0 Interrupt is cleared. 1 Interrupt is requested.
11 DC_FC_3	DC Frame Complete on channel #3 interrupt indication 0 Interrupt is cleared. 1 Interrupt is requested.
10 DC_FC_2	DC Frame Complete on channel #2 interrupt indication 0 Interrupt is cleared. 1 Interrupt is requested.
9 DC_FC_1	DC Frame Complete on channel #1 interrupt indication 0 Interrupt is cleared. 1 Interrupt is requested.
8 DC_FC_0	DC Frame Complete on channel #0 interrupt indication 0 Interrupt is cleared. 1 Interrupt is requested.
7 DP_ASF_ BRAKE	DP Async Flow Brake indication interrupt. This bit indicates on the interrupt that is a result of the async flow brake at the DP 0 Interrupt is cleared. 1 Interrupt is requested.

Table continues on the next page...

IPU_INT_STAT_15 field descriptions (continued)

Field	Description
6 DP_SF_BRAKE	DP Sync Flow Brake indication interrupt. This bit indicates on the interrupt that is a result of the Sync flow brake at the DP 0 Interrupt is cleared. 1 Interrupt is requested.
5 DP_ASF_END	DP Async Flow End indication interrupt. This bit indicates on the interrupt that is a result of the Async flow end at the DP 0 Interrupt is cleared. 1 Interrupt is requested.
4 DP_ASF_START	DP Async Flow Start indication interrupt. This bit indicates on the interrupt that is a result of the Async flow start at the DP 0 Interrupt is cleared. 1 Interrupt is requested.
3 DP_SF_END	DP Sync Flow End indication interrupt. This bit indicates on the interrupt that is a result of the Sync flow end at the DP 0 Interrupt is cleared. 1 Interrupt is requested.
2 DP_SF_START	DP Sync Flow Start indication interrupt. This bit indicates on the interrupt that is a result of the Sync flow start at the DP 0 Interrupt is cleared. 1 Interrupt is requested.
1 SNOOPING2_INT	IPU snooping 2 event indication interrupt. This bit indicates on the interrupt that is a result of the detection of a snooping 2 signal assertion coming to the IPU 0 Interrupt is cleared. 1 Interrupt is requested.
0 SNOOPING1_INT	IPU snooping 1 event indication interrupt. This bit indicates on the interrupt that is a result of the detection of a snooping 1 signal assertion coming to the 0 Interrupt is cleared. 1 Interrupt is requested.

45.51.67 Current Buffer Register 0 (IPU_CUR_BUF_0)

This register contains the current buffer status information bit for each DMA channel.

Address: IPU_CUR_BUF_0 is 1E00_0000h base + 23Ch offset = 1E00_023Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DMA_CH_CUR_BUF_31	0	DMA_CH_CUR_BUF_29	DMA_CH_CUR_BUF_28	DMA_CH_CUR_BUF_27	DMA_CH_CUR_BUF_26	DMA_CH_CUR_BUF_25	DMA_CH_CUR_BUF_24	DMA_CH_CUR_BUF_23	DMA_CH_CUR_BUF_22	DMA_CH_CUR_BUF_21	DMA_CH_CUR_BUF_20	DMA_CH_CUR_BUF_19	DMA_CH_CUR_BUF_18	DMA_CH_CUR_BUF_17	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DMA_CH_CUR_BUF_15	DMA_CH_CUR_BUF_14	DMA_CH_CUR_BUF_13	DMA_CH_CUR_BUF_12	DMA_CH_CUR_BUF_11	DMA_CH_CUR_BUF_10	DMA_CH_CUR_BUF_9	DMA_CH_CUR_BUF_8	0		DMA_CH_CUR_BUF_5	0	DMA_CH_CUR_BUF_3	DMA_CH_CUR_BUF_2	DMA_CH_CUR_BUF_1	DMA_CH_CUR_BUF_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CUR_BUF_0 field descriptions

Field	Description
31 DMA_CH_CUR_BUF_31	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
29 DMA_CH_CUR_BUF_29	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
28 DMA_CH_CUR_BUF_28	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

Table continues on the next page...

IPU_CUR_BUF_0 field descriptions (continued)

Field	Description
27 DMA_CH_CUR_BUF_27	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
26 DMA_CH_CUR_BUF_26	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
25 DMA_CH_CUR_BUF_25	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
24 DMA_CH_CUR_BUF_24	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
23 DMA_CH_CUR_BUF_23	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
22 DMA_CH_CUR_BUF_22	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
21 DMA_CH_CUR_BUF_21	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
20 DMA_CH_CUR_BUF_20	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
19 DMA_CH_CUR_BUF_19	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

Table continues on the next page...

IPU_CUR_BUF_0 field descriptions (continued)

Field	Description
18 DMA_CH_CUR_BUF_18	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
17 DMA_CH_CUR_BUF_17	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
16 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
15 DMA_CH_CUR_BUF_15	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
14 DMA_CH_CUR_BUF_14	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
13 DMA_CH_CUR_BUF_13	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
12 DMA_CH_CUR_BUF_12	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
11 DMA_CH_CUR_BUF_11	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
10 DMA_CH_CUR_BUF_10	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

Table continues on the next page...

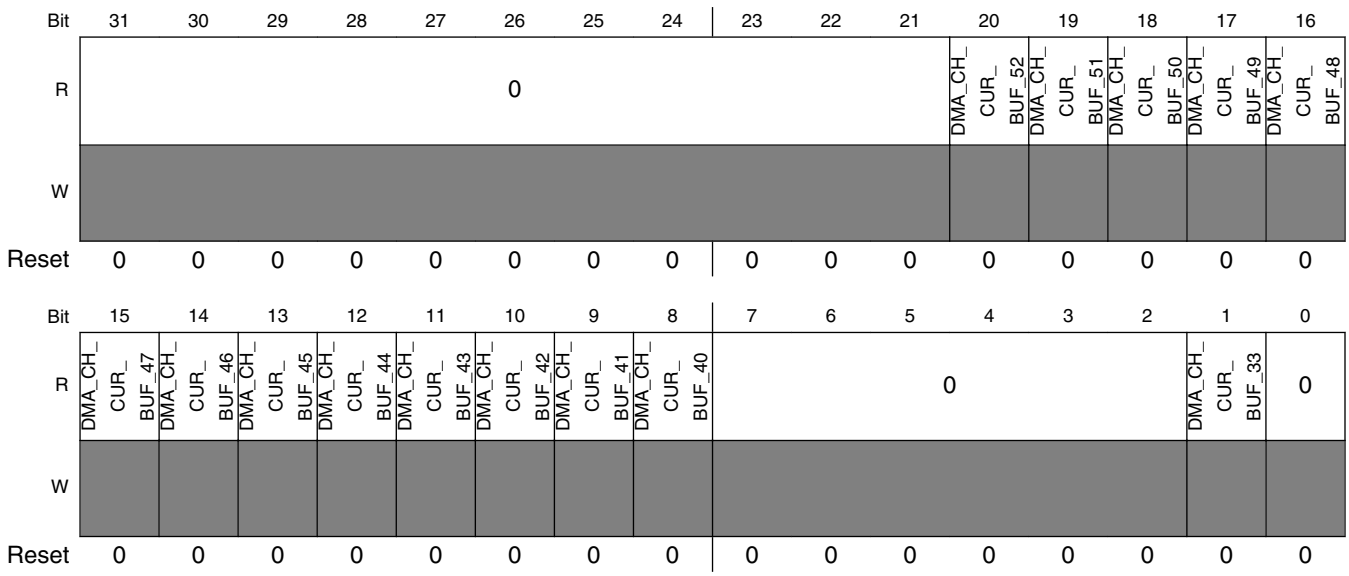
IPU_CUR_BUF_0 field descriptions (continued)

Field	Description
9 DMA_CH_CUR_BUF_9	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
8 DMA_CH_CUR_BUF_8	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
5 DMA_CH_CUR_BUF_5	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
4 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
3 DMA_CH_CUR_BUF_3	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
2 DMA_CH_CUR_BUF_2	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
1 DMA_CH_CUR_BUF_1	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
0 DMA_CH_CUR_BUF_0	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

45.51.68 Current Buffer Register 1 (IPU_CUR_BUF_1)

This register contains the current buffer status information bit for each DMA channel.

Address: IPU_CUR_BUF_1 is 1E00_0000h base + 240h offset = 1E00_0240h



IPU_CUR_BUF_1 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
20 DMA_CH_CUR_BUF_52	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
19 DMA_CH_CUR_BUF_51	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
18 DMA_CH_CUR_BUF_50	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

Table continues on the next page...

IPU_CUR_BUF_1 field descriptions (continued)

Field	Description
17 DMA_CH_CUR_BUF_49	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
16 DMA_CH_CUR_BUF_48	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
15 DMA_CH_CUR_BUF_47	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
14 DMA_CH_CUR_BUF_46	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
13 DMA_CH_CUR_BUF_45	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
12 DMA_CH_CUR_BUF_44	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
11 DMA_CH_CUR_BUF_43	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
10 DMA_CH_CUR_BUF_42	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
9 DMA_CH_CUR_BUF_41	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

Table continues on the next page...

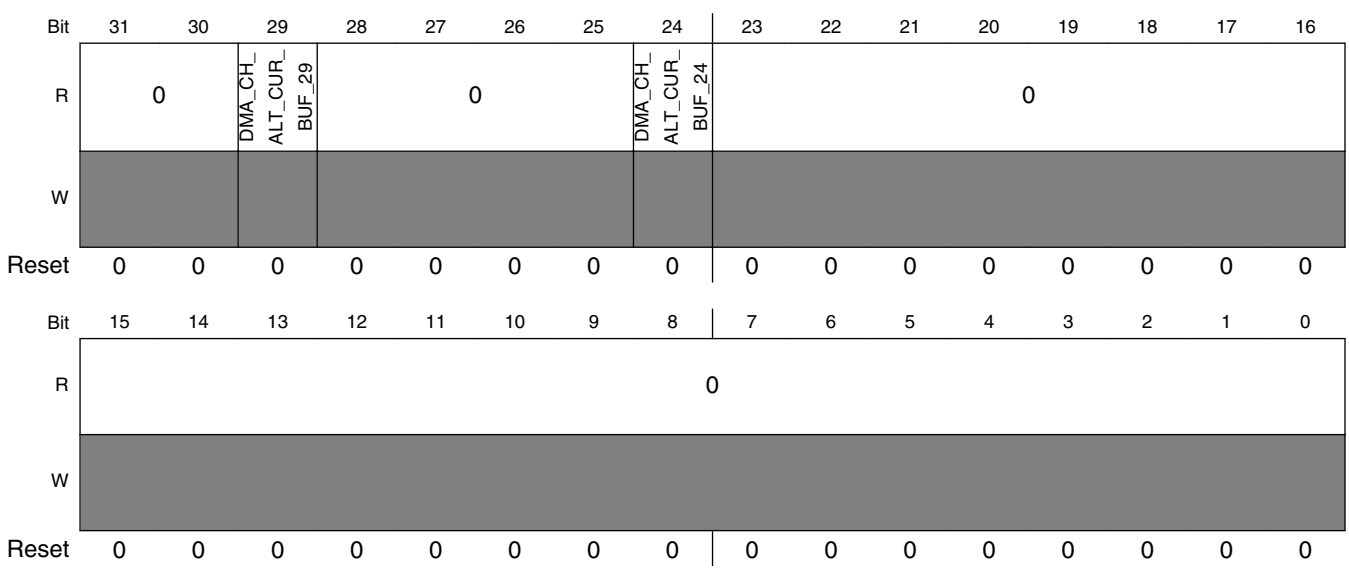
IPU_CUR_BUF_1 field descriptions (continued)

Field	Description
8 DMA_CH_CUR_BUF_40	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
7–2 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
1 DMA_CH_CUR_BUF_33	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

45.51.69 Alternate Current Buffer Register 0 (IPU_ALT_CUR_0)

This register contains the current buffer status information bit for each DMA channel.

Address: IPU_ALT_CUR_0 is 1E00_0000h base + 244h offset = 1E00_0244h



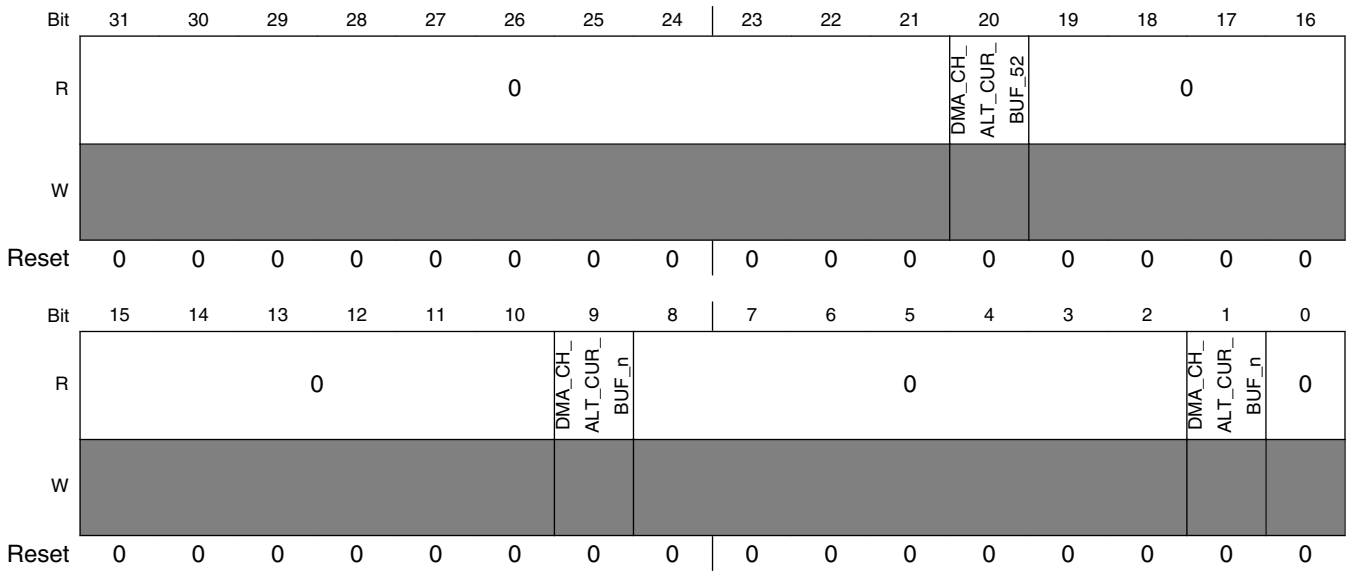
IPU_ALT_CUR_0 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
29 DMA_CH_ALT_ CUR_BUF_29	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
28–25 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
24 DMA_CH_ALT_ CUR_BUF_24	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
23–0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

45.51.70 Alternate Current Buffer Register 1 (IPU_ALT_CUR_1)

This register contains the current buffer status information bit for each DMA channel. The register is shown in VDI Plane Size Register 4 Alternate Current Buffer Register 1, and the register fields are described in VDI Plane Size Register 4 Alternate Current Buffer Register 1.

Address: IPU_ALT_CUR_1 is 1E00_0000h base + 248h offset = 1E00_0248h



IPU_ALT_CUR_1 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
20 DMA_CH_ALT_ CUR_BUF_52	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
19–10 Reserved	This read-only field is reserved and always has the value zero. Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

Table continues on the next page...

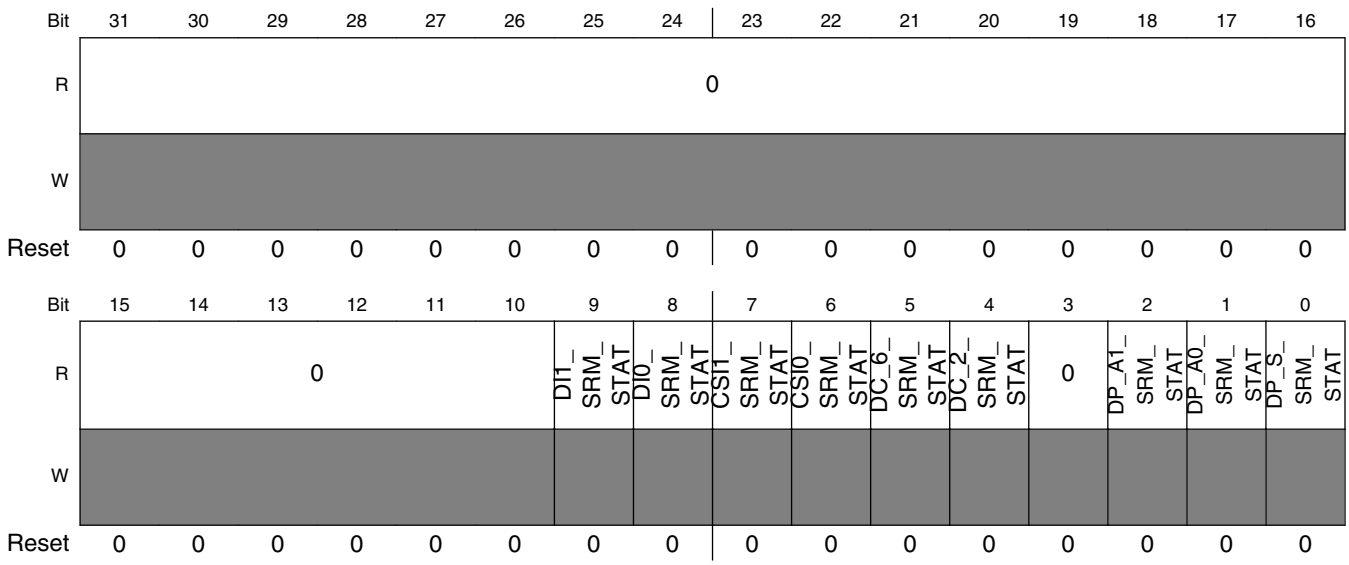
IPU_ALT_CUR_1 field descriptions (continued)

Field	Description
9 DMA_CH_ALT_ CUR_BUF_n	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
8–2 Reserved	This read-only field is reserved and always has the value zero. Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
1 DMA_CH_ALT_ CUR_BUF_n	Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.
0 Reserved	This read-only field is reserved and always has the value zero. Current Buffer. This bit indicates which buffer is in use by DMA when double buffer mode is selected. *n* Indicates the corresponding DMA channel number. 0 Current buffer used by DMA is buffer 0. 1 Current buffer used by DMA is buffer 1.

45.51.71 Shadow Registers Memory Status Register (IPU_SRM_STAT)

The register contains status bits of SRM updates. There is a bit for each block. The bit is set when the SRM is currently updating the module's registers. When the SRM completes updating the registers of the block the bit is cleared. SW should not update the block's registers while it is being updated by the SRM.

Address: IPU_SRM_STAT is 1E00_0000h base + 24Ch offset = 1E00_024Ch



IPU_SRM_STAT field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 DI1_SRM_STAT	DI1 SRM STAT This bit indicates that the SRM is currently updating the DI1 registers 1 SRM is busy updating the DI1 registers 0 SRM is not updating the DI1 registers
8 DI0_SRM_STAT	DI0 SRM STAT This bit indicates that the SRM is currently updating the DI0 registers 1 SRM is busy updating the DI0 registers 0 SRM is not updating the DI0 registers
7 CSI1_SRM_STAT	CSI1_SRM_STAT

Table continues on the next page...

IPU_SRM_STAT field descriptions (continued)

Field	Description
6 CSI0_SRM_ STAT	CSI1_SRM_STAT
5 DC_6_SRM_ STAT	DC group #6 SRM STAT This bit indicates that the SRM is currently updating the DC group #6 registers 1 SRM is busy updating the DC registers 0 SRM is not updating the DC registers
4 DC_2_SRM_ STAT	DC group #2 SRM STAT This bit indicates that the SRM is currently updating the DC group #2 registers 1 SRM is busy updating the DC group #6 registers 0 SRM is not updating the DC group #2 registers
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2 DP_A1_SRM_ STAT	DP ASYNC1 FLOW SRM STAT This bit indicates that the SRM is currently updating the DP async flow 1 registers 1 SRM is busy updating the DP sync flow registers 0 SRM is not updating the DP sync flow registers
1 DP_A0_SRM_ STAT	DP ASYNC0 FLOW SRM STAT This bit indicates that the SRM is currently updating the DP async flow 0 registers 1 SRM is busy updating the DP async flow 0 registers 0 SRM is not updating the DP async flow 0 registers
0 DP_S_SRM_ STAT	DP SYNC FLOW SRM STAT This bit indicates that the SRM is currently updating the DP sync flow registers 1 SRM is busy updating the DP sync flow registers 0 SRM is not updating the DP sync flow registers

45.51.72 Processing Status Tasks Register (IPU_PROC_TASKS_STAT)

This register contains status bits for IPU's tasks.

Address: IPU_PROC_TASKS_STAT is 1E00_0000h base + 250h offset = 1E00_0250h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MEM2PRP_TSTAT			PP_ROT_TSTAT		VF_ROT_TSTAT		ENC_ROT_TSTAT		PP_TSTAT		VF_TSTAT		ENC_TSTAT	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_PROC_TASKS_STAT field descriptions

Field	Description
31–15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 MEM2PRP_TSTAT	Status of the pre processing tasks (viewfinder and encoding) when the source is coming from the memory. 000 IDLE - Both pre processing tasks are idle 001 BOTH_ACTIVE - Both pre processing tasks are idle 010 ENC_ACTIVE - Encoding task is active 011 VF_ACTIVE - View finder task is active 100 BOTH_PAUSE - both tasks are paused 101 Reserved 110 Reserved 111 Reserve
11–10 PP_ROT_TSTAT	Status of the rotation for post processing task 00 IDLE - The task is idle 01 ACTIVE - The primary flow of this task is currently active 10 WAIT_FOR_ READY - The task is waiting for a buffer to be ready
9–8 VF_ROT_TSTAT	Status of the rotation for viewfinder task 00 IDLE - The task is idle 01 ACTIVE - The primary flow of this task is currently active 10 WAIT_FOR_ READY - The task is waiting for a buffer to be ready
7–6 ENC_ROT_TSTAT	Status of the rotation for encoding task 00 IDLE - The task is idle

Table continues on the next page...

IPU_PROC_TASKS_STAT field descriptions (continued)

Field	Description
	01 ACTIVE - The primary flow of this task is currently active 10 WAIT_FOR_ READY - The task is waiting for a buffer to be ready
5–4 PP_TSTAT	Status of the post processing task 00 IDLE - The task is idle 01 ACTIVE - The primary flow of this task is currently active 10 WAIT_FOR_ READY - The task is waiting for a buffer to be ready
3–2 VF_TSTAT	Status of the viewfinder task 00 IDLE - The task is idle 01 ACTIVE - The primary flow of this task is currently active 10 WAIT_FOR_ READY - The task is waiting for a buffer to be ready
1–0 ENC_TSTAT	Status of the encoding task 00 IDLE - The task is idle 01 ACTIVE - The primary flow of this task is currently active 10 WAIT_FOR_ READY - The task is waiting for a buffer to be ready

45.51.73 Display Tasks Status Register (IPU_DISP_TASKS_STAT)

This register contains status bits for IPU's tasks.

Address: IPU_DISP_TASKS_STAT is 1E00_0000h base + 254h offset = 1E00_0254h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DC_ ASYNC2_ CUR_FLOW	DC_ASYNCH2_ STAT			0		DC_ ASYNC1_ STAT		DP_ASYNC_ CUR_FLOW	DP_ASYNC_STAT		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DISP_TASKS_STAT field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11 DC_ASYNC2_ CUR_FLOW	Current asynchronous #2 flow via the DC 1 alternate flow 0 main flow
10–8 DC_ASYNCH2_ STAT	Status of the Asynchronous flow #2 through the DC 000 IDLE - the task is idle 001 PRIM_ACTIVE - The primary flow of this task is currently active 010 ALT_ACTIVE - The alternate flow of this task is currently active 011 UPDATE_PARAM - The FSU is busy updating parameters from the SRM 100 PAUSE - The task is paused 101 Reserved 110 Reserved 111 Reserved
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5–4 DC_ASYNC1_ STAT	Status of the Asynchronous flow #1 through the DC (ch 28) 00 IDLE - The task is idle 01 ACTIVE - This task is currently active 10 WAIT_FOR_ READY - The task is waiting for a buffer to be ready
3 DP_ASYNC_ CUR_FLOW	Current asynchronous flow via the DP 1 alternate flow 0 main flow
2–0 DP_ASYNC_ STAT	Status of the Asynchronous flow through the DP 000 IDLE - the task is idle 001 PRIM_ACTIVE - The primary flow of this task is currently active 010 ALT_ACTIVE - The alternate flow of this task is currently active 011 UPDATE_PARAM - The FSU is busy updating parameters from the SRM 100 PAUSE - The task is paused 101 Reserved 110 Reserved 111 Reserved

45.51.74 Triple Current Buffer Register 0 (IPU_TRIPLE_CUR_BUF_0)

This register contains the current buffer status information for triple buffer mode for each DMA channel.

- Hide VPU_SUB_FRAME_SYNC for all versions
- Show VPU_SUB_FRAME_SYNC for IPUv3H version.

- The table below tagged with other settings (like IPU3M_only) should be hidden in IPUv3H version.
- This requires some sophisticated conditional tag settings

Address: IPU_TRIPLE_CUR_BUF_0 is 1E00_0000h base + 258h offset = 1E00_0258h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				DMA_CH_TRIPLE_CUR_BUF_13		0				DMA_CH_TRIPLE_CUR_BUF_10		DMA_CH_TRIPLE_CUR_BUF_9		DMA_CH_TRIPLE_CUR_BUF_8	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_TRIPLE_CUR_BUF_0 field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved.
27–26 DMA_CH_TRIPLE_CUR_BUF_13	Current Buffer for triple buffer mode. This bits indicate which buffer is in use by the DMA when triple buffer mode is selected. Each pair of bits indicate the corresponding DMA channel (bits [1:0] correspond to ch #0; (bits [3:2] correspond to ch #1, etc.) 11 NA 00 Current buffer used by DMA is buffer 0. 01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.
25–22 Reserved	This read-only field is reserved and always has the value zero. Reserved.
21–20 DMA_CH_TRIPLE_CUR_BUF_10	Current Buffer for triple buffer mode. This bits indicate which buffer is in use by the DMA when triple buffer mode is selected. Each pair of bits indicate the corresponding DMA channel (bits [1:0] correspond to ch #0; (bits [3:2] correspond to ch #1, etc.) 11 NA 00 Current buffer used by DMA is buffer 0. 01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.
19–18 DMA_CH_TRIPLE_CUR_BUF_9	Current Buffer for triple buffer mode. This bits indicate which buffer is in use by the DMA when triple buffer mode is selected. Each pair of bits indicate the corresponding DMA channel (bits [1:0] correspond to ch #0; (bits [3:2] correspond to ch #1, etc.) 11 NA

Table continues on the next page...

IPU_TRIPLE_CUR_BUF_0 field descriptions (continued)

Field	Description
	00 Current buffer used by DMA is buffer 0. 01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.
17–16 DMA_CH_ TRIPLE_CUR_ BUF_8	Current Buffer for triple buffer mode. This bits indicate which buffer is in use by the DMA when triple buffer mode is selected. Each pair of bits indicate the corresponding DMA channel (bits [1:0] correspond to ch #0; (bits [3:2] correspond to ch #1, etc.) 11 NA 00 Current buffer used by DMA is buffer 0. 01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.
15–0 Reserved	This read-only field is reserved and always has the value zero. Reserved.

45.51.75 Triple Current Buffer Register 1 (IPU_TRIPLE_CUR_BUF_1)

This register contains the current buffer status information for triple buffer mode for each DMA channel.

Address: IPU_TRIPLE_CUR_BUF_1 is 1E00_0000h base + 25Ch offset = 1E00_025Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						DMA_CH_TRIPLE_CUR_BUF_28		DMA_CH_TRIPLE_CUR_BUF_27		0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DMA_CH_TRIPLE_CUR_BUF_23		0		DMA_CH_TRIPLE_CUR_BUF_21		0									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_TRIPLE_CUR_BUF_1 field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_TRIPLE_CUR_BUF_1 field descriptions (continued)

Field	Description
25–24 DMA_CH_ TRIPLE_CUR_ BUF_28	Current Buffer for triple buffer mode. This bits indicate which buffer is in use by the DMA when triple buffer mode is selected. Each pair of bits indicate the corresponding DMA channel (bits [1:0] correspond to ch #0; (bits [3:2] correspond to ch #1, etc.) 11 NA 00 Current buffer used by DMA is buffer 0. 01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.
23–22 DMA_CH_ TRIPLE_CUR_ BUF_27	Current Buffer for triple buffer mode. This bits indicate which buffer is in use by the DMA when triple buffer mode is selected. Each pair of bits indicate the corresponding DMA channel (bits [1:0] correspond to ch #0; (bits [3:2] correspond to ch #1, etc.) 11 NA 00 Current buffer used by DMA is buffer 0. 01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.
21–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–14 DMA_CH_ TRIPLE_CUR_ BUF_23	Current Buffer for triple buffer mode. This bits indicate which buffer is in use by the DMA when triple buffer mode is selected. Each pair of bits indicate the corresponding DMA channel (bits [1:0] correspond to ch #0; (bits [3:2] correspond to ch #1, etc.) 11 NA 00 Current buffer used by DMA is buffer 0. 01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.
13–12 Reserved	This read-only field is reserved and always has the value zero. Reserved.
11–10 DMA_CH_ TRIPLE_CUR_ BUF_21	Current Buffer for triple buffer mode. This bits indicate which buffer is in use by the DMA when triple buffer mode is selected. Each pair of bits indicate the corresponding DMA channel (bits [1:0] correspond to ch #0; (bits [3:2] correspond to ch #1, etc.) 11 NA 00 Current buffer used by DMA is buffer 0. 01 Current buffer used by DMA is buffer 1. 10 Current buffer used by DMA is buffer 2.
9–0 Reserved	This read-only field is reserved and always has the value zero. Reserved.

45.51.76 IPU Channels Buffer 0 Ready 0 Register (IPU_CH_BUF0_RDY0)

The register contains buffer 0 ready control information for 32 IPU's DMA channels (31-0). This register can be a write one to set or a write one to clear according to the **IPU_CH_BUF0_RDY0_CLR bit**.

The register is shown in IPU Channels Buffer 0 Ready 0 Register (IPU_CH_BUF0_RDY0), and the register fields are described in IPU Channels Buffer 0 Ready 0 Register (IPU_CH_BUF0_RDY0).

Address: IPU_CH_BUF0_RDY0 is 1E00_0000h base + 268h offset = 1E00_0268h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DMA_CH_BUF0_RDY_31	-	DMA_CH_BUF0_RDY_29	DMA_CH_BUF0_RDY_28	DMA_CH_BUF0_RDY_27	-		DMA_CH_BUF0_RDY_24	DMA_CH_BUF0_RDY_23	DMA_CH_BUF0_RDY_22	DMA_CH_BUF0_RDY_21	DMA_CH_BUF0_RDY_20	-	DMA_CH_BUF0_RDY_18	DMA_CH_BUF0_RDY_17	-
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DMA_CH_BUF0_RDY_15	DMA_CH_BUF0_RDY_14	DMA_CH_BUF0_RDY_13	DMA_CH_BUF0_RDY_12	DMA_CH_BUF0_RDY_11	DMA_CH_BUF0_RDY_10	DMA_CH_BUF0_RDY_9	DMA_CH_BUF0_RDY_8	DMA_CH_BUF0_RDY_7	DMA_CH_BUF0_RDY_6	DMA_CH_BUF0_RDY_5	DMA_CH_BUF0_RDY_4	DMA_CH_BUF0_RDY_3	DMA_CH_BUF0_RDY_2	DMA_CH_BUF0_RDY_1	DMA_CH_BUF0_RDY_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CH_BUF0_RDY0 field descriptions

Field	Description
31 DMA_CH_ BUF0_RDY_31	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
30 -	Reserved.
29 DMA_CH_ BUF0_RDY_29	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.

Table continues on the next page...

IPU_CH_BUF0_RDY0 field descriptions (continued)

Field	Description
28 DMA_CH_ BUF0_RDY_28	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
27 DMA_CH_ BUF0_RDY_27	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
26–25 -	Reserved.
24 DMA_CH_ BUF0_RDY_24	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
23 DMA_CH_ BUF0_RDY_23	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
22 DMA_CH_ BUF0_RDY_22	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
21 DMA_CH_ BUF0_RDY_21	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
20 DMA_CH_ BUF0_RDY_20	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
19 -	Reserved.
18 DMA_CH_ BUF0_RDY_18	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
17 DMA_CH_ BUF0_RDY_17	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
16 -	Reserved.
15 DMA_CH_ BUF0_RDY_15	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.

Table continues on the next page...

IPU_CH_BUF0_RDY0 field descriptions (continued)

Field	Description
14 DMA_CH_ BUF0_RDY_14	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
13 DMA_CH_ BUF0_RDY_13	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
12 DMA_CH_ BUF0_RDY_12	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
11 DMA_CH_ BUF0_RDY_11	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
10 DMA_CH_ BUF0_RDY_10	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
9 DMA_CH_ BUF0_RDY_9	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
8 DMA_CH_ BUF0_RDY_8	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
7 DMA_CH_ BUF0_RDY_7	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
6 DMA_CH_ BUF0_RDY_6	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
5 DMA_CH_ BUF0_RDY_5	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
4 DMA_CH_ BUF0_RDY_4	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
3 DMA_CH_ BUF0_RDY_3	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory.

Table continues on the next page...

IPU_CH_BUF0_RDY0 field descriptions (continued)

Field	Description
	0 Buffer 0 is not ready. 1 Buffer 0 is ready.
2 DMA_CH_ BUF0_RDY_2	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
1 DMA_CH_ BUF0_RDY_1	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
0 DMA_CH_ BUF0_RDY_0	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.

45.51.77 IPU Channels Buffer 0 Ready 1 Register (IPU_CH_BUF0_RDY1)

The register contains buffer 0 ready control information for 32 IPU's DMA channels (63-32). This register can be a write one to set or a write one to clear according to the IPU_CH_BUF0_RDY1_CLR bit.

Address: IPU_CH_BUF0_RDY1 is 1E00_0000h base + 26Ch offset = 1E00_026Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-											DMA_CH_BUF0_RDY_52	DMA_CH_BUF0_RDY_51	DMA_CH_BUF0_RDY_50	DMA_CH_BUF0_RDY_49	DMA_CH_BUF0_RDY_48
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DMA_CH_BUF0_RDY_47	DMA_CH_BUF0_RDY_46	DMA_CH_BUF0_RDY_45	DMA_CH_BUF0_RDY_44	DMA_CH_BUF0_RDY_43	DMA_CH_BUF0_RDY_42	DMA_CH_BUF0_RDY_41	DMA_CH_BUF0_RDY_40	-						DMA_CH_BUF0_RDY_33	-
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CH_BUF0_RDY1 field descriptions

Field	Description
31–21 -	Reserved.
20 DMA_CH_ BUF0_RDY_52	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
19 DMA_CH_ BUF0_RDY_51	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
18 DMA_CH_ BUF0_RDY_50	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
17 DMA_CH_ BUF0_RDY_49	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
16 DMA_CH_ BUF0_RDY_48	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
15 DMA_CH_ BUF0_RDY_47	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
14 DMA_CH_ BUF0_RDY_46	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
13 DMA_CH_ BUF0_RDY_45	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
12 DMA_CH_ BUF0_RDY_44	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
11 DMA_CH_ BUF0_RDY_43	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
10 DMA_CH_ BUF0_RDY_42	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.

Table continues on the next page...

IPU_CH_BUF0_RDY1 field descriptions (continued)

Field	Description
9 DMA_CH_ BUF0_RDY_41	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
8 DMA_CH_ BUF0_RDY_40	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
7–2 -	Reserved.
1 DMA_CH_ BUF0_RDY_33	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
0 -	Reserved.

45.51.78 IPU Channels Buffer 1 Ready 0 Register (IPU_CH_BUF1_RDY0)

The register contains buffer 1 ready control information for 32 IPU's DMA channels (31-0). This register can be a write one to set or a write one to clear according to the IPU_CH_BUF1_RDY0_CLR bit.

Address: IPU_CH_BUF1_RDY0 is 1E00_0000h base + 270h offset = 1E00_0270h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_CH_BUF1_RDY_31	-	DMA_CH_BUF1_RDY_29	DMA_CH_BUF1_RDY_28	DMA_CH_BUF1_RDY_27	DMA_CH_BUF1_RDY_26	DMA_CH_BUF1_RDY_25	DMA_CH_BUF1_RDY_24	DMA_CH_BUF1_RDY_23	DMA_CH_BUF1_RDY_22	DMA_CH_BUF1_RDY_21	DMA_CH_BUF1_RDY_20	DMA_CH_BUF1_RDY_19	DMA_CH_BUF1_RDY_18	DMA_CH_BUF1_RDY_17	-
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_CH_BUF1_RDY_15	DMA_CH_BUF1_RDY_14	DMA_CH_BUF1_RDY_13	DMA_CH_BUF1_RDY_12	DMA_CH_BUF1_RDY_	DMA_CH_BUF1_RDY_10	DMA_CH_BUF1_RDY_9	DMA_CH_BUF1_RDY_8	-		DMA_CH_BUF1_RDY_5	-	DMA_CH_BUF1_RDY_3	DMA_CH_BUF1_RDY_2	DMA_CH_BUF1_RDY_1	DMA_CH_BUF1_RDY_0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CH_BUF1_RDY0 field descriptions

Field	Description
31 DMA_CH_ BUF1_RDY_31	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
30 -	Reserved.
29 DMA_CH_ BUF1_RDY_29	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
28 DMA_CH_ BUF1_RDY_28	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
27 DMA_CH_ BUF1_RDY_27	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
26 DMA_CH_ BUF1_RDY_26	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
25 DMA_CH_ BUF1_RDY_25	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
24 DMA_CH_ BUF1_RDY_24	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
23 DMA_CH_ BUF1_RDY_23	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
22 DMA_CH_ BUF1_RDY_22	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
21 DMA_CH_ BUF1_RDY_21	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
20 DMA_CH_ BUF1_RDY_20	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.

Table continues on the next page...

IPU_CH_BUF1_RDY0 field descriptions (continued)

Field	Description
19 DMA_CH_ BUF1_RDY_19	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
18 DMA_CH_ BUF1_RDY_18	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
17 DMA_CH_ BUF1_RDY_17	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
16 -	Reserved.
15 DMA_CH_ BUF1_RDY_15	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
14 DMA_CH_ BUF1_RDY_14	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
13 DMA_CH_ BUF1_RDY_13	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
12 DMA_CH_ BUF1_RDY_12	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
11 DMA_CH_ BUF1_RDY_	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
10 DMA_CH_ BUF1_RDY_10	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
9 DMA_CH_ BUF1_RDY_9	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
8 DMA_CH_ BUF1_RDY_8	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.

Table continues on the next page...

IPU_CH_BUF1_RDY0 field descriptions (continued)

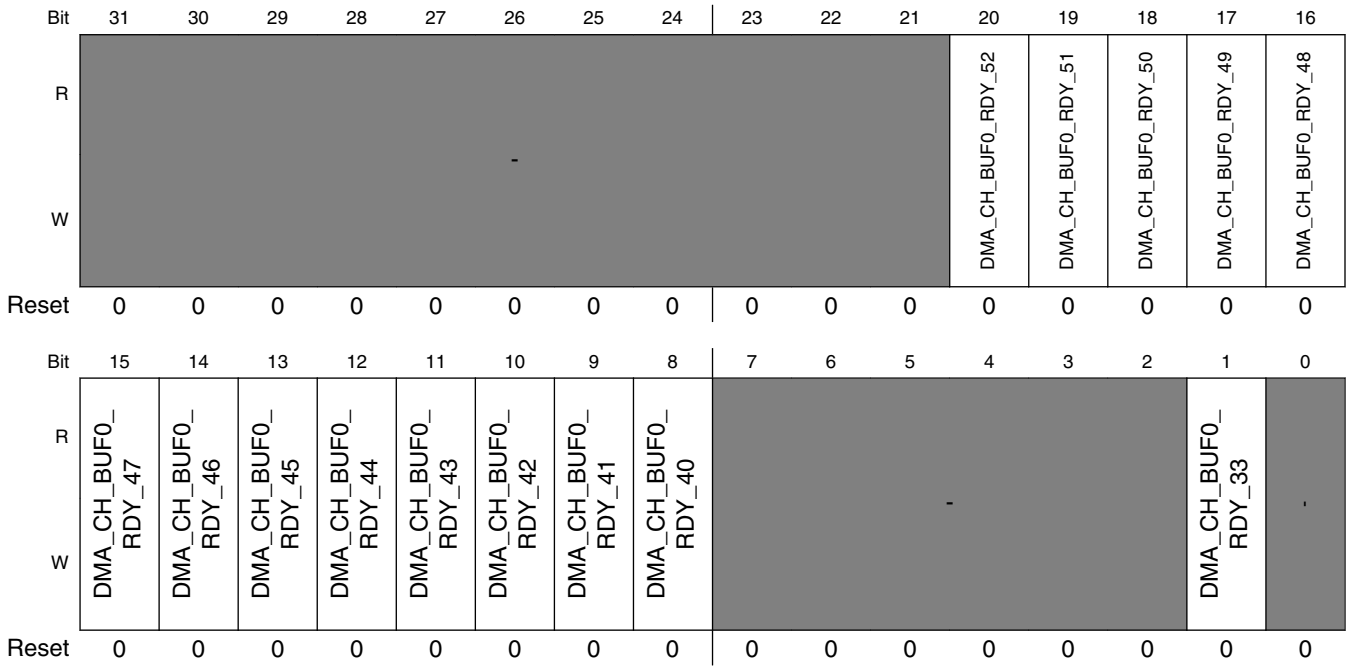
Field	Description
7–6 -	Reserved.
5 DMA_CH_ BUF1_RDY_5	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
4 -	Reserved.
3 DMA_CH_ BUF1_RDY_3	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
2 DMA_CH_ BUF1_RDY_2	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
1 DMA_CH_ BUF1_RDY_1	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
0 DMA_CH_ BUF1_RDY_0	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.

45.51.79 IPU Channels Buffer 1 Ready 1Register (IPU_CH_BUF1_RDY1)

The register contains buffer 0 ready control information for 32 IPU's DMA channels (63-32). This register can be a write one to set or a write one to clear according to the **IPU_CH_BUF1_RDY1_CLR bit**.

The register is shown in IPU Channels Buffer 1 Ready 1Register (IPU_CH_BUF1_RDY1), and the register fields are described in IPU Channels Buffer 1 Ready 1Register (IPU_CH_BUF1_RDY1).

Address: IPU_CH_BUF1_RDY1 is 1E00_0000h base + 274h offset = 1E00_0274h



IPU_CH_BUF1_RDY1 field descriptions

Field	Description
31–21 -	Reserved.
20 DMA_CH_ BUF0_RDY_52	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
19 DMA_CH_ BUF0_RDY_51	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
18 DMA_CH_ BUF0_RDY_50	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
17 DMA_CH_ BUF0_RDY_49	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
16 DMA_CH_ BUF0_RDY_48	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
15 DMA_CH_ BUF0_RDY_47	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.

Table continues on the next page...

IPU_CH_BUF1_RDY1 field descriptions (continued)

Field	Description
14 DMA_CH_ BUF0_RDY_46	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
13 DMA_CH_ BUF0_RDY_45	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
12 DMA_CH_ BUF0_RDY_44	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
11 DMA_CH_ BUF0_RDY_43	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
10 DMA_CH_ BUF0_RDY_42	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
9 DMA_CH_ BUF0_RDY_41	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
8 DMA_CH_ BUF0_RDY_40	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
7–2 -	Reserved.
1 DMA_CH_ BUF0_RDY_33	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 Buffer 1 is ready.
0 -	Reserved.

45.51.80 IPU Alternate Channels Buffer 0 Ready 0 Register (IPU_ALT_CH_BUF0_RDY0)

The register contains buffer 0 ready control information for 32 IPU's DMA channels (31-0). Writing "1" to each field will set each bit. Writing "0" to each field simultaneously will clear all the bits.

Address: IPU_ALT_CH_BUF0_RDY0 is 1E00_0000h base + 278h offset = 1E00_0278h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-		DMA_CH_ALT_BUF0_RDY_29	-				DMA_CH_ALT_BUF0_RDY_24	-							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

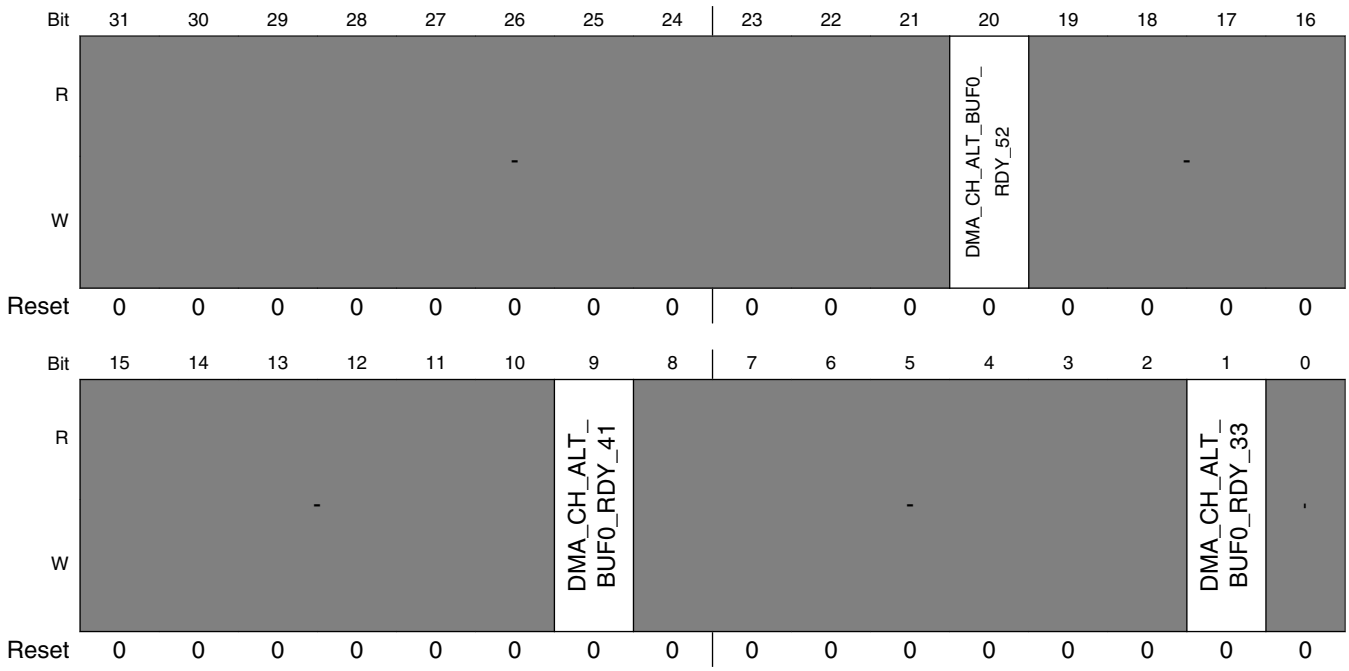
IPU_ALT_CH_BUF0_RDY0 field descriptions

Field	Description
31–30 -	Reserved.
29 DMA_CH_ALT_BUF0_RDY_29	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
28–25 -	Reserved.
24 DMA_CH_ALT_BUF0_RDY_24	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
23–0 -	Reserved.

45.51.81 IPU Alternate Channels Buffer 0 Ready 1 Register (IPU_ALT_CH_BUF0_RDY1)

The register contains buffer 0 ready control information for 32 IPU's DMA channels (63-32). Writing "1" to each field will set each bit. Writing "0" to each field simultaneously will clear all the bits.

Address: IPU_ALT_CH_BUF0_RDY1 is 1E00_0000h base + 27Ch offset = 1E00_027Ch



IPU_ALT_CH_BUF0_RDY1 field descriptions

Field	Description
31–21 -	Reserved.
20 DMA_CH_ALT_BUF0_RDY_52	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
19–10 -	Reserved.
9 DMA_CH_ALT_BUF0_RDY_41	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
8–2 -	Reserved.

Table continues on the next page...

IPU_ALT_CH_BUF0_RDY1 field descriptions (continued)

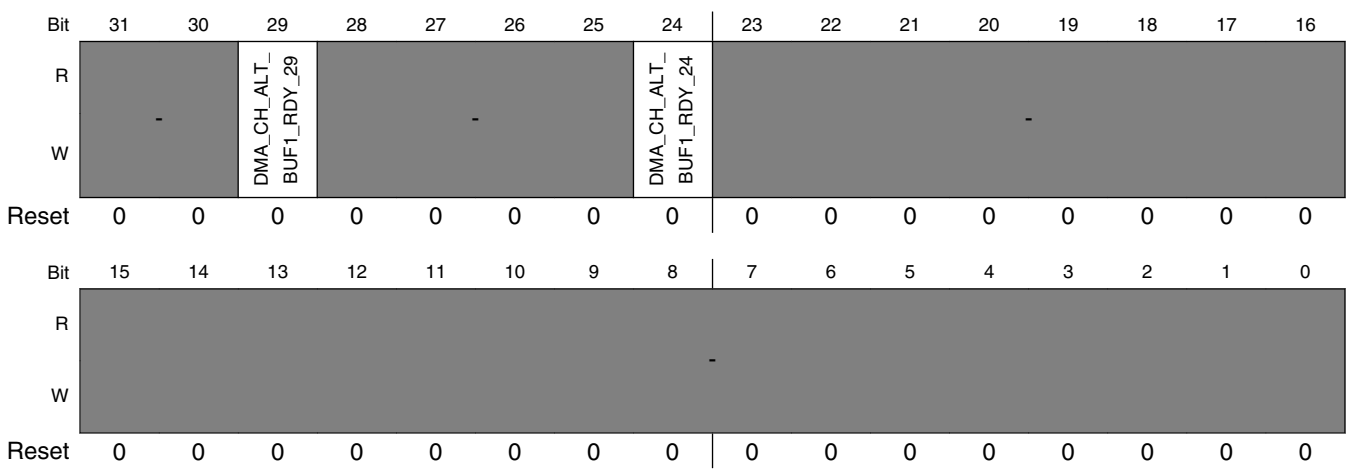
Field	Description
1 DMA_CH_ALT_ BUF0_RDY_33	Buffer 0 is ready. This bit indicates that ARM platform finished/writing reading buffer 0 in memory. 0 Buffer 0 is not ready. 1 Buffer 0 is ready.
0 -	Reserved.

45.51.82 IPU Alternate Channels Buffer1 Ready 0 Register (IPU_ALT_CH_BUF1_RDY0)

The register contains buffer 1 ready control information for 32 IPU's DMA channels (31-0). Writing "1" to each field will set each bit. Writing "0" to each field simultaneously will clear all the bits.

The register is shown in IPU Alternate Channels Buffer1 Ready 0 Register (IPU_ALT_CH_BUF1_RDY0), and the register fields are described in IPU Alternate Channels Buffer1 Ready 0 Register (IPU_ALT_CH_BUF1_RDY0).

Address: IPU_ALT_CH_BUF1_RDY0 is 1E00_0000h base + 280h offset = 1E00_0280h



IPU_ALT_CH_BUF1_RDY0 field descriptions

Field	Description
31–30 -	Reserved.
29 DMA_CH_ALT_ BUF1_RDY_29	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 buffer 1 is ready.

Table continues on the next page...

IPU_ALT_CH_BUF1_RDY0 field descriptions (continued)

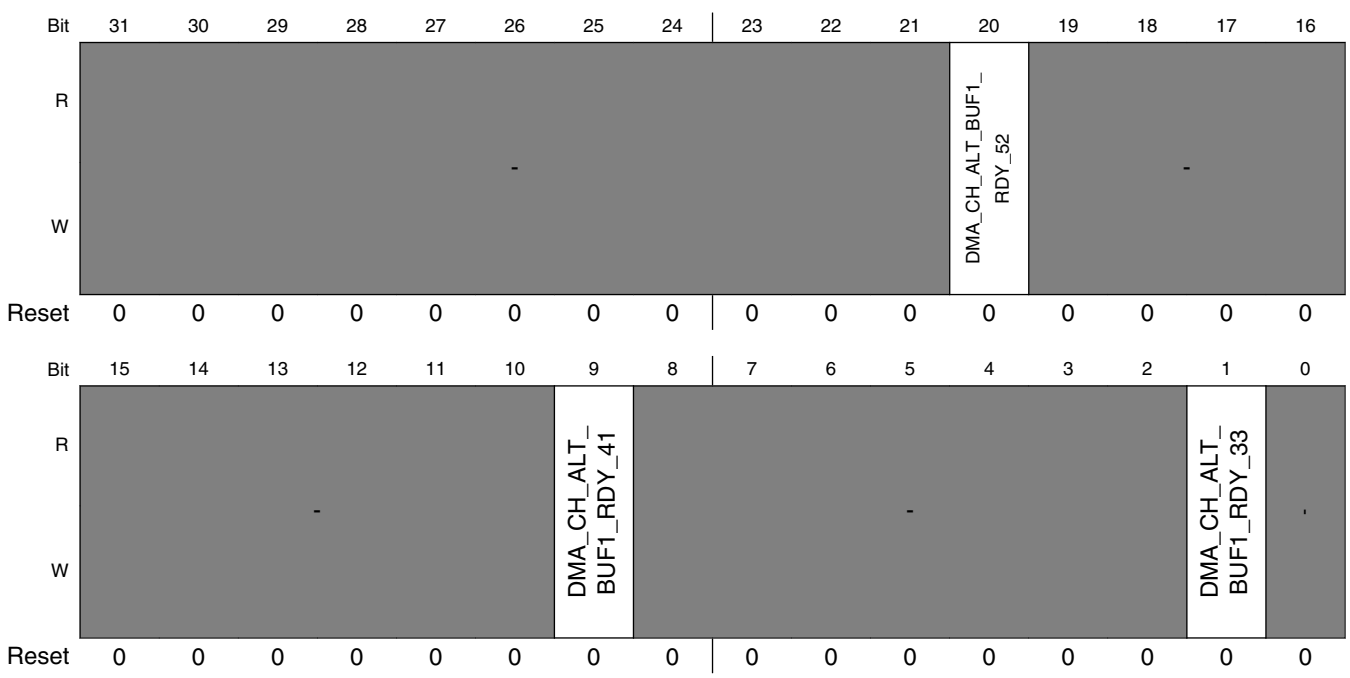
Field	Description
28–25 -	Reserved.
24 DMA_CH_ALT_BUF1_RDY_24	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 buffer 1 is ready.
23–0 -	Reserved.

45.51.83 IPU Alternate Channels Buffer 1 Ready 1 Register (IPU_ALT_CH_BUF1_RDY1)

The register contains buffer 0 ready control information for 32 IPU's DMA channels (63-32). Writing "1" to each field will set each bit. Writing "0" to each field simultaneously will clear all the bits.

The register is shown in IPU Alternate Channels Buffer 1 Ready 1 Register (IPU_ALT_CH_BUF1_RDY1), and the register fields are described in IPU Alternate Channels Buffer 1 Ready 1 Register (IPU_ALT_CH_BUF1_RDY1).

Address: IPU_ALT_CH_BUF1_RDY1 is 1E00_0000h base + 284h offset = 1E00_0284h



IPU_ALT_CH_BUF1_RDY1 field descriptions

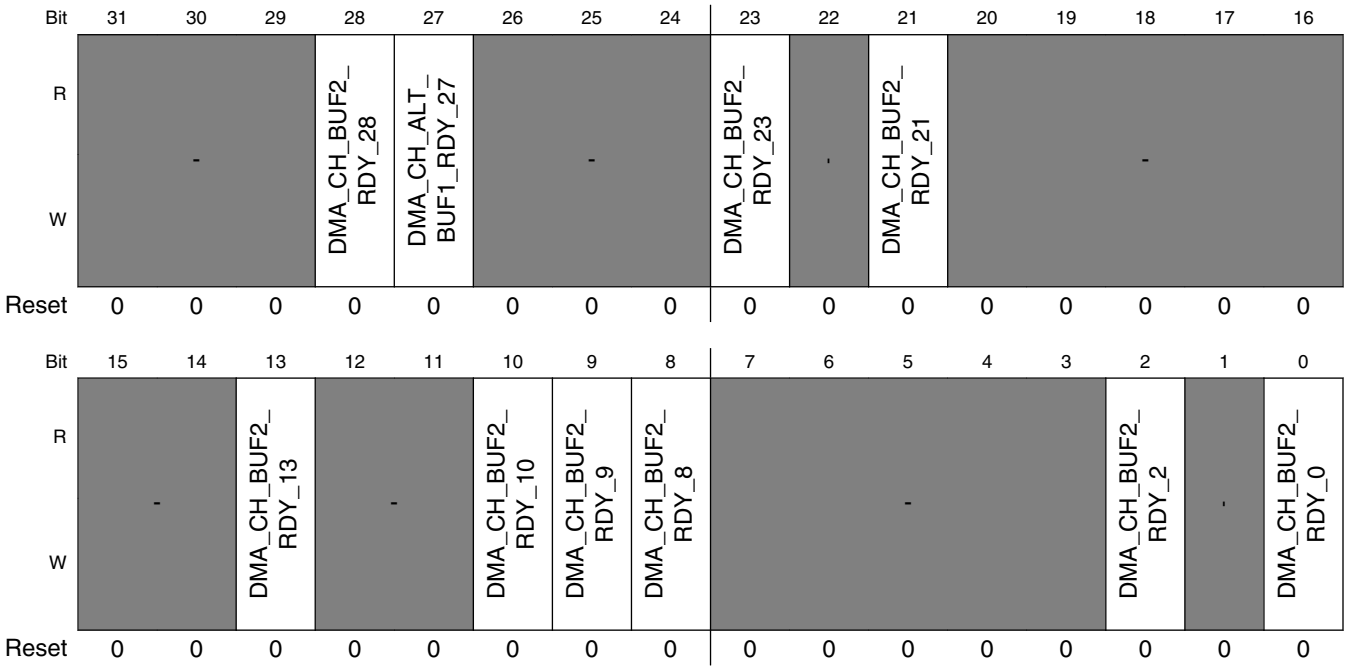
Field	Description
31–21 -	Reserved.
20 DMA_CH_ALT_ BUF1_RDY_52	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 buffer 1 is ready.
19–10 -	Reserved.
9 DMA_CH_ALT_ BUF1_RDY_41	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 buffer 1 is ready.
8–2 -	Reserved.
1 DMA_CH_ALT_ BUF1_RDY_33	Buffer 1 is ready. This bit indicates that ARM platform finished/writing reading buffer 1 in memory. 0 buffer 1 is not ready. 1 buffer 1 is ready.
0 -	Reserved.

45.51.84 IPU Channels Buffer 2 Ready 0 Register (IPU_CH_BUF2_RDY0)

The register contains buffer 2 ready control information for 32 IPU's DMA channels (31-0). This register can be a write one to set or a write one to clear according to the **IPU_CH_BUF2_RDY0_CLR bit**.

The register is shown in IPU Channels Buffer 2 Ready 0 Register (IPU_CH_BUF2_RDY0), and the register fields are described in IPU Channels Buffer 2 Ready 0 Register (IPU_CH_BUF2_RDY0).

Address: IPU_CH_BUF2_RDY0 is 1E00_0000h base + 288h offset = 1E00_0288h



IPU_CH_BUF2_RDY0 field descriptions

Field	Description
31–29 -	Reserved.
28 DMA_CH_ BUF2_RDY_28	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.
27 DMA_CH_ALT_ BUF1_RDY_27	buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 buffer 2 is not ready. 1 buffer 2 is ready.
26–24 -	Reserved.
23 DMA_CH_ BUF2_RDY_23	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.
22 -	Reserved.
21 DMA_CH_ BUF2_RDY_21	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.
20–14 -	Reserved.

Table continues on the next page...

IPU_CH_BUF2_RDY0 field descriptions (continued)

Field	Description
13 DMA_CH_ BUF2_RDY_13	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.
12–11 -	Reserved.
10 DMA_CH_ BUF2_RDY_10	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.
9 DMA_CH_ BUF2_RDY_9	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.
8 DMA_CH_ BUF2_RDY_8	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.
7–3 -	Reserved.
2 DMA_CH_ BUF2_RDY_2	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.
1 -	Reserved.
0 DMA_CH_ BUF2_RDY_0	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.

45.51.85 IPU Channels Buffer 2 Ready 1 Register (IPU_CH_BUF2_RDY1)

The register contains buffer 2 ready control information for 32 IPU's DMA channels (63-32). This register can be a write one to set or a write one to clear according to the **IPU_CH_BUF2_RDY1_CLR bit.**

The register is shown in IPU Channels Buffer 2 Ready 1 Register (IPU_CH_BUF2_RDY1), and the register fields are described in IPU Channels Buffer 2 Ready 1 Register (IPU_CH_BUF2_RDY1).

Address: IPU_CH_BUF2_RDY1 is 1E00_0000h base + 28Ch offset = 1E00_028Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DMA_CH_BUF2_RDY_*																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CH_BUF2_RDY1 field descriptions

Field	Description
31–0 DMA_CH_ BUF2_RDY_*	Buffer 2 is ready. This bit indicates that ARM platform finished/writing reading buffer 2 in memory. 0 Buffer 2 is not ready. 1 Buffer 2 is ready.

45.51.86 Interrupt Status Register 4 (IPU_INT_STAT_4)

IPU status registers are not stored in the SRM during power gating mode.This register contains part of IPU interrupts status bits. The status bits of NFACK (New Frame Ack) of DMA Channels interrupts [63:32] can be found in this register.

Address: IPU_INT_STAT_4 is 1E00_0000h base + 2C0h offset = 1E00_02C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											IDMAC_NFACK_52	IDMAC_NFACK_51	IDMAC_NFACK_50	IDMAC_NFACK_49	IDMAC_NFACK_48
W												w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_NFACK_47	IDMAC_NFACK_46	IDMAC_NFACK_45	IDMAC_NFACK_44	IDMAC_NFACK_43	IDMAC_NFACK_42	IDMAC_NFACK_41	IDMAC_NFACK_40	0						IDMAC_NFACK_33	0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c							w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_INT_STAT_4 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_INT_STAT_4 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
20 IDMAC_ NFACK_52	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
19 IDMAC_ NFACK_51	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
18 IDMAC_ NFACK_50	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
17 IDMAC_ NFACK_49	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
16 IDMAC_ NFACK_48	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
15 IDMAC_ NFACK_47	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
14 IDMAC_ NFACK_46	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
13 IDMAC_ NFACK_45	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
12 IDMAC_ NFACK_44	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number.

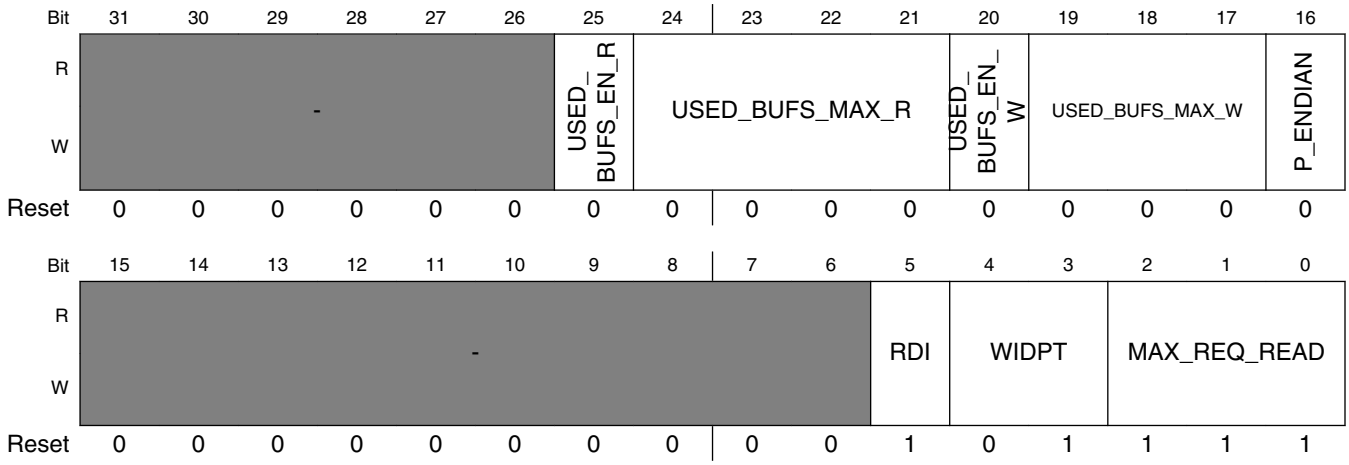
Table continues on the next page...

IPU_INT_STAT_4 field descriptions (continued)

Field	Description
	0 Interrupt is cleared. 1 Interrupt is requested.
11 IDMAC_ NFACK_43	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
10 IDMAC_ NFACK_42	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
9 IDMAC_ NFACK_41	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
8 IDMAC_ NFACK_40	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
7–2 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.
1 IDMAC_ NFACK_33	Enable New Frame Ack of Channel interrupt. This bit is the status bits of New Frame Ack of Channel #n. *n* Indicates the corresponding DMA channel number. 0 Interrupt is cleared. 1 Interrupt is requested.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved. 0 Interrupt is cleared. 1 Interrupt is requested.

45.51.87 IDMAC Configuration Register (IPU_IDMAC_CONF)

Address: IPU_IDMAC_CONF is 1E00_0000h base + 8000h offset = 1E00_8000h



IPU_IDMAC_CONF field descriptions

Field	Description
31–26 -	Reserved, should be cleared.
25 USED_BUFS_EN_R	Enables the limit on the number of pending non real time read requests.
24–21 USED_BUFS_MAX_R	Limit the number of pending non real time read requests. The value can be between 0 to 8. This field has no affect if USED_BUFS_EN_R is cleared
20 USED_BUFS_EN_W	Enables the limit on the number of pending non real time write requests.
19–17 USED_BUFS_MAX_W	Limit the number of pending non real time write requests. The value can be between 0 to 6. This field has no affect if USED_BUFS_EN_W is cleared
16 P_ENDIAN	Pixel Endianness. The pixel Endianness must not be changed while any of the IDMAC channels is enabled. 0 little endian 1 Big endian
15–6 -	Reserved, should be cleared.
5 RDI	Read Data Interleaving. This bit must match the slave read data interleaving support. If the AXI slave connected to the IPU supports read data interleaving then this bit must be set. If the AXI slave does not support read data interleaving then the IDMAC can utilize this and issue more address phases on read. In that case it is recommended to have this bit cleared. 0 The AXI slave does not support read data interleaving

Table continues on the next page...

IPU_IDMAC_CONF field descriptions (continued)

Field	Description
	1 The AXI slave supports read data interleaving
4–3 WIDPT	Write Interleaving Depth These 2 bits define the Write Interleaving Depth of the AXI port. This bits should be configured by the user according to the AXI slave's Write Interleaving Depth. WIDPT defines the maximal number of active bursts (yet to be responded) with different IDs. IDMAC will block data phase if the next data's ID is new (no such ID active) and the number of active IDs is equal to WIDPT. 00 Write Interleaving Depth of 1 01 Write Interleaving Depth of 2 10 Write Interleaving Depth of 3 11 Write Interleaving Depth of 4
2–0 MAX_REQ_ READ	Maximum Read Requests. This fields sets the maximum pending requests allowed in the AXI Read requests queue.

45.51.88 IDMAC Channel Enable 1 Register (IPU_IDMAC_CH_EN_1)

Address: IPU_IDMAC_CH_EN_1 is 1E00_0000h base + 8004h offset = 1E00_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IDMAC_CH_EN_31	-	IDMAC_CH_EN_29	IDMAC_CH_EN_28	IDMAC_CH_EN_27	IDMAC_CH_EN_26	IDMAC_CH_EN_25	IDMAC_CH_EN_24	IDMAC_CH_EN_23	IDMAC_CH_EN_22	IDMAC_CH_EN_21	IDMAC_CH_EN_20	IDMAC_CH_EN_19	IDMAC_CH_EN_18	IDMAC_CH_EN_17	-
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IDMAC_CH_EN_15	IDMAC_CH_EN_14	IDMAC_CH_EN_13	IDMAC_CH_EN_12	IDMAC_CH_EN_11	IDMAC_CH_EN_10	IDMAC_CH_EN_9	IDMAC_CH_EN_8	-		IDMAC_CH_EN_5	-	IDMAC_CH_EN_3	IDMAC_CH_EN_2	IDMAC_CH_EN_1	IDMAC_CH_EN_0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IDMAC_CH_EN_1 field descriptions

Field	Description
31 IDMAC_CH_ EN_31	IDMAC Channel enable bit [i]

Table continues on the next page...

IPU_IDMAC_CH_EN_1 field descriptions (continued)

Field	Description
	0 IDMAC channel is disabled 1 IDMAC channel is enabled
30 -	Reserved.
29 IDMAC_CH_EN_29	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
28 IDMAC_CH_EN_28	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
27 IDMAC_CH_EN_27	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
26 IDMAC_CH_EN_26	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
25 IDMAC_CH_EN_25	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
24 IDMAC_CH_EN_24	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
23 IDMAC_CH_EN_23	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
22 IDMAC_CH_EN_22	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
21 IDMAC_CH_EN_21	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
20 IDMAC_CH_EN_20	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
19 IDMAC_CH_EN_19	IDMAC Channel enable bit [i]

Table continues on the next page...

IPU_IDMAC_CH_EN_1 field descriptions (continued)

Field	Description
	0 IDMAC channel is disabled 1 IDMAC channel is enabled
18 IDMAC_CH_EN_18	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
17 IDMAC_CH_EN_17	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
16 -	Reserved.
15 IDMAC_CH_EN_15	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
14 IDMAC_CH_EN_14	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
13 IDMAC_CH_EN_13	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
12 IDMAC_CH_EN_12	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
11 IDMAC_CH_EN_11	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
10 IDMAC_CH_EN_10	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
9 IDMAC_CH_EN_9	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
8 IDMAC_CH_EN_8	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
7–6 -	Reserved.

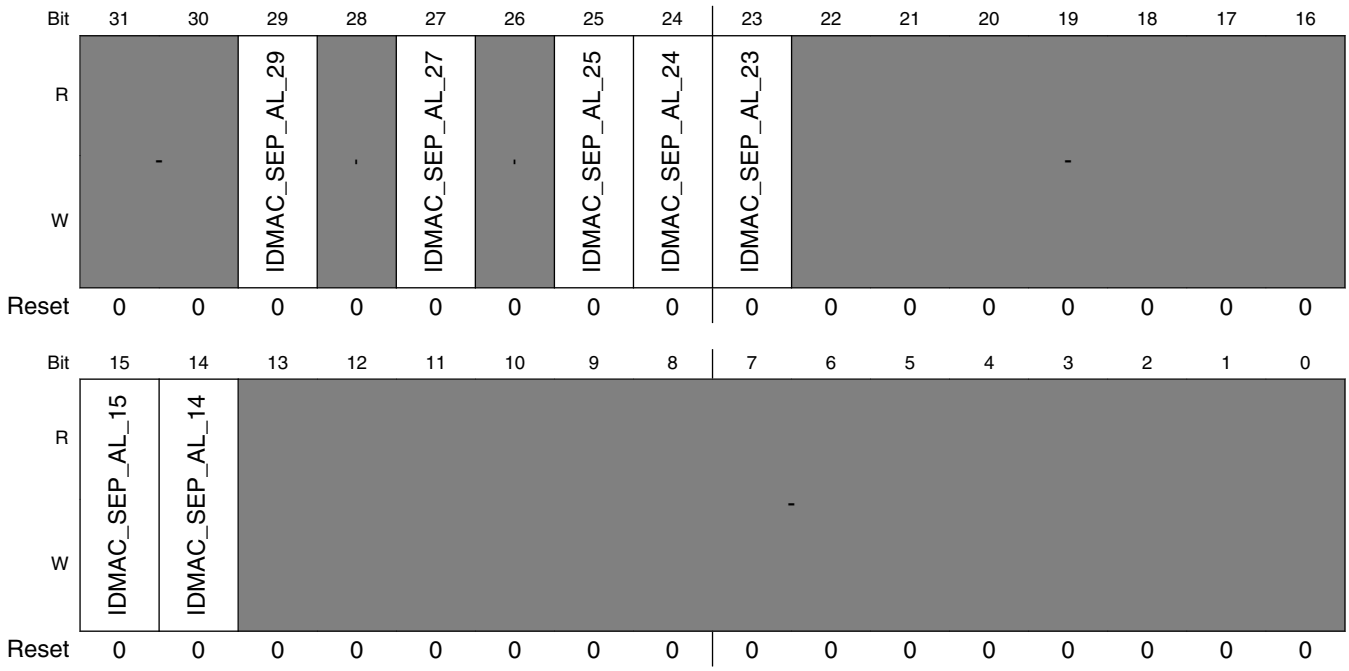
Table continues on the next page...

IPU_IDMAC_CH_EN_1 field descriptions (continued)

Field	Description
5 IDMAC_CH_EN_5	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
4 -	Reserved.
3 IDMAC_CH_EN_3	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
2 IDMAC_CH_EN_2	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
1 IDMAC_CH_EN_1	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
0 IDMAC_CH_EN_0	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled

45.51.89 IDMAC Separate Alpha Indication Register (IPU_IDMAC_SEP_ALPHA)

Address: IPU_IDMAC_SEP_ALPHA is 1E00_0000h base + 800Ch offset = 1E00_800Ch



IPU_IDMAC_SEP_ALPHA field descriptions

Field	Description
31–30 -	Reserved.
29 IDMAC_SEP_AL_29	IDMAC Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
28 -	Reserved.
27 IDMAC_SEP_AL_27	IDMAC Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding.

Table continues on the next page...

IPU_IDMAC_SEP_ALPHA field descriptions (continued)

Field	Description
	0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
26 -	Reserved.
25 IDMAC_SEP_AL_25	IDMAC Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
24 IDMAC_SEP_AL_24	IDMAC Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
23 IDMAC_SEP_AL_23	IDMAC Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
22–16 -	Reserved.
15 IDMAC_SEP_AL_15	IDMAC Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
14 IDMAC_SEP_AL_14	IDMAC Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel.

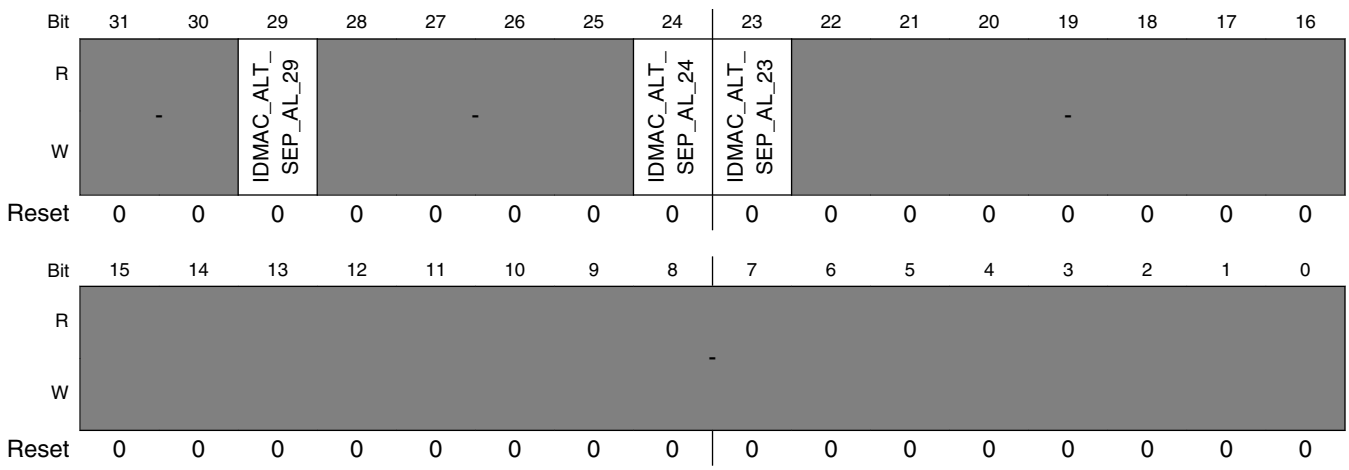
Table continues on the next page...

IPU_IDMAC_SEP_ALPHA field descriptions (continued)

Field	Description
	In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
13–0 -	Reserved.

45.51.90 IDMAC Alternate Separate Alpha Indication Register (IPU_IDMAC_ALT_SEP_ALPHA)

Address: IPU_IDMAC_ALT_SEP_ALPHA is 1E00_0000h base + 8010h offset = 1E00_8010h



IPU_IDMAC_ALT_SEP_ALPHA field descriptions

Field	Description
31–30 -	Reserved.
29 IDMAC_ALT_SEP_AL_29	IDMAC Alternate Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. For channels that may have alternate flow and may use separate alpha. This bit indicates the mode of the alpha for the alternate flow 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
28–25 -	Reserved.

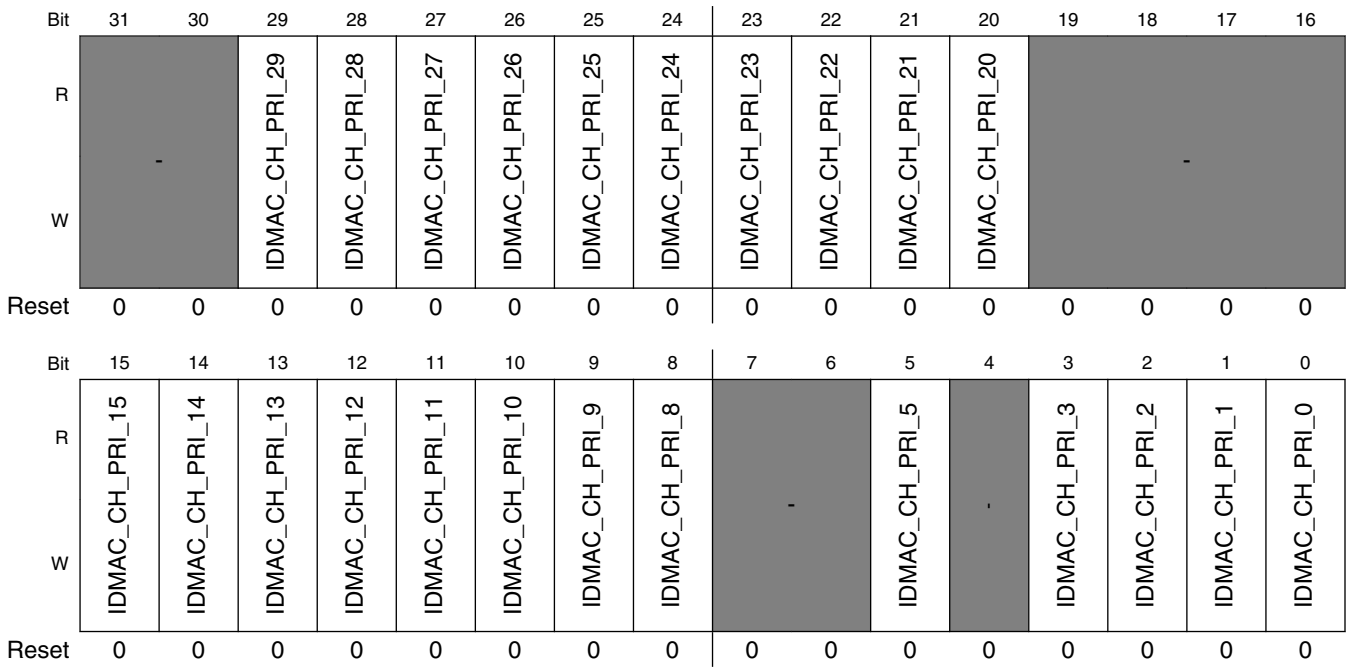
Table continues on the next page...

IPU_IDMAC_ALT_SEP_ALPHA field descriptions (continued)

Field	Description
24 IDMAC_ALT_ SEP_AL_24	IDMAC Alternate Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. For channels that may have alternate flow and may use separate alpha. This bit indicates the mode of the alpha for the alternate flow 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
23 IDMAC_ALT_ SEP_AL_23	IDMAC Alternate Separate alpha indication bit [i] A sub block may need to read data from the system's memory where the pixel data and the alpha transparency data are located in separate buffers. In that case the Alpha transparency data is read by the special Alpha channel. In a case where the alpha should be read from a separate buffer, the user should set the channels corresponding. For channels that may have alternate flow and may use separate alpha. This bit indicates the mode of the alpha for the alternate flow 0 Channel [i] does not read Alpha transparency data from a separate buffer. 1 Channel [i] reads Alpha transparency data from a separate buffer.
22–0 -	Reserved.

45.51.91 IDMAC Channel Priority 1 Register (IPU_IDMAC_CH_PRI_1)

Address: IPU_IDMAC_CH_PRI_1 is 1E00_0000h base + 8014h offset = 1E00_8014h



IPU_IDMAC_CH_PRI_1 field descriptions

Field	Description
31–30 -	Reserved.
29 IDMAC_CH_ PRI_29	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
28 IDMAC_CH_ PRI_28	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
27 IDMAC_CH_ PRI_27	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
26 IDMAC_CH_ PRI_26	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
25 IDMAC_CH_ PRI_25	IDMAC Channel enable bit [i]

Table continues on the next page...

IPU_IDMAC_CH_PRI_1 field descriptions (continued)

Field	Description
	0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
24 IDMAC_CH_ PRI_24	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
23 IDMAC_CH_ PRI_23	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
22 IDMAC_CH_ PRI_22	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
21 IDMAC_CH_ PRI_21	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
20 IDMAC_CH_ PRI_20	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
19–16 -	Reserved.
15 IDMAC_CH_ PRI_15	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
14 IDMAC_CH_ PRI_14	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
13 IDMAC_CH_ PRI_13	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
12 IDMAC_CH_ PRI_12	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
11 IDMAC_CH_ PRI_11	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
10 IDMAC_CH_ PRI_10	IDMAC Channel enable bit [i]

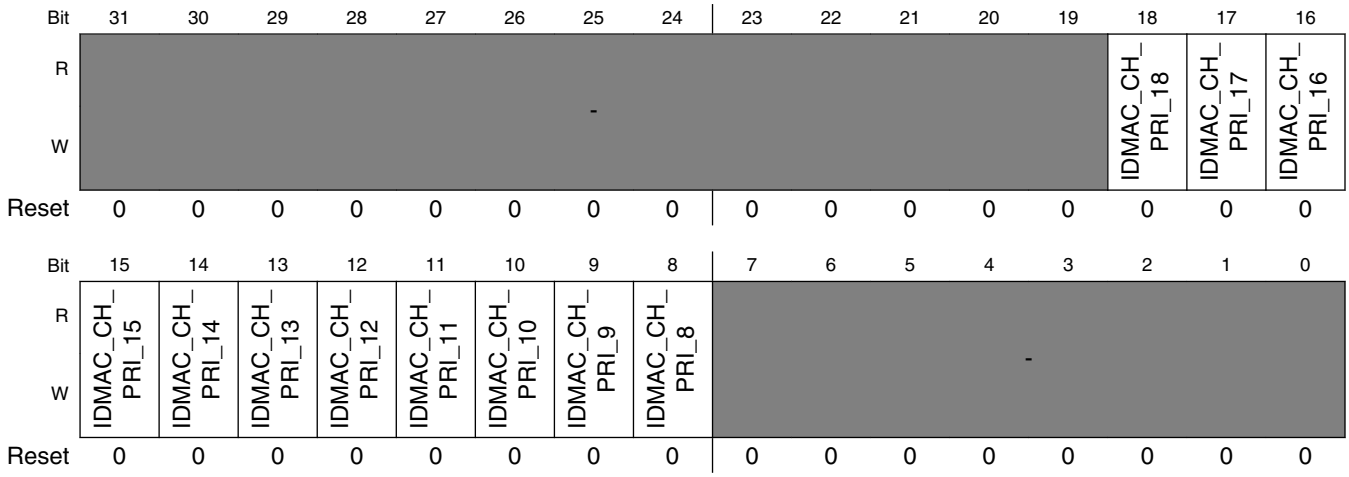
Table continues on the next page...

IPU_IDMAC_CH_PRI_1 field descriptions (continued)

Field	Description
	0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
9 IDMAC_CH_ PRI_9	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
8 IDMAC_CH_ PRI_8	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
7–6 -	Reserved.
5 IDMAC_CH_ PRI_5	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
4 -	Reserved.
3 IDMAC_CH_ PRI_3	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
2 IDMAC_CH_ PRI_2	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
1 IDMAC_CH_ PRI_1	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
0 IDMAC_CH_ PRI_0	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority

45.51.92 IDMAC Channel Priority 2 Register (IPU_IDMAC_CH_PRI_2)

Address: IPU_IDMAC_CH_PRI_2 is 1E00_0000h base + 8018h offset = 1E00_8018h



IPU_IDMAC_CH_PRI_2 field descriptions

Field	Description
31–19 -	Reserved.
18 IDMAC_CH_PRI_18	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
17 IDMAC_CH_PRI_17	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
16 IDMAC_CH_PRI_16	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
15 IDMAC_CH_PRI_15	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
14 IDMAC_CH_PRI_14	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
13 IDMAC_CH_PRI_13	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority

Table continues on the next page...

IPU_IDMAC_CH_PRI_2 field descriptions (continued)

Field	Description
12 IDMAC_CH_ PRI_12	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
11 IDMAC_CH_ PRI_11	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
10 IDMAC_CH_ PRI_10	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
9 IDMAC_CH_ PRI_9	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
8 IDMAC_CH_ PRI_8	IDMAC Channel enable bit [i] 0 IDMAC channel [i] is in low priority 1 IDMAC channel [i] is in high priority
7–0 -	Reserved.

45.51.93 IDMAC Channel Watermark Enable 1 Register (IPU_IDMAC_WM_EN_1)

Address: IPU_IDMAC_WM_EN_1 is 1E00_0000h base + 801Ch offset = 1E00_801Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-		IDMAC_WM_EN_29	IDMAC_WM_EN_28	IDMAC_WM_EN_27	IDMAC_WM_EN_26	IDMAC_WM_EN_25	IDMAC_WM_EN_24	IDMAC_WM_EN_23	-						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-	IDMAC_WM_EN_14	IDMAC_WM_EN_13	IDMAC_WM_EN_12	-	IDMAC_WM_EN_10	-	IDMAC_WM_EN_8	-				IDMAC_WM_EN_3	IDMAC_WM_EN_2	IDMAC_WM_EN_1	IDMAC_WM_EN_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IDMAC_WM_EN_1 field descriptions

Field	Description
31–30 -	Reserved.
29 IDMAC_WM_EN_29	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
28 IDMAC_WM_EN_28	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
27 IDMAC_WM_EN_27	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
26 IDMAC_WM_EN_26	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled

Table continues on the next page...

IPU_IDMAC_WM_EN_1 field descriptions (continued)

Field	Description
25 IDMAC_WM_EN_25	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
24 IDMAC_WM_EN_24	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
23 IDMAC_WM_EN_23	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
22–15 -	Reserved.
14 IDMAC_WM_EN_14	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
13 IDMAC_WM_EN_13	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
12 IDMAC_WM_EN_12	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
11 -	Reserved.
10 IDMAC_WM_EN_10	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
9 -	Reserved.
8 IDMAC_WM_EN_8	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
7–4 -	Reserved.
3 IDMAC_WM_EN_3	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled

Table continues on the next page...

IPU_IDMAC_WM_EN_1 field descriptions (continued)

Field	Description
2 IDMAC_WM_EN_2	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
1 IDMAC_WM_EN_1	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
0 IDMAC_WM_EN_0	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled

45.51.94 IDMAC Channel Watermark Enable 2 Register (IPU_IDMAC_WM_EN_2)

Address: IPU_IDMAC_WM_EN_2 is 1E00_0000h base + 8020h offset = 1E00_8020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	-															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-			IDMAC_WM_EN_44	IDMAC_WM_EN_43	IDMAC_WM_EN_42	IDMAC_WM_EN_41	IDMAC_WM_EN_40	-							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IDMAC_WM_EN_2 field descriptions

Field	Description
31–13 -	Reserved.
12 IDMAC_WM_EN_44	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
11 IDMAC_WM_EN_43	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled

Table continues on the next page...

IPU_IDMAC_WM_EN_2 field descriptions (continued)

Field	Description
10 IDMAC_WM_EN_42	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
9 IDMAC_WM_EN_41	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
8 IDMAC_WM_EN_40	IDMAC Watermark enable bit [i] 0 IDMAC channel [i] watermark feature is disabled 1 IDMAC channel [i] watermark feature is enabled
7–0 -	Reserved.

45.51.95 IDMAC Channel Lock Enable 1Register (IPU_IDMAC_LOCK_EN_1)

Address: IPU_IDMAC_LOCK_EN_1 is 1E00_0000h base + 8024h offset = 1E00_8024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	-										IDMAC_LOCK_EN_28		IDMAC_LOCK_EN_27		IDMAC_LOCK_EN_23	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IDMAC_LOCK_EN_22		IDMAC_LOCK_EN_21		IDMAC_LOCK_EN_20		IDMAC_LOCK_EN_15		IDMAC_LOCK_EN_14		IDMAC_LOCK_EN_12		IDMAC_LOCK_EN_11		IDMAC_LOCK_EN_5	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IDMAC_LOCK_EN_1 field descriptions

Field	Description
31–22 -	Reserved.
21–20 IDMAC_LOCK_EN_28	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.

Table continues on the next page...

IPU_IDMAC_LOCK_EN_1 field descriptions (continued)

Field	Description
19–18 IDMAC_LOCK_EN_27	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
17–16 IDMAC_LOCK_EN_23	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
15–14 IDMAC_LOCK_EN_22	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
13–12 IDMAC_LOCK_EN_21	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
11–10 IDMAC_LOCK_EN_20	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
9–8 IDMAC_LOCK_EN_15	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
7–6 IDMAC_LOCK_EN_14	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request.

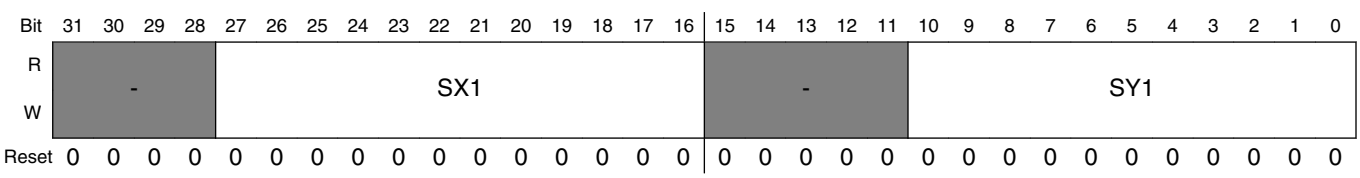
Table continues on the next page...

IPU_IDMAC_LOCK_EN_1 field descriptions (continued)

Field	Description
	10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
5–4 IDMAC_LOCK_EN_12	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
3–2 IDMAC_LOCK_EN_11	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
1–0 IDMAC_LOCK_EN_5	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.

45.51.96 IDMAC Scroll Coordinations Register 1 (IPU_IDMAC_SC_CORD_1)

Address: IPU_IDMAC_SC_CORD_1 is 1E00_0000h base + 804Ch offset = 1E00_804Ch



IPU_IDMAC_SC_CORD_1 field descriptions

Field	Description
31–28 -	Reserved, should be cleared.
27–16 SX1	Scroll X coordination (2nd set) This field indicates the X coordinate of the scroll. This parameter has an affect on continuos scroll mode only. Units are pixels. When the alpha buffer resides in a separate buffer and the alpha is different than 8 BPP this field should be 0.

Table continues on the next page...

IPU_IDMAC_SC_CORD_1 field descriptions (continued)

Field	Description
15–11 -	Reserved, should be cleared.
10–0 SY1	Scroll Y coordination (2nd set) This field indicates the Y coordinate of the scroll. This parameter has an affect on continuos scroll mode only. Units are pixels. When the alpha buffer resides in a separate buffer and the alpha is different than 8 BPP this field should be 0.

45.51.97 IDMAC Channel Busy 1 Register (IPU_IDMAC_CH_BUSY_1)

Address: IPU_IDMAC_CH_BUSY_1 is 1E00_0000h base + 8100h offset = 1E00_8100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IDMAC_CH_BUSY_31	-	IDMAC_CH_BUSY_29	IDMAC_CH_BUSY_28	IDMAC_CH_BUSY_27	IDMAC_CH_BUSY_26	IDMAC_CH_BUSY_25	IDMAC_CH_BUSY_24	IDMAC_CH_BUSY_23	IDMAC_CH_BUSY_22	IDMAC_CH_BUSY_21	IDMAC_CH_BUSY_20	-	IDMAC_CH_BUSY_18	IDMAC_CH_BUSY_17	-
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDMAC_CH_BUSY_15	IDMAC_CH_BUSY_14	IDMAC_CH_BUSY_13	IDMAC_CH_BUSY_12	IDMAC_CH_BUSY_11	IDMAC_CH_BUSY_10	IDMAC_CH_BUSY_9	IDMAC_CH_BUSY_8	-		IDMAC_CH_BUSY_5	-	IDMAC_CH_BUSY_3	IDMAC_CH_BUSY_2	IDMAC_CH_BUSY_1	IDMAC_CH_BUSY_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IDMAC_CH_BUSY_1 field descriptions

Field	Description
31 IDMAC_CH_BUSY_	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
30 -	Reserved.

Table continues on the next page...

IPU_IDMAC_CH_BUSY_1 field descriptions (continued)

Field	Description
29 IDMAC_CH_ BUSY_29	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
28 IDMAC_CH_ BUSY_28	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
27 IDMAC_CH_ BUSY_27	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
26 IDMAC_CH_ BUSY_26	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
25 IDMAC_CH_ BUSY_25	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
24 IDMAC_CH_ BUSY_24	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
23 IDMAC_CH_ BUSY_23	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy

Table continues on the next page...

IPU_IDMAC_CH_BUSY_1 field descriptions (continued)

Field	Description
22 IDMAC_CH_ BUSY_22	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
21 IDMAC_CH_ BUSY_21	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
20 IDMAC_CH_ BUSY_20	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
19 -	Reserved.
18 IDMAC_CH_ BUSY_18	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
17 IDMAC_CH_ BUSY_17	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
16 -	Reserved.
15 IDMAC_CH_ BUSY_15	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
14 IDMAC_CH_ BUSY_14	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC.

Table continues on the next page...

IPU_IDMAC_CH_BUSY_1 field descriptions (continued)

Field	Description
	0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
13 IDMAC_CH_BUSY_13	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
12 IDMAC_CH_BUSY_12	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
11 IDMAC_CH_BUSY_11	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
10 IDMAC_CH_BUSY_10	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
9 IDMAC_CH_BUSY_9	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
8 IDMAC_CH_BUSY_8	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
7–6 -	Reserved.
5 IDMAC_CH_BUSY_5	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC.

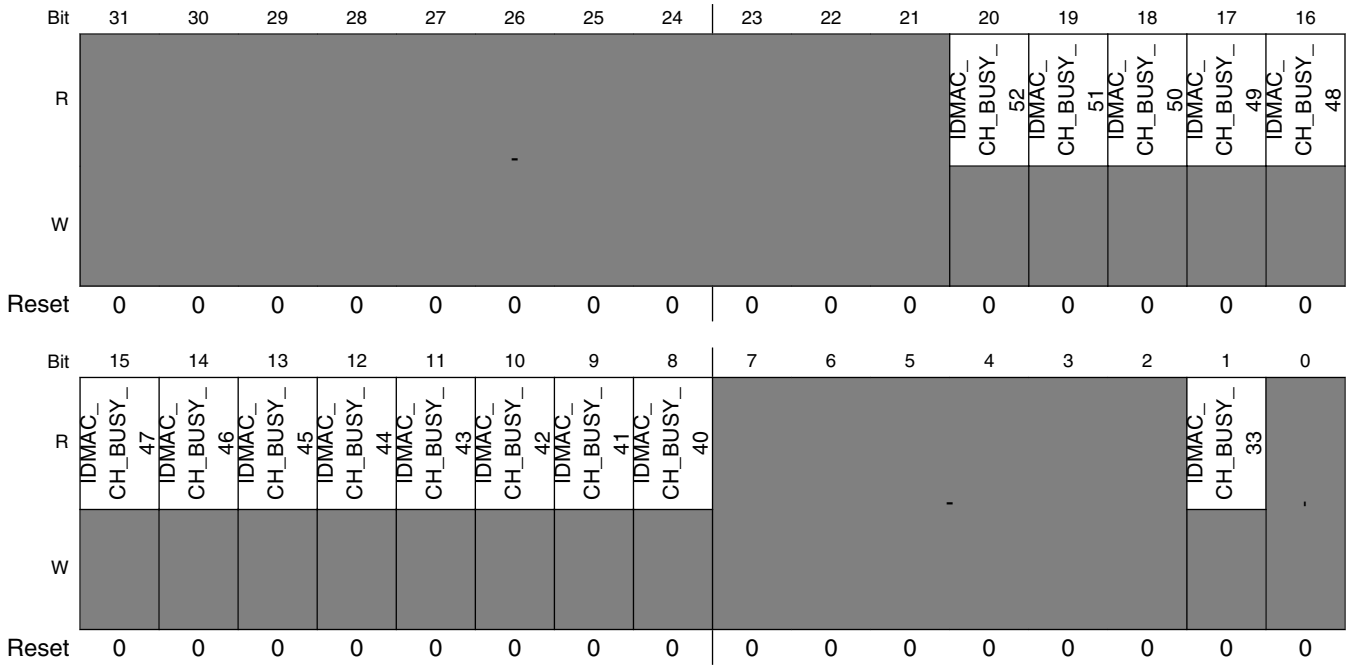
Table continues on the next page...

IPU_IDMAC_CH_BUSY_1 field descriptions (continued)

Field	Description
	0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
4 -	Reserved.
3 IDMAC_CH_ BUSY_3	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
2 IDMAC_CH_ BUSY_2	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
1 IDMAC_CH_ BUSY_1	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
0 IDMAC_CH_ BUSY_0	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy

45.51.98 IDMAC Channel Busy 2 Register (IPU_IDMAC_CH_BUSY_2)

Address: IPU_IDMAC_CH_BUSY_2 is 1E00_0000h base + 8104h offset = 1E00_8104h



IPU_IDMAC_CH_BUSY_2 field descriptions

Field	Description
31–21 -	Reserved.
20 IDMAC_CH_BUSY_52	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
19 IDMAC_CH_BUSY_51	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
18 IDMAC_CH_BUSY_50	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy

Table continues on the next page...

IPU_IDMAC_CH_BUSY_2 field descriptions (continued)

Field	Description
17 IDMAC_CH_ BUSY_49	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
16 IDMAC_CH_ BUSY_48	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
15 IDMAC_CH_ BUSY_47	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
14 IDMAC_CH_ BUSY_46	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
13 IDMAC_CH_ BUSY_45	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
12 IDMAC_CH_ BUSY_44	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
11 IDMAC_CH_ BUSY_43	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy

Table continues on the next page...

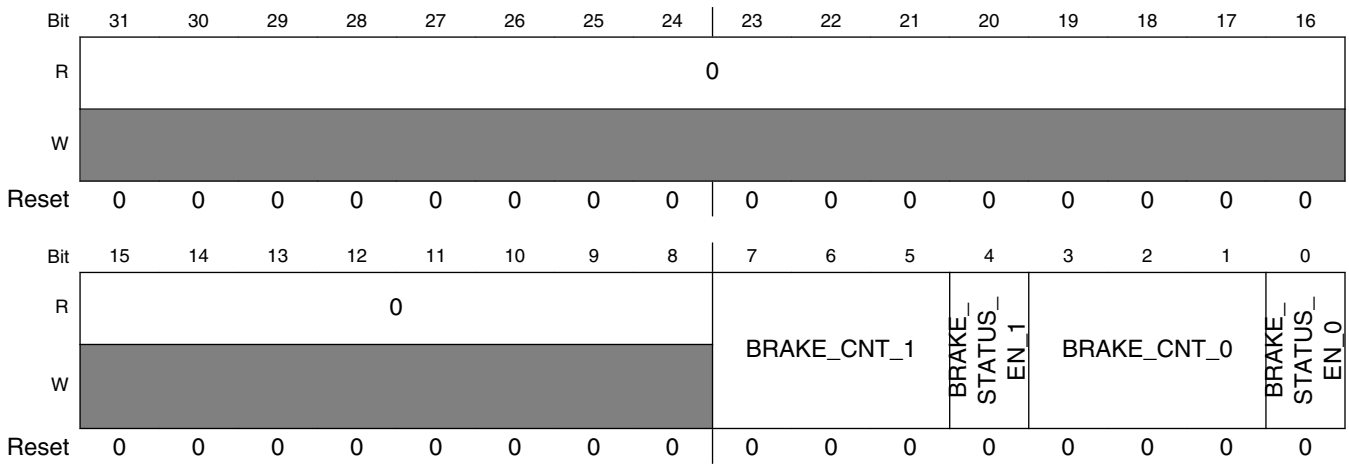
IPU_IDMAC_CH_BUSY_2 field descriptions (continued)

Field	Description
10 IDMAC_CH_ BUSY_42	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
9 IDMAC_CH_ BUSY_41	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
8 IDMAC_CH_ BUSY_40	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
7–2 -	Reserved.
1 IDMAC_CH_ BUSY_33	IDMAC Channel busy bit [i] This bit indicates if the channel is currently served by the IDMAC. This bit is self cleared by the IDMAC. 0 IDMAC channel [i] is not busy 1 IDMAC channel [i] is busy
0 -	Reserved.

45.51.99 DP Debug Control Register (IPU_DP_DEBUG_CNT)

This is the debug unit control register. This register is not stored in the SRM.

Address: IPU_DP_DEBUG_CNT is 1E00_0000h base + 180BCh offset = 1E01_80BCh

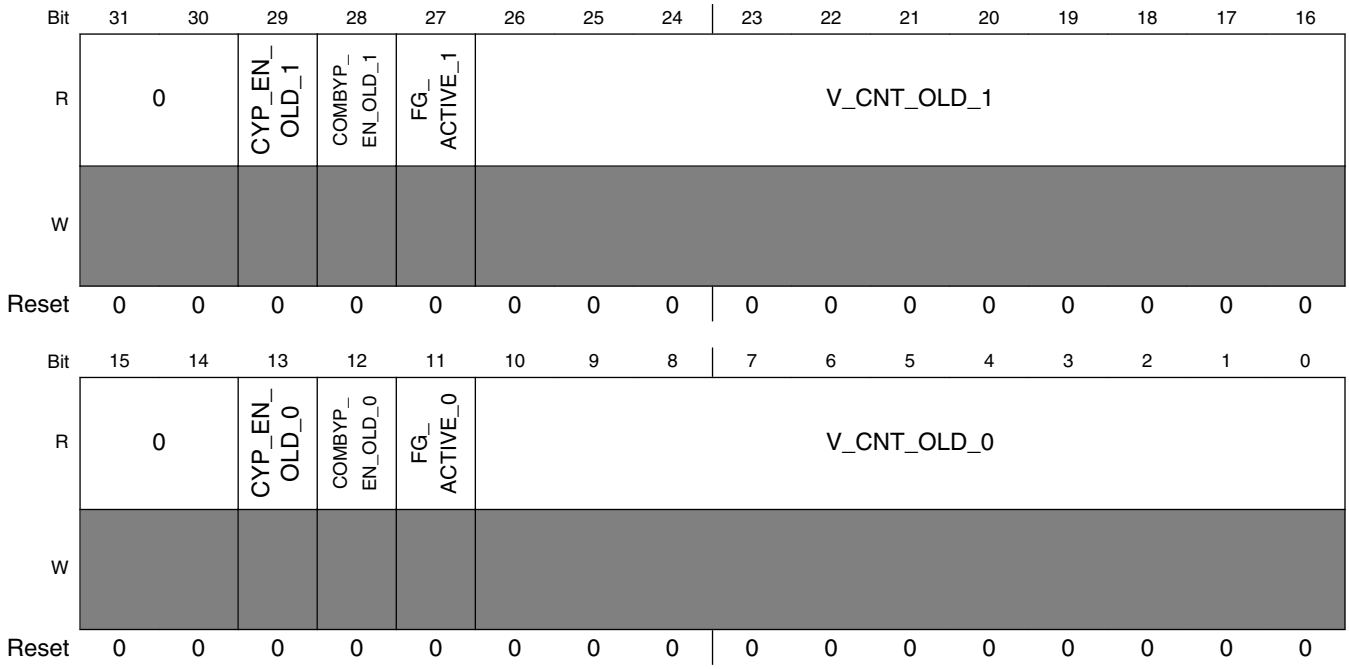


IPU_DP_DEBUG_CNT field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–5 BRAKE_CNT_1	The async flow can be broken multiple times. It possible to control which breaking event will cause the interrupt. This field counts the breaking events for unit #1
4 BRAKE_ STATUS_EN_1	This bit enables the break/status unit #1
3–1 BRAKE_CNT_0	The async flow can be broken multiple times. It possible to control which breaking event will cause the interrupt. This field counts the breaking events for unit #0
0 BRAKE_ STATUS_EN_0	This bit enables the break/status unit #0

45.51.100 DP Debug Status Register (IPU_DP_DEBUG_STAT)

Address: IPU_DP_DEBUG_STAT is 1E00_0000h base + 180C0h offset = 1E01_80C0h



IPU_DP_DEBUG_STAT field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved
29 CYP_EN_OLD_1	The async flow has been broken in the middle of a cursor (This filed is relevant for debug unit #1)
28 COMBYP_EN_OLD_1	the async1 flow has been broken in the middle of combining (This filed is relevant for debug unit #1)
27 FG_ACTIVE_1	Displaying the partial frame has been started (This filed is relevant for debug unit #1)
26–16 V_CNT_OLD_1	The exact row where the async flow has been broken (This filed is relevant for debug unit #0)
15–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 CYP_EN_OLD_0	The async flow has been broken in the middle of a cursor (This filed is relevant for debug unit #0)
12 COMBYP_EN_OLD_0	the async flow has been broken in the middle of combining (This filed is relevant for debug unit #0)

Table continues on the next page...

IPU_DP_DEBUG_STAT field descriptions (continued)

Field	Description
11 FG_ACTIVE_0	Displaying the partial frame has been started for async flow (This filed is relevant for debug unit #0)
10–0 V_CNT_OLD_0	The exact row where the async flow has been broken (This filed is relevant for debug unit #0)

45.51.101 IC Configuration Register (IPU_IC_CONF)

This register contains control parameter for IC 3 tasks (pre-processing for encoding, pre-processing for view-finder and post processing).

Address: IPU_IC_CONF is 1E00_0000h base + 20000h offset = 1E02_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CSI_MEM_WR_EN	RWS_EN	IC_KEY_COLOR_EN	IC_GLB_LOC_A	0							PP_ROT_EN	PP_CMB	PP_CSC2	PP_CSC1	PP_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			PRPVF_ROT_EN	PRPVF_CMB	PRPVF_CSC2	PRPVF_CSC1	PRPVF_EN	0					PRPENC_ROT_EN	PRPENC_CSC1	PRPENC_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IC_CONF field descriptions

Field	Description
31 CSI_MEM_WR_EN	CSI direct memory write enable. This bit enables writing data from sensor directly to memory even when a raw sensor is not attached. 0 CSI direct writing to memory is disabled. 1 CSI direct writing to memory is enabled.
30 RWS_EN	Raw sensor enable. This bit indicate if a Raw sensor is attached (Bayer format). This bit is used together with the CSI_MEM_WR_EN bit as follows: CSI_MEM_WR_EN=0, RWS_EN=0 - data is fed from the CSI to the IC for processing; CSI_MEM_WR_EN=1, RWS_EN=0 - data is fed from the CSI to the IC for processing and also for writing to the system memory; CSI_MEM_WR_EN=0, RWS_EN=1 - data is fed from the CSI to the system memory (via the IC) and from the system memory to the IC for processing; CSI_MEM_WR_EN=1, RWS_EN=1 - non-valid configuration.

Table continues on the next page...

IPU_IC_CONF field descriptions (continued)

Field	Description
	0 Raw sensor is not attached. 1 Raw sensor is attached.
29 IC_KEY_ COLOR_EN	Key Color enable. This bit enables the key color feature. 0 Key color is disabled. 1 Key color is enabled.
28 IC_GLB_LOC_A	Global Alpha. This bit select the source of Alpha parameter. 0 Alpha parameter is local. 1 Alpha parameter is global.
27–21 Reserved	This read-only field is reserved and always has the value zero. Reserved
20 PP_ROT_EN	Post-Processing Rotation Task enable. This bit enable Post-Processing Rotation Task. 0 Rotation is disabled. 1 Rotation is enabled.
19 PP_CMB	Post-Processing Task combining enable. This bit enables combining. 0 Combining is disabled. 1 Combining is enabled.
18 PP_CSC2	Post-Processing Task color conversion RGB-->YUV enable. This bit enables YUV-->RGB. Reserved 0 RGB-->YUV is disabled. 1 RGB-->YUV is enabled.
17 PP_CSC1	Post-Processing Task color conversion YUV-->RGB enable. This bit enables YUV-->RGB. 0 YUV-->RGB is disabled. 1 YUV-->RGB is enabled.
16 PP_EN	Post-Processing Task enable. This bit enables the Post-Processing task. 0 Task is disabled. 1 Task is enabled.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 PRPVF_ROT_ EN	Preprocessing Rotation Task for viewfinder enable. This bit enable Preprocessing Rotation Task for viewfinder. 0 Rotation is disabled. 1 Rotation is enabled.
11 PRPVF_CMB	Preprocessing Task for View-Finder combining enable. This bit enables combining. 0 Combining is disabled. 1 Combining is enabled.
10 PRPVF_CSC2	Reserved

Table continues on the next page...

IPU_IC_CONF field descriptions (continued)

Field	Description
9 PRPVF_CSC1	Pre-processing task for view-finder first color conversion enable. This bit enables first color conversion. 0 First color conversion is disabled. 1 First color conversion is enabled.
8 PRPVF_EN	Preprocessing Task for View-Finder enable. This bit enables the View-Finder task. 0 Task is disabled. 1 Task is enabled.
7–3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2 PRPENC_ROT_EN	Preprocessing Rotation Task for encoding enable. This bit enable Preprocessing Rotation Task for encoding. 0 Rotation is disabled. 1 Rotation is enabled.
1 PRPENC_CSC1	Preprocessing Task for encoding color conversion enable. This bit enables color conversion. 0 Color conversion is disabled. 1 Color conversion is enabled.
0 PRPENC_EN	Preprocessing Task for encoding enable. This bit enables the encoding task. 0 Task is disabled. 1 Task is enabled.

45.51.102 IC Preprocessing Encoder Resizing Coefficients Register (IPU_IC_PRP_ENC_RSC)

This register contains the resizing and downsizing parameters for Preprocessing task for encoding.

Address: IPU_IC_PRP_ENC_RSC is 1E00_0000h base + 20004h offset = 1E02_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PRPENC_DS_R_V		PRPENC_RS_R_V													
W																
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PRPENC_DS_R_H		PRPENC_RS_R_H													
W																
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IC_PRP_ENC_RSC field descriptions

Field	Description
31–30 PRPENC_DS_R_V	Preprocessing task for encoding Downsizing vertical Ratio. This field contains the downsizing vertical coefficient of Preprocessing for Encoding.
29–16 PRPENC_RS_R_V	Preprocessing task for encoding Resizing vertical Ratio. This field contains the resizing vertical coefficient of Preprocessing for Encoding. Resizing Ratio is equal to PRPENC_RS_R_V: M Where M = 2^13; SI - input size; SO - output size PRPENC_RS_R_V = floor(M*(SI-1)/(SO-1));
15–14 PRPENC_DS_R_H	Preprocessing task for encoding Downsizing horizontal Ratio. This field contains the downsizing horizontal coefficient of Preprocessing for Encoding. Values: 00 1 01 2 10 4 11 RSV

Table continues on the next page...

IPU_IC_PRP_ENC_RSC field descriptions (continued)

Field	Description
13–0 PRPENC_RS_R_H	Preprocessing task for encoding Resizing horizontal Ratio. This field contains the resizing horizontal coefficient of Preprocessing for Encoding. Resizing Ratio is equal to PRPENC_RS_R_H: M Where M = 2^13; SI - input size; SO - output size PRPENC_RS_R_H = floor(M*(SI-1)/(SO-1));

45.51.103 IC Preprocessing View-Finder Resizing Coefficients Register (IPU_IC_PRP_VF_RSC)

This register contains the resizing and downsizing parameters for preprocessing task for display.

Address: IPU_IC_PRP_VF_RSC is 1E00_0000h base + 20008h offset = 1E02_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PRPVF_DS_R_V		PRPVF_RS_R_V														PRPVF_DS_R_H		PRPVF_RS_R_H													
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IC_PRP_VF_RSC field descriptions

Field	Description
31–30 PRPVF_DS_R_V	Preprocessing task for encoding Downsizing vertical Ratio. This field contains the downsizing vertical coefficient of Preprocessing for View-Finder.
29–16 PRPVF_RS_R_V	Preprocessing task for encoding Resizing vertical Ratio. This field contains the resizing vertical coefficient of Preprocessing for View-Finder. Resizing Ratio is equal to PRPVF_RS_R_V: M Where M = 2^13; SI - input size; SO - output size PRPVF_RS_R_V = floor(M*(SI-1)/(SO-1));
15–14 PRPVF_DS_R_H	Preprocessing task for encoding Downsizing horizontal Ratio. This field contains the downsizing horizontal coefficient of Preprocessing for View-Finder. Values: 00 1 01 2 10 4 11 RSV
13–0 PRPVF_RS_R_H	Preprocessing task for view-finding resizing horizontal ratio. This field contains the resizing horizontal coefficient of preprocessing Task For View-finder. Resizing Ratio is equal to PRPVF_RS_R_H: M Where M = 2^13; SI - input size; SO - output size

Table continues on the next page...

IPU_IC_PRP_VF_RSC field descriptions (continued)

Field	Description
	PRPVF_RS_R_H = floor(M*(SI-1)/(SO-1));

45.51.104 IC Postprocessing Encoder Resizing Coefficients Register (IPU_IC_PP_RSC)

This register contains the resizing and downsizing parameters for Post-Processing task for display.

Address: IPU_IC_PP_RSC is 1E00_0000h base + 2000Ch offset = 1E02_000Ch

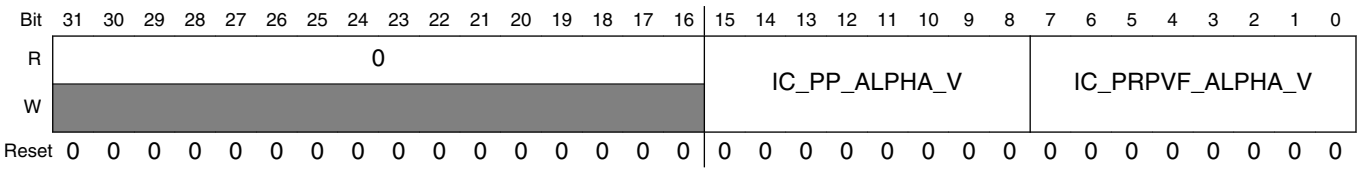
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PP_DS_R_V		PP_RS_R_V														PP_DS_R_H		PP_RS_R_H													
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IC_PP_RSC field descriptions

Field	Description
31–30 PP_DS_R_V	Post-Processing task Downsizing vertical Ratio. This field contains the downsizing vertical coefficient of Post-Processing.
29–16 PP_RS_R_V	Post-Processing task Resizing vertical Ratio. This field contains the resizing vertical coefficient of Post-Processing. Resizing Ratio is equal to PP_RS_R_V: M Where M = 2^13; SI - input size; SO - output size PP_RS_R_V = floor(M*(SI-1)/(SO-1));
15–14 PP_DS_R_H	Post-Processing task Downsizing horizontal Ratio. This field contains the downsizing horizontal coefficient of Post-Processing. 00 1 01 2 10 4 11 RSV
13–0 PP_RS_R_H	Post-Processing task Resizing horizontal Ratio. This field contains the resizing horizontal coefficient of Post-Processing. Resizing Ratio is equal to PP_RS_R_H: M Where M = 2^13; SI - input size; SO - output size PP_RS_R_H = floor(M*(SI-1)/(SO-1));

45.51.105 IC Combining Parameters Register 1 (IPU_IC_CMBP_1)

Address: IPU_IC_CMBP_1 is 1E00_0000h base + 20010h offset = 1E02_0010h

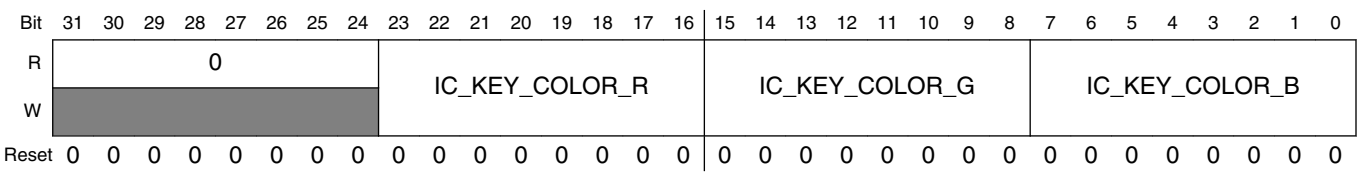


IPU_IC_CMBP_1 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 IC_PP_ALPHA_V	Post-Processing task Global Alpha. This field contains the Global Alpha value of Post-Processing.
7–0 IC_PRPVF_ALPHA_V	Preprocessing task for viewfinder Global Alpha. This field contains the Global Alpha value of Preprocessing for viewfinder.

45.51.106 IC Combining Parameters Register 2 (IPU_IC_CMBP_2)

Address: IPU_IC_CMBP_2 is 1E00_0000h base + 20014h offset = 1E02_0014h

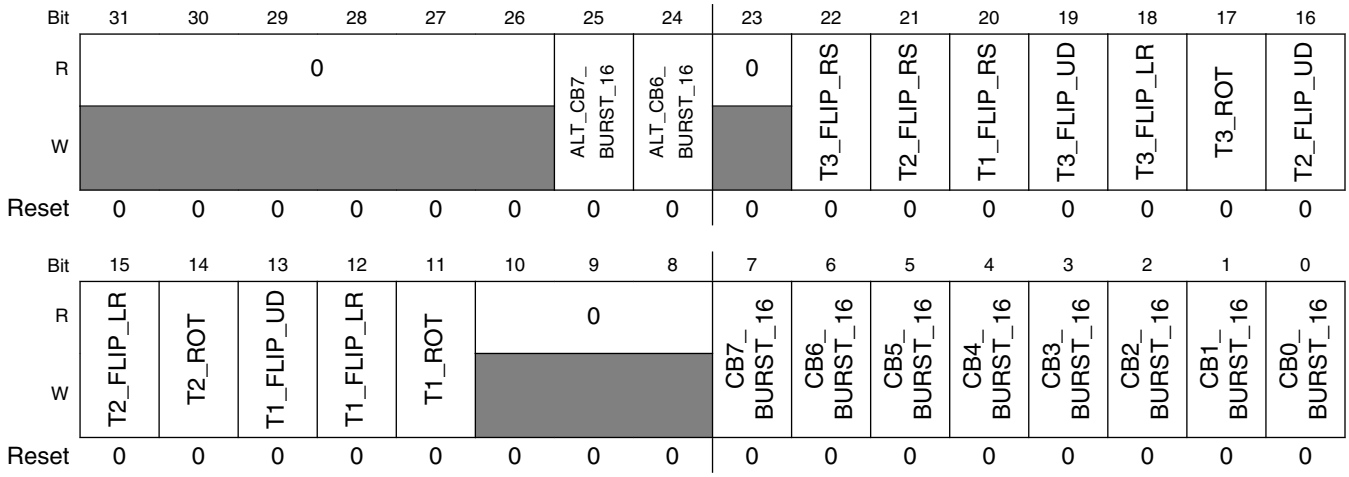


IPU_IC_CMBP_2 field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved.
23–16 IC_KEY_COLOR_R	Key Color Red.
15–8 IC_KEY_COLOR_G	Key Color Green.
7–0 IC_KEY_COLOR_B	Key Color Blue.

45.51.107 IC IDMAC Parameters 1 Register (IPU_IC_IDMAC_1)

Address: IPU_IC_IDMAC_1 is 1E00_0000h base + 20018h offset = 1E02_0018h



IPU_IC_IDMAC_1 field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value zero. Reserved
25 ALT_CB7_ BURST_16	Reserved
24 ALT_CB6_ BURST_16	Reserved
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22 T3_FLIP_RS	LEFT/RIGHT flip for Post Processing (PP) task; his bit affect the flipping done on the resizing unit. The value of this field must be identical to the corresponding channel's HF parameter in the IDMAC's CPMEM 1 horizontal flip enabled 0 no flip
21 T2_FLIP_RS	LEFT/RIGHT flip for View Finder (VF) task; this bit affect the flipping done on the resizing unit. The value of this field must be identical to the corresponding channel's HF parameter in the IDMAC's CPMEM 1 horizontal flip enabled 0 no flip
20 T1_FLIP_RS	LEFT/RIGHT flip for Encoding (ENC) task; this bit affect the flipping done on the resizing unit. The value of this field must be identical to the corresponding channel's HF parameter in the IDMAC's CPMEM

Table continues on the next page...

IPU_IC_IDMAC_1 field descriptions (continued)

Field	Description
	1 horizontal flip enabled 0 no flip
19 T3_FLIP_UD	UP/DOWN flip for Post Processing (PP) task The value of this field must be identical to the corresponding channel's VF parameters in the IDMAC's CPMEM 1 Vertical flip enable 0 no flip
18 T3_FLIP_LR	LEFT/RIGHT flip for Post Processing (PP) task; this bit affect the flipping done on the rotation unit The value of this field must be identical to the corresponding channel's HF parameter in the IDMAC's CPMEM 1 horizontal flip enabled 0 no flip
17 T3_ROT	Rotation for Post Processing (PP) task The value of this field must be identical to the corresponding channel's ROT parameters in the IDMAC's CPMEM 1 90 degree rotation clockwise 0 no rotation
16 T2_FLIP_UD	UP/DOWN flip for View Finder (VF) task The value of this field must be identical to the corresponding channel's VF parameters in the IDMAC's CPMEM 1 Vertical flip enable 0 no flip
15 T2_FLIP_LR	LEFT/RIGHT flip for View Finder (VF) task; this bit affect the flipping done on the rotation unit The value of this field must be identical to the corresponding channel's HF parameter in the IDMAC's CPMEM 1 horizontal flip enabled 0 no flip
14 T2_ROT	Rotation for View Finder (VF) task The value of this field must be identical to the corresponding channel's ROT parameters in the IDMAC's CPMEM 1 90 degree rotation clockwise 0 no rotation
13 T1_FLIP_UD	UP/DOWN flip for Encoding (ENC) task The value of this field must be identical to the corresponding channel's VF parameters in the IDMAC's CPMEM 1 Vertical flip enable 0 no flip
12 T1_FLIP_LR	LEFT/RIGHT flip for Encoding (ENC) task; this bit affect the flipping done on the rotation unit

Table continues on the next page...

IPU_IC_IDMAC_1 field descriptions (continued)

Field	Description
	The value of this field must be identical to the corresponding channel's HF parameter in the IDMAC's CPMEM 1 horizontal flip enabled 0 no flip
11 T1_ROT	Rotation for Encoding (ENC) task The value of this field must be identical to the corresponding channel's ROT parameters in the IDMAC's CPMEM 1 90 degree rotation clockwise 0 no rotation
10–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7 CB7_BURST_16	This bit defines the number of active cycles within a burst (burst size) coming from the IDMAC for IC's CB7 For pixel data the number of pixels should match the NPB[6:2] value on the IDMAC's CPMEM 0 Burst size is 8 pixels; The Matching NPB[6:2] should be 00111 1 Burst size is 16 pixels; The Matching NPB[6:2] should be 01111
6 CB6_BURST_16	This bit defines the number of active cycles within a burst (burst size) coming from the IDMAC for IC's CB6 For pixel data the number of pixels should match the NPB[6:2] value on the IDMAC's CPMEM 0 Burst size is 8 pixels; The Matching NPB[6:2] should be 00111 1 Burst size is 16 pixels; The Matching NPB[6:2] should be 01111
5 CB5_BURST_16	This bit defines the number of active cycles within a burst (burst size) coming from the IDMAC for IC's CB5 For pixel data the number of pixels should match the NPB[6:2] value on the IDMAC's CPMEM 0 Burst size is 8 pixels; The Matching NPB[6:2] should be 00111 1 Burst size is 16 pixels; The Matching NPB[6:2] should be 01111
4 CB4_BURST_16	This bit defines the number of active cycles within a burst (burst size) coming from the IDMAC for IC's CB4 For pixel data the number of pixels should match the NPB[6:2] value on the IDMAC's CPMEM 0 Burst size is 8 pixels; The Matching NPB[6:2] should be 00111 1 Burst size is 16 pixels; The Matching NPB[6:2] should be 01111
3 CB3_BURST_16	This bit defines the number of active cycles within a burst (burst size) coming from the IDMAC for IC's CB3 For pixel data the number of pixels should match the NPB[6:2] value on the IDMAC's CPMEM 0 Burst size is 8 pixels; The Matching NPB[6:2] should be 00111 1 Burst size is 16 pixels; The Matching NPB[6:2] should be 01111
2 CB2_BURST_16	This bit defines the number of active cycles within a burst (burst size) coming from the IDMAC for IC's CB2 For pixel data the number of pixels should match the NPB[6:2] value on the IDMAC's CPMEM

Table continues on the next page...

IPU_IC_IDMAC_1 field descriptions (continued)

Field	Description
	0 Burst size is 8 pixels; The Matching NPB[6:2] should be 00111 1 Burst size is 16 pixels; The Matching NPB[6:2] should be 01111
1 CB1_BURST_16	This bit defines the number of active cycles within a burst (burst size) coming from the IDMAC for IC's CB1 For pixel data the number of pixels should match the NPB[6:2] value on the IDMAC's CPMEM 0 Burst size is 8 pixels; The Matching NPB[6:2] should be 00111 1 Burst size is 16 pixels; The Matching NPB[6:2] should be 01111
0 CB0_BURST_16	This bit defines the number of active cycles within a burst (burst size) coming from the IDMAC for IC's CB0 For pixel data the number of pixels should match the NPB[6:2] value on the IDMAC's CPMEM 0 Burst size is 8 pixels; The Matching NPB[6:2] should be 00111 1 Burst size is 16 pixels; The Matching NPB[6:2] should be 01111

45.51.108 IC IDMAC Parameters 2 Register (IPU_IC_IDMAC_2)

Address: IPU_IC_IDMAC_2 is 1E00_0000h base + 2001Ch offset = 1E02_001Ch

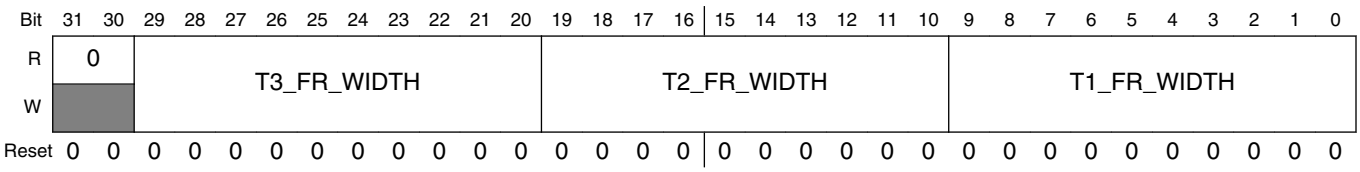
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		T3_FR_HEIGHT										T2_FR_HEIGHT										T1_FR_HEIGHT									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IC_IDMAC_2 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved
29–20 T3_FR_HEIGHT	Frame Height for Post Processing (PP) task The value of this field must be identical to the corresponding FH channel's parameters in the IDMAC's CPMEM. This parameter refers to the output's size -1
19–10 T2_FR_HEIGHT	Frame Height for View Finder (VF) task The value of this field must be identical to the corresponding FH channel's parameters in the IDMAC's CPMEM. This parameter refers to the output's size -1
9–0 T1_FR_HEIGHT	Frame Height for Encoding (ENC) task The value of this field must be identical to corresponding FH channel's parameters in the IDMAC's CPMEM. This parameter refers to the output's size -1

45.51.109 IC IDMAC Parameters 3Register (IPU_IC_IDMAC_3)

Address: IPU_IC_IDMAC_3 is 1E00_0000h base + 20020h offset = 1E02_0020h

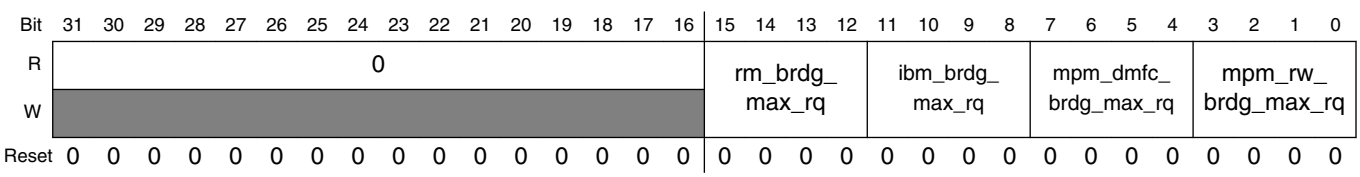


IPU_IC_IDMAC_3 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved
29–20 T3_FR_WIDTH	Frame Width for Post Processing (PP) task The value of this field must be identical to the corresponding FW channel's parameters in the IDMAC's CPMEM. This parameter refers to the output's size -1
19–10 T2_FR_WIDTH	Frame Width for View Finder (VF) task The value of this field must be identical to the corresponding FW channel's parameters in the IDMAC's CPMEM. This parameter refers to the output's size -1
9–0 T1_FR_WIDTH	Frame Width for Encoding (ENC) task The value of this field must be identical to corresponding FW channel's parameters in the IDMAC's CPMEM. This parameter refers to the output's size -1

45.51.110 IC IDMAC Parameters 4 Register (IPU_IC_IDMAC_4)

Address: IPU_IC_IDMAC_4 is 1E00_0000h base + 20024h offset = 1E02_0024h



IPU_IC_IDMAC_4 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–12 rm_brdg_max_rq	RM memory Bridge Max Requests 0000 Feature is disabled 0001 Max request is 1 1111 Max request are15

Table continues on the next page...

IPU_IC_IDMAC_4 field descriptions (continued)

Field	Description
11–8 ibm_brdg_max_rq	IBM memory Bridge Max Requests 0000 Feature is disabled 0001 Max request is 1 1111 Max request are15
7–4 mpm_dmfc_brdg_max_rq	MPM memory Bridge Max Requests for the IC DMFC interface 0000 Feature is disabled 0001 Max request is 1 1111 Max request are15
3–0 mpm_rw_brdg_max_rq	MPM memory Bridge Max Requests between MPM's read and writes 0000 Feature is disabled 0001 Max request is 1 1111 Max request are15

45.51.111 CSI0 Sensor Configuration Register (IPU_CSI0_SENS_CONF)

Address: IPU_CSI0_SENS_CONF is 1E00_0000h base + 30000h offset = 1E03_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CSI0_DATA_EN_POL	0	CSI0_FORCE_EOF	CSI0_JPEG_MODE	CSI0_JPEG8_EN	CSI0_DATA_DEST			CSI0_DIV_RATIO							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CSI0_EXT_VSYNC	CSI0_DATA_WIDTH				CSI0_SENS_DATA_FORMAT			CSI0_PACK_TIGHT	CSI0_SENS_PRTCL			CSI0_SENS_PIX_CLK_POL	CSI0_DATA_POL	CSI0_HSYNC_POL	CSI0_VSYNC_POL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI0_SENS_CONF field descriptions

Field	Description
31 CSI0_DATA_EN_POL	Invert IPP_IND_SENSB_DATA_EN input. This bit selects the polarity of IPP_IND_SENSB_DATA_EN signal. 0 IPP_IND_SENSB_DATA_EN is directly applied to internal circuitry. 1 IPP_IND_SENSB_DATA_EN is inverted before applied to internal circuitry.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.

Table continues on the next page...

IPU_CSI0_SENS_CONF field descriptions (continued)

Field	Description
29 CSI0_FORCE_EOF	Force End of frame This is a self clear bit allowing the user to force an End-of-frame event; This bit can be used in cases where the frame sent by the sensor was not completed. 1 force end of frame 0 no action
28 CSI0_JPEG_MODE	JPEG Mode - this bit defines the mode of the control signals when working in JPEG mode 1 The data is valid as long as HSYNC and VSYNC signals are active; HSYNC is valid for single frame 0 The frame starts withe the assertion of VSYNC. The frame ends on the next VSYNC or by setting the **CSI0_FORCE_EOF bit.**
27 CSI0_JPEG8_EN	JPEG8 enable bit 1 JPEG8 detection is enabled 0 JPEG8 is disabled
26–24 CSI0_DATA_DEST	These bits enable the destination of the data coming from the CSI. CSI0_DATA_DEST[0] - Reserved CSI0_DATA_DEST[1] - destination is IC CSI0_DATA_DEST[2] - destination is IDMAC via SMFC
23–16 CSI0_DIV_RATIO	DIV Ratio Clock division ratio minus 1. This field defines the division ratio of HSP_CLK into SENSB_MCLK: SENSB_MCLK rate = HSP_CLK rate /(DIV_RATIO+1)
15 CSI0_EXT_VSYNC	External VSYNC enable. This bits select between external and internal VSYNC. 0 Internal VSYNC mode. 1 External VSYNC mode.
14–11 CSI0_DATA_WIDTH	Data width. This field defines the number of bits per color. Values: 0000 4 bits per color 0000 Reserved 0001 8 bits per color 0010 9 bits per color 0010 Reserved 0011 10 bits per color 0100 11 bits per color 0100 Reserved 0101 12 bits per color 0101 Reserved 0110 13 bits per color 0110 Reserved 0111 14 bits per color 0111 Reserved 1000 15 bits per color 1000 Reserved 1001 16 bits per color

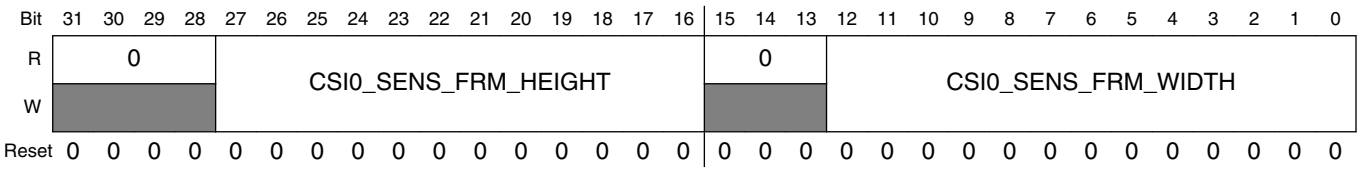
Table continues on the next page...

IPU_CSI0_SENS_CONF field descriptions (continued)

Field	Description
10–8 CSI0_SENS_ DATA_FORMAT	Data format from the sensor. This field defines the data format for the input of the CSI sensor. Values: 000 full RGB or YUV444 001 YUV422 (YUYV...) 010 YUV422 (UYVY...) 011 Bayer or Generic data 100 RGB565 101 RGB555 110 RGB444 111 JPEG
7 CSI0_PACK_ TIGHT	CSI0 Pack Tight When the data format is YUV or RGB and the component's width is 9-16 bits, it can be sent to the memory in 2 different ways. 1 Three 10 bits components are packed into a 32 bit word. Color extension/reduction is performed 0 Each component is written as a 16 bit word where the MSB is written to bit #15, color extension is done for the remaining least significant bits.
6–4 CSI0_SENS_ PRTCL	Sensor Protocol. This bit defines the Sensor timing/data mode protocol. Values: 000 Gated clock mode 001 Non-gated clock mode 010 CCIR progressive mode (BT.656) 011 CCIR interlaced mode (BT.656) 100 CCIR progressive (BT.1120 DDR mode: data arrives on every edge of the clock) 101 CCIR progressive (BT.1120 SDR mode: data arrives only on the positive edge of the clock) 110 CCIR interlaced mode (BT.1120 DDR mode: data arrives on every edge of the clock) 111 CCIR interlaced mode (BT.1120 SDR mode: data arrives only on the positive edge of the clock)
3 CSI0_SENS_ PIX_CLK_POL	Invert Pixel clock input. This bit selects the polarity of pixel clock. 0 pixel clock is directly applied to internal circuitry. 1 pixel clock is inverted before applied to internal circuitry.
2 CSI0_DATA_ POL	Invert data input. This bit selects the polarity of data input. 0 data lines are directly applied to internal circuitry. 1 data lines are inverted before applied to internal circuitry.
1 CSI0_HSYNC_ POL	Invert IPP_IND_SENSB_HSYNC input. This bit selects the polarity of IPP_IND_SENSB_HSYNC signal. 0 IPP_IND_SENSB_HSYNC is directly applied to internal circuitry. 1 IPP_IND_SENSB_HSYNC is inverted before applied to internal circuitry.
0 CSI0_VSYNC_ POL	Invert IPP_IND_SENSB_VSYNC input. This bit selects the polarity of IPP_IND_SENSB_VSYNC signal. 0 IPP_IND_SENSB_VSYNC is not inverted before applied to internal circuitry. 1 IPP_IND_SENSB_VSYNC is inverted before applied to internal circuitry.

45.51.112 CSI0 Sense Frame Size Register (IPU_CSI0_SENS_FRM_SIZE)

Address: IPU_CSI0_SENS_FRM_SIZE is 1E00_0000h base + 30004h offset = 1E03_0004h

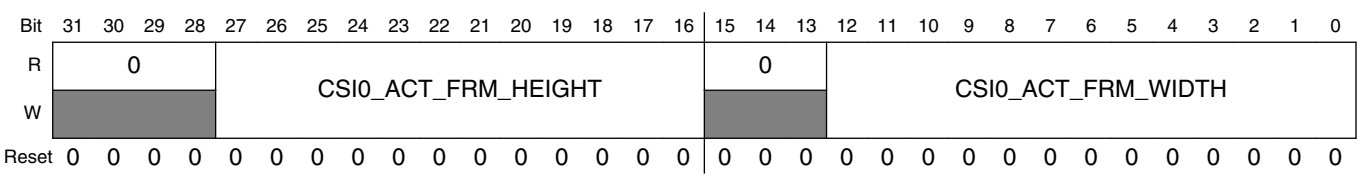


IPU_CSI0_SENS_FRM_SIZE field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
27–16 CSI0_SENS_ FRM_HEIGHT	Sensor frame height minus 1. This field defines the sensor frame rows number minus 1.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
12–0 CSI0_SENS_ FRM_WIDTH	Sensor frame width minus 1. This field defines the sensor frame column number minus 1.

45.51.113 CSI0 Actual Frame Size Register (IPU_CSI0_ACT_FRM_SIZE)

Address: IPU_CSI0_ACT_FRM_SIZE is 1E00_0000h base + 30008h offset = 1E03_0008h



IPU_CSI0_ACT_FRM_SIZE field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
27–16 CSI0_ACT_ FRM_HEIGHT	Actual frame height minus 1. This field defines the CSI output frame rows number minus 1.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.

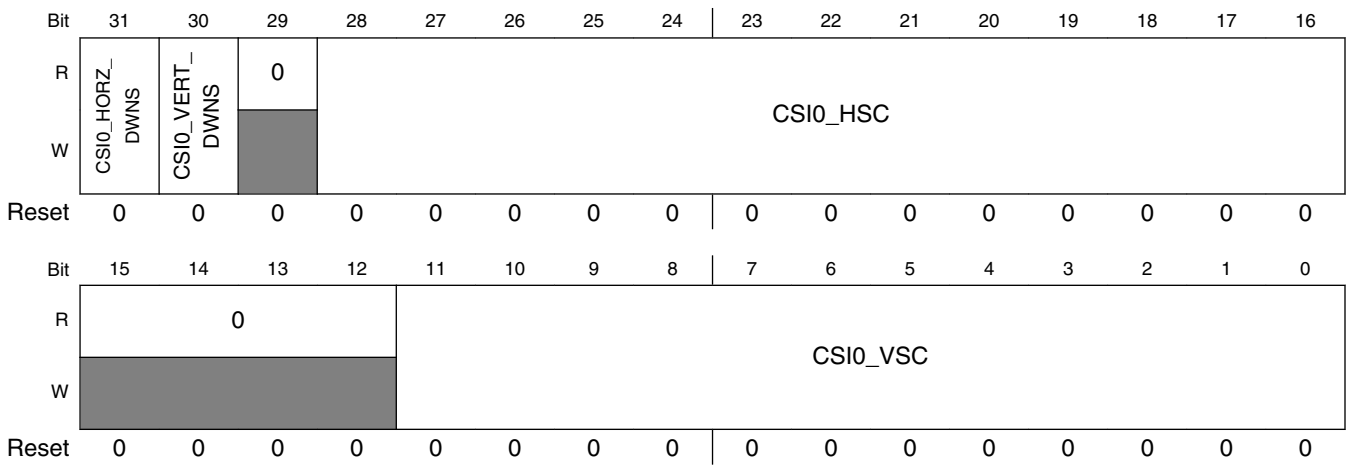
Table continues on the next page...

IPU_CSI0_ACT_FRM_SIZE field descriptions (continued)

Field	Description
12–0 CSI0_ACT_ FRM_WIDTH	Actual frame width minus 1. This field defines the CSI output frame columns number minus 1.

45.51.114 CSI0 Output Control Register (IPU_CSI0_OUT_FRM_CTRL)

Address: IPU_CSI0_OUT_FRM_CTRL is 1E00_0000h base + 3000Ch offset = 1E03_000Ch

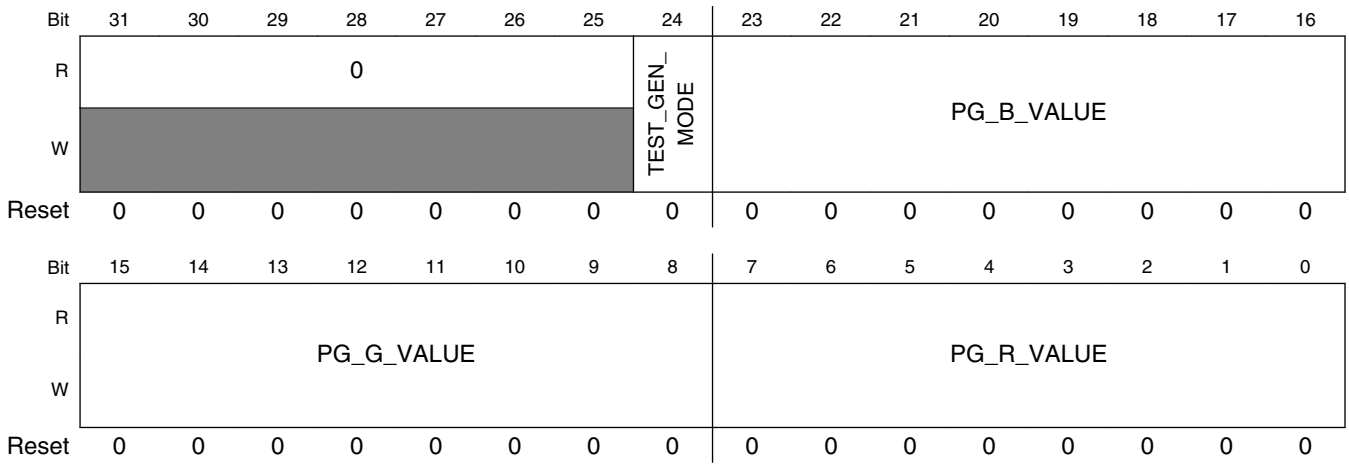


IPU_CSI0_OUT_FRM_CTRL field descriptions

Field	Description
31 CSI0_HORZ_ DWNS	Enable horizontal downsizing (decimation) by 2. 0 Downsizing disabled 1 Downsizing enabled
30 CSI0_VERT_ DWNS	Enable vertical downsizing (decimation) by 2. 0 Downsizing disabled 1 Downsizing enabled
29 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
28–16 CSI0_HSC	Horizontal skip. This field defines the number of columns to skip. In Interlaced mode this number refers to the number of lines per field.
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
11–0 CSI0_VSC	Vertical skip. This field defines the number of rows to skip.

45.51.115 CSIO Test Control Register (IPU_CSI0_TST_CTRL)

Address: IPU_CSI0_TST_CTRL is 1E00_0000h base + 30010h offset = 1E03_0010h

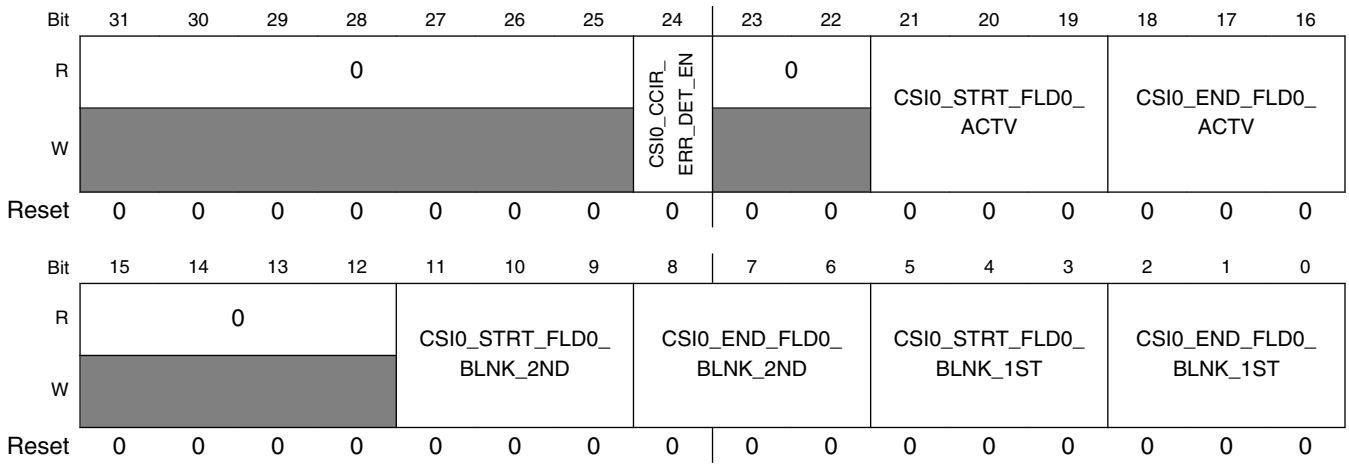


IPU_CSI0_TST_CTRL field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
24 TEST_GEN_ MODE	Test generator mode. This bit activates the signal generation. 0 Test signal generator is inactive. 1 Test signal generator is active.
23–16 PG_B_VALUE	Pattern generator B value. This field selects the B value for the generated pattern of even pixel.
15–8 PG_G_VALUE	Pattern generator G value. This field selects the G value for the generated pattern of even pixel.
7–0 PG_R_VALUE	Pattern generator R value. This field selects the R value for the generated pattern of even pixel.

45.51.116 CSIO CCIR Code Register 1 (IPU_CSI0_CCIR_CODE_1)

Address: IPU_CSI0_CCIR_CODE_1 is 1E00_0000h base + 30014h offset = 1E03_0014h



IPU_CSI0_CCIR_CODE_1 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
24 CSI0_CCIR_ERR_DET_EN	Enable error detection and correction for CCIR interlaced mode with protection bit. 0 Error detection and correction is disabled. 1 Error detection and correction is enabled.
23–22 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
21–19 CSI0_STRT_FLD0_ACTV	Start of field 0 active line command (interlaces mode). (In progressive mode, start of active line command mode).
18–16 CSI0_END_FLD0_ACTV	End of field 0 active line command (interlaces mode). (In progressive mode, end of active line command mode).
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
11–9 CSI0_STRT_FLD0_BLNK_2ND	Start of field 0 second blanking line command (interlaces mode). (In progressive mode this field is ignored).
8–6 CSI0_END_FLD0_BLNK_2ND	End of field 0 second blanking line command (interlaces mode). (In progressive mode this field is ignored).

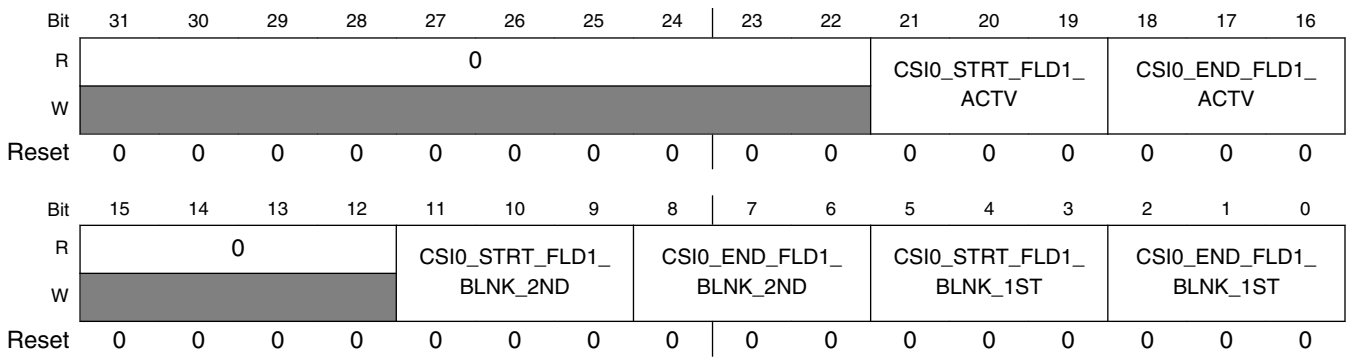
Table continues on the next page...

IPU_CSI0_CCIR_CODE_1 field descriptions (continued)

Field	Description
5–3 CSI0_STRT_FLD0_BLNK_1ST	Start of field 0 first blanking line command (interlaces mode). (In progressive mode this field indicates start of blanking line command).
2–0 CSI0_END_FLD0_BLNK_1ST	End of field 0 first blanking line command (interlaces mode). (In progressive mode this field is ignored).

45.51.117 CSIO CCIR Code Register 2 (IPU_CSI0_CCIR_CODE_2)

Address: IPU_CSI0_CCIR_CODE_2 is 1E00_0000h base + 30018h offset = 1E03_0018h



IPU_CSI0_CCIR_CODE_2 field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
21–19 CSI0_STRT_FLD1_ACTV	Start of field 1 active line command (interlaces mode). (In progressive mode this field is ignored).
18–16 CSI0_END_FLD1_ACTV	End of field 1 active line command (interlaces mode). (In progressive mode this field is ignored).
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
11–9 CSI0_STRT_FLD1_BLNK_2ND	Start of field 1 second blanking line command (interlaces mode). (In progressive mode this field is ignored).
8–6 CSI0_END_FLD1_BLNK_2ND	End of field 1 second blanking line command (interlaces mode). (In progressive mode this field is ignored).

Table continues on the next page...

IPU_CSI0_CCIR_CODE_2 field descriptions (continued)

Field	Description
5–3 CSI0_STRT_FLD1_BLNK_1ST	Start of field 1 first blanking line command (interlaces mode). (In progressive mode this field is ignored).
2–0 CSI0_END_FLD1_BLNK_1ST	End of field 1 first blanking line command (interlaces mode). (In progressive mode this field is ignored).

45.51.118 CSIO CCIR Code Register 3 (IPU_CSI0_CCIR_CODE_3)

Address: IPU_CSI0_CCIR_CODE_3 is 1E00_0000h base + 3001Ch offset = 1E03_001Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CSI0_CCIR_PRECOM																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI0_CCIR_CODE_3 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
29–0 CSI0_CCIR_PRECOM	CCIR pre command. This field defines the sequence which comes before the CCIR command. For BT.656 the code should be written to bits [23:0] while bits [29:24] are ignored (3X8bit) For BT.1120 the code should be written to bits [29:0] (3X10bit)

45.51.119 CSI0 Data Identifier Register (IPU_CSI0_DI)

Address: IPU_CSI0_DI is 1E00_0000h base + 30020h offset = 1E03_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CSI0_MIPI_DI3								CSI0_MIPI_DI2								CSI0_MIPI_DI1								CSI0_MIPI_DI0							
W																																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

IPU_CSI0_DI field descriptions

Field	Description
31–24 CSI0_MIPI_DI3	Reserved

Table continues on the next page...

IPU_CSI0_DI field descriptions (continued)

Field	Description
23–16 CSI0_MIPI_DI2	Reserved
15–8 CSI0_MIPI_DI1	Reserved
7–0 CSI0_MIPI_DI0	Reserved

45.51.120 CSI0 SKIP Register (IPU_CSI0_SKIP)

This register controls the frame skipping supported between CSI0 and the SMFC.

Address: IPU_CSI0_SKIP is 1E00_0000h base + 30024h offset = 1E03_0024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								CSI0_SKIP_ISP					CSI0_MAX_RATIO_SKIP_ISP		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						CSI0_ID_2_SKIP		CSI0_SKIP_SMFC					CSI0_MAX_RATIO_SKIP_SMFC		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI0_SKIP field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23–19 CSI0_SKIP_ISP	Reserved
18–16 CSI0_MAX_RATIO_SKIP_ISP	Reserved
15–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–8 CSI0_ID_2_SKIP	Reserved
7–3 CSI0_SKIP_SMFC	CSI0 SKIP SMFC These 5 bits define the skipping pattern of the frames send to the SMFC. Skipping is done for a set of frames. The number of frames in a set is defined at CSI0_MAX_RATIO_SKIP_SMFC. when CSI0_MAX_RATIO_SKIP_SMFC = 1 => CSI0_SKIP_SMFC[0] is used; other bits are ignored

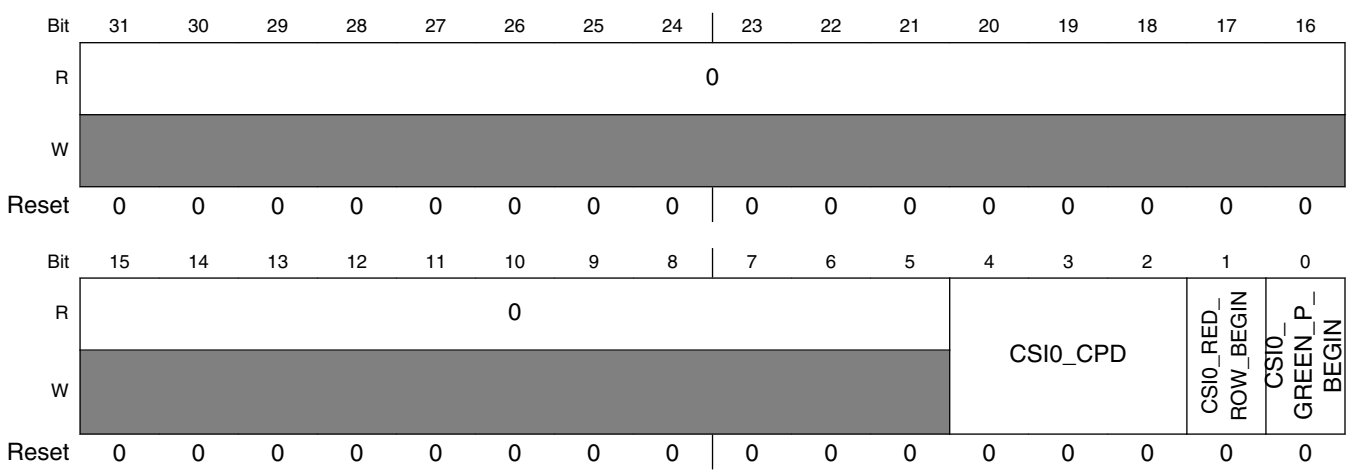
Table continues on the next page...

IPU_CSI0_SKIP field descriptions (continued)

Field	Description
	when CSI0_MAX_RATIO_SKIP_SMFC = 2 => CSI0_SKIP_SMFC[1:0] are used; other bits are ignored when CSI0_MAX_RATIO_SKIP_SMFC = 3 => CSI0_SKIP_SMFC[2:0] are used; other bits are ignored when CSI0_MAX_RATIO_SKIP_SMFC = 4 => CSI0_SKIP_SMFC[3:0] are used; other bits are ignored when CSI0_MAX_RATIO_SKIP_SMFC = 5 => CSI0_SKIP_SMFC[4:0] are used; Setting bit #n of CSI0_SKIP_SMFC means that the #n frame in the set is skipped. For example: if CSI0_MAX_RATIO_SKIP_SMFC = 4 and CSI0_SKIP_SMFC = 11010 Frames #0 & Frame #2 will not be skipped as bit0 and bit2 are cleared Frames #1 & Frame #3 will be skipped as bit1 and bit3 are set bit #4 is ignored as CSI0_MAX_RATIO_SKIP_SMFC is set to 4
2–0 CSI0_MAX_ RATIO_SKIP_ SMFC	CSI0 Maximum Ratio Skip for SMFC These bits define the number of frames in a skipping set. The skipping number is equal to CSI0_MAX_RATIO_SKIP_SMFC+1; The maximum value of this bits is 5. When set to 0 the skipping is disabled.

45.51.121 CSI0 Compander Control Register (IPU_CSIO_CPD_CTRL)

Address: IPU_CSIO_CPD_CTRL is 1E00_0000h base + 30028h offset = 1E03_0028h



IPU_CSIO_CPD_CTRL field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
4–2 CSI0_CPD	CSI0_CPD These bits enable the compander in the path to different destination. CSI0_CPD[0] - CSI0_CPD[1] - Enable for the compander for data sent to the IC

Table continues on the next page...

IPU_CSIO_CPD_CTRL field descriptions (continued)

Field	Description
	CSI0_CPD[2] - Enable for the compander for data sent to the IDMAC via SMFC If all the 3 bits are zero the compander is disabled Reserved
1 CSI0_RED_ROW_BEGIN	Color of first row in the frame. Reserved 0 First row in the frame is GBGB. 1 First row in the frame is GRGR.
0 CSI0_GREEN_P_BEGIN	Color of first component in the frame. Reserved 0 First component in the frame is blue or red, depending from RED_ROW bit. 1 First component in the frame is green

45.51.122 CSI0 Red Component Compander Constants Register <i> (IPU_CSIO_CPD_RC_i)

These registers contain CONSTANT <i> parameters used for companding of red component.

Address: IPU_CSIO_CPD_RC_i is 1E00_0000h base + 3002Ch offset = 1E03_002Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CSI0_CPD_RC_<2*i+1>									0							CSI0_CPD_RC_<2*i>							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSIO_CPD_RC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 CSI0_CPD_RC_<2*i+1>	CONSTANT <2*i+1> parameter of Compander, Red component. Reserved
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 CSI0_CPD_RC_<2*i>	CONSTANT <2*i> parameter of Compander, Red component. Reserved

45.51.123 CSI0 Red Component Compander SLOPE Register <i> (IPU_CSIO_CPD_RS_i)

These registers contain SLOPE <i> parameters used for companding of red component.

Address: IPU_CSIO_CPD_RS_i is 1E00_0000h base + 3004Ch offset = 1E03_004Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSI0_CPD_RS<4*i+3>								CSI0_CPD_RS<4*i+2>								CSI0_CPD_RS<4*i+1>								CSI0_CPD_RS<4*i>							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSIO_CPD_RS_i field descriptions

Field	Description
31–24 CSI0_CPD_ RS<4*i+3>	Reserved
23–16 CSI0_CPD_ RS<4*i+2>	Reserved
15–8 CSI0_CPD_ RS<4*i+1>	Reserved
7–0 CSI0_CPD_ RS<4*i>	Reserved

45.51.124 CSI0 GR Component Compander Constants Register <i> (IPU_CSIO_CPD_GRC_i)

These registers contain CONSTANTi parameters used for companding of green components in GRGR rows.

Address: IPU_CSIO_CPD_GRC_i is 1E00_0000h base + 3005Ch offset = 1E03_005Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CSI0_CPD_GRC<2*i+1>									0							CSI0_CPD_GRC<2*i>							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSIO_CPD_GRC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_CSIO_CPD_GRC_i field descriptions (continued)

Field	Description
24–16 CSI0_CPD_ GRC<2*i+1>	Reserved
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 CSI0_CPD_ GRC<2*i>	Reserved

45.51.125 CSI0 GR Component Compander SLOPE Register <i> (IPU_CSIO_CPD_GRS_i)

These registers contain SLOPEi parameters used for companding of green components in GRGR rows.

Address: IPU_CSIO_CPD_GRS_i is 1E00_0000h base + 3007Ch offset = 1E03_007Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSI0_CPD_GRS<4*i+3>								CSI0_CPD_GRS<4*i+2>								CSI0_CPD_GRS<4*i+1>								CSI0_CPD_GRS<4*i>							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSIO_CPD_GRS_i field descriptions

Field	Description
31–24 CSI0_CPD_ GRS<4*i+3>	Reserved
23–16 CSI0_CPD_ GRS<4*i+2>	Reserved
15–8 CSI0_CPD_ GRS<4*i+1>	Reserved
7–0 CSI0_CPD_ GRS<4*i>	Reserved

45.51.126 CSI0 GB Component Compander Constants Register <i> (IPU_CSIO_CPD_GBC_i)

These registers contain CONSTANTi parameters used for companding of green components in GBGB rows.

Address: IPU_CSIO_CPD_GBC_i is 1E00_0000h base + 3008Ch offset = 1E03_008Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CSI0_CPD_GBC<2*i+1>									0							CSI0_CPD_GBC<2*i>							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSIO_CPD_GBC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 CSI0_CPD_ GBC<2*i+1>	Reserved
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 CSI0_CPD_ GBC<2*i>	Reserved

45.51.127 CSI0 GB Component Compander SLOPE Register <i> (IPU_CSIO_CPD_GBS_i)

These registers contain SLOPEi parameters used for companding of green components in GBGB rows.

Address: IPU_CSIO_CPD_GBS_i is 1E00_0000h base + 300ACh offset = 1E03_00ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CSI0_CPD_GBS<4*i+3>								CSI0_CPD_GBS<4*i+2>								CSI0_CPD_GBS<4*i+1>								CSI0_CPD_GBS<4*i>							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSIO_CPD_GBS_i field descriptions

Field	Description
31–24 CSI0_CPD_ GBS<4*i+3>	Reserved
23–16 CSI0_CPD_ GBS<4*i+2>	Reserved
15–8 CSI0_CPD_ GBS<4*i+1>	Reserved
7–0 CSI0_CPD_ GBS<4*i>	Reserved

45.51.128 CSI0 Blue Component Compander Constants Register <i> (IPU_CSIO_CPD_BC_i)

These registers contain CONSTANTi parameters used for companding of blue component.

Address: IPU_CSIO_CPD_BC_i is 1E00_0000h base + 300BCh offset = 1E03_00BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CSI0_CPD_BC<2*i+1>									0							CSI0_CPD_BC<2*i>							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSIO_CPD_BC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 CSI0_CPD_ BC<2*i+1>	Reserved
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 CSI0_CPD_ BC<2*i>	Reserved

45.51.129 CSI0 Blue Component Compander SLOPE Register <i> (IPU_CSIO_CPD_BS_i)

These registers contain SLOPEi parameters used for companding of red component.

Address: IPU_CSIO_CPD_BS_i is 1E00_0000h base + 300DCh offset = 1E03_00DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSI0_CPD_BS<4*i+3>								CSI0_CPD_BS<4*i+2>								CSI0_CPD_BS<4*i+1>								CSI0_CPD_BS<4*i>							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSIO_CPD_BS_i field descriptions

Field	Description
31–24 CSI0_CPD_ BS<4*i+3>	Reserved
23–16 CSI0_CPD_ BS<4*i+2>	Reserved
15–8 CSI0_CPD_ BS<4*i+1>	Reserved
7–0 CSI0_CPD_ BS<4*i>	Reserved I

45.51.130 CSI0 Compander Offset Register 1 (IPU_CSI0_CPD_OFFSET1)

These registers contain Offset parameters used for companding.

Address: IPU_CSI0_CPD_OFFSET1 is 1E00_0000h base + 300ECh offset = 1E03_00ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		CSI0_CPD_B_OFFSET										CSI0_GB_OFFSET										CSI0_GR_OFFSET									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI0_CPD_OFFSET1 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

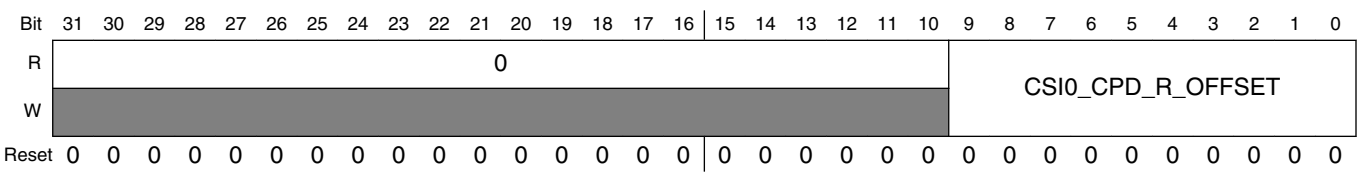
IPU_CSI0_CPD_OFFSET1 field descriptions (continued)

Field	Description
29–20 CSI0_CPD_B_ OFFSET	Reserved
19–10 CSI0_GB_ OFFSET	Reserved
9–0 CSI0_GR_ OFFSET	Reserved

45.51.131 CSI0 Compander Offset Register 2 (IPU_CSI0_CPD_OFFSET2)

This register contain Offset parameters used for companding.

Address: IPU_CSI0_CPD_OFFSET2 is 1E00_0000h base + 300F0h offset = 1E03_00F0h



IPU_CSI0_CPD_OFFSET2 field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–0 CSI0_CPD_R_ OFFSET	CSI0 Red component offset The value is between -512 to 511. The value is added to the red component before companding. Clipping: If the result of the red components value + the offset is smaller than 0, the result is zero If the result of the red components value + the offset is greater than 1023, the result is 1023 Reserved

45.51.132 CSI1 Sensor Configuration Register (IPU_CSI1_SENS_CONF)

Address: IPU_CSI1_SENS_CONF is 1E00_0000h base + 38000h offset = 1E03_8000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CSI0_DATA_EN_POL	0	CSI1_FORCE_EOF	CSI1_JPEG_MODE	CSI1_JPEG8_EN	CSI1_DATA_DEST			CSI1_DIV_RATIO							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CSI1_EXT_VSYNC	CSI1_DATA_WIDTH				CSI1_SENS_DATA_FORMAT			CSI1_PACK_TIGHT	CSI1_SENS_PRTCL			CSI1_SENS_PIX_CLK_POL	CSI1_DATA_POL	CSI1_HSYNC_POL	CSIT_VSYNC_POL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_SENS_CONF field descriptions

Field	Description
31 CSI0_DATA_EN_POL	Invert IPP_IND_SENSB_DATA_EN input. This bit selects the polarity of IPP_IND_SENSB_DATA_EN signal. 0 IPP_IND_SENSB_DATA_EN is directly applied to internal circuitry. 1 IPP_IND_SENSB_DATA_EN is inverted before applied to internal circuitry.
30 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
29 CSI1_FORCE_EOF	Force End of frame This is a self clear bit allowing the user to force an End-of-frame event; This bit can be used in cases where the frame sent by the sensor was not completed. 1 force end of frame 0 no action
28 CSI1_JPEG_MODE	JPEG Mode - this bit defines the mode of the control signals when working in JPEG mode 1 The data is valid as long as HSYNC and VSYNC signals are active; HSYNC is valid for single frame 0 The frame starts withe the assertion of VSYNC. The frame ends on the next VSYNC or by setting the **CSI0_FORCE_EOF bit**
27 CSI1_JPEG8_EN	JPEG8 enable bit 1 JPEG8 detection is enabled 0 JPEG8 is disabled
26–24 CSI1_DATA_DEST	These bits enable the destination of the data coming from the CSI. CSI1_DATA_DEST[0] - Reserved CSI1_DATA_DEST[1] - destination is IC

Table continues on the next page...

IPU_CSI1_SENS_CONF field descriptions (continued)

Field	Description
	CSI1_DATA_DEST[2] - destination is IDMAC via SMFC
23–16 CSI1_DIV_RATIO	DIV Ratio Clock division ratio minus 1. This field defines the division ratio of HSP_CLK into SENSB_MCLK: SENSB_MCLK rate = HSP_CLK rate /(DIV_RATIO+1)
15 CSI1_EXT_VSYNC	External VSYNC enable. This bits select between external and internal VSYNC. 0 Internal VSYNC mode. 1 External VSYNC mode.
14–11 CSI1_DATA_WIDTH	Data width. This fields defines the number of bits per color. Values: 0000 4 bits per color 0000 Reserved 0001 8 bits per color 0010 9 bits per color 0010 Reserved 0011 10 bits per color 0100 11 bits per color 0100 Reserved 0101 12 bits per color 0101 Reserved 0110 13 bits per color 0110 Reserved 0111 14 bits per color 0111 Reserved 1000 15 bits per color 1000 Reserved 1001 16 bits per color
10–8 CSI1_SENS_DATA_FORMAT	Data format from the sensor. This field defines the data format for the input of the CSI sensor. Values: 000 full RGB or YUV444 001 YUV422 (YUYV...) 010 YUV422 (UYVY...) 011 Bayer or Generic data 100 RGB565 101 RGB555 110 RGB444 111 JPEG
7 CSI1_PACK_TIGHT	**CSI1 Pack Tight** When the data format is YUV or RGB and the component's width is 9-16 bits, it can be sent to the memory in 2 different ways 1 Three 10 bits components are packed into a 32 bit word. Color extension/reduction is performed 0 Each component is written as a 16 bit word where the MSB is written to bit #15, color extension is done for the remaining least significant bits.

Table continues on the next page...

IPU_CSI1_SENS_CONF field descriptions (continued)

Field	Description
6–4 CSI1_SENS_ PRTCL	Sensor Protocol. This bit defines the Sensor timing/data mode protocol. Values: 000 Gated clock mode 001 Non-gated clock mode 010 CCIR progressive mode (BT.656) 011 CCIR interlaced mode (BT.656) 100 CCIR progressive (BT.1120 DDR mode: data arrives on every edge of the clock) 101 CCIR progressive (BT.1120 SDR mode: data arrives only on the positive edge of the clock) 110 CCIR interlaced mode (BT.1120 DDR mode: data arrives on every edge of the clock) 111 CCIR interlaced mode (BT.1120 SDR mode: data arrives only on the positive edge of the clock)
3 CSI1_SENS_ PIX_CLK_POL	Invert Pixel clock input. This bit selects the polarity of pixel clock. 0 pixel clock is directly applied to internal circuitry. 1 pixel clock is inverted before applied to internal circuitry.
2 CSI1_DATA_ POL	Invert data input. This bit selects the polarity of data input. 0 data lines are directly applied to internal circuitry. 1 data lines are inverted before applied to internal circuitry.
1 CSI1_HSYNC_ POL	Invert IPP_IND_SENSB_HSYNC input. This bit selects the polarity of IPP_IND_SENSB_HSYNC signal. 0 IPP_IND_SENSB_HSYNC is directly applied to internal circuitry. 1 IPP_IND_SENSB_HSYNC is inverted before applied to internal circuitry.
0 CSI1_VSYNC_ POL	Invert IPP_IND_SENSB_VSYNC input. This bit selects the polarity of IPP_IND_SENSB_VSYNC signal. 0 IPP_IND_SENSB_VSYNC is not inverted before applied to internal circuitry. 1 IPP_IND_SENSB_VSYNC is inverted before applied to internal circuitry.

45.51.133 CSI1 Sense Frame Size Register (IPU_CSI1_SENS_FRM_SIZE)

Address: IPU_CSI1_SENS_FRM_SIZE is 1E00_0000h base + 38004h offset = 1E03_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				CSI1_SENS_FRM_HEIGHT												0			CSI1_SENS_FRM_WIDTH													
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_SENS_FRM_SIZE field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.

Table continues on the next page...

IPU_CSI1_SENS_FRM_SIZE field descriptions (continued)

Field	Description
27–16 CSI1_SENS_ FRM_HEIGHT	Sensor frame height minus 1. This field defines the sensor frame rows number minus 1.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
12–0 CSI1_SENS_ FRM_WIDTH	Sensor frame width minus 1. This field defines the sensor frame column number minus 1.

45.51.134 CSI1 Actual Frame Size Register (IPU_CSI1_ACT_FRM_SIZE)

Address: IPU_CSI1_ACT_FRM_SIZE is 1E00_0000h base + 38008h offset = 1E03_8008h

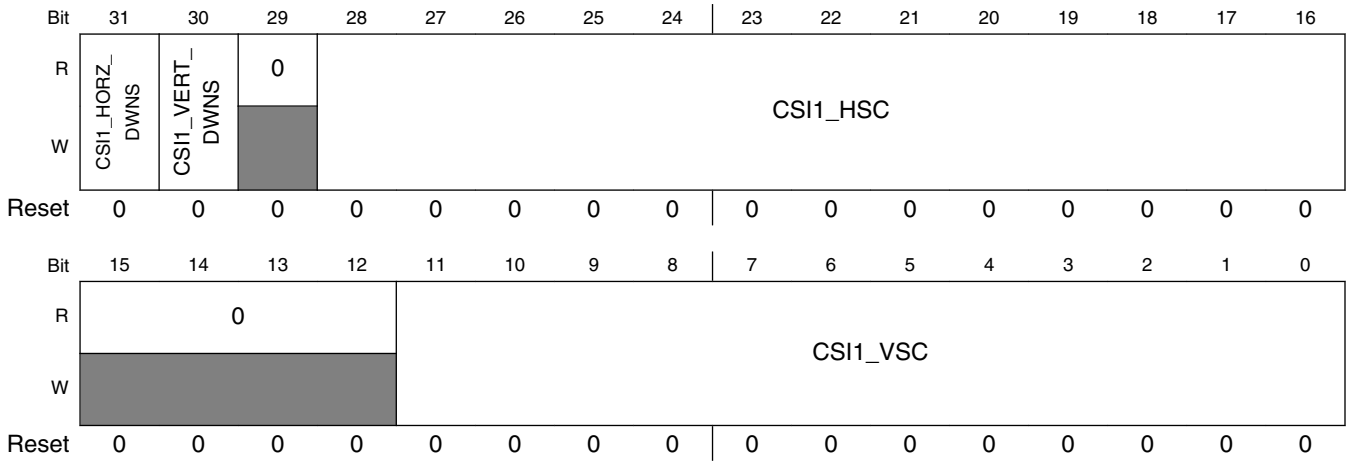
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				CSI1_ACT_FRM_HEIGHT												0			CSI1_ACT_FRM_WIDTH													
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_ACT_FRM_SIZE field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
27–16 CSI1_ACT_ FRM_HEIGHT	Actual frame height minus 1. This field defines the CSI output frame rows number minus 1.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
12–0 CSI1_ACT_ FRM_WIDTH	Actual frame width minus 1. This field defines the CSI output frame columns number minus 1.

45.51.135 CSI1 Output Control Register (IPU_CSI1_OUT_FRM_CTRL)

Address: IPU_CSI1_OUT_FRM_CTRL is 1E00_0000h base + 3800Ch offset = 1E03_800Ch

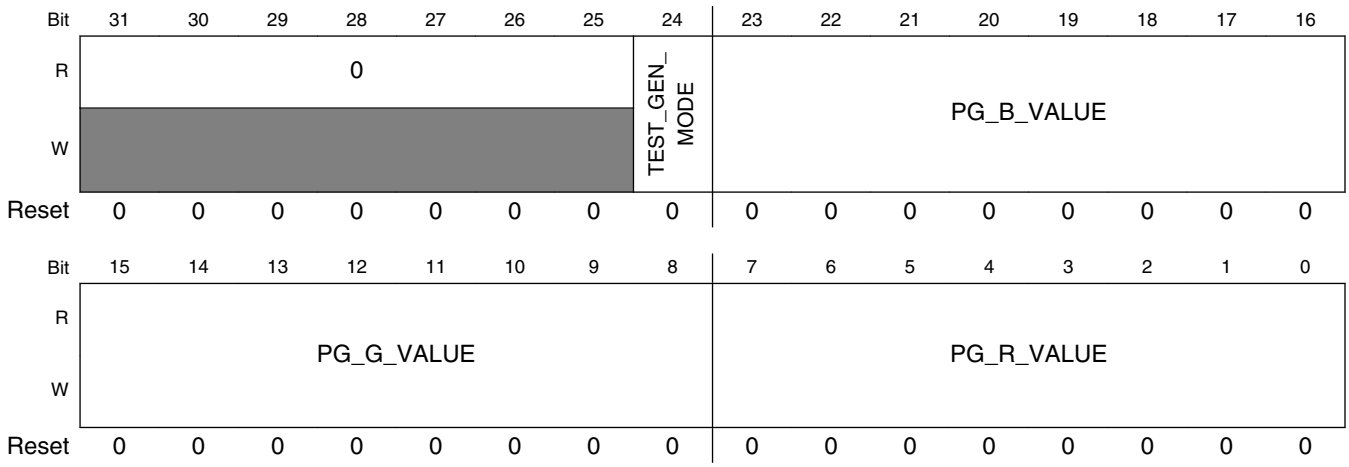


IPU_CSI1_OUT_FRM_CTRL field descriptions

Field	Description
31 CSI1_HORZ_DWNS	Enable horizontal downsizing (decimation) by 2. 0 Downsizing disabled 1 Downsizing enabled
30 CSI1_VERT_DWNS	Enable vertical downsizing (decimation) by 2. 0 Downsizing disabled 1 Downsizing enabled
29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28–16 CSI1_HSC	Horizontal skip. This field defines the number of columns to skip. In Interlaced mode this number refers to the number of lines per field
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
11–0 CSI1_VSC	Vertical skip. This field defines the number of rows to skip.

45.51.136 CSI1 Test Control Register (IPU_CSI1_TST_CTRL)

Address: IPU_CSI1_TST_CTRL is 1E00_0000h base + 38010h offset = 1E03_8010h



IPU_CSI1_TST_CTRL field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
24 TEST_GEN_ MODE	Test generator mode. This bit activates the signal generation. 0 Test signal generator is inactive. 1 Test signal generator is active.
23–16 PG_B_VALUE	Pattern generator B value. This field selects the B value for the generated pattern of even pixel.
15–8 PG_G_VALUE	Pattern generator G value. This field selects the G value for the generated pattern of even pixel.
7–0 PG_R_VALUE	Pattern generator R value. This field selects the R value for the generated pattern of even pixel.

45.51.137 CSI1 CCIR Code Register 1 (IPU_CSI1_CCIR_CODE_1)

Address: IPU_CSI1_CCIR_CODE_1 is 1E00_0000h base + 38014h offset = 1E03_8014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							CSI1_CCIR_ERR_DET_EN	0		CSI1_STRT_FLD0_ACTV			CSI1_END_FLD0_ACTV		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				CSI1_STRT_FLD0_BLNK_2ND			CSI1_END_FLD0_BLNK_2ND			CSI1_STRT_FLD0_BLNK_1ST			CSI1_END_FLD0_BLNK_1ST		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_CCIR_CODE_1 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
24 CSI1_CCIR_ERR_DET_EN	Enable error detection and correction for CCIR interlaced mode with protection bit. 0 Error detection and correction is disabled. 1 Error detection and correction is enabled.
23–22 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
21–19 CSI1_STRT_FLD0_ACTV	Start of field 0 active line command (interlaces mode). (In progressive mode, start of active line command mode).
18–16 CSI1_END_FLD0_ACTV	End of field 0 active line command (interlaces mode). (In progressive mode, end of active line command mode).
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
11–9 CSI1_STRT_FLD0_BLNK_2ND	Start of field 0 second blanking line command (interlaces mode). (In progressive mode this field is ignored).
8–6 CSI1_END_FLD0_BLNK_2ND	End of field 0 second blanking line command (interlaces mode). (In progressive mode this field is ignored).

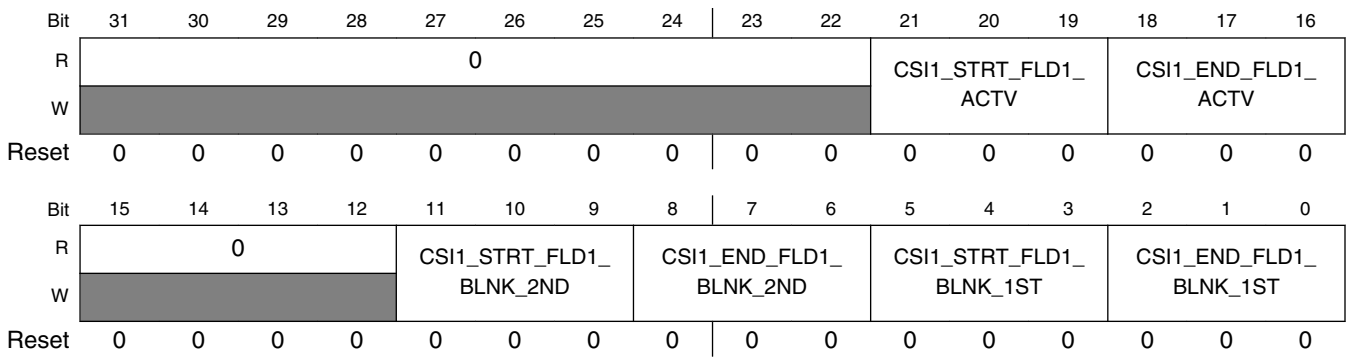
Table continues on the next page...

IPU_CSI1_CCIR_CODE_1 field descriptions (continued)

Field	Description
5–3 CSI1_STRT_FLD0_BLNK_1ST	Start of field 0 first blanking line command (interlaces mode). (In progressive mode this field indicates start of blanking line command).
2–0 CSI1_END_FLD0_BLNK_1ST	End of field 0 first blanking line command (interlaces mode). (In progressive mode this field is ignored).

45.51.138 CSI1 CCIR Code Register 2 (IPU_CSI1_CCIR_CODE_2)

Address: IPU_CSI1_CCIR_CODE_2 is 1E00_0000h base + 38018h offset = 1E03_8018h



IPU_CSI1_CCIR_CODE_2 field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
21–19 CSI1_STRT_FLD1_ACTV	Start of field 1 active line command (interlaces mode). (In progressive mode this field is ignored).
18–16 CSI1_END_FLD1_ACTV	End of field 1 active line command (interlaces mode). (In progressive mode this field is ignored).
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
11–9 CSI1_STRT_FLD1_BLNK_2ND	Start of field 1 second blanking line command (interlaces mode). (In progressive mode this field is ignored).
8–6 CSI1_END_FLD1_BLNK_2ND	End of field 1 second blanking line command (interlaces mode). (In progressive mode this field is ignored).

Table continues on the next page...

IPU_CSI1_CCIR_CODE_2 field descriptions (continued)

Field	Description
5–3 CSI1_STRT_ FLD1_BLNK_ 1ST	Start of field 1 first blanking line command (interlaces mode). (In progressive mode this field is ignored).
2–0 CSI1_END_ FLD1_BLNK_ 1ST	End of field 1 first blanking line command (interlaces mode). (In progressive mode this field is ignored).

45.51.139 CSI1 CCIR Code Register 3 (IPU_CSI1_CCIR_CODE_3)

Address: IPU_CSI1_CCIR_CODE_3 is 1E00_0000h base + 3801Ch offset = 1E03_801Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																CSI1_CCIR_PRECOM																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_CCIR_CODE_3 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
29–0 CSI1_CCIR_ PRECOM	CCIR pre command. This field defines the sequence which comes before the CCIR command. For BT.656 the code should be written to bits [23:0] while bits [29:24] are ignored (3X8bit) For BT.1120 the code should be written to bits [29:0] (3X10bit)

45.51.140 CSI1 Data Identifier Register (IPU_CSI1_DI)

Address: IPU_CSI1_DI is 1E00_0000h base + 38020h offset = 1E03_8020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CSI1_MIPI_DI3								CSI1_MIPI_DI2								CSI0_MIPI_DI1								CSI1_MIPI_DI0							
W																																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

IPU_CSI1_DI field descriptions

Field	Description
31–24 CSI1_MIPI_DI3	Reserved

Table continues on the next page...

IPU_CSI1_DI field descriptions (continued)

Field	Description
23–16 CSI1_MIPI_DI2	Reserved
15–8 CSI0_MIPI_DI1	Reserved
7–0 CSI1_MIPI_DI0	Reserved

45.51.141 CSI1 SKIP Register (IPU_CSI1_SKIP)

This register control the frame skipping supported between CSI1 and the SMFC.

Address: IPU_CSI1_SKIP is 1E00_0000h base + 38024h offset = 1E03_8024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								CSI1_SKIP_ISP					CSI1_MAX_RATIO_SKIP_ISP		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						CSI1_ID_2_SKIP		CSI1_SKIP_SMFC					CSI1_MAX_RATIO_SKIP_SMFC		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_SKIP field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23–19 CSI1_SKIP_ISP	Reserved
18–16 CSI1_MAX_RATIO_SKIP_ISP	Reserved
15–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–8 CSI1_ID_2_SKIP	Reserved
7–3 CSI1_SKIP_SMFC	CSI1 SKIP SMFC These 5 bits define the skipping pattern of the frames send to the SMFC. Skipping is done for a set of frames. The number of frames in a set is defined at CSI1_MAX_RATIO_SKIP_SMFC. when CSI1_MAX_RATIO_SKIP_SMFC = 1 => CSI1_SKIP_SMFC[0] is used; other bits are ignored

Table continues on the next page...

IPU_CSI1_SKIP field descriptions (continued)

Field	Description
	when CSI1_MAX_RATIO_SKIP_SMFC = 2 => CSI1_SKIP_SMFC[1:0] are used; other bits are ignored when CSI1_MAX_RATIO_SKIP_SMFC = 3 => CSI1_SKIP_SMFC[2:0] are used; other bits are ignored when CSI1_MAX_RATIO_SKIP_SMFC = 4 => CSI1_SKIP_SMFC[3:0] are used; other bits are ignored when CSI1_MAX_RATIO_SKIP_SMFC = 5 => CSI1_SKIP_SMFC[4:0] are used; Setting bit #n of CSI1_SKIP_SMFC means that the #n frame in the set is skipped. For example: if CSI1_MAX_RATIO_SKIP_SMFC = 4 and CSI1_SKIP_SMFC = 11010 Frames #0 & Frame #2 will not be skipped as bit0 and bit2 are cleared Frames #1 & Frame #3 will be skipped as bit1 and bit3 are set bit #4 is ignored as CSI1_MAX_RATIO_SKIP_SMFC is set to 4
2–0 CSI1_MAX_ RATIO_SKIP_ SMFC	CSI1 Maximum Ratio Skip for SMFC These bits define the number of frames in a skipping set. These bits define the number of frames in a skipping set. The skipping number is equal to CSI1_MAX_RATIO_SKIP_SMFC+1; The maximum value of this bits is 5. When set to 0 the skipping is disabled.

45.51.142 CSI1 Compander Control Register (IPU_CSI1_CPD_CTRL)

Address: IPU_CSI1_CPD_CTRL is 1E00_0000h base + 38028h offset = 1E03_8028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											0			0	0
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

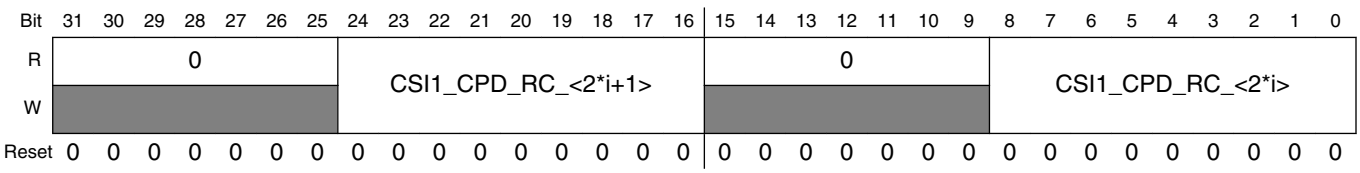
IPU_CSI1_CPD_CTRL field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared.
4–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

45.51.143 CSI1 Red Component Compander Constants Register <i> (IPU_CSI1_CPD_RC_i)

These registers contain CONSTANT <i> parameters used for companding of red component.

Address: IPU_CSI1_CPD_RC_i is 1E00_0000h base + 3802Ch offset = 1E03_802Ch



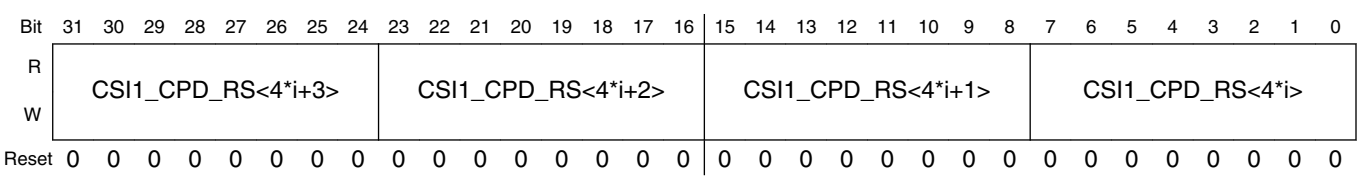
IPU_CSI1_CPD_RC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 CSI1_CPD_RC_<2*i+1>	CONSTANT <2*i+1> parameter of Compander, Red component. Reserved
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 CSI1_CPD_RC_<2*i>	CONSTANT <2*i> parameter of Compander, Red component. Reserved

45.51.144 CSI1 Red Component Compander SLOPE Register <i> (IPU_CSI1_CPD_RS_i)

These registers contain SLOPE <i> parameters used for companding of red component.

Address: IPU_CSI1_CPD_RS_i is 1E00_0000h base + 3804Ch offset = 1E03_804Ch



IPU_CSI1_CPD_RS_i field descriptions

Field	Description
31–24 CSI1_CPD_RS<4*i+3>	SLOPE<4*i+3> parameter of Compander, Red component. Reserved

Table continues on the next page...

IPU_CSI1_CPD_RS_i field descriptions (continued)

Field	Description
23–16 CSI1_CPD_ RS<4*i+2>	SLOPE<4*i+2> parameter of Compander, Red component. Reserved
15–8 CSI1_CPD_ RS<4*i+1>	SLOPE<4*i+1> parameter of Compander, Red component. Reserved
7–0 CSI1_CPD_ RS<4*i>	SLOPE<4*i> parameter of Compander, Red component. Reserved

45.51.145 CSI1 GR Component Compander Constants Register <i> (IPU_CSI1_CPD_GRC_i)

These registers contain CONSTANTi parameters used for companding of green components in GRGR rows.

Address: IPU_CSI1_CPD_GRC_i is 1E00_0000h base + 3805Ch offset = 1E03_805Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CSI1_CPD_GRC<2*i+1>									0							CSI1_CPD_GRC<2*i>							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_CPD_GRC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 CSI1_CPD_ GRC<2*i+1>	CONST<2*i+1> parameter of Compander, GR component. If the input format is RGB/YUV then CSI1_CPD_GRC should be equal to CSI1_CPD_GBC Reserved
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 CSI1_CPD_ GRC<2*i>	CONSTANT<2*i> parameter of Compander, GR component. If the input format is RGB/YUV then CSI1_CPD_GRC should be equal to CSI1_CPD_GBC Reserved

45.51.146 CSI1 GR Component Compander SLOPE Register <i> (IPU_CSI1_CPD_GRS_i)

These registers contain SLOPEi parameters used for companding of green components in GRGR rows.

Address: IPU_CSI1_CPD_GRS_i is 1E00_0000h base + 3807Ch offset = 1E03_807Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSI1_CPD_GRS<4*i+3>								CSI1_CPD_GRS<4*i+2>								CSI1_CPD_GRS<4*i+1>								CSI1_CPD_GRS<4*i>							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_CPD_GRS_i field descriptions

Field	Description
31–24 CSI1_CPD_ GRS<4*i+3>	SLOPE<4*i+3> parameter of Compander, GR component. If the input format is RGB/YUV then CSI1_CPD_GRS should be equal to CSI1_CPD_GBS Reserved
23–16 CSI1_CPD_ GRS<4*i+2>	SLOPE<4*i+2> parameter of Compander, GR component. If the input format is RGB/YUV then CSI1_CPD_GRS should be equal to CSI1_CPD_GBS Reserved
15–8 CSI1_CPD_ GRS<4*i+1>	SLOPE<4*i+1> parameter of Compander, GR component. If the input format is RGB/YUV then CSI1_CPD_GRS should be equal to CSI1_CPD_GBS Reserved
7–0 CSI1_CPD_ GRS<4*i>	SLOPE<4*i> parameter of Compander, GR component. If the input format is RGB/YUV then CSI1_CPD_GRS should be equal to CSI1_CPD_GBS Reserved

45.51.147 CSI1 GB Component Compander Constants Register <i> (IPU_CSI1_CPD_GBC_i)

These registers contain CONSTANTi parameters used for companding of green components in GBGB rows.

Address: IPU_CSI1_CPD_GBC_i is 1E00_0000h base + 3808Ch offset = 1E03_808Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CSI1_CPD_GBC<2*i+1>								0								CSI1_CPD_GBC<2*i>							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_CPD_GBC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 CSI1_CPD_ GBC<2*i+1>	CONSTi+1 parameter of Compander, GB component. If the input format is RGB/YUV then CSI1_CPD_GBC should be equal to CSI1_CPD_GRC Reserved
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 CSI1_CPD_ GBC<2*i>	CONSTANTi parameter of Compander, GB component. If the input format is RGB/YUV then CSI1_CPD_GBC should be equal to CSI1_CPD_GRC Reserved

45.51.148 CSI1 GB Component Compander SLOPE Register <i> (IPU_CSI1_CPD_GBS_i)

These registers contain SLOPEi parameters used for companding of green components in GBGB rows.

Address: IPU_CSI1_CPD_GBS_i is 1E00_0000h base + 380ACh offset = 1E03_80ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSI1_CPD_GBS<4*i+3>								CSI1_CPD_GBS<4*i+2>								CSI1_CPD_GBS<4*i+1>								CSI1_CPD_GBS<4*i>							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_CPD_GBS_i field descriptions

Field	Description
31–24 CSI1_CPD_ GBS<4*i+3>	SLOPE<4*i+3> parameter of Compander, GB component. If the input format is RGB/YUV then CSI1_CPD_GBS should be equal to CSI1_CPD_GRS Reserved
23–16 CSI1_CPD_ GBS<4*i+2>	SLOPE<4*i+2> parameter of Compander, GB component. If the input format is RGB/YUV then CSI1_CPD_GBS should be equal to CSI1_CPD_GRS Reserved
15–8 CSI1_CPD_ GBS<4*i+1>	SLOPE<4*i+1> parameter of Compander, GB component. If the input format is RGB/YUV then CSI1_CPD_GBS should be equal to CSI1_CPD_GRS Reserved
7–0 CSI1_CPD_ GBS<4*i>	SLOPE<4*i> parameter of Compander, GB component. If the input format is RGB/YUV then CSI1_CPD_GBS should be equal to CSI1_CPD_GRS Reserved

45.51.149 CSI1 Blue Component Compander Constants Register <i> (IPU_CSI1_CPD_BC_i)

These registers contend CONSTANTi parameters used for companding of blue component.

Address: IPU_CSI1_CPD_BC_i is 1E00_0000h base + 380BCh offset = 1E03_80BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CSI1_CPD_BC<2*i+1>									0							CSI1_CPD_BC<2*i>							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_CPD_BC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 CSI1_CPD_BC<2*i+1>	CONSTANT<2*i+1> parameter of Compander, Blue component. Reserved
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 CSI1_CPD_BC<2*i>	CONSTANT<2*i> parameter of Compander, Blue component. Reserved

45.51.150 CSI1 Blue Component Compander SLOPE Register <i> (IPU_CSI1_CPD_BS_i)

This registers contain SLOPEi parameters used for companding of red component.

Address: IPU_CSI1_CPD_BS_i is 1E00_0000h base + 380DCh offset = 1E03_80DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CSI1_CPD_BS<4*i+3>								CSI1_CPD_BS<4*i+2>								CSI1_CPD_BS<4*i+1>								CSI1_CPD_BS<4*i>							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_CPD_BS_i field descriptions

Field	Description
31–24 CSI1_CPD_BS<4*i+3>	SLOPE<4*i+3> parameter of Compander, Blue component. Reserved

Table continues on the next page...

IPU_CSI1_CPD_BS_i field descriptions (continued)

Field	Description
23–16 CSI1_CPD_ BS<4*i+2>	SLOPE<4*i+2> parameter of Compander, Blue component. Reserved
15–8 CSI1_CPD_ BS<4*i+1>	SLOPE<4*i+1> parameter of Compander, Blue component. Reserved
7–0 CSI1_CPD_ BS<4*i>	SLOPE<4*i> parameter of Compander, Blue component. Reserved

45.51.151 CSI1 Compander Offset Register 1 (IPU_CSI1_CPD_OFFSET1)

These registers contain Offset parameters used for companding.

Address: IPU_CSI1_CPD_OFFSET1 is 1E00_0000h base + 380ECh offset = 1E03_80ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		CSI1_CPD_B_OFFSET										CSI1_CPD_GB_OFFSET										CSI1_CPD_GR_OFFSET									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_CSI1_CPD_OFFSET1 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved
29–20 CSI1_CPD_B_ OFFSET	CSI1 Blue component offset The value is between -512 to 511. The value is added to the blue component before companding. Clipping: If the result of the blue components value + the offset is smaller than 0, the result is zero If the result of the blue components value + the offset is greater than 1023, the result is 1023 Reserved
19–10 CSI1_CPD_GB_ OFFSET	CSI1 Green Blue component offset The value is between -512 to 511. The value is added to the blue component before companding. Clipping: If the result of the green-blue components value + the offset is smaller than 0, the result is zero If the result of the green-blue components value + the offset is greater than 1023, the result is 1023 If the input format is RGB/YUV then CSI1_GB_OFFSET must be equal to CSI1_GR_OFFSET Reserved

Table continues on the next page...

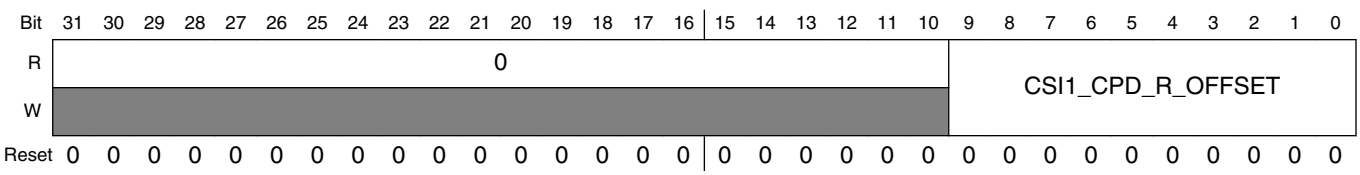
IPU_CSI1_CPD_OFFSET1 field descriptions (continued)

Field	Description
9–0 CSI1_CPD_GR_OFFSET	CSI1 Green Red component offset The value is between -512 to 511. The value is added to the green-red component before companding. Clipping: If the result of the green-red components value + the offset is smaller than 0, the result is zero If the result of the green-red components value + the offset is greater than 1023, the result is 1023 Reserved

45.51.152 CSI1 Compander Offset Register 2 (IPU_CSI1_CPD_OFFSET2)

These registers contain Offset parameters used for companding.

Address: IPU_CSI1_CPD_OFFSET2 is 1E00_0000h base + 380F0h offset = 1E03_80F0h



IPU_CSI1_CPD_OFFSET2 field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–0 CSI1_CPD_R_OFFSET	CSI1 Red component offset The value is between -512 to 511. The value is added to the red component before companding. Clipping: If the result of the red components value + the offset is smaller than 0, the result is zero If the result of the red components value + the offset is greater than 1023, the result is 1023 Reserved

45.51.153 DI0 General Register (IPU_DI0_GENERAL)

Address: IPU_DI0_GENERAL is 1E00_0000h base + 40000h offset = 1E04_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	di0_pin8_pin15_sel	di0_disp_y_sel			DI0_CLOCK_STOP_MODE				DI0_DISP_CLOCK_INIT	di0_mask_sel	di0_vsync_ext	di0_clk_ext	DI0_WATCHDOG_MODE		di0_polarity_disp_clk	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	di0_sync_count_sel				di0_err_treatment	di0_erm_vsync_sel	di0_polarity_cs1	di0_polarity_cs0	di0_polarity_<i+1>							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_GENERAL field descriptions

Field	Description
31 di0_pin8_pin15_sel	This bit routes PIN8 over PIN15 1 PIN8 is routed to PIN15, PIN8 is also routed to PIN8 0 PIN15 is routed to PIN15, PIN8 is routed to PIN8
30–28 di0_disp_y_sel	DI0 Display Vertical coordinate (Y) select. This field defines which one of the 8 counters will be used as a display's line counter. 000 counter #1 is selected 111 counter #8 is selected
27–24 DI0_CLOCK_STOP_MODE	DI clock stop mode When performing a clock change. The DI stops the clock to the display. These field defines when the clock will be stopped. Stopping at EOL/EOF is supported for the case where the data is coming from the IDMAC (DMA access). In case that only direct accesses is performed, the user should set this field to 0000 0001-1001 stop at the next event of one of the counters (counter #1 to counter #9) 0000 stop at the next edge of the display clock 1100 stop at EOL (end of a line), but if stop request is during blanking interval, stop now 1101 stop at EOF (end of a frame), but if stop request is during blanking interval, stop now 1110 stop at EOL (end of a line), but if stop request is during blanking interval, stop at the end of the next line 1111 stop at EOF (end of a frame), but if stop request is during blanking interval, stop at the end of the next frame
23 DI0_DISP_CLOCK_INIT	Display clock's initial mode For synchronization error conditions the display clock can be stopped on the next VSYNC

Table continues on the next page...

IPU_DI0_GENERAL field descriptions (continued)

Field	Description
	1 The display's clock is running after the next VSYNC (indicating new frame) 0 The display's clock is stopped after the next VSYNC (indicating new frame)
22 di0_mask_sel	DI0 Mask select. IPP_PIN_2 output of the DI that functions as MASK signal can come from 2 sources: counter #2 or extracted from the MASK data coming from the memory. 1 IPP_PIN_2 is coming from extracted MASK data coming from the memory 0 IPP_PIN_2 is coming from counter #2
21 di0_vsync_ext	DI0 External VSYNC. This bit selects the source of the VSYNC signal 1 External to the IPU 0 Internally generated by the IPU
20 di0_clk_ext	DI0 External Clock. This bit selects the source of the DI0's clock 1 The source of the clock is external to the IPU 0 The clock is internally generated by the IPU
19–18 DI0_ WATCHDOG_ MODE	DI0 watchdog mode In case of a display error where the DI clock is stopped (defined at di0_err_treatment). An internal watchdog counts DI clocks. If this timer reached its pre defined value the DI will skip the current frame and restart on the frame. This 2 bits define the number of DI clock cycles that the timer counts. 00 The timer counts 4 DI cycles 01 The timer counts 16 DI cycles 10 The timer counts 64 DI cycles 11 The timer counts 128 DI cycles
17 di0_polarity_ disp_clk	DI0 Output Clock's polarity This bits define the polarity of the DI0's clock. 1 The output clock is active high 0 The output clock is active low
16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–12 di0_sync_count_ sel	For synchronous flow error: selects synchronous flow synchronization counter in DI:
11 di0_err_ treatment	In case of synchronous flow error there are 2 ways to handle the display 1 to wait (stop clock) 0 Drive the last component
10 di0_erm_vsync_ sel	DI0 error recovery block's VSYNC source select The error recovery block detect a case where the DI's VSYNC is asserted before the EOF. This bit selects the source of the VSYNC signal monitored by this mechanism.

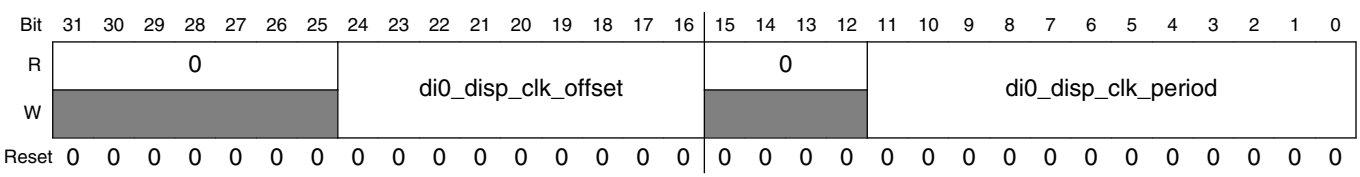
Table continues on the next page...

IPU_DI0_GENERAL field descriptions (continued)

Field	Description
	1 vsync_post - an internal VSYNC signal asserted 2 lines after the DI's VSYNC 0 vsync_pre - an internal VSYNC signal asserted 2 lines before the DI's VSYNC
9 di0_polarity_cs1	DI0 Chip Select's 1 polarity This bits define the polarity of the DI's CS1. 1 The CS1 is active high 0 The CS1 is active low
8 di0_polarity_cs0	DI0 Chip Select's 0 polarity This bits define the polarity of the DI's CS0. 1 The CS0 is active high 0 The CS0 is active low
7–0 di0_polarity_<i+1>	DI0 output pin's polarity This bits define the polarity of each of the DI's outputs. 1 The output pin is active high 0 The output pin is active low

45.51.154 DI0 Base Sync Clock Gen 0 Register (IPU_DI0_BS_CLKGEN0)

Address: IPU_DI0_BS_CLKGEN0 is 1E00_0000h base + 40004h offset = 1E04_0004h

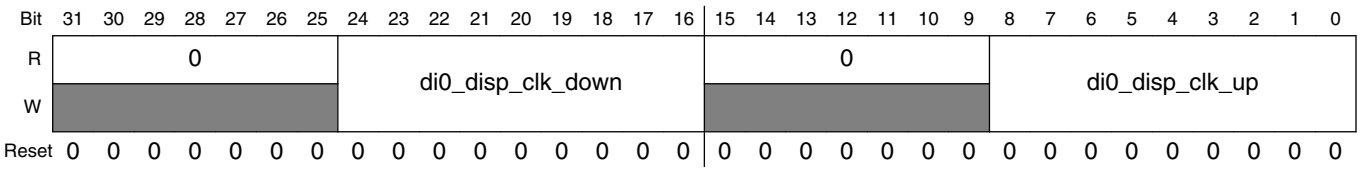


IPU_DI0_BS_CLKGEN0 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 di0_disp_clk_offset	DI0 Display Clock Offset The DI has the ability to delay the display's clock This field defines the amount of IPU's clock cycles added as delay on this clock.
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved.
11–0 di0_disp_clk_period	DI0 Display Clock Period This field defines the Display interface clock period for display write access. This parameter contains an integer part (bits 11:4) and a fractional part (bits 3:0). It defines a fractional division ratio of the HSP_CLK clock for generation of the display's interface clock.

45.51.155 DI0 Base Sync Clock Gen 1 Register (IPU_DI0_BS_CLKGEN1)

Address: IPU_DI0_BS_CLKGEN1 is 1E00_0000h base + 40008h offset = 1E04_0008h

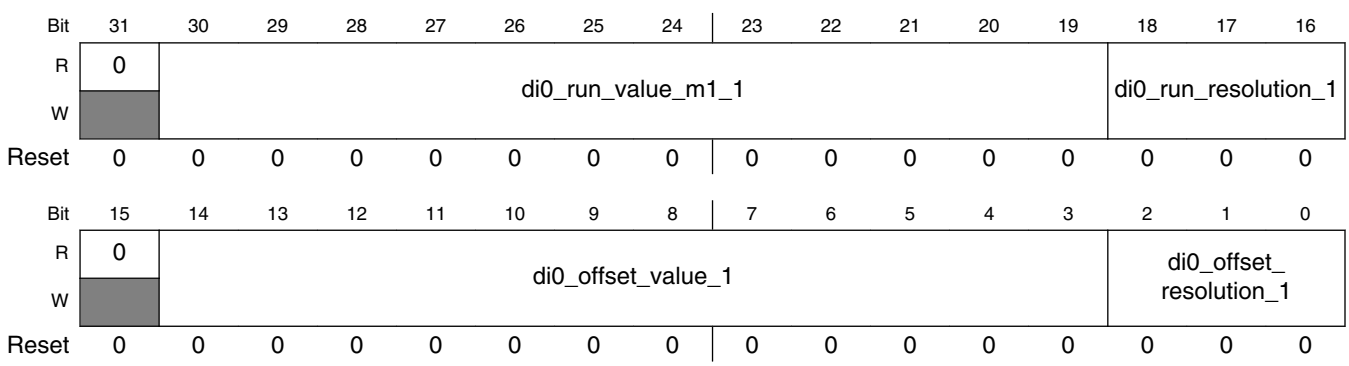


IPU_DI0_BS_CLKGEN1 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 di0_disp_clk_down	DI0 display clock falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is a time interval between display's access start point and display's interface clock falling edge.
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 di0_disp_clk_up	DI0 display clock rising edge position This parameter contains an integer part (bits 8:1) and a fractional part (bit 0). The position value is a time interval between display's access start point and display's interface clock rising edge.

45.51.156 DI0 Sync Wave Gen 1 Register 0 (IPU_DI0_SW_GEN0_1)

Address: IPU_DI0_SW_GEN0_1 is 1E00_0000h base + 4000Ch offset = 1E04_000Ch



IPU_DI0_SW_GEN0_1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–19 di0_run_value_m1_1	DI0 counter #1 pre defined value This fields defines the counter #1 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_1 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.
18–16 di0_run_resolution_1	DI0 counter #1 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 NA 011 NA 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSIs according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–3 di0_offset_value_1	DI0 counter #1 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di0_offset_resolution_1	DI0 counter #1 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock 010 NA 011 NA 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSIs according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.

45.51.157 DI0 Sync Wave Gen 2 Register 0 (IPU_DI0_SW_GEN0_2)

Address: IPU_DI0_SW_GEN0_2 is 1E00_0000h base + 40010h offset = 1E04_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_run_value_m1_2												di0_run_resolution_2		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_offset_value_2												di0_offset_ resolution_2		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN0_2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–19 di0_run_value_ m1_2	DI0 counter #2 pre defined value This fields defines the counter #2 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_2 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.
18–16 di0_run_ resolution_2	DI0 counter #2 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock 010 The Counter is triggered by counter #1 011 NA 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–3 di0_offset_ value_2	DI0 counter #2 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di0_offset_ resolution_2	DI0 counter #2 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled

Table continues on the next page...

IPU_DI0_SW_GEN0_2 field descriptions (continued)

Field	Description
	001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 NA 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSIs according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.

45.51.158 DI0 Sync Wave Gen 3 Register 0 (IPU_DI0_SW_GEN0_3)

Address: IPU_DI0_SW_GEN0_3 is 1E00_0000h base + 40014h offset = 1E04_0014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_run_value_m1_3												di0_run_resolution_3		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_offset_value_3												di0_offset_ resolution_3		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN0_3 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–19 di0_run_value_ m1_3	DI0 counter #3 pre defined value This fields defines the counter #3 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_3 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.
18–16 di0_run_ resolution_3	DI0 counter #3 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit.

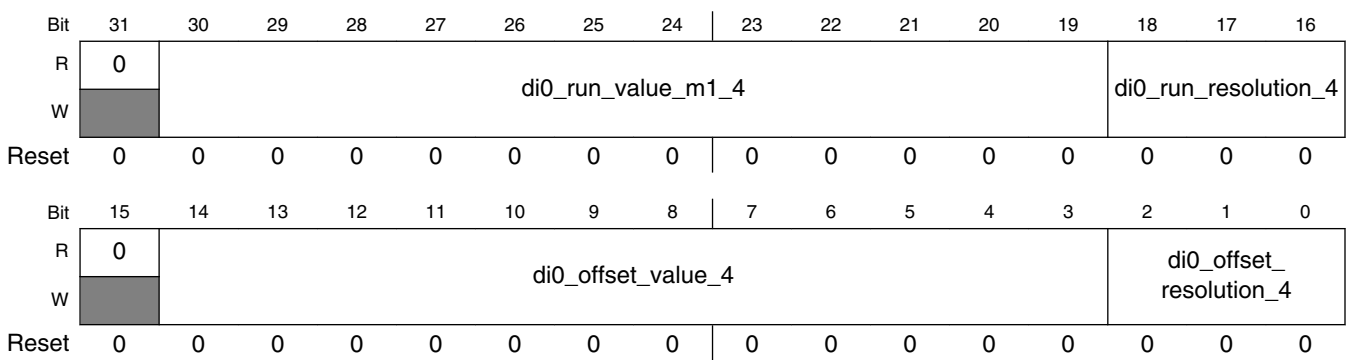
Table continues on the next page...

IPU_DI0_SW_GEN0_3 field descriptions (continued)

Field	Description
	— 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–3 di0_offset_ value_3	DI0 counter #3 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di0_offset_ resolution_3	DI0 counter #3 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.

45.51.159 DI0 Sync Wave Gen 4 Register 0 (IPU_DI0_SW_GEN0_4)

Address: IPU_DI0_SW_GEN0_4 is 1E00_0000h base + 40018h offset = 1E04_0018h



IPU_DI0_SW_GEN0_4 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DI0_SW_GEN0_4 field descriptions (continued)

Field	Description
30–19 di0_run_value_m1_4	DI0 counter #4 pre defined value This fields defines the counter #4 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_4 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.
18–16 di0_run_resolution_4	DI0 counter #4 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–3 di0_offset_value_4	DI0 counter #4 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di0_offset_resolution_4	DI0 counter #4 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.

45.51.160 DI0 Sync Wave Gen 5 Register 0 (IPU_DI0_SW_GEN0_5)

Address: IPU_DI0_SW_GEN0_5 is 1E00_0000h base + 4001Ch offset = 1E04_001Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_run_value_m1_5												di0_run_resolution_5		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_offset_value_5												di0_offset_ resolution_5		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN0_5 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–19 di0_run_value_ m1_5	DI0 counter #5 pre defined value This fields defines the counter #5 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_5 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.
18–16 di0_run_ resolution_5	DI0 counter #5 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–3 di0_offset_ value_5	DI0 counter #5 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di0_offset_ resolution_5	DI0 counter #5 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 -The Counter is triggered by counter #1

Table continues on the next page...

IPU_DI0_SW_GEN0_5 field descriptions (continued)

Field	Description
	011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 External VSYNC 111 Counter is always on.

45.51.161 DI0 Sync Wave Gen 6 Register 0 (IPU_DI0_SW_GEN0_6)

Address: IPU_DI0_SW_GEN0_6 is 1E00_0000h base + 40020h offset = 1E04_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_run_value_m1_6												di0_run_resolution_6		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_offset_value_6												di0_offset_ resolution_6		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN0_6 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–19 di0_run_value_ m1_6	DI0 counter #6 pre defined value This fields defines the counter #6 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_6 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.
18–16 di0_run_ resolution_6	DI0 counter #6 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DI0_SW_GEN0_6 field descriptions (continued)

Field	Description
14–3 di0_offset_ value_6	DI0 counter #6 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di0_offset_ resolution_6	DI0 counter #6 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

45.51.162 DI0 Sync Wave Gen 7 Register 0 (IPU_DI0_SW_GEN0_7)

Address: IPU_DI0_SW_GEN0_7 is 1E00_0000h base + 40024h offset = 1E04_0024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_run_value_m1_7												di0_run_resolution_7		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_offset_value_7												di0_offset_ resolution_1		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN0_7 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–19 di0_run_value_ m1_7	DI0 counter #7 pre defined value This fields defines the counter #7 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_7 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.
18–16 di0_run_ resolution_7	DI0 counter #1 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled

Table continues on the next page...

IPU_DI0_SW_GEN0_7 field descriptions (continued)

Field	Description
	001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–3 di0_offset_ value_7	DI0 counter #7 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di0_offset_ resolution_1	DI0 counter #7 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

45.51.163 DI0 Sync Wave Gen 8 Register 0 (IPU_DI0_SW_GEN0_8)

Address: IPU_DI0_SW_GEN0_8 is 1E00_0000h base + 40028h offset = 1E04_0028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_run_value_m1_8												di0_run_resolution_8		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

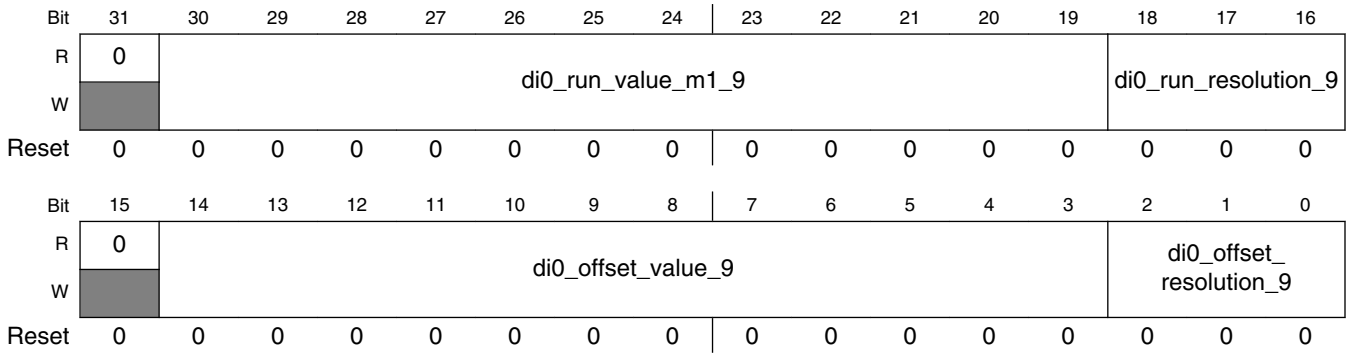
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_offset_value_8												di0_offset_ resolution_8		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN0_8 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–19 di0_run_value_m1_8	DI0 counter #8 pre defined value This fields defines the counter #8 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_8 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.
18–16 di0_run_resolution_8	DI0 counter #8 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–3 di0_offset_value_8	DI0 counter #8 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di0_offset_resolution_8	DI0 counter #8 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

45.51.164 DI0 Sync Wave Gen 9 Register 0 (IPU_DI0_SW_GEN0_9)

Address: IPU_DI0_SW_GEN0_9 is 1E00_0000h base + 4002Ch offset = 1E04_002Ch



IPU_DI0_SW_GEN0_9 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–19 di0_run_value_ m1_9	DI0 counter #9 pre defined value This fields defines the counter #9 pre defines value. When the counter is auto reload mode (di0_cnt_auto_reload_9 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di0_step_repeat field.
18–16 di0_run_ resolution_9	DI0 counter #9 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–3 di0_offset_ value_9	DI0 counter #9 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di0_offset_ resolution_9	DI0 counter #9 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1

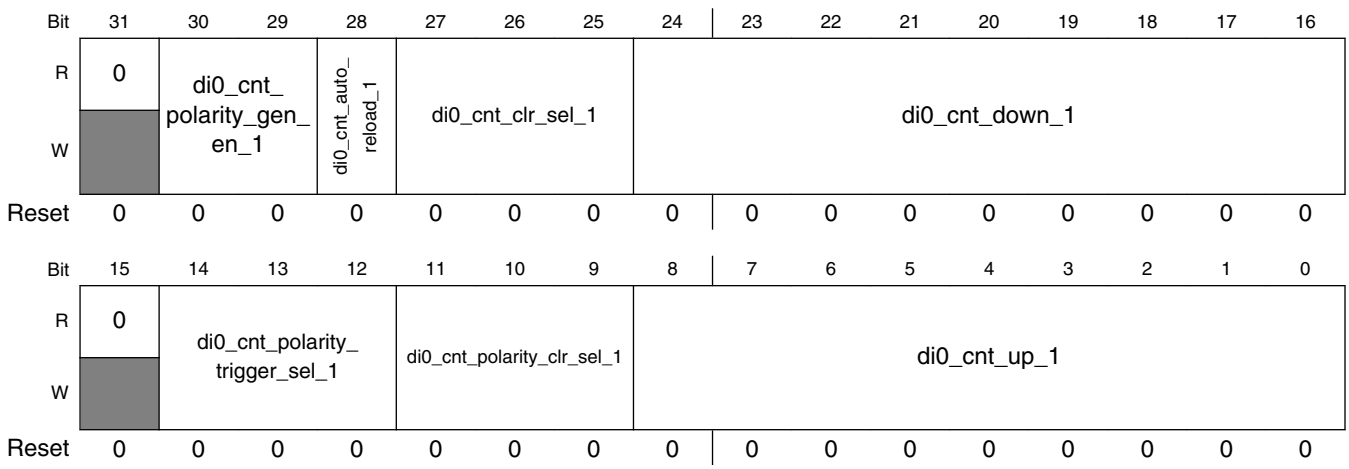
Table continues on the next page...

IPU_DI0_SW_GEN0_9 field descriptions (continued)

Field	Description
	011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

45.51.165 DI0 Sync Wave Gen 1 Register 1 (IPU_DI0_SW_GEN1_1)

Address: IPU_DI0_SW_GEN1_1 is 1E00_0000h base + 40030h offset = 1E04_0030h



IPU_DI0_SW_GEN1_1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–29 di0_cnt_polarity_ gen_en_1	DI0 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di0_cnt_auto_ reload_1	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_1 field

Table continues on the next page...

IPU_DI0_SW_GEN1_1 field descriptions (continued)

Field	Description
27–25 di0_cnt_clr_sel_1	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 Reserved 011 Reserved 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
24–16 di0_cnt_down_1	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 di0_cnt_polarity_trigger_sel_1	DI0 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 Reserved 011 Reserved 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
11–9 di0_cnt_polarity_clr_sel_1	DI0 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Reserved 011 Reserved 100 Reserved 101 Reserved 110 Reserved 111 Reserved
8–0 di0_cnt_up_1	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16).

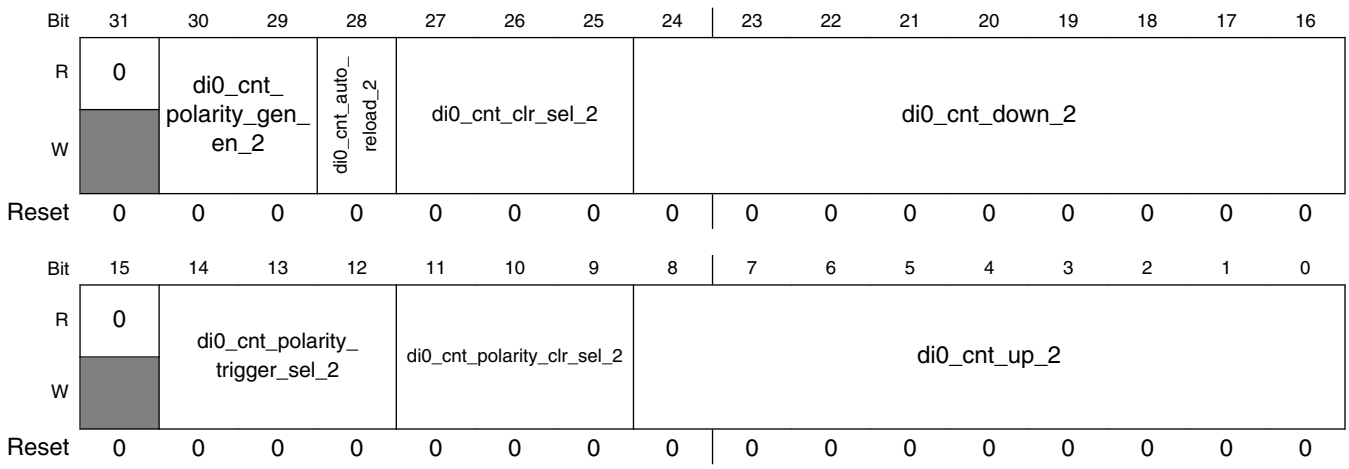
Table continues on the next page...

IPU_DI0_SW_GEN1_1 field descriptions (continued)

Field	Description
	The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.166 DI0 Sync Wave Gen 2 Register 1 (IPU_DI0_SW_GEN1_2)

Address: IPU_DI0_SW_GEN1_2 is 1E00_0000h base + 40034h offset = 1E04_0034h



IPU_DI0_SW_GEN1_2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–29 di0_cnt_polarity_ gen_en_2	DI0 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di0_cnt_auto_ reload_2	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_<i> field
27–25 di0_cnt_clr_sel_ 2	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock.

Table continues on the next page...

IPU_DI0_SW_GEN1_2 field descriptions (continued)

Field	Description
	010 The Counter is triggered by counter #1 011 Reserved 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
24–16 di0_cnt_down_2	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 di0_cnt_polarity_trigger_sel_2	DI0 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 Reserved 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
11–9 di0_cnt_polarity_clr_sel_2	DI0 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Reserved 011 Reserved 100 Reserved 101 Reserved 110 Reserved 111 Reserved
8–0 di0_cnt_up_2	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.167 DI0 Sync Wave Gen 3 Register 1 (IPU_DI0_SW_GEN1_3)

Address: IPU_DI0_SW_GEN1_3 is 1E00_0000h base + 40038h offset = 1E04_0038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_cnt_polarity_gen_en_3		di0_cnt_auto_reload_3	di0_cnt_clr_sel_3				di0_cnt_down_3							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_cnt_polarity_trigger_sel_3			di0_cnt_polarity_clr_sel_3				di0_cnt_up_3							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN1_3 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–29 di0_cnt_polarity_gen_en_3	DI0 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di0_cnt_auto_reload_3	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_<i> field
27–25 di0_cnt_clr_sel_3	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. —

Table continues on the next page...

IPU_DI0_SW_GEN1_3 field descriptions (continued)

Field	Description
	110 External VSYNC 111 Counter is always on.
24–16 di0_cnt_down_3	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 di0_cnt_polarity_trigger_sel_3	DI0 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
11–9 di0_cnt_polarity_clr_sel_3	DI0 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Reserved 101 Reserved 110 Reserved 111 Reserved
8–0 di0_cnt_up_3	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.168 DI0 Sync Wave Gen 4 Register 1 (IPU_DI0_SW_GEN1_4)

Address: IPU_DI0_SW_GEN1_4 is 1E00_0000h base + 4003Ch offset = 1E04_003Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_cnt_polarity_gen_en_4		di0_cnt_auto_reload_4	di0_cnt_clr_sel_4				di0_cnt_down_4							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_cnt_polarity_trigger_sel_4			di0_cnt_polarity_clr_sel_4				di0_cnt_up_4							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN1_4 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–29 di0_cnt_polarity_gen_en_4	DI0 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di0_cnt_auto_reload_4	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_<i> field
27–25 di0_cnt_clr_sel_4	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. —

Table continues on the next page...

IPU_DI0_SW_GEN1_4 field descriptions (continued)

Field	Description
	110 External VSYNC 111 Counter is always on.
24–16 di0_cnt_down_4	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 di0_cnt_polarity_trigger_sel_4	DI0 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
11–9 di0_cnt_polarity_clr_sel_4	DI0 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Reserved 110 Reserved 111 Reserved
8–0 di0_cnt_up_4	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.169 DI0 Sync Wave Gen 5 Register 1 (IPU_DI0_SW_GEN1_5)

Address: IPU_DI0_SW_GEN1_5 is 1E00_0000h base + 40040h offset = 1E04_0040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_cnt_polarity_gen_en_5		di0_cnt_auto_reload_5	di0_cnt_clr_sel_5				di0_cnt_down_5							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_cnt_polarity_trigger_sel_5			di0_cnt_polarity_clr_sel_5				di0_cnt_up_5							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN1_5 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–29 di0_cnt_polarity_gen_en_5	DI0 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di0_cnt_auto_reload_5	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_<i> field
27–25 di0_cnt_clr_sel_5	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 External VSYNC 111 Counter is always on.

Table continues on the next page...

IPU_DI0_SW_GEN1_5 field descriptions (continued)

Field	Description
24–16 di0_cnt_down_5	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 di0_cnt_polarity_trigger_sel_5	DI0 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 External VSYNC 111 Counter is always on.
11–9 di0_cnt_polarity_clr_sel_5	DI0 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Output is inverted if the output of counter #4 is set 110 Reserved 111 Reserved
8–0 di0_cnt_up_5	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.170 DI0 Sync Wave Gen 6 Register 1 (IPU_DI0_SW_GEN1_6)

Address: IPU_DI0_SW_GEN1_6 is 1E00_0000h base + 40044h offset = 1E04_0044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_cnt_polarity_gen_en_6		di0_cnt_auto_reload_6	di0_cnt_clr_sel_6				di0_cnt_down_6							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_cnt_polarity_trigger_sel_6			di0_cnt_polarity_clr_sel_6				di0_cnt_up_6							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN1_6 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–29 di0_cnt_polarity_gen_en_6	DI0 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di0_cnt_auto_reload_6	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_<i> field
27–25 di0_cnt_clr_sel_6	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

Table continues on the next page...

IPU_DI0_SW_GEN1_6 field descriptions (continued)

Field	Description
24–16 di0_cnt_down_6	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 di0_cnt_polarity_trigger_sel_6	DI0 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
11–9 di0_cnt_polarity_clr_sel_6	DI0 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Output is inverted if the output of counter #4 is set 110 Output is inverted if the output of counter #5 is set 111 Reserved
8–0 di0_cnt_up_6	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.171 DI0 Sync Wave Gen 7 Register 1 (IPU_DI0_SW_GEN1_7)

Address: IPU_DI0_SW_GEN1_7 is 1E00_0000h base + 40048h offset = 1E04_0048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_cnt_polarity_gen_en_7		di0_cnt_auto_reload_7	di0_cnt_clr_sel_7				di0_cnt_down_7							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_cnt_polarity_trigger_sel_7			di0_cnt_polarity_clr_sel_7				di0_cnt_up_7							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN1_7 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–29 di0_cnt_polarity_gen_en_7	DI0 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di0_cnt_auto_reload_7	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_<i> field
27–25 di0_cnt_clr_sel_7	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

Table continues on the next page...

IPU_DI0_SW_GEN1_7 field descriptions (continued)

Field	Description
24–16 di0_cnt_down_7	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 di0_cnt_polarity_trigger_sel_7	DI0 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
11–9 di0_cnt_polarity_clr_sel_7	DI0 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Output is inverted if the output of counter #4 is set 110 Output is inverted if the output of counter #5 is set 111 Output is inverted if the output of counter #6 is set
8–0 di0_cnt_up_7	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.172 DI0 Sync Wave Gen 8 Register 1 (IPU_DI0_SW_GEN1_8)

Address: IPU_DI0_SW_GEN1_8 is 1E00_0000h base + 4004Ch offset = 1E04_004Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di0_cnt_polarity_gen_en_8		di0_cnt_auto_reload_8	di0_cnt_clr_sel_8				di0_cnt_down_8							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di0_cnt_polarity_trigger_sel_8			di0_cnt_polarity_clr_sel_8				di0_cnt_up_8							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SW_GEN1_8 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–29 di0_cnt_polarity_gen_en_8	DI0 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di0_cnt_auto_reload_8	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_<i> field
27–25 di0_cnt_clr_sel_8	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

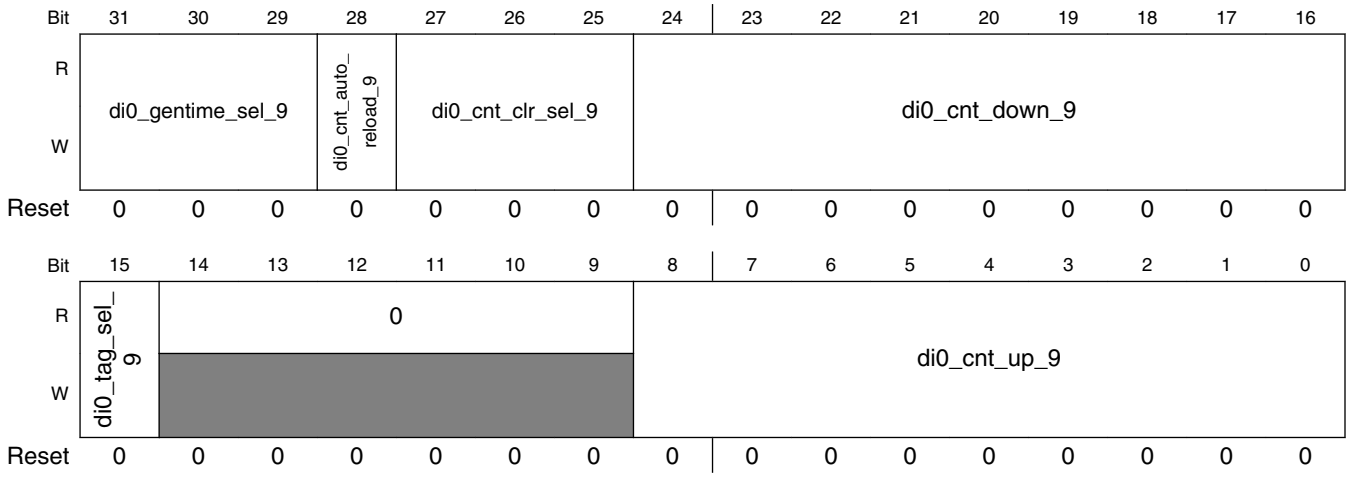
Table continues on the next page...

IPU_DI0_SW_GEN1_8 field descriptions (continued)

Field	Description
24–16 di0_cnt_down_8	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 di0_cnt_polarity_trigger_sel_8	DI0 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
11–9 di0_cnt_polarity_clr_sel_8	DI0 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Output is inverted if the output of counter #4 is set 110 Output is inverted if the output of counter #5 is set 111 Output is inverted if the output of counter #6 is set
8–0 di0_cnt_up_8	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.173 DI0 Sync Wave Gen 9 Register 1 (IPU_DI0_SW_GEN1_9)

Address: IPU_DI0_SW_GEN1_9 is 1E00_0000h base + 40050h offset = 1E04_0050h



IPU_DI0_SW_GEN1_9 field descriptions

Field	Description
31–29 di0_gentime_sel_9	Counter #9 main waveform select This field defines the counter that counter #9's auxiliary waveform will be attached too. 000 Counter #9's waveform is attached to counter #1's waveform 001 Counter #9's waveform is attached to counter #2's waveform 010 Counter #9's waveform is attached to counter #3's waveform 011 Counter #9's waveform is attached to counter #4's waveform 100 Counter #9's waveform is attached to counter #5's waveform 101 Counter #9's waveform is attached to counter #6's waveform 110 Counter #9's waveform is attached to counter #7's waveform 111 Counter #9's waveform is attached to counter #8's waveform
28 di0_cnt_auto_reload_9	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di0_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di0_step_repeat_<i> field
27–25 di0_cnt_clr_sel_9	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

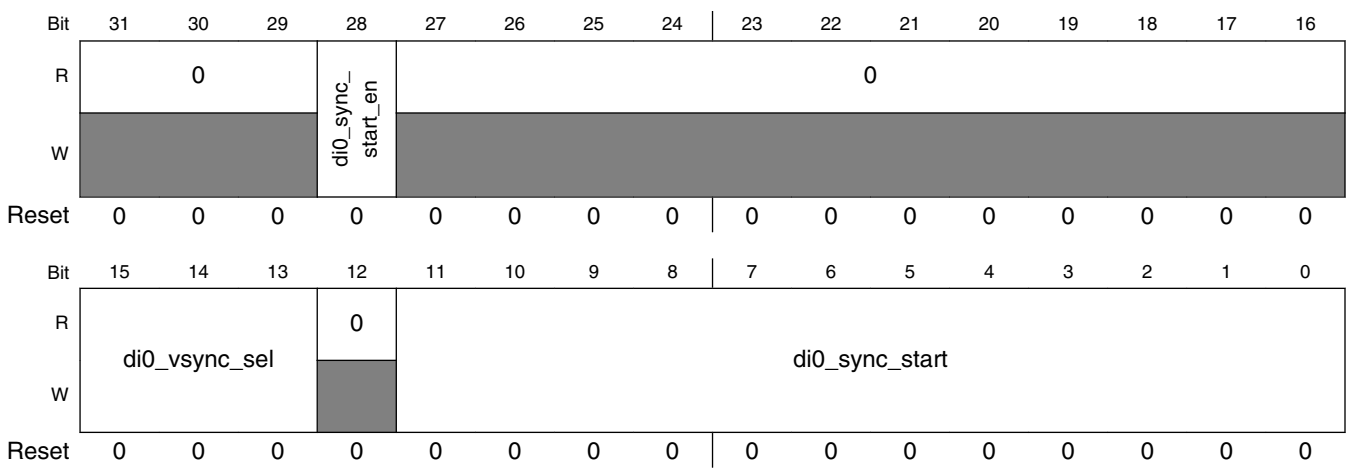
Table continues on the next page...

IPU_DI0_SW_GEN1_9 field descriptions (continued)

Field	Description
24–16 di0_cnt_down_9	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 di0_tag_sel_9	Counter #9 can send a synchronous tag when counter #9 reach its predefined value or when it's triggering counter reaches its pre defined value. 1 tag source is counter #9 0 Tag's source is the triggering counter.
14–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8–0 di0_cnt_up_9	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.174 DI0 Sync Assistance Gen Register (IPU_DI0_SYNC_AS_GEN)

Address: IPU_DI0_SYNC_AS_GEN is 1E00_0000h base + 40054h offset = 1E04_0054h



IPU_DI0_SYNC_AS_GEN field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28 di0_sync_start_en	di0_sync_start_en

Table continues on the next page...

IPU_DI0_SYNC_AS_GEN field descriptions (continued)

Field	Description
27–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–13 di0_vsync_sel	VSYNC select This field defines which of the counters functions as VSYNC signal 000 VSYNC is coming from counter #1 001 VSYNC is coming from counter #2 111 VSYNC is coming from counter #8
12 Reserved	This read-only field is reserved and always has the value zero. Reserved.
11–0 di0_sync_start	DI0 Sync start This field defines the number of low (including blanking rows) on the which the DI0 starts preparing the data for the next frame.

45.51.175 DI0 Data Wave Gen <i> Register (IPU_DI0_DW_GEN_i)

The DI0_DW_GEN_<i> register holds pointers for the waveform generators.

These registers have different bit arrangements for parallel and serial display. When using a parallel display VDI Plane Size Register 4 DI0 Data Wave Gen <i> Register is applicable. When using a serial interface VDI Plane Size Register 4 DI0 Data Wave Gen <i> Register is applicable.

Table 45-223. Register Field Descriptions for Serial Display

Field	Description
31-24 di0_serial_period_<i>	DI0 Serial Period <i> This field defines the period of the time base serial display clock. The units are the internal DI clock
23-16 di0_start_period_<i>	DI0 start period This field defines the amount of cycles between the point where the access is ready to be launched to the actual point where the time base serial display clock restarts. The units are the internal DI clock
15-14 di0_cst_<i>	DI0 Chip Select pointer for waveform <i> This field points to a register that defines the waveform of the CS pin. For serial displays the down value as defined on DI0_DW_SET*_<i> is measured from the assertion of the last serial display time base clock. 00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>

Table continues on the next page...

Table 45-223. Register Field Descriptions for Serial Display (continued)

Field	Description
13-9	Reserved
8-4 di0_serial_valid_bits<i>	DI0 Serial valid bits. This field defines the amount of valid bits to be transmitted within the 32 bits internal word aligned to bit[0]. The actual amount of valid bits is di0_serial_valid_bits_<i> + 1
3-2 di0_serial_rs_<i>	DI0 Serial RS This field points to a register that defines the waveform of the RS pin. For serial displays the down value as defined on DI0_DW_SET*_<i> is measured from the assertion of the last serial display time base clock. 00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>
1-0 di0_serial_clk_<i>	DI0 serial clock<i> This field points to a register that defines the waveform of the Serial clock pin. 00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>

Address: IPU_DI0_DW_GEN_i is 1E00_0000h base + 40058h offset = 1E04_0058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	di0_access_size_<i>								di0_componnent_size_<i>								di0_cst_<i>		di0_pt_6_<i>		di0_pt_5_<i>		di0_pt_4_<i>		di0_pt_3_<i>		di0_pt_2_<i>		di0_pt_1_<i>		di0_pt_0_<i>	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_DW_GEN_i field descriptions

Field	Description
31–24 di0_access_size_<i>	DI0 Access Size <i> This field defines the amount of IPU cycles between any 2 accesses (an access may be a pixel or generic data that may have more one component)
23–16 di0_componnent_size_<i>	DI0 component Size This field defines the amount of IPU cycles between any 2 components
15–14 di0_cst_<i>	DI0 Chip Select pointer for waveform <i> This field points to a register that defines the waveform of the CS pin. The CS is automatically mapped to a specific display 00 The waveform is defined according to the settings on DI0_DW_SET0_<i>

Table continues on the next page...

IPU_DI0_DW_GEN_i field descriptions (continued)

Field	Description
	01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>
13–12 di0_pt_6_<i>	DI0 PIN_17 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_17 pin. 00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>
11–10 di0_pt_5_<i>	DI0 PIN_16 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_16 pin. 00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>
9–8 di0_pt_4_<i>	DI0 PIN_15 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_15 pin. 00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>
7–6 di0_pt_3_<i>	DI0 PIN_14 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_14 pin. 00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>
5–4 di0_pt_2_<i>	DI0 PIN_13 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_13 pin. 00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>
3–2 di0_pt_1_<i>	DI0 PIN_12 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_12 pin. 00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>

Table continues on the next page...

IPU_DI0_DW_GEN_i field descriptions (continued)

Field	Description
1–0 di0_pt_0_<i>	DI0 PIN_11 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_11 pin. 00 The waveform is defined according to the settings on DI0_DW_SET0_<i> 01 The waveform is defined according to the settings on DI0_DW_SET1_<i> 10 The waveform is defined according to the settings on DI0_DW_SET2_<i> 11 The waveform is defined according to the settings on DI0_DW_SET3_<i>

45.51.176 DI0 Data Wave Set 0 <i> Register (IPU_DI0_DW_SET0_i)

Address: IPU_DI0_DW_SET0_i is 1E00_0000h base + 40088h offset = 1E04_0088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							di0_data_cnt_down0_<i>									0							di0_data_cnt_up0_<i>								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_DW_SET0_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 di0_data_cnt_down0_<i>	Waveform's falling edge position. This field defines the Waveform's falling edge position. The Waveform is mapped to a point according to the corresponding di0_pt_*_<i>
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 di0_data_cnt_up0_<i>	Waveform's rising edge position. This field defines the Waveform's rising edge position. The Waveform is mapped to a point according to the corresponding di0_pt_*_<i>

45.51.177 DI0 Data Wave Set 1 <i> Register (IPU_DI0_DW_SET1_i)

Address: IPU_DI0_DW_SET1_i is 1E00_0000h base + 400B8h offset = 1E04_00B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							di0_data_cnt_down1_<i>									0							di0_data_cnt_up1_<i>								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_DW_SET1_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 di0_data_cnt_down1_<i>	Waveform's falling edge position. This field defines the Waveform's falling edge position. The Waveform is mapped to a pint according to the corresponding di0_pt_*_<i>
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 di0_data_cnt_up1_<i>	Waveform's rising edge position. This field defines the Waveform's rising edge position. The Waveform is mapped to a pint according to the corresponding di0_pt_*_<i>

45.51.178 DI0 Data Wave Set 2 <i> Register (IPU_DI0_DW_SET2_i)

Address: IPU_DI0_DW_SET2_i is 1E00_0000h base + 400E8h offset = 1E04_00E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							di0_data_cnt_down2_<i>									0							di0_data_cnt_up2_<i>								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_DW_SET2_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 di0_data_cnt_down2_<i>	Waveform's falling edge position. This field defines the Waveform's falling edge position. The Waveform is mapped to a pint according to the corresponding di0_pt_*_<i>
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 di0_data_cnt_up2_<i>	Waveform's rising edge position. This field defines the Waveform's rising edge position. The Waveform is mapped to a pint according to the corresponding di0_pt_*_<i>

45.51.179 DI0 Data Wave Set 3 <i> Register (IPU_DI0_DW_SET3_i)

Address: IPU_DI0_DW_SET3_i is 1E00_0000h base + 40118h offset = 1E04_0118h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							di0_data_cnt_down3_<i>									0							di0_data_cnt_up3_<i>								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_DW_SET3_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 di0_data_cnt_down3_<i>	Waveform's falling edge position. This field defines the Waveform's falling edge position. The Waveform is mapped to a pint according to the corresponding di0_pt_*_<i>
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 di0_data_cnt_up3_<i>	Waveform's rising edge position. This field defines the Waveform's rising edge position. The Waveform is mapped to a pint according to the corresponding di0_pt_*_<i>

45.51.180 DI0 Step Repeat <i> Registers (IPU_DI0_STP_REP_i)

Address: IPU_DI0_STP_REP_i is 1E00_0000h base + 40148h offset = 1E04_0148h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W					di0_step_repeat_<2*i>																di0_step_repeat_<2*i-1>											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_STP_REP_i field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved
27–16 di0_step_repeat_<2*i>	Step Repeat <i> This fields defines the amount of repetitions that will be performed by the counter <i>
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved

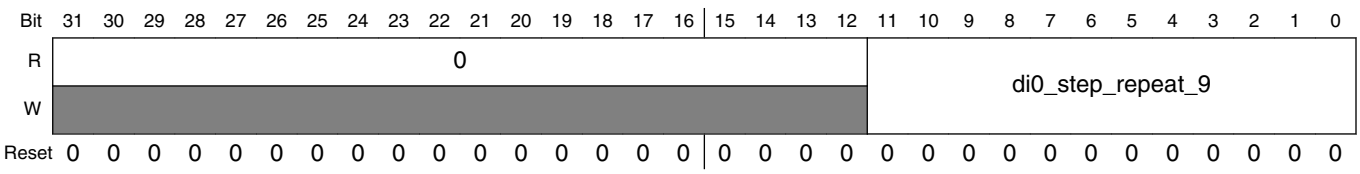
Table continues on the next page...

IPU_DI0_STP_REP_i field descriptions (continued)

Field	Description
11–0 di0_step_ repeat_<2*i-1>	Step Repeat <i> This fields defines the amount of repetitions that will be performed by the counter <i>

45.51.181 DI0 Step Repeat 9 Registers (IPU_DI0_STP_REP_9)

Address: IPU_DI0_STP_REP_9 is 1E00_0000h base + 40158h offset = 1E04_0158h

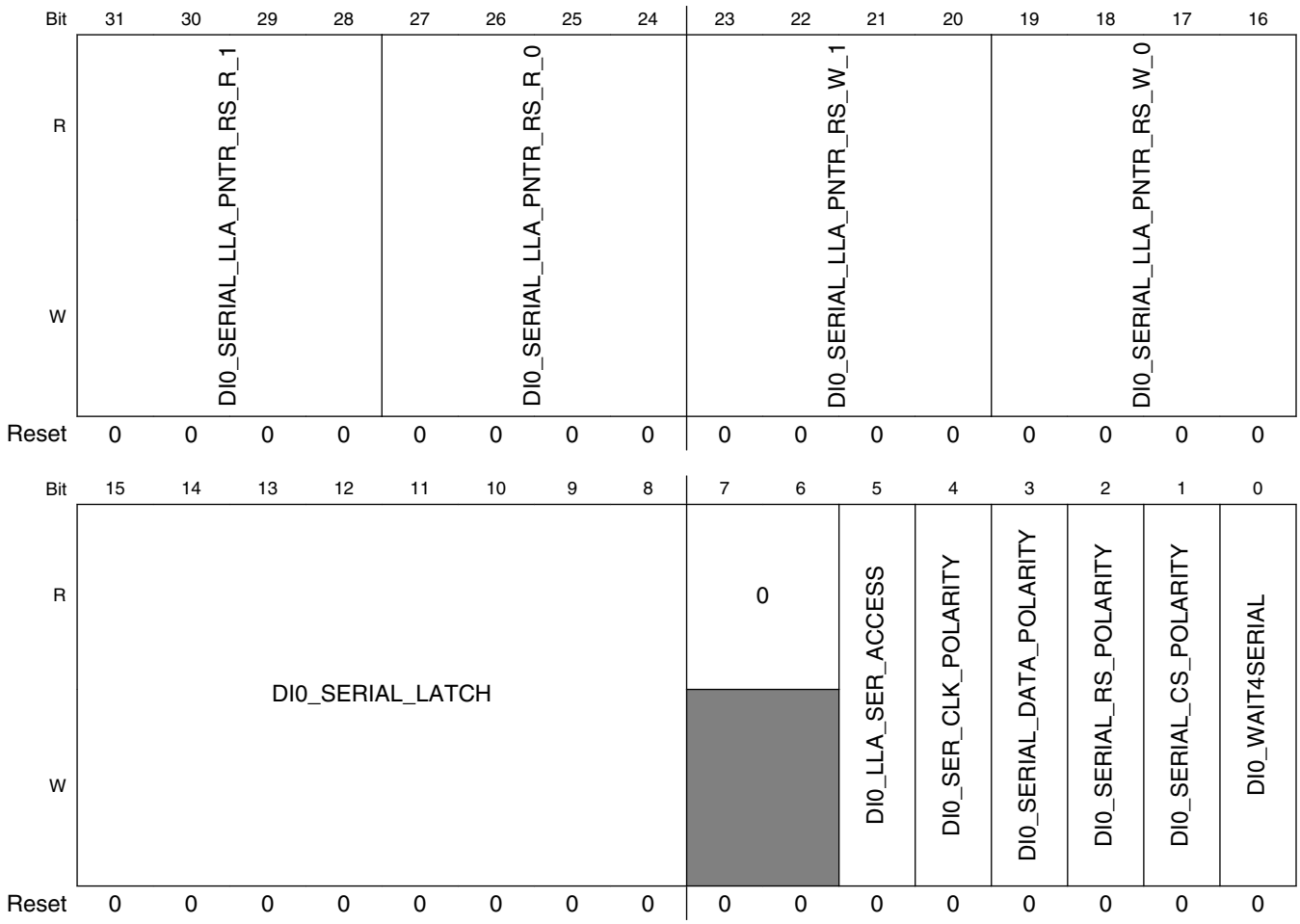


IPU_DI0_STP_REP_9 field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11–0 di0_step_ repeat_9	Step Repeat 9 This fields defines the amount of repetitions that will be performed by the counter 9

45.51.182 DI0 Serial Display Control Register (IPU_DI0_SER_CONF)

Address: IPU_DI0_SER_CONF is 1E00_0000h base + 4015Ch offset = 1E04_015Ch



IPU_DI0_SER_CONF field descriptions

Field	Description
31–28 DI0_SERIAL_LLA_PNTR_RS_R_1	RS 3 waveform pointer for read low level access This pointer defines which waveform set will be chosen when the read low level access is targeted to RS group 1. 0000 Waveform set #1 0001 Waveform set #2 1011 Waveform set #12 1100 Reserved 1101 Reserved 1100 Reserved

Table continues on the next page...

IPU_DI0_SER_CONF field descriptions (continued)

Field	Description
27–24 DI0_SERIAL_LLA_PNTR_RS_R_0	RS 2 waveform pointer for read low level access This pointer defines which waveform set will be chosen when the read low level access is targeted to RS group 0. 0000 Waveform set #1 0001 Waveform set #2 1011 Waveform set #12 1100 Reserved 1101 Reserved 1100 Reserved
23–20 DI0_SERIAL_LLA_PNTR_RS_W_1	RS 1 waveform pointer for write low level access This pointer defines which waveform set will be chosen when the low level write access is targeted to RS group 1. 0000 Waveform set #1 0001 Waveform set #2 1011 Waveform set #12 1100 Reserved 1101 Reserved 1100 Reserved
19–16 DI0_SERIAL_LLA_PNTR_RS_W_0	RS 0 waveform pointer for write low level access This pointer defines which waveform set will be chosen when the low level write access is targeted to RS group 0. 0000 Waveform set #1 0001 Waveform set #2 1011 Waveform set #12 1100 Reserved 1101 Reserved 1100 Reserved
15–8 DI0_SERIAL_LATCH	DI0 Serial Latch This field defines how many cycles to insert between serial read accesses start to data sampling in the
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5 DI0_LLA_SER_ACCESS	Direct Low Level Access to Serial display 1 ARM platform access is performed via a direct path to the serial display in LLA mode, in this mode only the ARM platform in LLA mode can access the serial port 0 ARM platform access to the serial display port is not done directly, hence other source are allowed to access the serial port. The arbitration is done automatically
4 DI0_SER_CLK_POLARITY	Serial Clock Polarity The output polarity of the SER_CLK pin 1 The clock is inverted 0 The clock is not inverted

Table continues on the next page...

IPU_DI0_SER_CONF field descriptions (continued)

Field	Description
3 DI0_SERIAL_ DATA_ POLARITY	Serial Data Polarity The output polarity of the SER_DATA pin 1 The data is inverted 0 The data is not inverted
2 DI0_SERIAL_ RS_POLARITY	Serial RS Polarity The output polarity of the SER_RS pin 1 The RS is inverted 0 The RS is not inverted
1 DI0_SERIAL_ CS_POLARITY	Serial Chip Select Polarity The output polarity of the SER_CS pin 1 The CS is inverted 0 The CS is not inverted
0 DI0_ WAIT4SERIAL	Wait for Serial When the parallel display share pins with the serial port. Accessing the parallel port is not allowed till the serial port completes its access. 1 The parallel port should wait to the serial port as the pins are shared 0 The parallel port should not wait to the serial port as the pins are not shared

45.51.183 DI0 Special Signals Control Register (IPU_DI0_SSC)

Address: IPU_DI0_SSC is 1E00_0000h base + 40160h offset = 1E04_0160h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								DI0_PIN17_ERM	DI0_PIN16_ERM	DI0_PIN15_ERM	DI0_PIN14_ERM	DI0_PIN13_ERM	DI0_PIN12_ERM	DI0_PIN11_ERM	DI0_CS_ERM
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										DI0_WAIT_ON	0	DI0_BYTE_EN_RD_IN	DI0_BYTE_EN_PNTR		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_SSC field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved.
23 DI0_PIN17_ERM	DI0 PIN17 error recovery mode. This bit defines the error recovery mode of the PIN17 pin. 1 The PIN17 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC 0 Nothing is done to the PIN17 pin following a display error detection
22 DI0_PIN16_ERM	DI0 PIN16 error recovery mode. This bit defines the error recovery mode of the PIN16 pin. 1 The PIN16 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC 0 Nothing is done to the PIN16 pin following a display error detection
21 DI0_PIN15_ERM	DI0 PIN15 error recovery mode. This bit defines the error recovery mode of the PIN15 pin. 1 The PIN15 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC 0 Nothing is done to the PIN15 pin following a display error detection
20 DI0_PIN14_ERM	DI0 PIN14 error recovery mode. This bit defines the error recovery mode of the PIN14 pin. 1 The PIN14 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC 0 Nothing is done to the PIN14 pin following a display error detection
19 DI0_PIN13_ERM	DI0 PIN13 error recovery mode. This bit defines the error recovery mode of the PIN13 pin. 1 The PIN13 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC 0 Nothing is done to the PIN13 pin following a display error detection
18 DI0_PIN12_ERM	DI0 PIN12 error recovery mode. This bit defines the error recovery mode of the PIN12 pin. 1 The PIN12 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC 0 Nothing is done to the PIN12 pin following a display error detection
17 DI0_PIN11_ERM	DI0 PIN11 error recovery mode. This bit defines the error recovery mode of the PIN11 pin. 1 The PIN11 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC 0 Nothing is done to the PIN11 pin following a display error detection
16 DI0_CS_ERM	DI0 GLUELOGIC error recovery mode. This bit defines the error recovery mode of the GLUELOGIC.

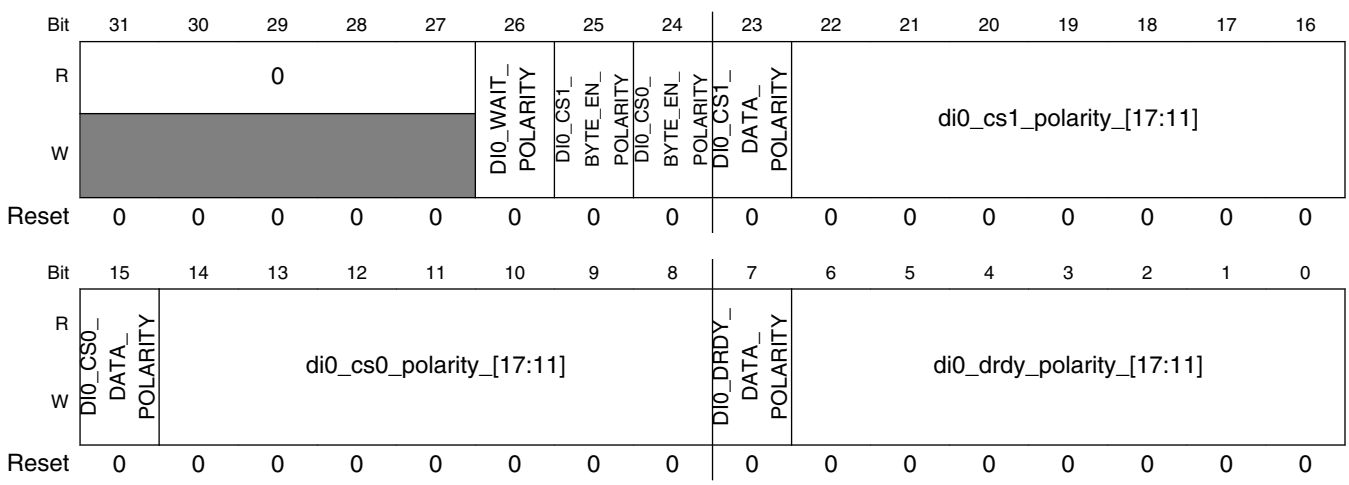
Table continues on the next page...

IPU_DI0_SSC field descriptions (continued)

Field	Description
	1 The GLUELOGIC is release in case of a synchronous display error. The release will be done on the next VSYNC 0 Nothing is done to the GLUELOGIC following a display error detection
15–6 Reserved	This read-only field is reserved and always has the value zero. Reserved.
5 DI0_WAIT_ON	Wait On This field defines the DC's response to WAIT signal 1 The DC holds the flow as long as WAIT is asserted 0 The DC continues the flow regardless the WAIT signal
4 Reserved	This read-only field is reserved and always has the value zero. Reserved
3 DI0_BYTE_EN_RD_IN	Byte Enable Read In This bit selects the source of the byte enable pins 1 The write byte enable signals are routed via bits [17:16] of the display's data, The read byte enable signals are routed via bits [19:18 of the display's data 0 The byte enable signals are routed via bits [17:16] of the display's data for both read and write
2–0 DI0_BYTE_EN_PNTR	Byte Enable Pointer This pointer selects the pin asserted along with the byte enables signals 000 wave form of byte enable as pin_11 001 wave form of byte enable as pin_12 111 wave form of byte enable as suitable CS pin

45.51.184 DI0 Polarity Register (IPU_DI0_POL)

Address: IPU_DI0_POL is 1E00_0000h base + 40164h offset = 1E04_0164h



IPU_DI0_POL field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26 DI0_WAIT_ POLARITY	WAIT polarity This bit defines the polarity of the wait signal input coming from the displa1 1 active high 0 active low
25 DI0_CS1_ BYTE_EN_ POLARITY	Byte Enable associated with CS1 polarity This bit defines the polarity of the byte enable signals to the display 1 active high 0 active low
24 DI0_CS0_ BYTE_EN_ POLARITY	Byte Enable associated with CS0 polarity This bit defines the polarity of the byte enable signals to the display 1 active high 0 active low
23 DI0_CS1_ DATA_ POLARITY	Data Polarity associated with CS1
22–16 di0_cs1_ polarity_[17:11]	DI0 output pin's polarity for CS1 This bits define the polarity of each of the DI's outputs when CS1 is asserted 1 The output pin is active high 0 The output pin is active low
15 DI0_CS0_ DATA_ POLARITY	Data Polarity associated with CS0
14–8 di0_cs0_ polarity_[17:11]	DI0 output pin's polarity for CS1 This bits define the polarity of each of the DI's outputs when CS1 is asserted 1 The output pin is active high 0 The output pin is active low
7 DI0_DRDY_ DATA_ POLARITY	Data Polarity associated with DRDY
6–0 di0_drdy_ polarity_[17:11]	DI0 output dynamic pin's polarity for synchronous access This bits define the polarity of each of the DI's outputs when synchronous display access is asserted The pins' default polarity is the same as defined in the di0_drdy_polarity_[17:11] bits 1 The output pin is active high 0 The output pin is active low

45.51.185 DI0 Active Window 0 Register (IPU_DI0_AW0)

Address: IPU_DI0_AW0 is 1E00_0000h base + 40168h offset = 1E04_0168h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DI0_AW_TRIG_SEL				DI0_AW_HEND												DI0_AW_HCOUNT_SEL				DI0_AW_HSTART											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_AW0 field descriptions

Field	Description
31–28 DI0_AW_TRIG_SEL	This field selects the trigger for sending data during the display's active window 000 disabled 001 The trigger is the same trigger that triggers the displays clock. 010 The trigger is counter #1 011 The trigger is counter #2 100 The trigger is counter #3 101 The trigger is counter #4 110 The trigger is counter #5 111 The trigger is always on.
27–16 DI0_AW_HEND	This field defines the horizontal end of the active window
15–12 DI0_AW_HCOUNT_SEL	GM: This field selects the counter that counts the horizontal position of the display's active window 0000 disabled 0001 reserved 0010 The counter is counter #1 0011 The counter is counter #2 0100 The counter is counter #3 0101 The counter is counter #4 0110 The counter is counter #5 1001 The counter is counter #8
11–0 DI0_AW_HSTART	This field defines the horizontal start of the active window DI0_AW_HSTART < DI0_AW_HEND

45.51.186 DI0 Active Window 1 Register (IPU_DI0_AW1)

Address: IPU_DI0_AW1 is 1E00_0000h base + 4016Ch offset = 1E04_016Ch

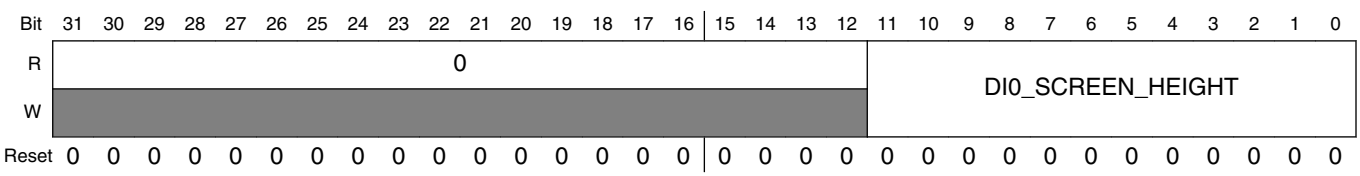
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DI0_AW_VEND												DI0_AW_VCOUNT_SEL				DI0_AW_VSTART											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI0_AW1 field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved
27–16 DI0_AW_VEND	This field defines the vertical end of the active window
15–12 DI0_AW_ VCOUNT_SEL	This field selects the counter that counts the vertical position of the display's active window 0000 disabled 0001 reserved 0010 The counter is counter #1 0011 The counter is counter #2 0100 The counter is counter #3 0101 The counter is counter #4 0110 The counter is counter #5 1001 The counter is counter #8
11–0 DI0_AW_ VSTART	This field defines the vertical start of the active window DI0_AW_VSTART < DI0_AW_VEND

45.51.187 DI0 Screen Configuration Register (IPU_DI0_SCR_CONF)

Address: IPU_DI0_SCR_CONF is 1E00_0000h base + 40170h offset = 1E04_0170h

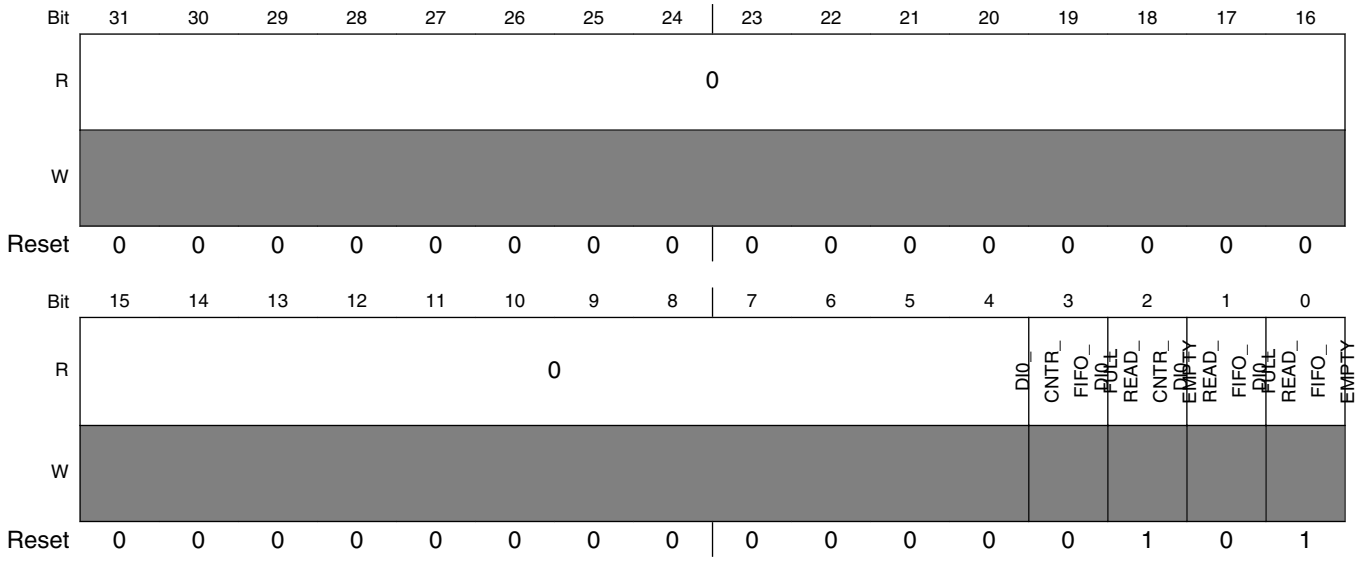


IPU_DI0_SCR_CONF field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero. Reserved.
11–0 DI0_SCREEN_ HEIGHT	This field defines the number of display rows (Number_of_ROWS = DI0_SCREEN_HEIGHT+1) This field is used for VSYNC calculation and for anti-tearing

45.51.188 DI0 Status Register (IPU_DI0_STAT)

Address: IPU_DI0_STAT is 1E00_0000h base + 40174h offset = 1E04_0174h

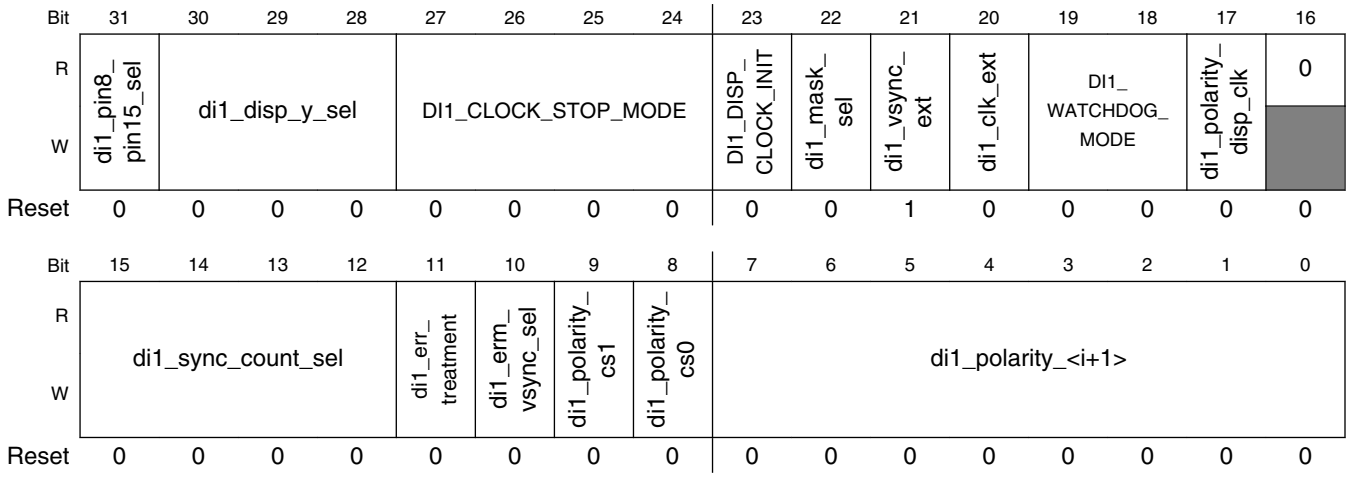


IPU_DI0_STAT field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value zero. Reserved
3 DI0_CNTR_ FIFO_FULL	This bit indicates a full state of the DI0 FIFO. This FIFO is part of the DI0 synchronizer.
2 DI0_READ_ CNTR_EMPTY	This bit indicates an empty state of the DI0 FIFO. This FIFO is part of the DI0 synchronizer.
1 DI0_READ_ FIFO_FULL	This bit indicates a full state of the DI0 FIFO when performing a read. This FIFO is part of the DI0 synchronizer.
0 DI0_READ_ FIFO_EMPTY	This bit indicates an empty state of the DI0 FIFO when performing a read. This FIFO is part of the DI0 synchronizer.

45.51.189 DI1General Register (IPU_DI1_GENERAL)

Address: IPU_DI1_GENERAL is 1E00_0000h base + 48000h offset = 1E04_8000h



IPU_DI1_GENERAL field descriptions

Field	Description
31 di1_pin8_pin15_sel	This bit routes PIN8 over PIN15 1 PIN8 is routed to PIN15, PIN8 is also routed to PIN8 0 PIN15 is routed to PIN15, PIN8 is routed to PIN8
30–28 di1_disp_y_sel	DI1 Display Vertical coordinate (Y) select. This field defines which one of the 8 counters will be used as a display's line counter. 000 counter #1 is selected 111 counter #8 is selected
27–24 DI1_CLOCK_STOP_MODE	DI clock stop mode When performing a clock change. The DI stops the clock to the display. These field defines when the clock will be stopped Stopping at EOL/EOF is supported for the case where the data is coming from the IDMAC (DMA access). In case that only direct accesses is performed, the user should set this field to 0000 0001-1001 stop at the next event of one of the counters (counter #1 to counter #9) 0000 stop at the next edge of the display clock 1100 stop at EOL (end of a line), but if stop request is during blanking interval, stop now 1101 stop at EOF (end of a frame), but if stop request is during blanking interval, stop now 1110 stop at EOL (end of a line), but if stop request is during blanking interval, stop at the end of the next line 1111 stop at EOF (end of a frame), but if stop request is during blanking interval, stop at the end of the next frame
23 DI1_DISP_CLOCK_INIT	Display clock's initial mode For synchronization error conditions the display clock can be stopped on the next VSYNC

Table continues on the next page...

IPU_DI1_GENERAL field descriptions (continued)

Field	Description
	1 The display's clock is running after the next VSYNC (indicating new frame) 0 The display's clock is stopped after the next VSYNC (indicating new frame)
22 di1_mask_sel	DI1 Mask select. IPP_PIN_2 output of the DI that functions as MASK signal can come from 2 sources: counter #2 or extracted from the MASK data coming from the memory. 1 IPP_PIN_2 is coming from extracted MASK data coming from the memory 0 IPP_PIN_2 is coming from counter #2
21 di1_vsync_ext	DI1 External VSYNC. This bit selects the source of the VSYNC signal 1 External to the IPU 0 Internally generated by the IPU
20 di1_clk_ext	DI1 External Clock. This bit selects the source of the DI's clock 1 The source of the clock is external to the IPU 0 The clock is internally generated by the IPU
19–18 DI1_ WATCHDOG_ MODE	DI1 watchdog mode In case of a display error where the DI clock is stopped (defined at di0_err_treatment). An internal watchdog counts DI clocks. If this timer reached its pre defined value the DI will skip the current frame and restart on the frame. This 2 bits define the number of DI clock cycles that the timer counts. 00 The timer counts 4 DI cycles 01 The timer counts 16 DI cycles 10 The timer counts 64 DI cycles 11 The timer counts 128 DI cycles
17 di1_polarity_ disp_clk	DI1 Output Clock's polarity This bits define the polarity of the DI's clock. 1 The output clock is active high 0 The output clock is active low
16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–12 di1_sync_count_ sel	For synchronous flow error: selects synchronous flow synchronization counter in DI:
11 di1_err_ treatment	In case of synchronous flow error there are 2 ways to handle the display 1 to wait (i.e. stop clock) 0 Drive the last component
10 di1_erm_vsync_ sel	DI1 error recovery module's VSYNC source select The error recovery block detect a case where the DI's VSYNC is asserted before the EOF. This bit selects the source of the VSYNC signal monitored by this mechanism.

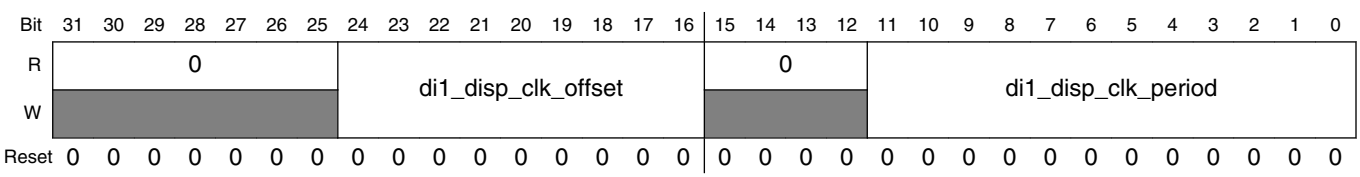
Table continues on the next page...

IPU_DI1_GENERAL field descriptions (continued)

Field	Description
	1 vsync_post - an internal VSYNC signal asserted 2 lines after the DI's VSYNC 0 vsync_pre - an internal VSYNC signal asserted 2 lines before the DI's VSYNC
9 di1_polarity_cs1	DI1 Chip Select's 1 polarity This bits define the polarity of the DI's CS1. 1 The CS1 is active high 0 The CS1 is active low
8 di1_polarity_cs0	DI1 Chip Select's 0 polarity This bits define the polarity of the DI's CS0. 1 The CS0 is active high 0 The CS0 is active low
7–0 di1_polarity_<i+1>	DI1 output pin's polarity This bits define the polarity of each of the DI's outputs. 1 The output pin is active high 0 The output pin is active low

45.51.190 DI1 Base Sync Clock Gen 0 Register (IPU_DI1_BS_CLKGEN0)

Address: IPU_DI1_BS_CLKGEN0 is 1E00_0000h base + 48004h offset = 1E04_8004h

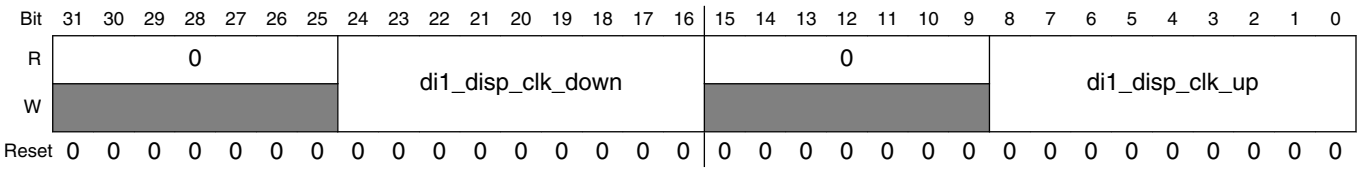


IPU_DI1_BS_CLKGEN0 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 di1_disp_clk_offset	DI1 Display Clock Offset The DI has the ability to delay the display's clock This field defines the amount of IPU's clock cycles added as delay on this clock.
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved.
11–0 di1_disp_clk_period	DI1 Display Clock Period This field defines the Display interface clock period for display write access. This parameter contains an integer part (bits 11:4) and a fractional part (bits 3:0). It defines a fractional division ratio of the HSP_CLK clock for generation of the display's interface clock.

45.51.191 DI1 Base Sync Clock Gen 1 Register (IPU_DI1_BS_CLKGEN1)

Address: IPU_DI1_BS_CLKGEN1 is 1E00_0000h base + 48008h offset = 1E04_8008h

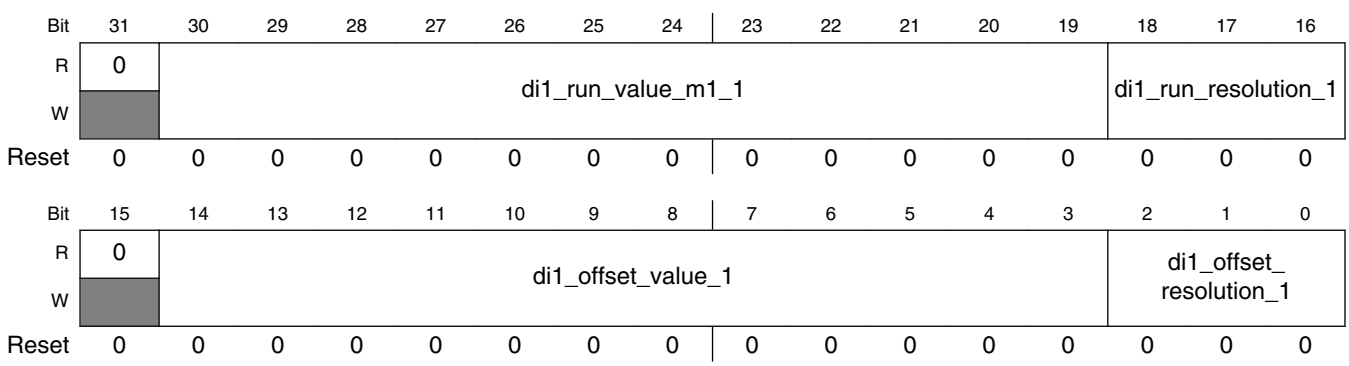


IPU_DI1_BS_CLKGEN1 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 di1_disp_clk_down	DI1 display clock falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is a time interval between display's access start point and display 's interface clock falling edge.
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 di1_disp_clk_up	DI1 display clock rising edge position This parameter contains an integer part (bits 8:1) and a fractional part (bit 0). The position value is a time interval between display's access start point and display 's interface clock rising edge.

45.51.192 DI1 Sync Wave Gen 1 Register 0 (IPU_DI1_SW_GEN0_1)

Address: IPU_DI1_SW_GEN0_1 is 1E00_0000h base + 4800Ch offset = 1E04_800Ch



IPU_DI1_SW_GEN0_1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–19 di1_run_value_m1_1	DI1 counter #1 pre defined value This fields defines the counter #1 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_1 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.
18–16 di1_run_resolution_1	DI1 counter #1 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 NA 011 NA 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–3 di1_offset_value_1	DI1 counter #1 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di1_offset_resolution_1	DI1 counter #1 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 NA 011 NA 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.

45.51.193 DI1 Sync Wave Gen 2 Register 0 (IPU_DI1_SW_GEN0_2)

Address: IPU_DI1_SW_GEN0_2 is 1E00_0000h base + 48010h offset = 1E04_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_run_value_m1_2												di1_run_resolution_2		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_offset_value_2												di1_offset_ resolution_2		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN0_2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–19 di1_run_value_ m1_2	DI1 counter #2 pre defined value This fields defines the counter #2 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_2 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.
18–16 di1_run_ resolution_2	DI1 counter #2 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 NA 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–3 di1_offset_ value_2	DI1 counter #2 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di1_offset_ resolution_2	DI1 counter #2 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled

Table continues on the next page...

IPU_DI1_SW_GEN0_2 field descriptions (continued)

Field	Description
	001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 NA 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.

45.51.194 DI1 Sync Wave Gen 3 Register 0 (IPU_DI1_SW_GEN0_3)

Address: IPU_DI1_SW_GEN0_3 is 1E00_0000h base + 48014h offset = 1E04_8014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_run_value_m1_3												di1_run_resolution_3		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_offset_value_3												di1_offset_ resolution_3		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN0_3 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–19 di1_run_value_ m1_3	DI1 counter #3 pre defined value This fields defines the counter #3 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_3 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.
18–16 di1_run_ resolution_3	DI1 counter #3 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit.

Table continues on the next page...

IPU_DI1_SW_GEN0_3 field descriptions (continued)

Field	Description
	— 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–3 di1_offset_value_3	DI1 counter #3 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di1_offset_resolution_3	DI1 counter #3 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 NA 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.

45.51.195 DI1 Sync Wave Gen 4 Register 0 (IPU_DI1_SW_GEN0_4)

Address: IPU_DI1_SW_GEN0_4 is 1E00_0000h base + 48018h offset = 1E04_8018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_run_value_m1_4												di1_run_resolution_4		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_offset_value_4												di1_offset_resolution_4		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN0_4 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.

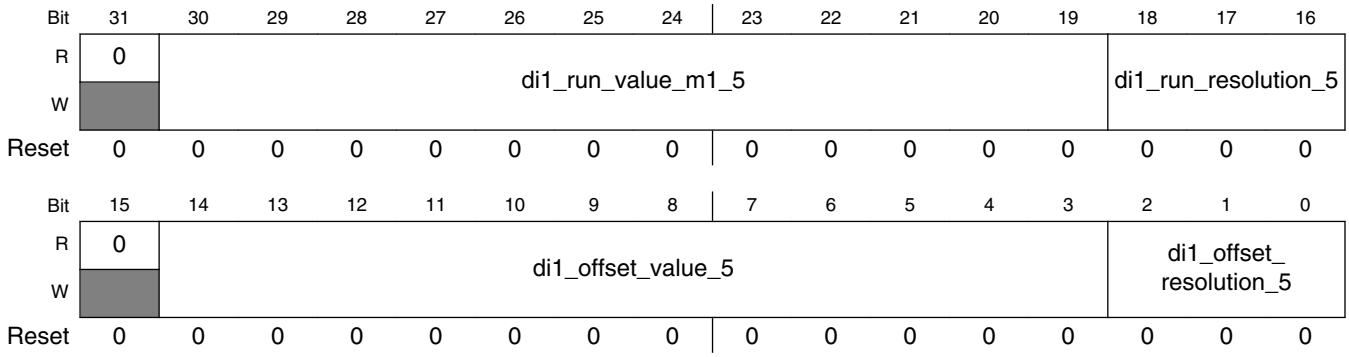
Table continues on the next page...

IPU_DI1_SW_GEN0_4 field descriptions (continued)

Field	Description
30–19 di1_run_value_m1_4	DI1 counter #4 pre defined value This fields defines the counter #4 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_4 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.
18–16 di1_run_resolution_4	DI1 counter #4 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–3 di1_offset_value_4	DI1 counter #4 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di1_offset_resolution_4	DI1 counter #4 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.

45.51.196 DI1 Sync Wave Gen 5 Register 0 (IPU_DI1_SW_GEN0_5)

Address: IPU_DI1_SW_GEN0_5 is 1E00_0000h base + 4801Ch offset = 1E04_801Ch



IPU_DI1_SW_GEN0_5 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30–19 di1_run_value_ m1_5	DI1 counter #5 pre defined value This fields defines the counter #5 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_5 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.
18–16 di1_run_ resolution_5	DI1 counter #5 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 External VSYNC 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–3 di1_offset_ value_5	DI1 counter #5 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di1_offset_ resolution_5	DI1 counter #5 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 -The Counter is triggered by counter #1

Table continues on the next page...

IPU_DI1_SW_GEN0_5 field descriptions (continued)

Field	Description
	011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 External VSYNC 111 Counter is always on.

45.51.197 DI1 Sync Wave Gen 6 Register 0 (IPU_DI1_SW_GEN0_6)

Address: IPU_DI1_SW_GEN0_6 is 1E00_0000h base + 48020h offset = 1E04_8020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_run_value_m1_6												di1_run_resolution_6		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_offset_value_6												di1_offset_ resolution_6		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN0_6 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–19 di1_run_value_ m1_6	DI1 counter #6 pre defined value This fields defines the counter #6 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_6 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.
18–16 di1_run_ resolution_6	DI1 counter #6 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IPU_DI1_SW_GEN0_6 field descriptions (continued)

Field	Description
14–3 di1_offset_ value_6	DI1 counter #6 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di1_offset_ resolution_6	DI1 counter #6 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

45.51.198 DI1 Sync Wave Gen 7 Register 0 (IPU_DI1_SW_GEN0_7)

Address: IPU_DI1_SW_GEN0_7 is 1E00_0000h base + 48024h offset = 1E04_8024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_run_value_m1_7												di1_run_resolution_7		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_offset_value_7												di1_offset_ resolution_1		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN0_7 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–19 di1_run_value_ m1_7	DI1 counter #7 pre defined value This fields defines the counter #7 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_7 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.
18–16 di1_run_ resolution_7	DI1 counter #1 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled

Table continues on the next page...

IPU_DI1_SW_GEN0_7 field descriptions (continued)

Field	Description
	001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–3 di1_offset_ value_7	DI1 counter #7 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di1_offset_ resolution_1	DI1 counter #7 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

45.51.199 DI1 Sync Wave Gen 8 Register 0 (IPU_DI1_SW_GEN0_8)

Address: IPU_DI1_SW_GEN0_8 is 1E00_0000h base + 48028h offset = 1E04_8028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_run_value_m1_8												di1_run_resolution_8		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_offset_value_8												di1_offset_ resolution_8		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN0_8 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–19 di1_run_value_m1_8	DI1 counter #8 pre defined value This fields defines the counter #8 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_8 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.
18–16 di1_run_resolution_8	DI1 counter #8 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–3 di1_offset_value_8	DI1 counter #8 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di1_offset_resolution_8	DI1 counter #8 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

45.51.200 DI1Sync Wave Gen 9 Register 0 (IPU_DI1_SW_GEN0_9)

Address: IPU_DI1_SW_GEN0_9 is 1E00_0000h base + 4802Ch offset = 1E04_802Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_run_value_m1_9												di1_run_resolution_9		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_offset_value_9												di1_offset_ resolution_9		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN0_9 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–19 di1_run_value_ m1_9	DI1 counter #9 pre defined value This fields defines the counter #9 pre defines value. When the counter is auto reload mode (di1_cnt_auto_reload_9 bit is set), the counter automatically restarts. otherwise the counter restarts for the amount of time defined on the corresponding di1_step_repeat field.
18–16 di1_run_ resolution_9	DI1 counter #9 Run Resolution This field defines the trigger causing the counter to increment. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–3 di1_offset_ value_9	DI1 counter #9 offset value The counter can start counting after a pre defined delay This field defines the amount of cycles that the counter will be delayed by
2–0 di1_offset_ resolution_9	DI1 counter #9 offset Resolution This field defines the trigger causing the offset counter to increment 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1

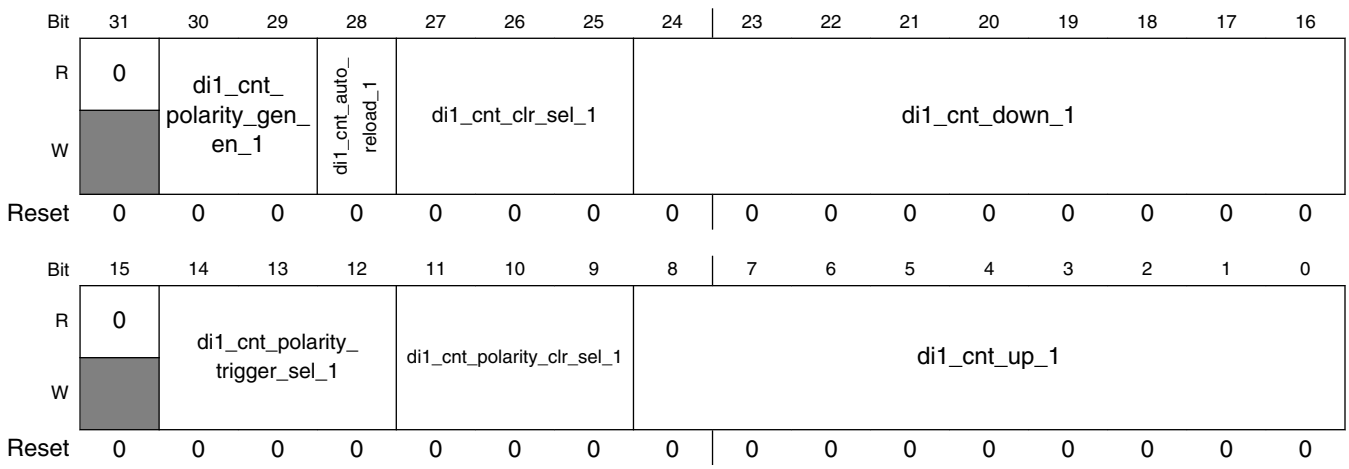
Table continues on the next page...

IPU_DI1_SW_GEN0_9 field descriptions (continued)

Field	Description
	011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

45.51.201 DI1 Sync Wave Gen 1 Register 1 (IPU_DI1_SW_GEN1_1)

Address: IPU_DI1_SW_GEN1_1 is 1E00_0000h base + 48030h offset = 1E04_8030h



IPU_DI1_SW_GEN1_1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–29 di1_cnt_polarity_ gen_en_1	DI1 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di1_cnt_auto_ reload_1	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_1 field

Table continues on the next page...

IPU_DI1_SW_GEN1_1 field descriptions (continued)

Field	Description
27–25 di1_cnt_clr_sel_1	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 Reserved 011 Reserved 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
24–16 di1_cnt_down_1	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 di1_cnt_polarity_trigger_sel_1	DI1 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 Reserved 011 Reserved 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
11–9 di1_cnt_polarity_clr_sel_1	DI1 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Reserved 011 Reserved 100 Reserved 101 Reserved 110 Reserved 111 Reserved
8–0 di1_cnt_up_1	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16).

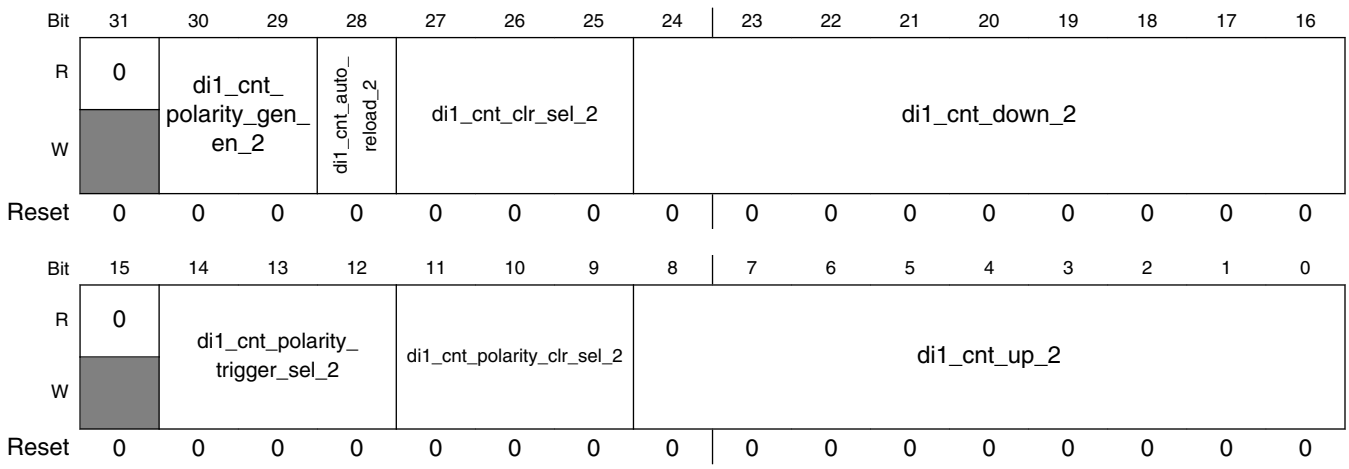
Table continues on the next page...

IPU_DI1_SW_GEN1_1 field descriptions (continued)

Field	Description
	The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.202 DI1 Sync Wave Gen 2 Register 1 (IPU_DI1_SW_GEN1_2)

Address: IPU_DI1_SW_GEN1_2 is 1E00_0000h base + 48034h offset = 1E04_8034h



IPU_DI1_SW_GEN1_2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–29 di1_cnt_polarity_ gen_en_2	DI1 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di1_cnt_auto_ reload_2	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_<i> field
27–25 di1_cnt_clr_sel_ 2	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock.

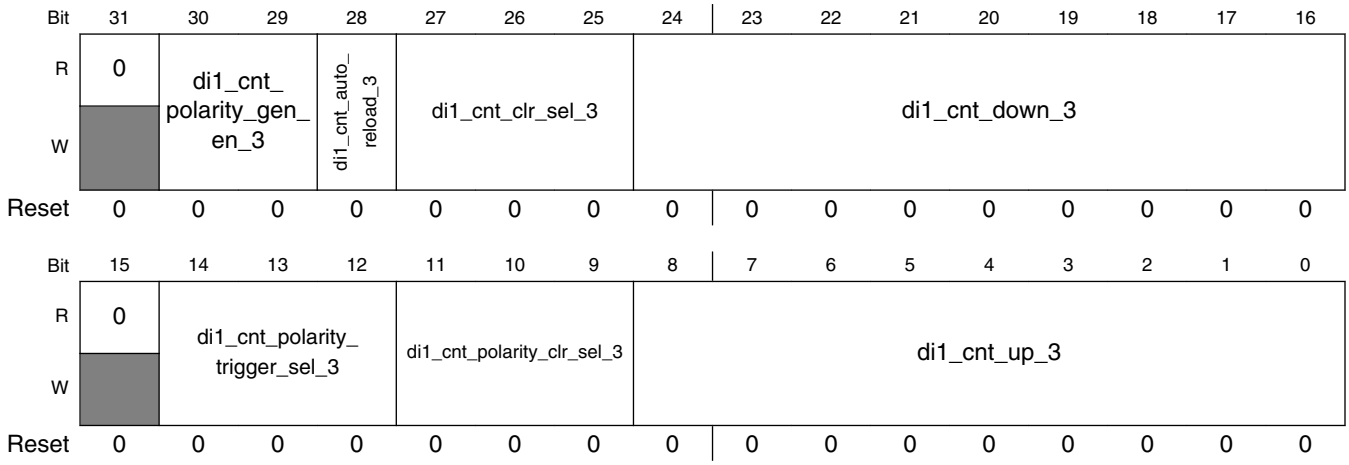
Table continues on the next page...

IPU_DI1_SW_GEN1_2 field descriptions (continued)

Field	Description
	010 The Counter is triggered by counter #1 011 Reserved 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
24–16 di1_cnt_down_2	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 di1_cnt_polarity_trigger_sel_2	DI1 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 Reserved 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
11–9 di1_cnt_polarity_clr_sel_2	DI1 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Reserved 011 Reserved 100 Reserved 101 Reserved 110 Reserved 111 Reserved
8–0 di1_cnt_up_2	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.203 DI1 Sync Wave Gen 3 Register 1 (IPU_DI1_SW_GEN1_3)

Address: IPU_DI1_SW_GEN1_3 is 1E00_0000h base + 48038h offset = 1E04_8038h



IPU_DI1_SW_GEN1_3 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–29 di1_cnt_polarity_gen_en_3	DI1 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di1_cnt_auto_reload_3	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_<i> field
27–25 di1_cnt_clr_sel_3	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. —

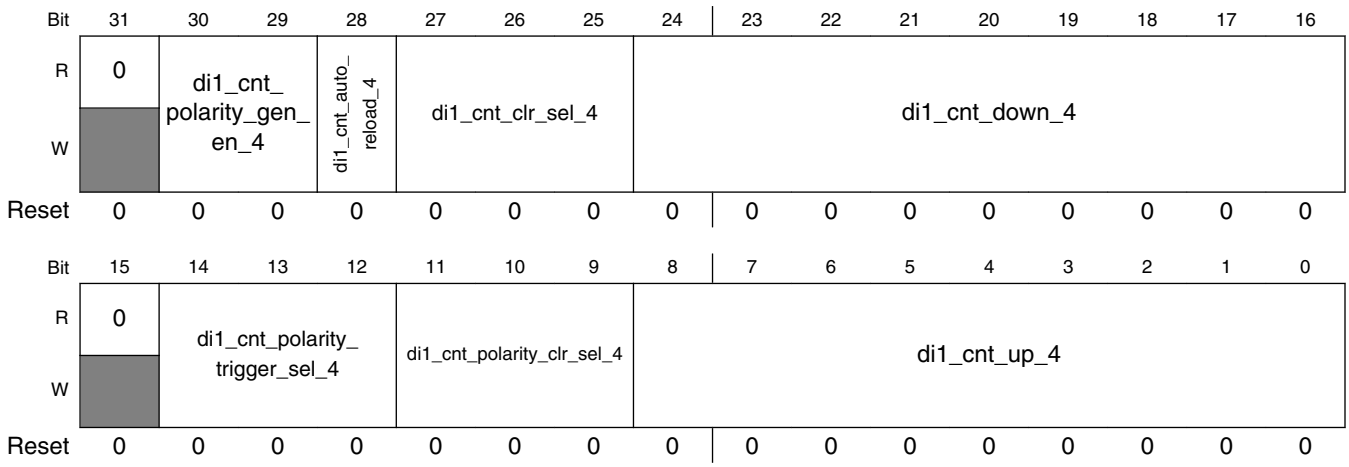
Table continues on the next page...

IPU_DI1_SW_GEN1_3 field descriptions (continued)

Field	Description
	110 External VSYNC 111 Counter is always on.
24–16 di1_cnt_down_3	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 di1_cnt_polarity_trigger_sel_3	DI1 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 Reserved 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
11–9 di1_cnt_polarity_clr_sel_3	DI1 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Reserved 101 Reserved 110 Reserved 111 Reserved
8–0 di1_cnt_up_3	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.204 DI1 Sync Wave Gen 4 Register 1 (IPU_DI1_SW_GEN1_4)

Address: IPU_DI1_SW_GEN1_4 is 1E00_0000h base + 4803Ch offset = 1E04_803Ch



IPU_DI1_SW_GEN1_4 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–29 di1_cnt_polarity_gen_en_4	DI1 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di1_cnt_auto_reload_4	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_<i> field
27–25 di1_cnt_clr_sel_4	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. —

Table continues on the next page...

IPU_DI1_SW_GEN1_4 field descriptions (continued)

Field	Description
	110 External VSYNC 111 Counter is always on.
24–16 di1_cnt_down_4	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 di1_cnt_polarity_trigger_sel_4	DI1 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 - Counter is disabled 001 - The counter is triggered by the same trigger that triggers the displays clock. 010 - The Counter is triggered by counter #1 011 - The Counter is triggered by counter #2 100 - The Counter is triggered by counter #3 101 - CSI VSYNC. The VSYNC is a trigger coming from one of the CSI's according to the CSI_VSYNC_DEST bit. — 110 External VSYNC 111 Counter is always on.
11–9 di1_cnt_polarity_clr_sel_4	DI1 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Reserved 110 Reserved 111 Reserved
8–0 di1_cnt_up_4	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.205 DI1 Sync Wave Gen 5 Register 1 (IPU_DI1_SW_GEN1_5)

Address: IPU_DI1_SW_GEN1_5 is 1E00_0000h base + 48040h offset = 1E04_8040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_cnt_polarity_gen_en_5		di1_cnt_auto_reload_5	di1_cnt_clr_sel_5				di1_cnt_down_5							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_cnt_polarity_trigger_sel_5			di1_cnt_polarity_clr_sel_5				di1_cnt_up_5							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN1_5 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–29 di1_cnt_polarity_gen_en_5	DI1 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di1_cnt_auto_reload_5	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_<i> field
27–25 di1_cnt_clr_sel_5	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 External VSYNC 111 Counter is always on.

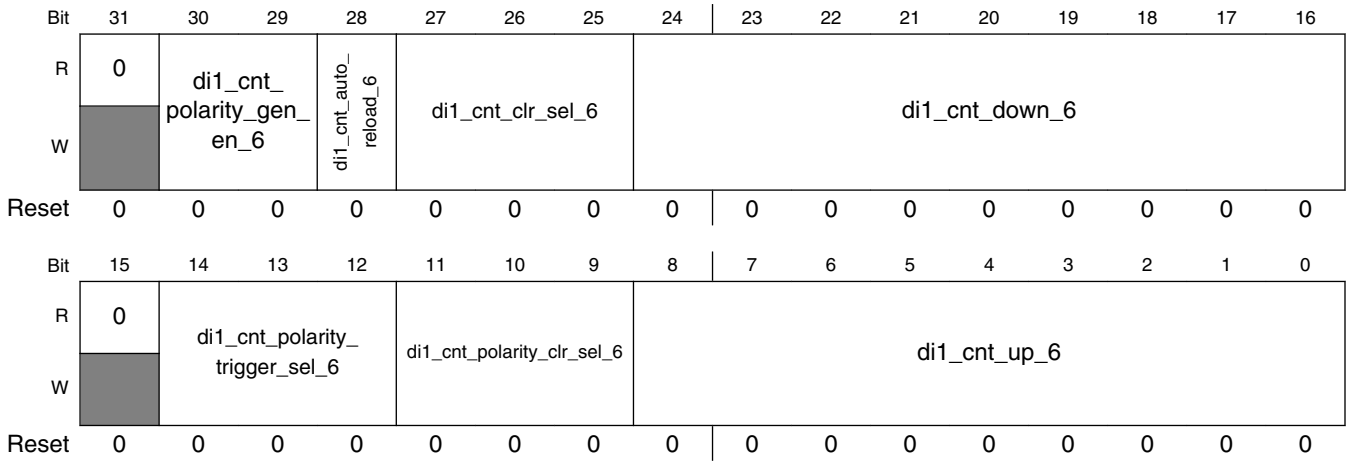
Table continues on the next page...

IPU_DI1_SW_GEN1_5 field descriptions (continued)

Field	Description
24–16 di1_cnt_down_5	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 di1_cnt_polarity_trigger_sel_5	DI1 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 External VSYNC 111 Counter is always on.
11–9 di1_cnt_polarity_clr_sel_5	DI1 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Output is inverted if the output of counter #4 is set 110 Reserved 111 Reserved
8–0 di1_cnt_up_5	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.206 DI1 Sync Wave Gen 6 Register 1 (IPU_DI1_SW_GEN1_6)

Address: IPU_DI1_SW_GEN1_6 is 1E00_0000h base + 48044h offset = 1E04_8044h



IPU_DI1_SW_GEN1_6 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–29 di1_cnt_polarity_gen_en_6	DI1 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di1_cnt_auto_reload_6	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_<i> field
27–25 di1_cnt_clr_sel_6	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

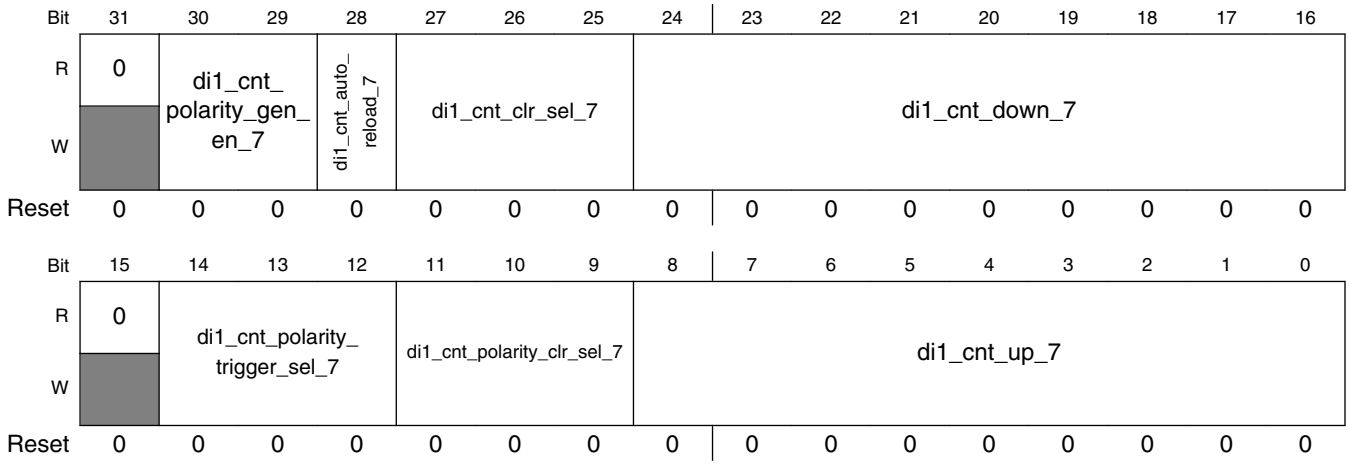
Table continues on the next page...

IPU_DI1_SW_GEN1_6 field descriptions (continued)

Field	Description
24–16 di1_cnt_down_6	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 di1_cnt_polarity_trigger_sel_6	DI1 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
11–9 di1_cnt_polarity_clr_sel_6	DI1 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Output is inverted if the output of counter #4 is set 110 Output is inverted if the output of counter #5 is set 111 Reserved
8–0 di1_cnt_up_6	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.207 DI1Sync Wave Gen 7 Register 1 (IPU_DI1_SW_GEN1_7)

Address: IPU_DI1_SW_GEN1_7 is 1E00_0000h base + 48048h offset = 1E04_8048h



IPU_DI1_SW_GEN1_7 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–29 di1_cnt_polarity_gen_en_7	DI1 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di1_cnt_auto_reload_7	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_<i> field
27–25 di1_cnt_clr_sel_7	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

Table continues on the next page...

IPU_DI1_SW_GEN1_7 field descriptions (continued)

Field	Description
24–16 di1_cnt_down_7	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 di1_cnt_polarity_trigger_sel_7	DI1 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
11–9 di1_cnt_polarity_clr_sel_7	DI1 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Output is inverted if the output of counter #4 is set 110 Output is inverted if the output of counter #5 is set 111 Output is inverted if the output of counter #6 is set
8–0 di1_cnt_up_7	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.208 DI1 Sync Wave Gen 8 Register 1 (IPU_DI1_SW_GEN1_8)

Address: IPU_DI1_SW_GEN1_8 is 1E00_0000h base + 4804Ch offset = 1E04_804Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	di1_cnt_polarity_gen_en_8		di1_cnt_auto_reload_8	di1_cnt_clr_sel_8				di1_cnt_down_8							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	di1_cnt_polarity_trigger_sel_8			di1_cnt_polarity_clr_sel_8				di1_cnt_up_8							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SW_GEN1_8 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–29 di1_cnt_polarity_gen_en_8	DI1 Counter polarity generator enable The counter's output polarity can be changed on the fly. 00 Dynamic polarity change is disabled 01 The counters UP and DOWN value are calculated according to the trigger selected by cnt_polarity_trigger_sel field. When selecting this mode the UP and DOWN values has to be a multiple of 2. 10 Dynamic polarity change is enabled 11 Outputs polarity is dynamically changed every time the counter reaches its pre defined value
28 di1_cnt_auto_reload_8	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_<i> field
27–25 di1_cnt_clr_sel_8	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

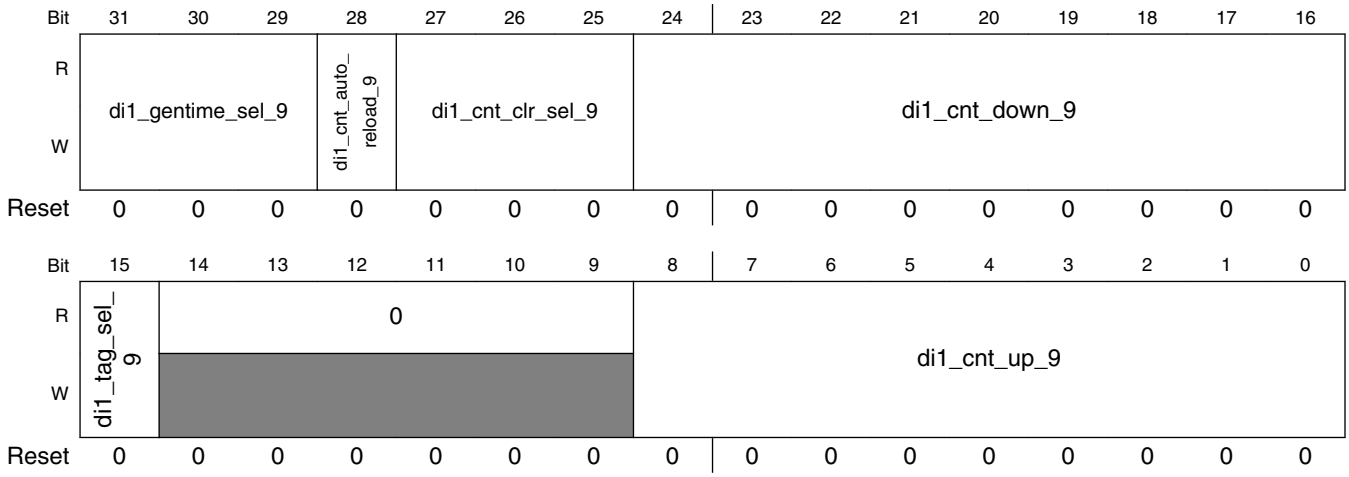
Table continues on the next page...

IPU_DI1_SW_GEN1_8 field descriptions (continued)

Field	Description
24–16 di1_cnt_down_8	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14–12 di1_cnt_polarity_trigger_sel_8	DI1 Counter's toggling trigger select This field selects the counter's trigger causing the output to toggle 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.
11–9 di1_cnt_polarity_clr_sel_8	DI1 counter's polarity Clear select This field selects the input to the counter telling the counter wether to invert the output 000 Output is always inverted 001 Output is kept the same (no inversion) 010 Output is inverted if the output of counter #1 is set 011 Output is inverted if the output of counter #2 is set 100 Output is inverted if the output of counter #3 is set 101 Output is inverted if the output of counter #4 is set 110 Output is inverted if the output of counter #5 is set 111 Output is inverted if the output of counter #6 is set
8–0 di1_cnt_up_8	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.209 DI1 Sync Wave Gen 9 Register 1 (IPU_DI1_SW_GEN1_9)

Address: IPU_DI1_SW_GEN1_9 is 1E00_0000h base + 48050h offset = 1E04_8050h



IPU_DI1_SW_GEN1_9 field descriptions

Field	Description
31–29 di1_gentime_sel_9	Counter #9 main waveform select This field defines the counter that counter #9's auxiliary waveform will be attached too. 000 Counter #9's waveform is attached to counter #1's waveform 001 Counter #9's waveform is attached to counter #2's waveform 010 Counter #9's waveform is attached to counter #3's waveform 011 Counter #9's waveform is attached to counter #4's waveform 100 Counter #9's waveform is attached to counter #5's waveform 101 Counter #9's waveform is attached to counter #6's waveform 110 Counter #9's waveform is attached to counter #7's waveform 111 Counter #9's waveform is attached to counter #8's waveform
28 di1_cnt_auto_reload_9	Counter auto reload mode 1 The counter will automatically be reloaded forever, ignoring the value of the di1_step_repeat_<i> field 0 The counter will not be automatically reloaded, It will be reloaded for the amount of repeat times defined on the di1_step_repeat_<i> field
27–25 di1_cnt_clr_sel_9	Counter Clear select This field defines the source of the signals that clears the counter. 000 Counter is disabled 001 The counter is triggered by the same trigger that triggers the displays clock. 010 The Counter is triggered by counter #1 011 The Counter is triggered by counter #2 100 The Counter is triggered by counter #3 101 The Counter is triggered by counter #4 110 The Counter is triggered by counter #5 111 Counter is always on.

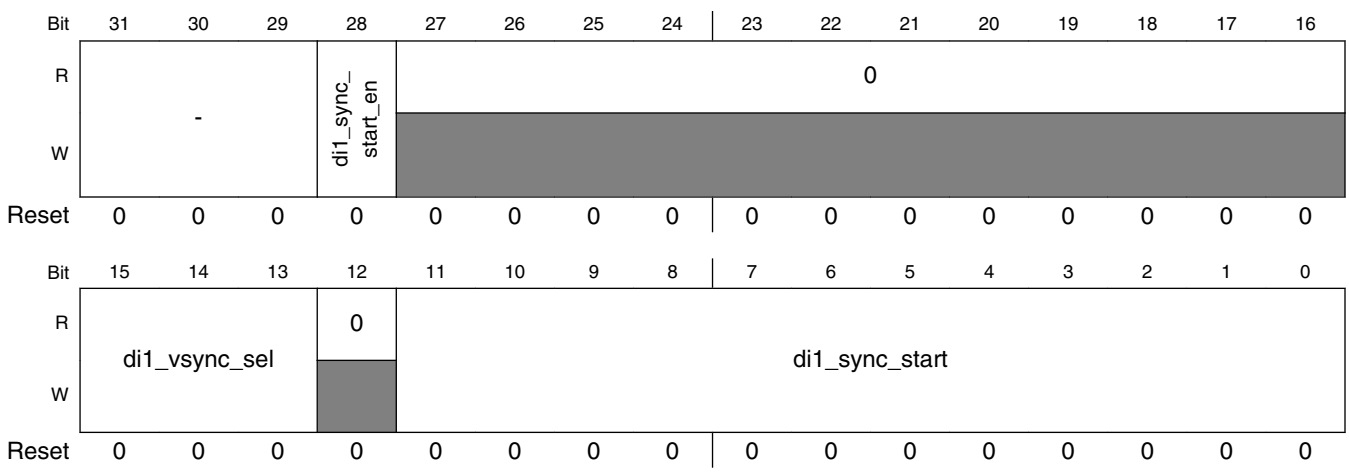
Table continues on the next page...

IPU_DI1_SW_GEN1_9 field descriptions (continued)

Field	Description
24–16 di1_cnt_down_9	Counter falling edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's falling edge.
15 di1_tag_sel_9	Counter #9 can send a synchronous tag when counter #9 reach its predefined value or when it's triggering counter reaches its pre defined value. 1 tag source is counter #9 0 Tag's source is the triggering counter.
14–9 Reserved	This read-only field is reserved and always has the value zero. Reserved
8–0 di1_cnt_up_9	Counter rising edge position This parameter contains an integer part (bits 24:17) and a fractional part (bit 16). The position value is the amount of cycles between the trigger's start point and the waveform's rising edge.

45.51.210 DI1 Sync Assistance Gen Register (IPU_DI1_SYNC_AS_GEN)

Address: IPU_DI1_SYNC_AS_GEN is 1E00_0000h base + 48054h offset = 1E04_8054h



IPU_DI1_SYNC_AS_GEN field descriptions

Field	Description
31–29 -	Reserve
28 di1_sync_start_en	di1_sync_start_en

Table continues on the next page...

IPU_DI1_SYNC_AS_GEN field descriptions (continued)

Field	Description
27–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–13 di1_vsync_sel	VSYNC select This field defines which of the counters functions as VSYNC signal 000 VSYNC is coming from counter #1 001 VSYNC is coming from counter #2 111 VSYNC is coming from counter #8
12 Reserved	This read-only field is reserved and always has the value zero. Reserved.
11–0 di1_sync_start	DI1 Sync start This field defines the number of low (including blanking rows) on the which the DI1 starts preparing the data for the next frame.

45.51.211 DI1 Data Wave Gen <i> Register (IPU_DI1_DW_GEN_i)

The DI1_DW_GEN_<i> register holds pointers for the waveform generators.

These registers have different bit arrangements for parallel and serial display. When using a parallel display VDI Plane Size Register 4 DI1 Data Wave Gen <i> Register is applicable. When using a serial interface VDI Plane Size Register 4 DI1 Data Wave Gen <i> Register is applicable.

Table 45-260. Register Field Descriptions for Serial display

Field	Description
31-24 di1_serial_period_<i>	DI1 Serial Period <i> This field defines the period of the time base serial display clock. The units are the internal DI clock
23-16 di1_start_period_<i>	DI1 start period This field defines the amount of cycles between the point where the access is ready to be launched to the actual point where the time base serial display clock restarts. The units are the internal DI clock.
15-14 di1_cst_<i>	DI1 Chip Select pointer for waveform <i> This field points to a register that defines the waveform of the CS pin. For serial displays the down value as defined on DI1_DW_SET*_<i> is measured from the assertion of the last serial display time base clock. 00 The waveform is defined according to the settings on DI1_DW_SET0_<i> 01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 The waveform is defined according to the settings on DI1_DW_SET3_<i>

Table continues on the next page...

Table 45-260. Register Field Descriptions for Serial display (continued)

Field	Description
13-9	Reserved
8-4 di1_serial_valid_bits_<i>	DI1 Serial valid bits. This field defines the amount of valid bits to be transmitted within the 32 bits internal word aligned to bit[0]. The actual amount of valid bits is di1_serial_valid_bits_<i> + 1
3-2 di1_serial_rs_<i>	DI1 Serial RS This field points to a register that defines the waveform of the RS pin. For serial displays the down value as defined on DI1_DW_SET*_<i> is measured from the assertion of the last serial display time base clock. 00 The waveform is defined according to the settings on DI1_DW_SET0_<i> 01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 The waveform is defined according to the settings on DI1_DW_SET3_<i>
1-0 di1_serial_clk_<i>	DI1 serial clock<i> This field points to a register that defines the waveform of the Serial clock pin. 00 The waveform is defined according to the settings on DI1_DW_SET0_<i> 01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 The waveform is defined according to the settings on DI1_DW_SET3_<i>

Address: IPU_DI1_DW_GEN_i is 1E00_0000h base + 48058h offset = 1E04_8058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	di1_access_size_<i>								di1_componnent_size_<i>								di1_cst_<i>		di1_pt_6_<i>		di1_pt_5_<i>		di1_pt_4_<i>		di1_pt_3_<i>		di1_pt_2_<i>		di1_pt_1_<i>		di1_pt_0_<i>	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_DW_GEN_i field descriptions

Field	Description
31–24 di1_access_size_<i>	DI1 Access Size <i> This field defines the amount of IPU cycles between any 2 accesses (an access may be a pixel or generic data that may have more one component)
23–16 di1_componnent_size_<i>	DI1 component Size This field defines the amount of IPU cycles between any 2 components
15–14 di1_cst_<i>	DI1 Chip Select pointer for waveform <i> This field points to a register that defines the waveform of the CS pin. The CS is automatically mapped to a specific display 00 The waveform is defined according to the settings on DI1_DW_SET0_<i>

Table continues on the next page...

IPU_DI1_DW_GEN_i field descriptions (continued)

Field	Description
	01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 The waveform is defined according to the settings on DI1_DW_SET3_<i>
13–12 di1_pt_6_<i>	DI1 PIN_17 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_17 pin. 00 The waveform is defined according to the settings on DI1_DW_SET0_<i> 01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 The waveform is defined according to the settings on DI1_DW_SET3_<i>
11–10 di1_pt_5_<i>	DI1 PIN_16 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_16 pin. 00 The waveform is defined according to the settings on DI1_DW_SET0_<i> 01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 The waveform is defined according to the settings on DI1_DW_SET3_<i>
9–8 di1_pt_4_<i>	DI1 PIN_15 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_15 pin. 00 The waveform is defined according to the settings on DI1_DW_SET0_<i> 01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 The waveform is defined according to the settings on DI1_DW_SET3_<i>
7–6 di1_pt_3_<i>	DI1 PIN_14 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_14 pin. 00 The waveform is defined according to the settings on DI1_DW_SET0_<i> 01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 The waveform is defined according to the settings on DI1_DW_SET3_<i>
5–4 di1_pt_2_<i>	DI1 PIN_13 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_13 pin. 00 The waveform is defined according to the settings on DI1_DW_SET0_<i> 01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 The waveform is defined according to the settings on DI1_DW_SET3_<i>
3–2 di1_pt_1_<i>	DI1 PIN_12 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_12 pin. 00 The waveform is defined according to the settings on DI1_DW_SET0_<i> 01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 The waveform is defined according to the settings on DI1_DW_SET3_<i>

Table continues on the next page...

IPU_DI1_DW_GEN_i field descriptions (continued)

Field	Description
1–0 di1_pt_0_<i>	DI1 PIN_11 pointer for waveform <i> This field points to a register that defines the waveform of the PIN_11 pin. 00 The waveform is defined according to the settings on DI1_DW_SET0_<i> 01 The waveform is defined according to the settings on DI1_DW_SET1_<i> 10 The waveform is defined according to the settings on DI1_DW_SET2_<i> 11 - The waveform is defined according to the settings on DI1_DW_SET3_<i>

45.51.212 DI1 Data Wave Set 0 <i> Register (IPU_DI1_DW_SET0_i)

Address: IPU_DI1_DW_SET0_i is 1E00_0000h base + 48088h offset = 1E04_8088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							di1_data_cnt_down0_<i>									0							di1_data_cnt_up0_<i>								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_DW_SET0_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 di1_data_cnt_down0_<i>	Waveform's falling edge position. This field defines the Waveform's falling edge position. The Waveform is mapped to a pint according to the corresponding di1_pt_*_<i>
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 di1_data_cnt_up0_<i>	Waveform's rising edge position. This field defines the Waveform's rising edge position. The Waveform is mapped to a pint according to the corresponding di1_pt_*_<i>

45.51.213 DI1 Data Wave Set 1 <i> Register (IPU_DI1_DW_SET1_i)

Address: IPU_DI1_DW_SET1_i is 1E00_0000h base + 480B8h offset = 1E04_80B8h

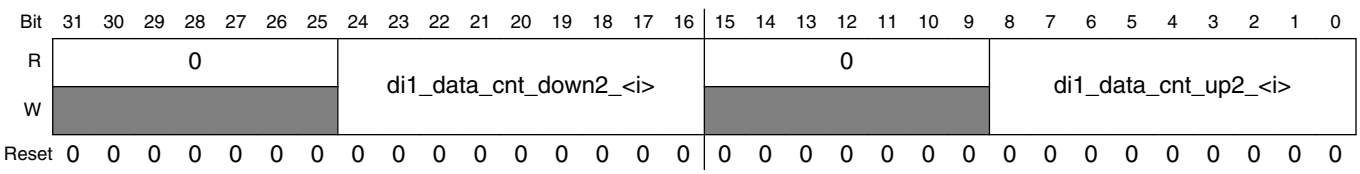
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							di1_data_cnt_down1_<i>									0							di1_data_cnt_up1_<i>								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_DW_SET1_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 di1_data_cnt_down1_<i>	Waveform's falling edge position. This field defines the Waveform's falling edge position. The Waveform is mapped to a pint according to the corresponding di1_pt_*_<i>
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 di1_data_cnt_up1_<i>	Waveform's rising edge position. This field defines the Waveform's rising edge position. The Waveform is mapped to a pint according to the corresponding di1_pt_*_<i>

45.51.214 DI1 Data Wave Set 2 <i> Register (IPU_DI1_DW_SET2_i)

Address: IPU_DI1_DW_SET2_i is 1E00_0000h base + 480E8h offset = 1E04_80E8h

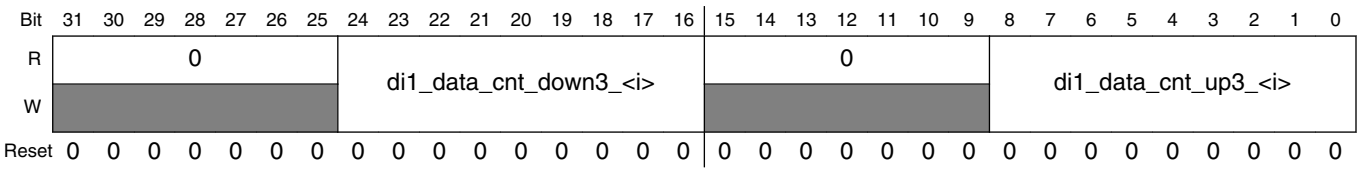


IPU_DI1_DW_SET2_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 di1_data_cnt_down2_<i>	Waveform's falling edge position. This field defines the Waveform's falling edge position. The Waveform is mapped to a pint according to the corresponding di1_pt_*_<i>
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 di1_data_cnt_up2_<i>	Waveform's rising edge position. This field defines the Waveform's rising edge position. The Waveform is mapped to a pint according to the corresponding di1_pt_*_<i>

45.51.215 DI1 Data Wave Set 3 <i> Register (IPU_DI1_DW_SET3_i)

Address: IPU_DI1_DW_SET3_i is 1E00_0000h base + 48118h offset = 1E04_8118h

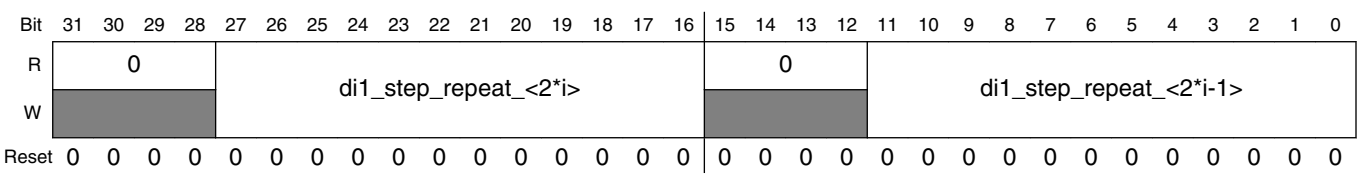


IPU_DI1_DW_SET3_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 di1_data_cnt_down3_<i>	Waveform's falling edge position. This field defines the Waveform's falling edge position. The Waveform is mapped to a pint according to the corresponding di1_pt_*_<i>
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 di1_data_cnt_up3_<i>	Waveform's rising edge position. This field defines the Waveform's rising edge position. The Waveform is mapped to a pint according to the corresponding di1_pt_*_<i>

45.51.216 DI1 Step Repeat <i> Registers (IPU_D1_STP_REP_i)

Address: IPU_D1_STP_REP_i is 1E00_0000h base + 48148h offset = 1E04_8148h



IPU_D1_STP_REP_i field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved
27–16 di1_step_repeat_<2*i>	Step Repeat <i> This fields defines the amount of repetitions that will be performed by the counter <i>
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved

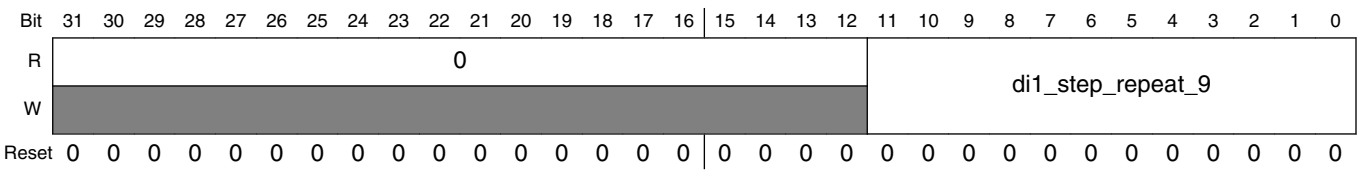
Table continues on the next page...

IPU_D1_STP_REP_i field descriptions (continued)

Field	Description
11–0 di1_step_ repeat_<2*i-1>	Step Repeat <i> This fields defines the amount of repetitions that will be performed by the counter <i>

45.51.217 DI1Step Repeat 9 Registers (IPU_DI1_STP_REP_9)

Address: IPU_DI1_STP_REP_9 is 1E00_0000h base + 48158h offset = 1E04_8158h

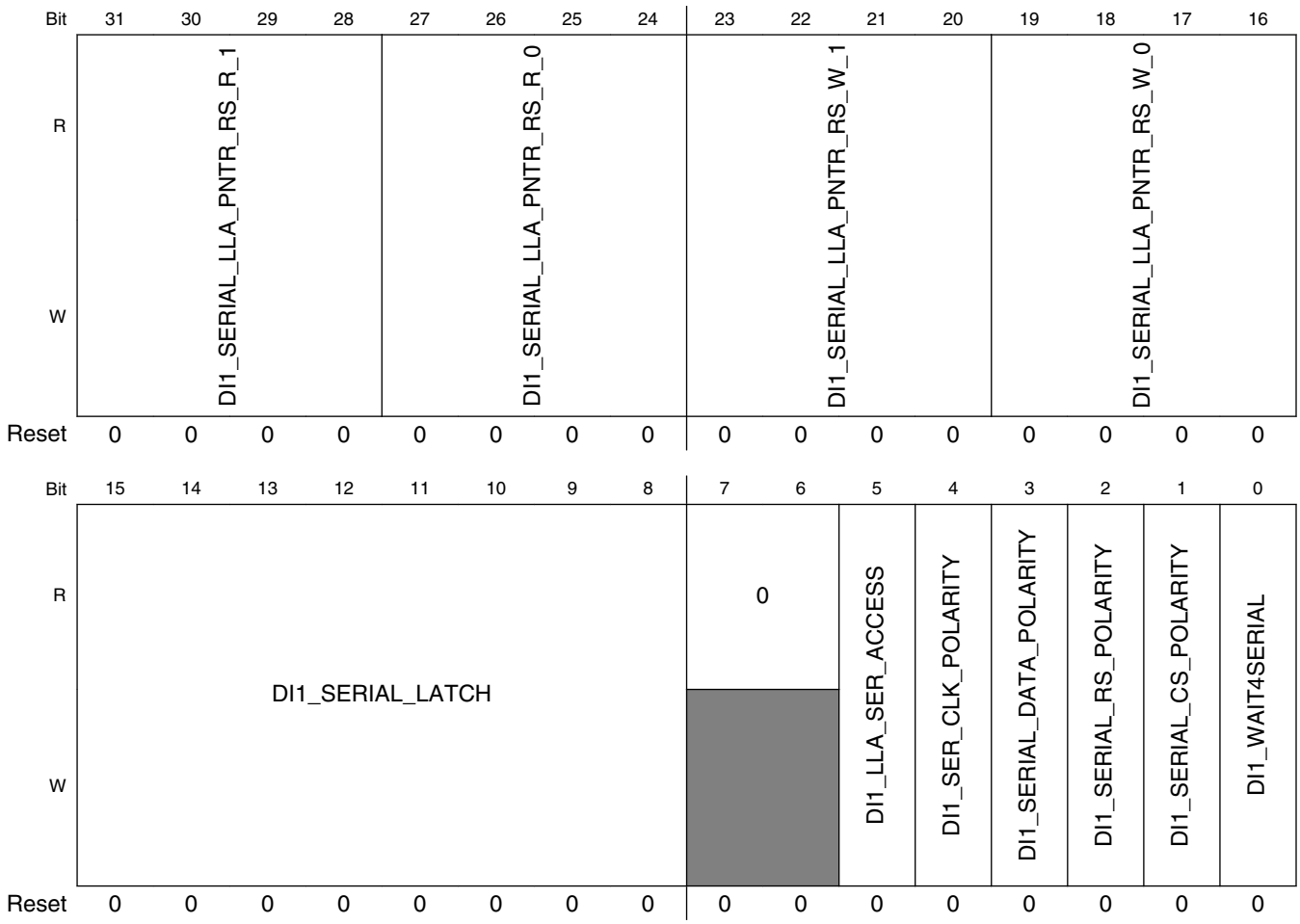


IPU_DI1_STP_REP_9 field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11–0 di1_step_ repeat_9	Step Repeat 9 This fields defines the amount of repetitions that will be performed by the counter 9.

45.51.218 DI1 Serial Display Control Register (IPU_DI1_SER_CONF)

Address: IPU_DI1_SER_CONF is 1E00_0000h base + 4815Ch offset = 1E04_815Ch



IPU_DI1_SER_CONF field descriptions

Field	Description
31–28 DI1_SERIAL_LLA_PNTR_RS_R_1	RS 3 waveform pointer for read low level access This pointer defines which waveform set will be chosen when the read low level access is targeted to RS group 1. 0000 Waveform set #1 0001 Waveform set #2 1011 Waveform set #12 1100 Reserved 1101 Reserved 1100 Reserved

Table continues on the next page...

IPU_DI1_SER_CONF field descriptions (continued)

Field	Description
27–24 DI1_SERIAL_LLA_PNTR_RS_R_0	RS 2 waveform pointer for low level access This pointer defines which waveform set will be chosen when the read low level access is targeted to RS group 0. 0000 Waveform set #1 0001 Waveform set #2 1011 Waveform set #12 1100 Reserved 1101 Reserved 1100 Reserved
23–20 DI1_SERIAL_LLA_PNTR_RS_W_1	RS 1 waveform pointer for write low level access This pointer defines which waveform set will be chosen when the write low level access is targeted to RS group 1. 0000 Waveform set #1 0001 Waveform set #2 1011 Waveform set #12 1100 Reserved 1101 Reserved 1100 Reserved
19–16 DI1_SERIAL_LLA_PNTR_RS_W_0	RS 0 waveform pointer for write low level access This pointer defines which waveform set will be chosen when the write low level access is targeted to RS group 0. 0000 Waveform set #1 0001 Waveform set #2 1011 Waveform set #12 1100 Reserved 1101 Reserved 1100 Reserved
15–8 DI1_SERIAL_LATCH	DI1 Serial Latch This field defines how many cycles to insert between serial read accesses start to data sampling in the
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5 DI1_LLA_SER_ACCESS	Direct Low Level Access to Serial display 1 ARM platform access is performed via a direct path to the serial display in LLA mode, in this mode only the ARM platform in LLA mode can access the serial port 0 ARM platform access to the serial display port is not done directly, hence other source are allowed to access the serial port. The arbitration is done automatically
4 DI1_SER_CLK_POLARITY	Serial Clock Polarity The output polarity of the SER_CLK pin 1 The clock is inverted 0 The clock is not inverted

Table continues on the next page...

IPU_DI1_SER_CONF field descriptions (continued)

Field	Description
3 DI1_SERIAL_ DATA_ POLARITY	Serial Data Polarity The output polarity of the SER_DATA pin 1 The data is inverted 0 The data is not inverted
2 DI1_SERIAL_ RS_POLARITY	Serial RS Polarity The output polarity of the SER_RS pin 1 The RS is inverted 0 The RS is not inverted
1 DI1_SERIAL_ CS_POLARITY	Serial Chip Select Polarity The output polarity of the SER_CS pin 1 The CS is inverted 0 The CS is not inverted
0 DI1_ WAIT4SERIAL	Wait for Serial When the parallel display share pins with the serial port. Accessing the parallel port is not allowed till the serial port completes its access. 1 The parallel port should wait to the serial port as the pins are shared 0 The parallel port should not wait to the serial port as the pins are not shared

45.51.219 DI1 Special Signals Control Register (IPU_DI1_SSC)

Address: IPU_DI1_SSC is 1E00_0000h base + 48160h offset = 1E04_8160h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								DI1_PIN17_ERM	DI1_PIN16_ERM	DI1_PIN15_ERM	DI1_PIN14_ERM	DI1_PIN13_ERM	DI1_PIN12_ERM	DI1_PIN11_ERM	DI1_CS_ERM
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										DI1_WAIT_ON	DI1_BYTE_EN_POLARITY	DI1_BYTE_EN_RD_IN	DI1_BYTE_EN_PNTR		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_SSC field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved.
23 DI1_PIN17_ERM	DI1 PIN17 error recovery mode. This bit defines the error recovery mode of the PIN17 pin. 1 The PIN17 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC. 0 Nothing is done to the PIN17 pin following a display error detection.
22 DI1_PIN16_ERM	DI1 PIN16 error recovery mode. This bit defines the error recovery mode of the PIN16 pin. 1 The PIN16 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC. 0 Nothing is done to the PIN16 pin following a display error detection.
21 DI1_PIN15_ERM	DI1 PIN15 error recovery mode. This bit defines the error recovery mode of the PIN15 pin. 1 The PIN15 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC. 0 Nothing is done to the PIN15 pin following a display error detection.
20 DI1_PIN14_ERM	DI1 PIN14 error recovery mode. This bit defines the error recovery mode of the PIN14 pin. 1 The PIN14 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC. 0 Nothing is done to the PIN14 pin following a display error detection.
19 DI1_PIN13_ERM	DI1 PIN13 error recovery mode. This bit defines the error recovery mode of the PIN13 pin. 1 The PIN13 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC. 0 Nothing is done to the PIN13 pin following a display error detection.
18 DI1_PIN12_ERM	DI1 PIN12 error recovery mode. This bit defines the error recovery mode of the PIN12 pin. 1 The PIN12 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC. 0 Nothing is done to the PIN12 pin following a display error detection.
17 DI1_PIN11_ERM	DI1 PIN11 error recovery mode. This bit defines the error recovery mode of the PIN11 pin. 1 The PIN11 pin is cleared in case of a synchronous display error. The clear will be done on the next VSYNC. 0 Nothing is done to the PIN11 pin following a display error detection.
16 DI1_CS_ERM	DI1 GLUELOGIC error recovery mode. This bit defines the error recovery mode of the GLUELOGIC.

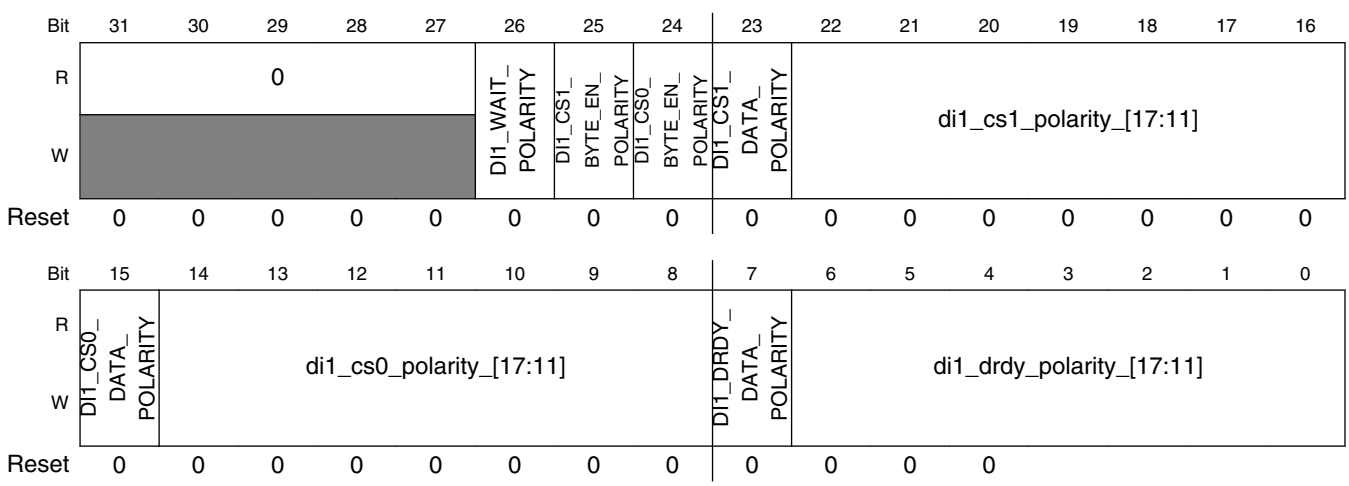
Table continues on the next page...

IPU_DI1_SSC field descriptions (continued)

Field	Description
	1 The GLUELOGIC is release in case of a synchronous display error. The release will be done on the next VSYNC. 0 Nothing is done to the GLUELOGIC following a display error detection.
15–6 Reserved	This read-only field is reserved and always has the value zero. Reserved.
5 DI1_WAIT_ON	Wait On This field defines the DC's response to WAIT signal 1 The DC holds the flow as long as WAIT is asserted. 0 The DC continues the flow regardless the WAIT signal.
4 DI1_BYTE_EN_POLARITY	Byte Enable polarity This bit defines the polarity of the byte enable signals to the display. 1 active high. 0 active low.
3 DI1_BYTE_EN_RD_IN	Byte Enable Read In This bit selects the source of the byte enable pins 1 The write byte enable signals are routed via bits [17:16] of the display's data, The read byte enable signals are routed via bits [19:18 of the display's data 0 The byte enable signals are routed via bits [17:16] of the display's data for both read and write
2–0 DI1_BYTE_EN_PNTR	Byte Enable Pointer This pointer selects the pin asserted along with the byte enables signals 000 wave form of byte enable as pin_11 001 wave form of byte enable as pin_12 111 wave form of byte enable as suitable CS pin

45.51.220 DI1 Polarity Register (IPU_DI1_POL)

Address: IPU_DI1_POL is 1E00_0000h base + 48164h offset = 1E04_8164h



IPU_DI1_POL field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26 DI1_WAIT_ POLARITY	WAIT polarity This bit defines the polarity of the wait signal input coming from the displa1 1 active high 0 active low
25 DI1_CS1_ BYTE_EN_ POLARITY	Byte Enable associated with CS1 polarity This bit defines the polarity of the byte enable signals to the display 1 active high 0 active low
24 DI1_CS0_ BYTE_EN_ POLARITY	Byte Enable associated with CS0 polarity This bit defines the polarity of the byte enable signals to the display 1 active high 0 active low
23 DI1_CS1_ DATA_ POLARITY	Data Polarity associated with CS1
22–16 di1_cs1_ polarity_[17:11]	DI1 output pin's polarity for CS1 This bits define the polarity of each of the DI's outputs when CS1 is asserted 1 The output pin is active high 0 The output pin is active low
15 DI1_CS0_ DATA_ POLARITY	Data Polarity associated with CS0
14–8 di1_cs0_ polarity_[17:11]	DI1 output pin's polarity for CS1 This bits define the polarity of each of the DI's outputs when CS1 is asserted 1 The output pin is active high 0 The output pin is active low
7 DI1_DRDY_ DATA_ POLARITY	Data Polarity associated with DRDY
6–0 di1_drdy_ polarity_[17:11]	DI1 output dynamic pin's polarity for synchronous access This bits define the polarity of each of the DI's outputs when synchronous display access is asserted The pins' default polarity is the same as defined in the di0_drdy_polarity_[17:11] bits 1 The output pin is active high 0 The output pin is active low

45.51.221 DI1Active Window 0 Register (IPU_DI1_AW0)

Address: IPU_DI1_AW0 is 1E00_0000h base + 48168h offset = 1E04_8168h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DI1_AW_TRIG_SEL				DI1_AW_HEND												DI1_AW_HCOUNT_SEL				DI1_AW_HSTART											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_AW0 field descriptions

Field	Description
31–28 DI1_AW_TRIG_SEL	This field selects the trigger for sending data during the display's active window 000 disabled 001 The trigger is the same trigger that triggers the displays clock. 010 The trigger is counter #1 011 The trigger is counter #2 100 The trigger is counter #3 101 The trigger is counter #4 110 The trigger is counter #5 111 The trigger is always on.
27–16 DI1_AW_HEND	This field defines the horizontal end of the active window
15–12 DI1_AW_HCOUNT_SEL	This field selects the counter that counts the horizontal position of the display's active window 0000 disabled 0001 reserved 0010 The counter is counter #1 0011 The counter is counter #2 0100 The counter is counter #3 0101 The counter is counter #4 0110 The counter is counter #5 1001 The counter is counter #8
11–0 DI1_AW_HSTART	This field defines the horizontal start of the active window DI1_AW_HSTART < DI1_AW_HEND

45.51.222 DI1 Active Window 1 Register (IPU_DI1_AW1)

Address: IPU_DI1_AW1 is 1E00_0000h base + 4816Ch offset = 1E04_816Ch

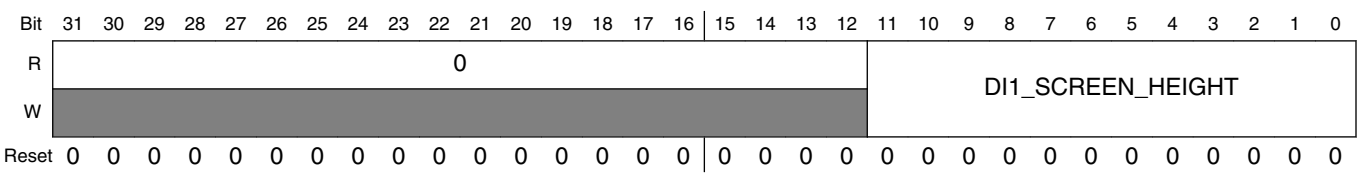
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DI1_AW_VEND												DI1_AW_VCOUNT_SEL				DI1_AW_VSTART											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DI1_AW1 field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved
27–16 DI1_AW_VEND	This field defines the vertical end of the active window
15–12 DI1_AW_ VCOUNT_SEL	This field selects the counter that counts the vertical position of the display's active window 0000 disabled 0001 reserved 0010 The counter is counter #1 0011 The counter is counter #2 0100 The counter is counter #3 0101 The counter is counter #4 0110 The counter is counter #5 1001 The counter is counter #8
11–0 DI1_AW_ VSTART	This field defines the vertical start of the active window DI1_AW_VSTART < DI1_AW_VEND

45.51.223 DI1 Screen Configuration Register (IPU_DI1_SCR_CONF)

Address: IPU_DI1_SCR_CONF is 1E00_0000h base + 48170h offset = 1E04_8170h

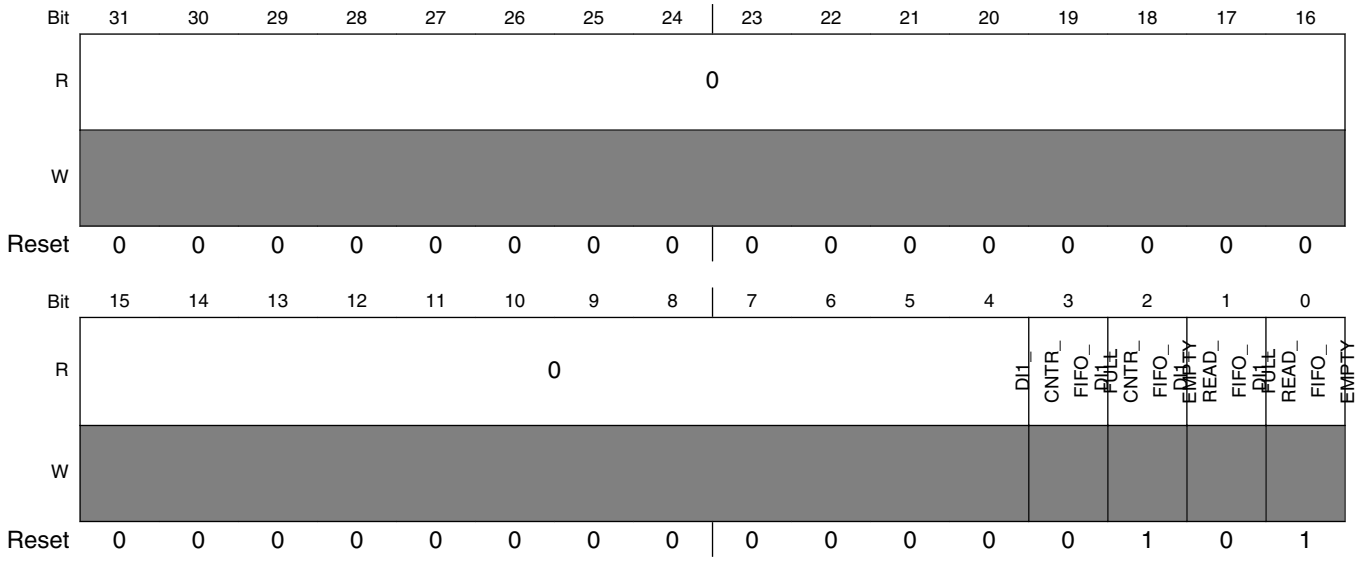


IPU_DI1_SCR_CONF field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero. Reserved.
11–0 DI1_SCREEN_ HEIGHT	This field defines the number of display rows (Number_of_ROWS = DI1_SCREEN_HEIGHT+1) This field is used for VSYNC calculation and for anti-tearing

45.51.224 DI1 Status Register (IPU_DI1_STAT)

Address: IPU_DI1_STAT is 1E00_0000h base + 48174h offset = 1E04_8174h



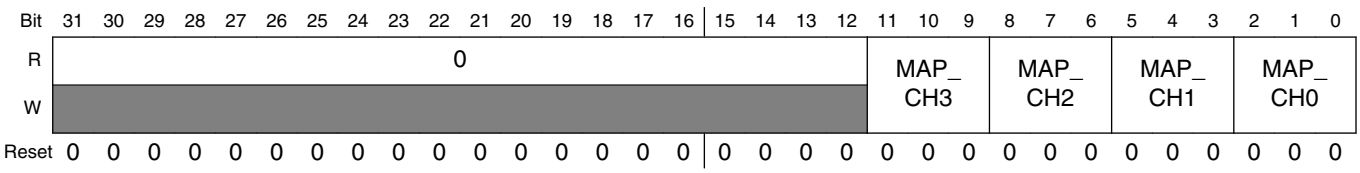
IPU_DI1_STAT field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value zero. Reserved
3 DI1_CNTR_ FIFO_FULL	This bit indicates a full state of the DI1 FIFO. This FIFO is part of the DI1 synchronizer.
2 DI1_CNTR_ FIFO_EMPTY	This bit indicates an empty state of the DI1 FIFO. This FIFO is part of the DI1 synchronizer.
1 DI1_READ_ FIFO_FULL	This bit indicates a full state of the DI1 FIFO when performing a read. This FIFO is part of the DI1 synchronizer.
0 DI1_READ_ FIFO_EMPTY	This bit indicates an empty state of the DI1 FIFO when performing a read. This FIFO is part of the DI1 synchronizer.

45.51.225 SMFC Mapping Register (IPU_SMFC_MAP)

The purpose of this register is to map CSI frames to IDMAC channels.

Address: IPU_SMFC_MAP is 1E00_0000h base + 50000h offset = 1E05_0000h



IPU_SMFC_MAP field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero. Reserved.
11–9 MAP_CH3	DMASMFC channel 3mapping bits. 000 CSI0, ID=0 mapped to DMASMFC channel 3. 001 CSI0, ID=1 mapped to DMASMFC channel 3. 010 CSI0, ID=2 mapped to DMASMFC channel 3. 011 CSI0, ID=3 mapped to DMASMFC channel 3. 100 CSI1, ID=0 mapped to DMASMFC channel 3. 101 CSI1, ID=1 mapped to DMASMFC channel 3. 110 CSI1, ID=2 mapped to DMASMFC channel 3. 111 CSI1, ID=3 mapped to DMASMFC channel 3.
8–6 MAP_CH2	DMASMFC channel 2 mapping bits. 000 CSI0, ID=0 mapped to DMASMFC channel 2. 001 CSI0, ID=1 mapped to DMASMFC channel 2. 010 CSI0, ID=2 mapped to DMASMFC channel 2. 011 CSI0, ID=3 mapped to DMASMFC channel 2. 100 CSI1, ID=0 mapped to DMASMFC channel 2. 101 CSI1, ID=1 mapped to DMASMFC channel 2. 110 CSI1, ID=2 mapped to DMASMFC channel 2. 111 CSI1, ID=3 mapped to DMASMFC channel 2.
5–3 MAP_CH1	DMASMFC channel 1 mapping bits. 000 CSI0, ID=0 mapped to DMASMFC channel 1. 001 CSI0, ID=1 mapped to DMASMFC channel 1. 010 CSI0, ID=2 mapped to DMASMFC channel 1. 011 CSI0, ID=3 mapped to DMASMFC channel 1. 100 CSI1, ID=0 mapped to DMASMFC channel 1. 101 CSI1, ID=1 mapped to DMASMFC channel 1. 110 CSI1, ID=2 mapped to DMASMFC channel 1. 111 CSI1, ID=3 mapped to DMASMFC channel 1.
2–0 MAP_CH0	DMASMFC channel 0 mapping bits.

Table continues on the next page...

IPU_SMFC_MAP field descriptions (continued)

Field	Description
	000 CSI0, ID=0 mapped to DMASMFC channel 0. 001 CSI0, ID=1 mapped to DMASMFC channel 0. 010 CSI0, ID=2 mapped to DMASMFC channel 0. 011 CSI0, ID=3 mapped to DMASMFC channel 0. 100 CSI1, ID=0 mapped to DMASMFC channel 0. 101 CSI1, ID=1 mapped to DMASMFC channel 0. 110 CSI1, ID=2 mapped to DMASMFC channel 0. 111 CSI1, ID=3 mapped to DMASMFC channel 0.

45.51.226 SMFC Watermark Control Register (IPU_SMFC_WMC)

The purpose of this register is to control watermarks levels of DMA channels. The bit setting given relative to FIFO size and not in number of words since FIFO size depend from number of enabled DMA channels.

Address: IPU_SMFC_WMC is 1E00_0000h base + 50004h offset = 1E05_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				WM3_CLR			WM3_SET			WM2_CLR			WM2_SET			0				WM1_CLR			WM1_SET			WM0_CLR			WM0_SET		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	1	0

IPU_SMFC_WMC field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved.
27–25 WM3_CLR	Watermark "clear" level of DMASMFC channel 3. 000 clear watermark level when FIFO is full on 1/8 of their size. 001 clear watermark level when FIFO is full on 2/8 of their size. 110 clear watermark level when FIFO is full on 7/8 of their size. 111 clear watermark level when FIFO is full.
24–22 WM3_SET	Watermark "set" level of DMASMFC channel 3 000 set watermark level when FIFO is full on 1/8 of their size. 001 set watermark level when FIFO is full on 2/8 of their size. 110 set watermark level when FIFO is full on 7/8 of their size. 111 set watermark level when FIFO is full.
21–19 WM2_CLR	Watermark "clear" level of DMASMFC channel 2. 000 clear watermark level when FIFO is full on 1/8 of their size. 001 clear watermark level when FIFO is full on 2/8 of their size. 110 clear watermark level when FIFO is full on 7/8 of their size. 111 clear watermark level when FIFO is full.

Table continues on the next page...

IPU_SMFC_WMC field descriptions (continued)

Field	Description
18–16 WM2_SET	Watermark "set" level of DMASMFC channel 2. 000 set watermark level when FIFO is full on 1/8 of their size. 001 set watermark level when FIFO is full on 2/8 of their size. 110 set watermark level when FIFO is full on 7/8 of their size. 111 set watermark level when FIFO is full.
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved.
11–9 WM1_CLR	Watermark "clear" level of DMASMFC channel 1. 000 clear watermark level when FIFO is full on 1/8 of their size. 001 clear watermark level when FIFO is full on 2/8 of their size. 110 clear watermark level when FIFO is full on 7/8 of their size. 111 clear watermark level when FIFO is full.
8–6 WM1_SET	Watermark "set" level of DMASMFC channel 1. 000 set watermark level when FIFO is full on 1/8 of their size. 001 set watermark level when FIFO is full on 2/8 of their size. 110 set watermark level when FIFO is full on 7/8 of their size. 111 set watermark level when FIFO is full.
5–3 WM0_CLR	Watermark "clear" level of DMASMFC channel 0. 000 clear watermark level when FIFO is full on 1/8 of their size. 001 clear watermark level when FIFO is full on 2/8 of their size. 110 clear watermark level when FIFO is full on 7/8 of their size. 111 clear watermark level when FIFO is full.
2–0 WM0_SET	Watermark "set" level of DMASMFC channel 0. 000 set watermark level when FIFO is full on 1/8 of their size. 001 set watermark level when FIFO is full on 2/8 of their size. 110 set watermark level when FIFO is full on 7/8 of their size. 111 set watermark level when FIFO is full.

45.51.227 SMFC Burst Size Register (IPU_SMFC_BS)

This register holds the burst size value for each DMASMFC channel. The burst size is the number of IDMAC's active accesses that will done for each IDMAC's burst. This number is a function of PFS, BPP & NPB parameters in the IDMAC's CPMEM. These are the parameters corresponding to the IDMAC's channel used. The table below describes what should be the burst size according to PFS, BPP & NPB settings

Table 45-277. SMFC Burst Size

BPP	PFS	BURST_SIZE
8	6	NPB[6:4]
16	6	NPB[6:4]
All other	All other	NPB[6:2]

Address: IPU_SMFC_BS is 1E00_0000h base + 50008h offset = 1E05_0008h

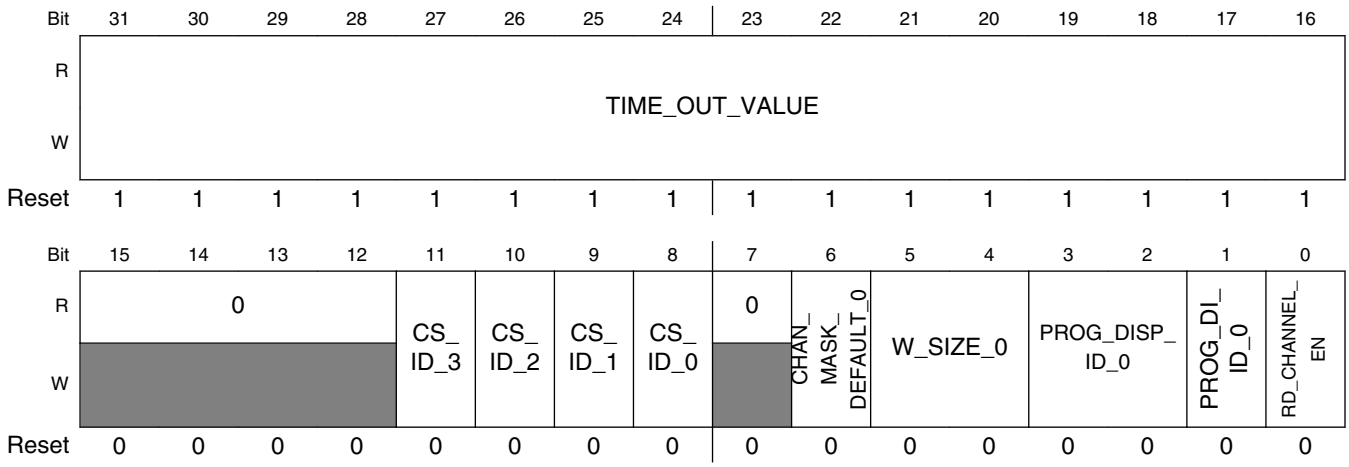
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																BURST3_SIZE				BURST2_SIZE				BURST1_SIZE				BURST0_SIZE			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_SMFC_BS field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–12 BURST3_SIZE	Burst Size of SMFCDMA channel 3. The value programed here should be BURST_SIZE-1 (for example: for a burst size of 1, write 0 to these bits)
11–8 BURST2_SIZE	Burst Size of SMFCDMA channel 2. The value programed here should be BURST_SIZE-1 (for example: for a burst size of 1, write 0 to these bits)
7–4 BURST1_SIZE	Burst Size of SMFCDMA channel 1. The value programed here should be BURST_SIZE-1 (for example: for a burst size of 1, write 0 to these bits)
3–0 BURST0_SIZE	Burst Size of SMFCDMA channel 0. The value programed here should be BURST_SIZE-1 (for example: for a burst size of 1, write 0 to these bits)

45.51.228 DC Read Channel Configuration Register (IPU_DC_READ_CH_CONF)

Address: IPU_DC_READ_CH_CONF is 1E00_0000h base + 58000h offset = 1E05_8000h



IPU_DC_READ_CH_CONF field descriptions

Field	Description
31–16 TIME_OUT_ VALUE	Time out value. In case of a error during read accesses to the display, where no response from the display was received. A time-out counter will terminate the current access and perform the next commend defined in the microcode. This field defines the amount of the hsp_clk cycles counted before the time-out event is issued. This event is tied to the interrupt controller and can generate an error interrupt.
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11 CS_ID_3	This bit maps an asynchronous display to a chip select 1 display #3 is connected to CS1 0 display #3 is connected to CS0
10 CS_ID_2	This bit maps an asynchronous display to a chip select 1 display #2 is connected to CS1 0 display #2 is connected to CS0
9 CS_ID_1	This bit maps an asynchronous display to a chip select 1 display #1 is connected to CS1 0 display #1 is connected to CS0
8 CS_ID_0	This bit maps an asynchronous display to a chip select 1 display #0 is connected to CS1 0 display #0 is connected to CS0

Table continues on the next page...

IPU_DC_READ_CH_CONF field descriptions (continued)

Field	Description
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6 CHAN_MASK_DEFAULT_0	Event mask bit for the read channel When more then one event is used during a flow (EOF, EOL, NL, NF, EOFIELD, etc.) masks all the event besides the event that is defined as the highest priority event 1 All the events are used - no mask 0 Only the highest priority event is used, the rest are masked
5–4 W_SIZE_0	Word Size The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used
3–2 PROG_DISP_ID_0	The field defines which one of the 4 displays can be read. 00 display #0 01 display #1 10 display #2 11 display #3
1 PROG_DI_ID_0	This bit select the DI which a read transaction can be performed through 1 DI #1 0 DI #0
0 RD_CHANNEL_EN	This bit enables the read channel. 1 The Read channel is enabled 0 The Read channel is disabled

45.51.229 DC Read Channel Start Address Register (IPU_DC_READ_SH_ADDR)

Address: IPU_DC_READ_SH_ADDR is 1E00_0000h base + 58004h offset = 1E05_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																ST_ADDR_0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_READ_SH_ADDR field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 ST_ADDR_0	This field defines the start address within the display's memory space where the read transactions will be done from.

45.51.230 DC Routine Link Register 0 Channel 0 (IPU_DC_RL0_CH_0)

Address: IPU_DC_RL0_CH_0 is 1E00_0000h base + 58008h offset = 1E05_8008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NL_START_CHAN_0								0				COD_NL_PRIORITY_CHAN_0				COD_NF_START_CHAN_0								0				COD_NF_PRIORITY_CHAN_0			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL0_CH_0 field descriptions

Field	Description
31–24 COD_NL_START_CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the new line event (NL) resides
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_NL_PRIORITY_CHAN_0	This field defines the priority of the new line (NL) event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_NF_START_CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the new Frame event (NF) resides
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NF_PRIORITY_CHAN_0	This field defines the priority of the new frame (NF) event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable

Table continues on the next page...

IPU_DC_RL0_CH_0 field descriptions (continued)

Field	Description
	0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.231 DC Routine Link Register 1 Channel 0 (IPU_DC_RL1_CH_0)

Address: IPU_DC_RL1_CH_0 is 1E00_0000h base + 5800Ch offset = 1E05_800Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COD_NFIELD_START_CHAN_0								0				COD_NFIELD_PRIORITY_CHAN_0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_EOF_START_CHAN_0								0				COD_EOF_PRIORITY_CHAN_0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL1_CH_0 field descriptions

Field	Description
31–24 COD_NFIELD_START_CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the new field event resides
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_NFIELD_PRIORITY_CHAN_0	This field defines the priority of the new field event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOF_START_CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the end-of-frame event (EOF) resides

Table continues on the next page...

IPU_DC_RL1_CH_0 field descriptions (continued)

Field	Description
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_EOF_ PRIORITY_ CHAN_0	This field defines the priority of the end-of-frame event (EOF) event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.232 DC Routine Link Registe3 Channel 0 (IPU_DC_RL3_CH_0)

Address: IPU_DC_RL3_CH_0 is 1E00_0000h base + 58014h offset = 1E05_8014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COD_NEW_CHAN_START_CHAN_0								0				COD_NEW_CHAN_PRIORITY_CHAN_0				COD_NEW_ADDR_START_CHAN_0								0				COD_NEW_ADDR_PRIORITY_CHAN_0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL3_CH_0 field descriptions

Field	Description
31–24 COD_NEW_ CHAN_START_ CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_NEW_ CHAN_ PRIORITY_ CHAN_0	This field defines the priority of the new channel event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

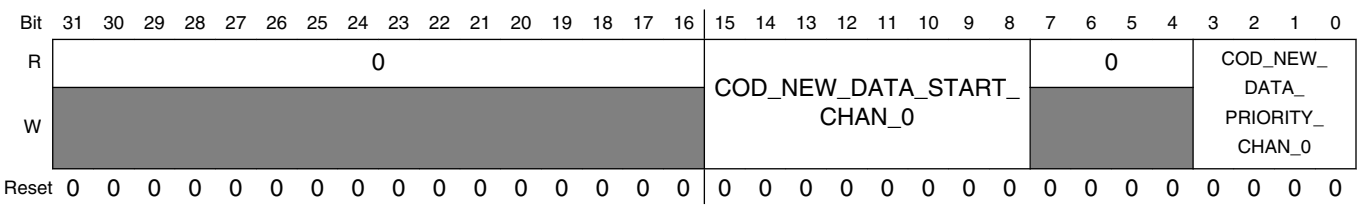
Table continues on the next page...

IPU_DC_RL3_CH_0 field descriptions (continued)

Field	Description
15–8 COD_NEW_ ADDR_START_ CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NEW_ ADDR_ PRIORITY_ CHAN_0	This field defines the priority of the new address event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.233 DC Routine Link Register 4 Channel 0 (IPU_DC_RL4_CH_0)

Address: IPU_DC_RL4_CH_0 is 1E00_0000h base + 58018h offset = 1E05_8018h



IPU_DC_RL4_CH_0 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 COD_NEW_ DATA_START_ CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NEW_ DATA_ PRIORITY_ CHAN_0	This field defines the priority of the new data event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest)

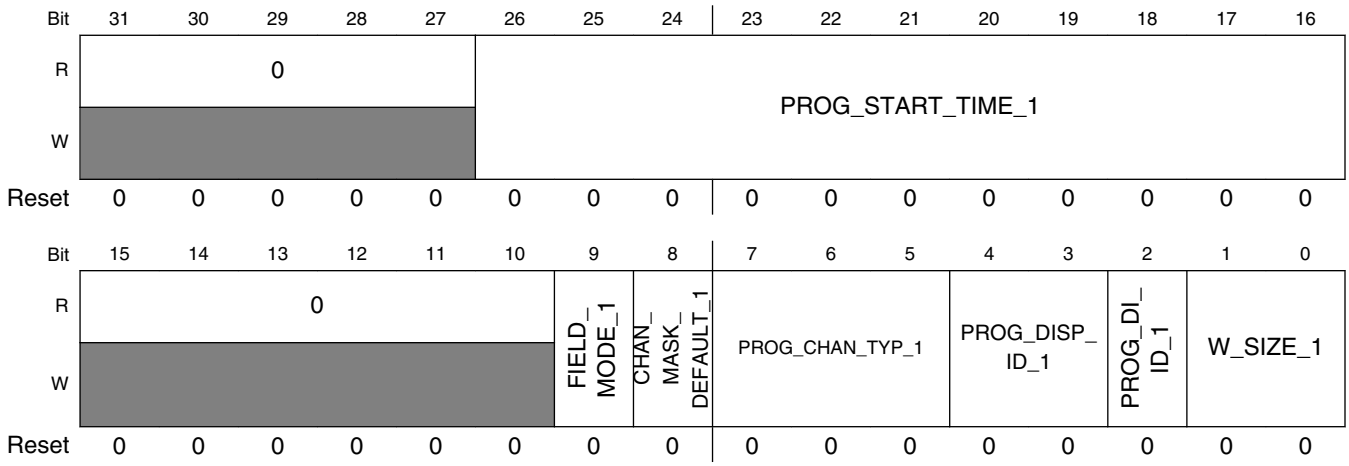
Table continues on the next page...

IPU_DC_RL4_CH_0 field descriptions (continued)

Field	Description
	0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.234 DC Write Channel 1 Configuration Register (IPU_DC_WR_CH_CONF_1)

Address: IPU_DC_WR_CH_CONF_1 is 1E00_0000h base + 5801Ch offset = 1E05_801Ch



IPU_DC_WR_CH_CONF_1 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26–16 PROG_START_TIME_1	This field defines the delay between display 's vertical synchronization pulse and the start time point of DC's channel 1 window. The delay is defined in pairs of rows. It is used for tearing elimination
15–10 Reserved	This read-only field is reserved and always has the value zero. Reserved.
9 FIELD_MODE_1	Field mode bit for channel #1 This bit defines if the channel works in field mode or frame mode; This bit is relevant if the flow is sync flow 1 Field mode 0 Frame mode
8 CHAN_MASK_DEFAULT_1	Event mask bit for channel #1 When more then one event is used during a flow (EOF, EOL, NL, NF, EOFIELD, etc.) masks all the event besides the event that is defined as the highest priority event

Table continues on the next page...

IPU_DC_WR_CH_CONF_1 field descriptions (continued)

Field	Description
	1 All the events are used - no mask 0 Only the highest priority event is used, the rest are masked
7–5 PROG_CHAN_TYP_1	This field define the mode of operation of channel #1 000 Disable 001 Reserved 010 Reserved 100 Normal mode without anti-tearing. For sync display this is the only mode allowed 101 Normal mode with anti-tearing 110 Reserved 111 Additional command channel is added to the flow handled by DC channel #1
4–3 PROG_DISP_ID_1	The field defines which one of the 4 displays is associated with channel #1. 00 display #0 01 display #1 10 display #2 11 display #3
2 PROG_DI_ID_1	This bit select the DI which a transaction associated with channel #1 can be performed to 1 DI #1 0 DI #0
1–0 W_SIZE_1	Word Size associated with channel #1 The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used

45.51.235 DC Routine Link Register2 Channel 0 (IPU_DC_RL2_CH_0)

Address: IPU_DC_RL2_CH_0 is 1E00_0000h base + 58020h offset = 1E05_8020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COD_EOFIELD_START_CHAN_0								0				COD_EOFIELD_PRIORITY_CHAN_0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

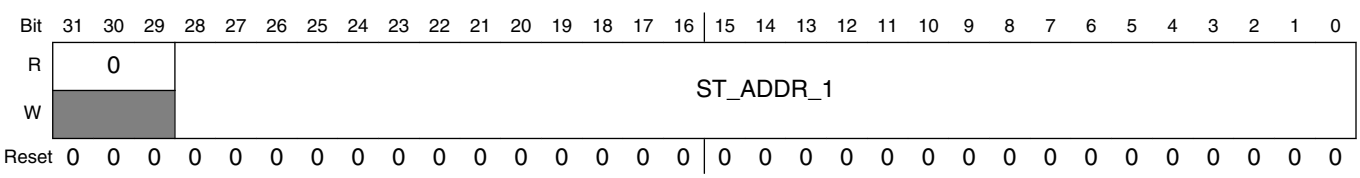
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_EOL_START_CHAN_0								0				COD_EOL_PRIORITY_CHAN_0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL2_CH_0 field descriptions

Field	Description
31–24 COD_EOFIELD_START_CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the end-of-field event resides
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_EOFIELD_PRIORITY_CHAN_0	This field defines the priority of the end-of-field event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOL_START_CHAN_0	This field is a pointer to the address within the microcode memory where the routine that handles the end-of-line event (EOL) resides
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_EOL_PRIORITY_CHAN_0	This field defines the priority of the end-of-line event (EOL) event The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.236 DC Write Channel 1 Address Configuration Register (IPU_DC_WR_CH_ADDR_1)

Address: IPU_DC_WR_CH_ADDR_1 is 1E00_0000h base + 58020h offset = 1E05_8020h

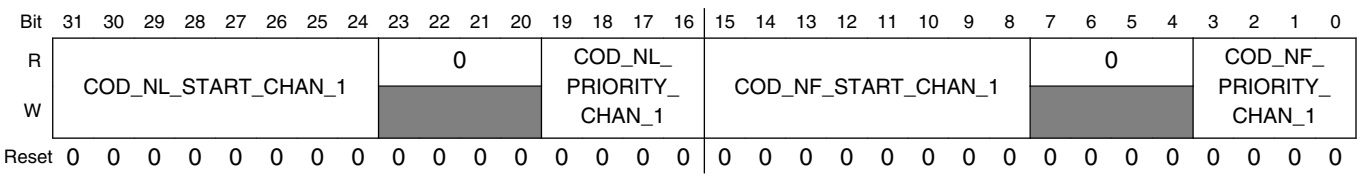


IPU_DC_WR_CH_ADDR_1 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 ST_ADDR_1	This field defines the start address within the display's memory space where the write transactions will be done to for channel #1.

45.51.237 DC Routine Link Register 0 Channel 1 (IPU_DC_RL0_CH_1)

Address: IPU_DC_RL0_CH_1 is 1E00_0000h base + 58024h offset = 1E05_8024h



IPU_DC_RL0_CH_1 field descriptions

Field	Description
31–24 COD_NL_ START_CHAN_ 1	This field is a pointer to the address within the microcode memory where the routine that handles the new line event resides (associated with channel #1)
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_NL_ PRIORITY_ CHAN_1	This field defines the priority of the new line event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_NF_ START_CHAN_ 1	This field is a pointer to the address within the microcode memory where the routine that handles the new frame event resides (associated with channel #1)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NF_ PRIORITY_ CHAN_1	This field defines the priority of the new frame event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable

Table continues on the next page...

IPU_DC_RL0_CH_1 field descriptions (continued)

Field	Description
	0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.238 DC Routine Link Register 1 Channel 1 (IPU_DC_RL1_CH_1)

Address: IPU_DC_RL1_CH_1 is 1E00_0000h base + 58028h offset = 1E05_8028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COD_NFIELD_START_CHAN_1								0				COD_NFIELD_PRIORITY_CHAN_1			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_EOF_START_CHAN_1								0				COD_EOF_PRIORITY_CHAN_1			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL1_CH_1 field descriptions

Field	Description
31–24 COD_NFIELD_START_CHAN_1	This field is a pointer to the address within the microcode memory where the routine that handles the new field event resides (associated with channel #1)
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_NFIELD_PRIORITY_CHAN_1	This field defines the priority of the new field event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOF_START_CHAN_1	This field is a pointer to the address within the microcode memory where the routine that handles the end of frame event resides (associated with channel #1)

Table continues on the next page...

IPU_DC_RL1_CH_1 field descriptions (continued)

Field	Description
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_EOF_PRIORITY_CHAN_1	This field defines the priority of the end of frame event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.239 DC Routine Link Register 2 Channel 2 (IPU_DC_RL2_CH_2)

Address: IPU_DC_RL2_CH_2 is 1E00_0000h base + 58028h offset = 1E05_8028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COD_EOFIELD_START_CHAN_2								0				COD_EOFIELD_PRIORITY_CHAN_2			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_EOL_START_CHAN_2								0				COD_EOL_PRIORITY_CHAN_2			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL2_CH_2 field descriptions

Field	Description
31–24 COD_EOFIELD_START_CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the end of field event resides (associated with channel #2)
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_EOFIELD_PRIORITY_CHAN_2	This field defines the priority of the end of field event (associated with channel #2) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest)

Table continues on the next page...

IPU_DC_RL2_CH_2 field descriptions (continued)

Field	Description
	1110 Reserved 1111 Reserved
15–8 COD_EOL_START_CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the end of line event resides (associated with channel #2)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_EOL_PRIORITY_CHAN_2	This field defines the priority of the end of line event (associated with channel #2) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.240 DC Routine Link Register 2 Channel 1 (IPU_DC_RL2_CH_1)

Address: IPU_DC_RL2_CH_1 is 1E00_0000h base + 58030h offset = 1E05_8030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COD_EOFIELD_START_CHAN_1								0				COD_EOFIELD_PRIORITY_CHAN_1			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_EOL_START_CHAN_1								0				COD_EOL_PRIORITY_CHAN_1			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL2_CH_1 field descriptions

Field	Description
31–24 COD_EOFIELD_START_CHAN_1	This field is a pointer to the address within the microcode memory where the routine that handles the end of field event resides (associated with channel #1)
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_RL2_CH_1 field descriptions (continued)

Field	Description
19–16 COD_EOFIELD_PRIORITY_CHAN_1	This field defines the priority of the end of field event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOL_START_CHAN_1	This field is a pointer to the address within the microcode memory where the routine that handles the end of line event resides (associated with channel #1)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_EOL_PRIORITY_CHAN_1	This field defines the priority of the end of line event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.241 DC Routine Link Register 3 Channel 1 (IPU_DC_RL3_CH_1)

Address: IPU_DC_RL3_CH_1 is 1E00_0000h base + 58032h offset = 1E05_8032h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_CHAN_START_CHAN_1								0				COD_NEW_CHAN_PRIORITY_CHAN_1				COD_NEW_ADDR_START_CHAN_1								0				COD_NEW_ADDR_PRIORITY_CHAN_1			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL3_CH_1 field descriptions

Field	Description
31–24 COD_NEW_CHAN_START_CHAN_1	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #1)

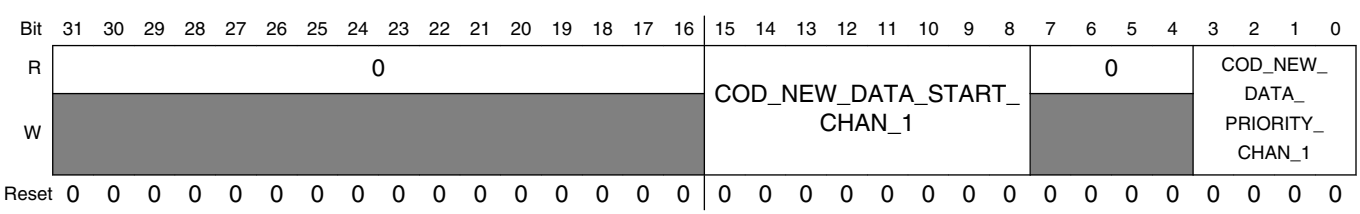
Table continues on the next page...

IPU_DC_RL3_CH_1 field descriptions (continued)

Field	Description
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_NEW_ CHAN_ PRIORITY_ CHAN_1	This field defines the priority of the new channel event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_NEW_ ADDR_START_ CHAN_1	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #1)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NEW_ ADDR_ PRIORITY_ CHAN_1	This field defines the priority of the new address event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.242 DC Routine Link Register 4 Channel 1 (IPU_DC_RL4_CH_1)

Address: IPU_DC_RL4_CH_1 is 1E00_0000h base + 58034h offset = 1E05_8034h



IPU_DC_RL4_CH_1 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.

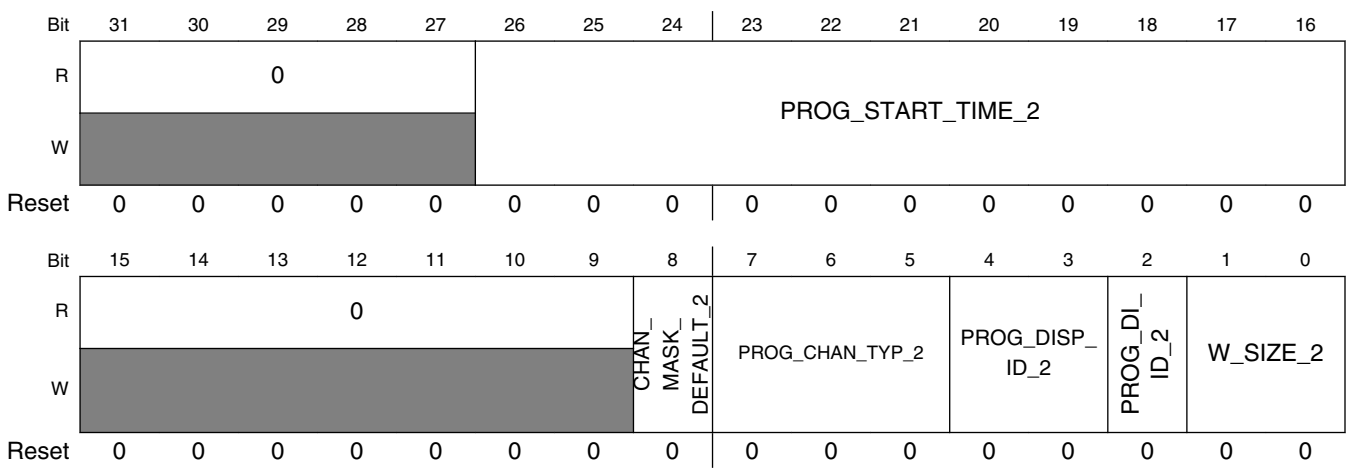
Table continues on the next page...

IPU_DC_RL4_CH_1 field descriptions (continued)

Field	Description
15–8 COD_NEW_DATA_START_CHAN_1	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #1)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NEW_DATA_PRIORITY_CHAN_1	This field defines the priority of the new data event (associated with channel #1) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.243 DC Write Channel 2 Configuration Register (IPU_DC_WR_CH_CONF_2)

Address: IPU_DC_WR_CH_CONF_2 is 1E00_0000h base + 58038h offset = 1E05_8038h



IPU_DC_WR_CH_CONF_2 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26–16 PROG_START_TIME_2	This field defines the delay between display 's vertical synchronization pulse and the start time point of DC's channel 2 window. The delay is defined in pairs of rows. It is used for tearing elimination

Table continues on the next page...

IPU_DC_WR_CH_CONF_2 field descriptions (continued)

Field	Description
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8 CHAN_MASK_DEFAULT_2	Event mask bit for channel #2 When more then one event is used during a flow (EOF, EOL, NL, NF, EOFIELD, etc.) masks all the event besides the event that is defined as the highest priority event 1 All the events are used - no mask 0 Only the highest priority event is used, the rest are masked
7–5 PROG_CHAN_TYP_2	This field define the mode of operation of channel #2 000 Disable 001 Reserved 010 Reserved 100 Normal mode without anti-tearing. For sync display this is the only mode allowed 101 Normal mode with anti-tearing 110 Reserved 111 Additional command channel is added to the flow handled by DC channel #2
4–3 PROG_DISP_ID_2	The field defines which one of the 4 displays is associated with channel #2. 00 display #0 01 display #1 10 display #2 11 display #3
2 PROG_DI_ID_2	This bit select the DI which a transaction associated with channel #2 can be performed to 1 DI #1 0 DI #0
1–0 W_SIZE_2	Word Size The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used

45.51.244 DC Write Channel 2 Address Configuration Register (IPU_DC_WR_CH_ADDR_2)

Address: IPU_DC_WR_CH_ADDR_2 is 1E00_0000h base + 5803Ch offset = 1E05_803Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																ST_ADDR_2															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_WR_CH_ADDR_2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 ST_ADDR_2	This field defines the start address within the display's memory space where the write transactions will be done to for channel #2.

45.51.245 DC Routine Link Register 0 Channel 2 (IPU_DC_RL0_CH_2)

Address: IPU_DC_RL0_CH_2 is 1E00_0000h base + 58040h offset = 1E05_8040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NL_START_CHAN_2								0				COD_NL_PRIORITY_CHAN_2				COD_NF_START_CHAN_2								0				COD_NF_PRIORITY_CHAN_2			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL0_CH_2 field descriptions

Field	Description
31–24 COD_NL_START_CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the new line event resides (associated with channel #2)
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_NL_PRIORITY_CHAN_2	This field defines the priority of the new line event (associated with channel #2) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_NF_START_CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the new frame event resides (associated with channel #2)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NF_PRIORITY_CHAN_2	This field defines the priority of the new frame event (associated with channel #2) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable

Table continues on the next page...

IPU_DC_RL0_CH_2 field descriptions (continued)

Field	Description
	0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.246 DC Routine Link Register 1 Channel 2 (IPU_DC_RL1_CH_2)

Address: IPU_DC_RL1_CH_2 is 1E00_0000h base + 58044h offset = 1E05_8044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COD_NFIELD_START_CHAN_2								0				COD_NFIELD_PRIORITY_CHAN_2			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_EOF_START_CHAN_2								0				COD_EOF_PRIORITY_CHAN_2			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL1_CH_2 field descriptions

Field	Description
31–24 COD_NFIELD_START_CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the new field event resides (associated with channel #2)
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_NFIELD_PRIORITY_CHAN_2	This field defines the priority of the new field event (associated with channel #2) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOF_START_CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the end of frame event resides (associated with channel #2)

Table continues on the next page...

IPU_DC_RL1_CH_2 field descriptions (continued)

Field	Description
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_EOF_ PRIORITY_ CHAN_2	This field defines the priority of the end of frame event (associated with channel #2) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.247 DC Routine Link Register 3 Channel 2 (IPU_DC_RL3_CH_2)

Address: IPU_DC_RL3_CH_2 is 1E00_0000h base + 5804Ch offset = 1E05_804Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_CHAN_START_CHAN_2								0				COD_NEW_CHAN_PRIORITY_CHAN_2				COD_NEW_ADDR_START_CHAN_2								0				COD_NEW_ADDR_PRIORITY_CHAN_2			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL3_CH_2 field descriptions

Field	Description
31–24 COD_NEW_ CHAN_START_ CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #2)
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_NEW_ CHAN_ PRIORITY_ CHAN_2	This field defines the priority of the end of line event (associated with channel #2) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

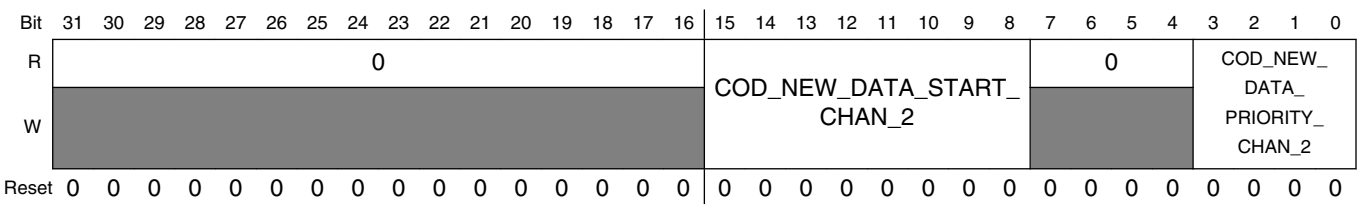
Table continues on the next page...

IPU_DC_RL3_CH_2 field descriptions (continued)

Field	Description
15–8 COD_NEW_ ADDR_START_ CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #2)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NEW_ ADDR_ PRIORITY_ CHAN_2	This field defines the priority of the end of line event (associated with channel #2) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.248 DC Routine Link Register 4 Channel 2 (IPU_DC_RL4_CH_2)

Address: IPU_DC_RL4_CH_2 is 1E00_0000h base + 58050h offset = 1E05_8050h



IPU_DC_RL4_CH_2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 COD_NEW_ DATA_START_ CHAN_2	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #2)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NEW_ DATA_ PRIORITY_ CHAN_2	This field defines the priority of the end of line event (associated with channel #2) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest)

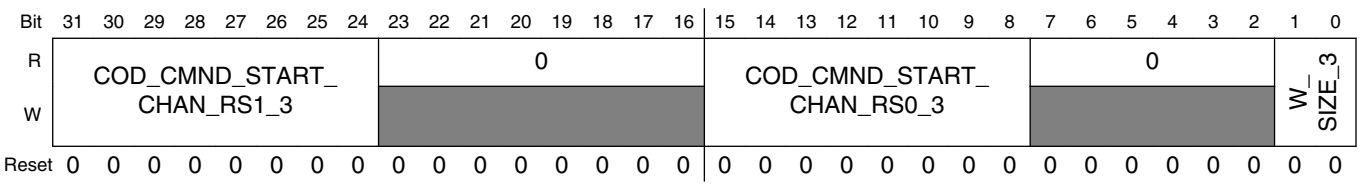
Table continues on the next page...

IPU_DC_RL4_CH_2 field descriptions (continued)

Field	Description
	0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.249 DC Command Channel 3 Configuration Register (IPU_DC_CMD_CH_CONF_3)

Address: IPU_DC_CMD_CH_CONF_3 is 1E00_0000h base + 58054h offset = 1E05_8054h

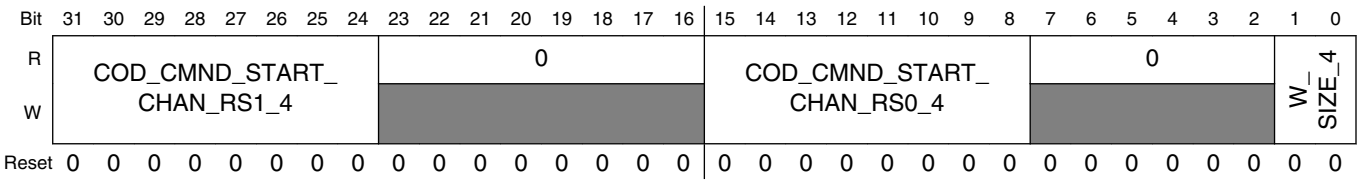


IPU_DC_CMD_CH_CONF_3 field descriptions

Field	Description
31–24 COD_CMND_START_CHAN_RS1_3	This field is a pointer to the address within the microcode memory where the routine that handles the command start event resides (associated with channel #3); This field is relevant when RS is equal to 1
23–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 COD_CMND_START_CHAN_RS0_3	This field is a pointer to the address within the microcode memory where the routine that handles the command start event resides (associated with channel #3); This field is relevant when RS is equal to 0
7–2 Reserved	This read-only field is reserved and always has the value zero. Reserved.
1–0 W_SIZE_3	Word Size associated with channel #3 The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used

45.51.250 DC Command Channel 4 Configuration Register (IPU_DC_CMD_CH_CONF_4)

Address: IPU_DC_CMD_CH_CONF_4 is 1E00_0000h base + 58058h offset = 1E05_8058h

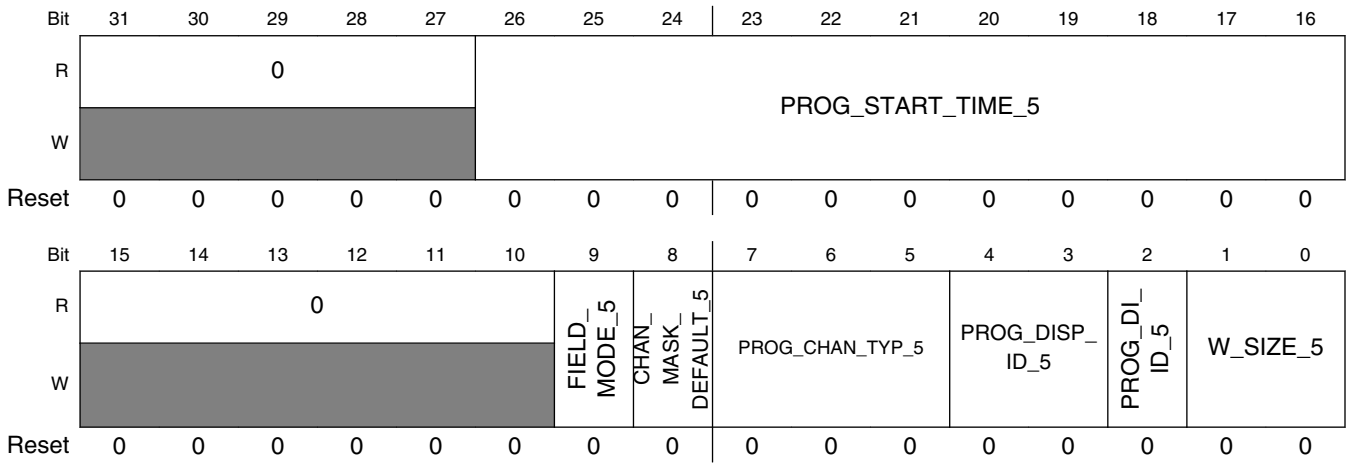


IPU_DC_CMD_CH_CONF_4 field descriptions

Field	Description
31–24 COD_CMND_START_CHAN_RS1_4	This field is a pointer to the address within the microcode memory where the routine that handles the command start event resides (associated with channel #4); This field is relevant when RS is equal to 1
23–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 COD_CMND_START_CHAN_RS0_4	This field is a pointer to the address within the microcode memory where the routine that handles the command start event resides (associated with channel #4); This field is relevant when RS is equal to 0
7–2 Reserved	This read-only field is reserved and always has the value zero. Reserved.
1–0 W_SIZE_4	Word Size associated with channel #4 The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used

45.51.251 DC Write Channel 5Configuration Register (IPU_DC_WR_CH_CONF_5)

Address: IPU_DC_WR_CH_CONF_5 is 1E00_0000h base + 5805Ch offset = 1E05_805Ch



IPU_DC_WR_CH_CONF_5 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26–16 PROG_START_TIME_5	This field defines the delay between display 's vertical synchronization pulse and the start time point of DC's channel 5 window. The delay is defined in pairs of rows. It is used for tearing elimination
15–10 Reserved	This read-only field is reserved and always has the value zero. Reserved.
9 FIELD_MODE_5	Field mode bit for channel #5 This bit defines if the channel works in field mode or frame mode; This bit is relevant if the flow is sync flow 1 Field mode 0 Frame mode
8 CHAN_MASK_DEFAULT_5	Event mask bit for channel #5 When more then one event is used during a flow (EOF, EOL, NL, NF, EOFIELD, etc.) masks all the event besides the event that is defined as the highest priority event 1 All the events are used - no mask 0 Only the highest priority event is used, the rest are masked
7–5 PROG_CHAN_TYP_5	This field define the mode of operation of channel #5 000 Disable 001 Reserved 010 Reserved 100 Normal mode without anti-tearing. For sync display this is the only mode allowed

Table continues on the next page...

IPU_DC_WR_CH_CONF_5 field descriptions (continued)

Field	Description
	101 Normal mode with anti-tearing 110 Reserved 111 Additional command channel is added to the flow handled by DC channel #5
4–3 PROG_DISP_ID_5	The field defines which one of the 4 displays is associated with channel #5. 00 display #0 01 display #1 10 display #2 11 display #3
2 PROG_DI_ID_5	This bit select the DI which a transaction associated with channel #5 can be performed to. When channel 28 is connected to DI0, channel 23 must be connected to DI1 even if ch23 is not used. This is done by writing 1 to this bit. 1 DI #1 0 DI #0
1–0 W_SIZE_5	Word Size associated with channel #5 The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used

45.51.252 DC Write Channel 5Address Configuration Register (IPU_DC_WR_CH_ADDR_5)

Address: IPU_DC_WR_CH_ADDR_5 is 1E00_0000h base + 58060h offset = 1E05_8060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																ST_ADDR_5															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_WR_CH_ADDR_5 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 ST_ADDR_5	This field defines the start address within the display's memory space where the write transactions will be done to for channel #5.

45.51.253 DC Routine Link Register 0 Channel 5 (IPU_DC_RL0_CH_5)

Address: IPU_DC_RL0_CH_5 is 1E00_0000h base + 58064h offset = 1E05_8064h

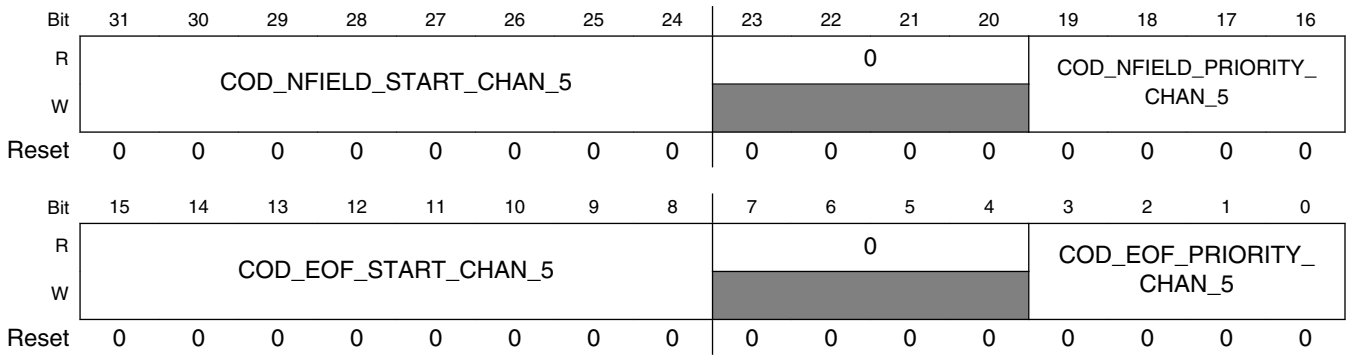
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NL_START_CHAN_5								0				COD_NL_PRIORITY_CHAN_5				COD_NF_START_CHAN_5								0				COD_NF_PRIORITY_CHAN_5			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL0_CH_5 field descriptions

Field	Description
31–24 COD_NL_START_CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the new line event resides (associated with channel #5)
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_NL_PRIORITY_CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_NF_START_CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the new frame event resides (associated with channel #5)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NF_PRIORITY_CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.254 DC Routine Link Register 1 Channel 5 (IPU_DC_RL1_CH_5)

Address: IPU_DC_RL1_CH_5 is 1E00_0000h base + 58068h offset = 1E05_8068h



IPU_DC_RL1_CH_5 field descriptions

Field	Description
31–24 COD_NFIELD_START_CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the new field event resides (associated with channel #5)
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_NFIELD_PRIORITY_CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOF_START_CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the end of frame event resides (associated with channel #5)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_EOF_PRIORITY_CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest)

Table continues on the next page...

IPU_DC_RL1_CH_5 field descriptions (continued)

Field	Description
	1110 Reserved 1111 Reserved

45.51.255 DC Routine Link Register 2 Channel 5 (IPU_DC_RL2_CH_5)

Address: IPU_DC_RL2_CH_5 is 1E00_0000h base + 5806Ch offset = 1E05_806Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COD_EOFIELD_START_CHAN_5								0				COD_EOFIELD_PRIORITY_CHAN_5			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_EOL_START_CHAN_5								0				COD_EOL_PRIORITY_CHAN_5			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL2_CH_5 field descriptions

Field	Description
31–24 COD_EOFIELD_START_CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the end of field event resides (associated with channel #5)
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_EOFIELD_PRIORITY_CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOL_START_CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the end of line event resides (associated with channel #5)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.

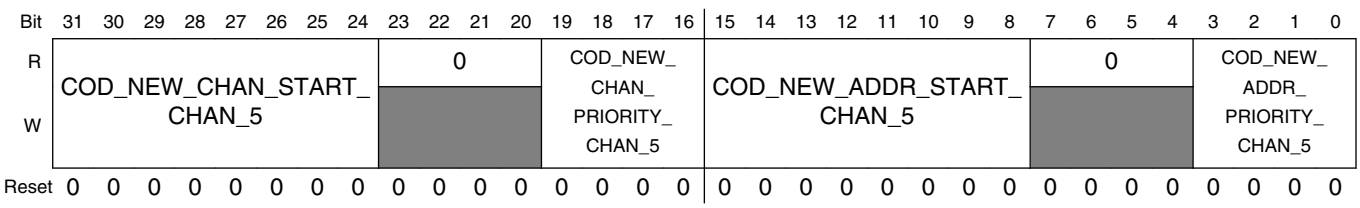
Table continues on the next page...

IPU_DC_RL2_CH_5 field descriptions (continued)

Field	Description
3–0 COD_EOL_ PRIORITY_ CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.256 DC Routine Link Register3 Channel 5 (IPU_DC_RL3_CH_5)

Address: IPU_DC_RL3_CH_5 is 1E00_0000h base + 58070h offset = 1E05_8070h



IPU_DC_RL3_CH_5 field descriptions

Field	Description
31–24 COD_NEW_ CHAN_START_ CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #5)
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_NEW_ CHAN_ PRIORITY_ CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_NEW_ ADDR_START_ CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #5)

Table continues on the next page...

IPU_DC_RL3_CH_5 field descriptions (continued)

Field	Description
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NEW_ADDR_PRIORITY_CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.257 DC Routine Link Register 4 Channel 5 (IPU_DC_RL4_CH_5)

Address: IPU_DC_RL4_CH_5 is 1E00_0000h base + 58074h offset = 1E05_8074h

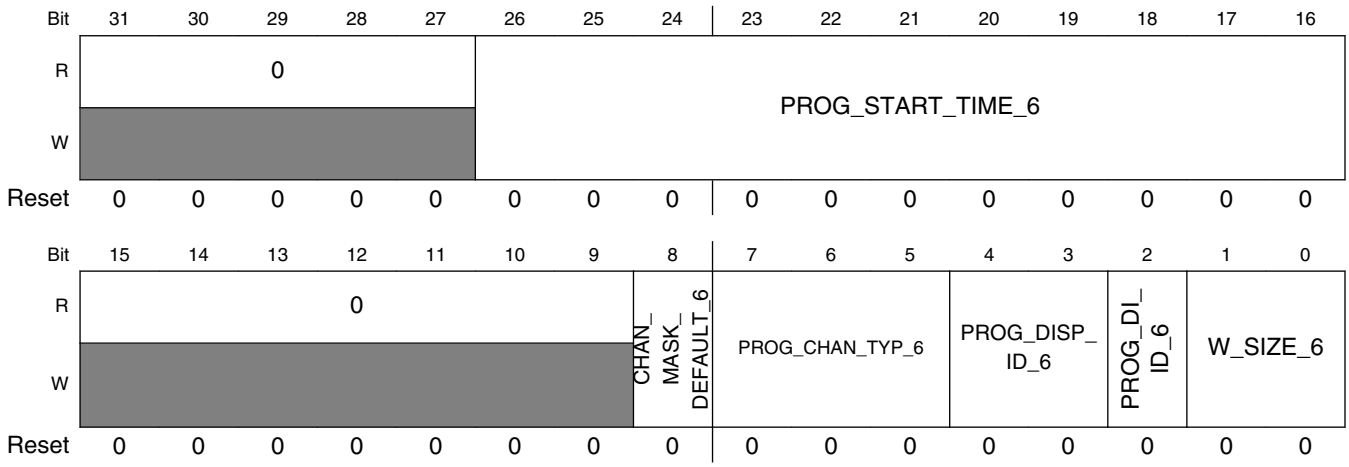
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COD_NEW_DATA_START_CHAN_5								0				COD_NEW_DATA_PRIORITY_CHAN_5			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL4_CH_5 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 COD_NEW_DATA_START_CHAN_5	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #5)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NEW_DATA_PRIORITY_CHAN_5	This field defines the priority of the new line event (associated with channel #5) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.258 DC Write Channel 6 Configuration Register (IPU_DC_WR_CH_CONF_6)

Address: IPU_DC_WR_CH_CONF_6 is 1E00_0000h base + 58078h offset = 1E05_8078h



IPU_DC_WR_CH_CONF_6 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26–16 PROG_START_TIME_6	This field defines the delay between display 's vertical synchronization pulse and the start time point of DC's channel 6 window. The delay is defined in pairs of rows. It is used for tearing elimination
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8 CHAN_MASK_DEFAULT_6	Event mask bit for channel #6 When more then one event is used during a flow (EOF, EOL, NL, NF, EOFIELD, etc.) masks all the event besides the event that is defined as the highest priority event 1 All the events are used - no mask 0 Only the highest priority event is used, the rest are masked
7–5 PROG_CHAN_TYP_6	This field define the mode of operation of channel #6 000 Disable 001 Reserved 010 Reserved 100 Normal mode without anti-tearing. For sync display this is the only mode allowed 101 Normal mode with anti-tearing 110 Reserved 111 Additional command channel is added to the flow handled by DC channel #6
4–3 PROG_DISP_ID_6	The field defines which one of the 4 displays is associated with channel #6. 00 display #0 01 display #1

Table continues on the next page...

IPU_DC_WR_CH_CONF_6 field descriptions (continued)

Field	Description
	10 display #2 11 display #3
2 PROG_DI_ID_6	This bit select the DI which a transaction associated with channel #6 can be performed to 1 DI #1 0 DI #0
1–0 W_SIZE_6	Word Size associated with channel #6 The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used

45.51.259 DC Write Channel 6 Address Configuration Register (IPU_DC_WR_CH_ADDR_6)

Address: IPU_DC_WR_CH_ADDR_6 is 1E00_0000h base + 5807Ch offset = 1E05_807Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																ST_ADDR_6															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_WR_CH_ADDR_6 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 ST_ADDR_6	This field defines the start address within the display's memory space where the write transactions will be done to for channel #6.

45.51.260 DC Routine Link Register 0Channel 6 (IPU_DC_RL0_CH_6)

Address: IPU_DC_RL0_CH_6 is 1E00_0000h base + 58080h offset = 1E05_8080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NL_START_CHAN_6								0				COD_NL_PRIORITY_CHAN_6				COD_NF_START_CHAN_6								0				COD_NF_PRIORITY_CHAN_6			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL0_CH_6 field descriptions

Field	Description
31–24 COD_NL_START_CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the new line event resides (associated with channel #6)
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_NL_PRIORITY_CHAN_6	This field defines the priority of the new line event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_NF_START_CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the new frame event resides (associated with channel #6)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NF_PRIORITY_CHAN_6	This field defines the priority of the new frame event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.261 DC Routine Link Register 1 Channel 6 (IPU_DC_RL1_CH_6)

Address: IPU_DC_RL1_CH_6 is 1E00_0000h base + 58084h offset = 1E05_8084h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COD_NFIELD_START_CHAN_6								0				COD_NFIELD_PRIORITY_CHAN_6			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_EOF_START_CHAN_6								0				COD_EOF_PRIORITY_CHAN_6			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL1_CH_6 field descriptions

Field	Description
31–24 COD_NFIELD_START_CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the new field event resides (associated with channel #6)
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_NFIELD_PRIORITY_CHAN_6	This field defines the priority of the new field event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOF_START_CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the end of frame event resides (associated with channel #6)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_EOF_PRIORITY_CHAN_6	This field defines the priority of the new field event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest)

Table continues on the next page...

IPU_DC_RL1_CH_6 field descriptions (continued)

Field	Description
	1110 Reserved 1111 Reserved

45.51.262 DC Routine Link Register 2 Channel 6 (IPU_DC_RL2_CH_6)

Address: IPU_DC_RL2_CH_6 is 1E00_0000h base + 58088h offset = 1E05_8088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	COD_EOFIELD_START_CHAN_6								0				COD_EOFIELD_PRIORITY_CHAN_6			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_EOL_START_CHAN_6								0				COD_EOL_PRIORITY_CHAN_6			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL2_CH_6 field descriptions

Field	Description
31–24 COD_EOFIELD_START_CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the end of field event resides (associated with channel #6)
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_EOFIELD_PRIORITY_CHAN_6	This field defines the priority of the new field event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_EOL_START_CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the end of line event resides (associated with channel #6)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.

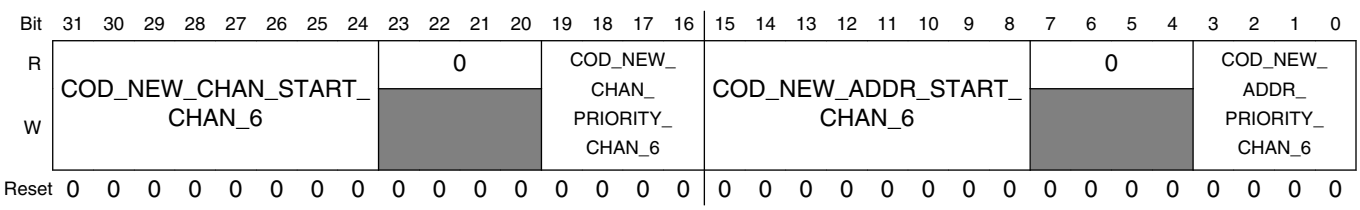
Table continues on the next page...

IPU_DC_RL2_CH_6 field descriptions (continued)

Field	Description
3–0 COD_EOL_ PRIORITY_ CHAN_6	This field defines the priority of the new field event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.263 DC Routine Link Register 3 Channel 6 (IPU_DC_RL3_CH_6)

Address: IPU_DC_RL3_CH_6 is 1E00_0000h base + 5808Ch offset = 1E05_808Ch



IPU_DC_RL3_CH_6 field descriptions

Field	Description
31–24 COD_NEW_ CHAN_START_ CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #6)
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–16 COD_NEW_ CHAN_ PRIORITY_ CHAN_6	This field defines the priority of the new field event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
15–8 COD_NEW_ ADDR_START_ CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #6)

Table continues on the next page...

IPU_DC_RL3_CH_6 field descriptions (continued)

Field	Description
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NEW_ADDR_PRIORITY_CHAN_6	This field defines the priority of the new field event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.264 DC Routine Link Register 4 Channel 6 (IPU_DC_RL4_CH_6)

Address: IPU_DC_RL4_CH_6 is 1E00_0000h base + 58090h offset = 1E05_8090h

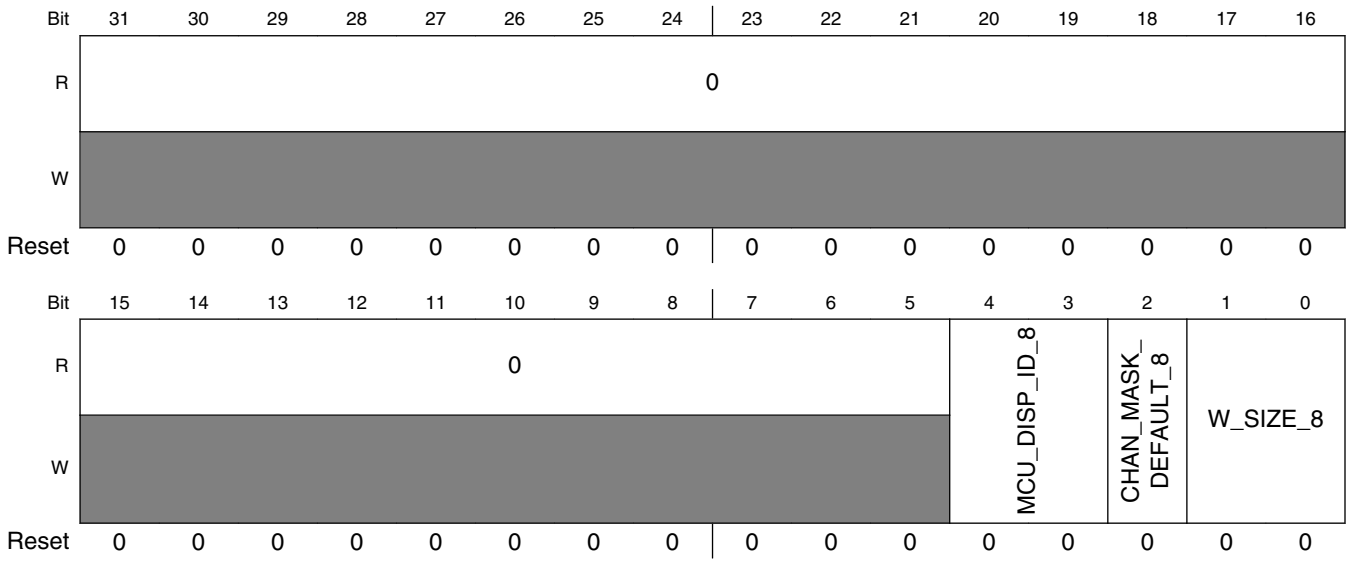
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COD_NEW_DATA_START_CHAN_6								0				COD_NEW_DATA_PRIORITY_CHAN_6			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL4_CH_6 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 COD_NEW_DATA_START_CHAN_6	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #6)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NEW_DATA_PRIORITY_CHAN_6	This field defines the priority of the new field event (associated with channel #6) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.265 DC Write Channel 8 Configuration 1Register (IPU_DC_WR_CH_CONF1_8)

Address: IPU_DC_WR_CH_CONF1_8 is 1E00_0000h base + 58094h offset = 1E05_8094h

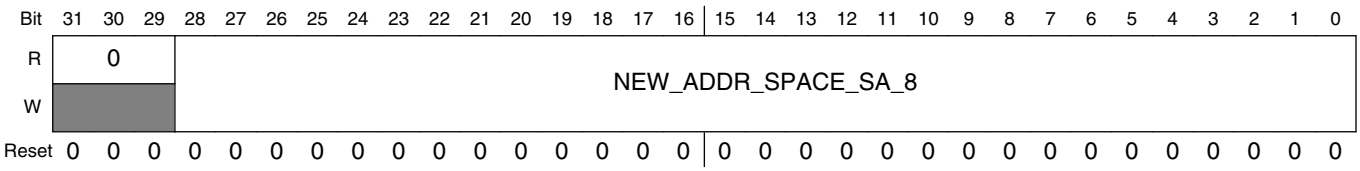


IPU_DC_WR_CH_CONF1_8 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved.
4–3 MCU_DISP_ID_8	The field defines which one of the 4 displays is associated with channel #8. 00 display #0 01 display #1 10 display #2 11 display #3
2 CHAN_MASK_DEFAULT_8	Event mask bit for channel #8 When more then one event is used during a flow (EOF, EOL, NL, NF, EOFIELD, etc.) masks all the event besides the event that is defined as the highest priority event 1 All the events are used - no mask 0 Only the highest priority event is used, the rest are masked
1–0 W_SIZE_8	Word Size associated with channel #8 The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used

45.51.266 DC Write Channel 8 Configuration 2 Register (IPU_DC_WR_CH_CONF2_8)

Address: IPU_DC_WR_CH_CONF2_8 is 1E00_0000h base + 58098h offset = 1E05_8098h

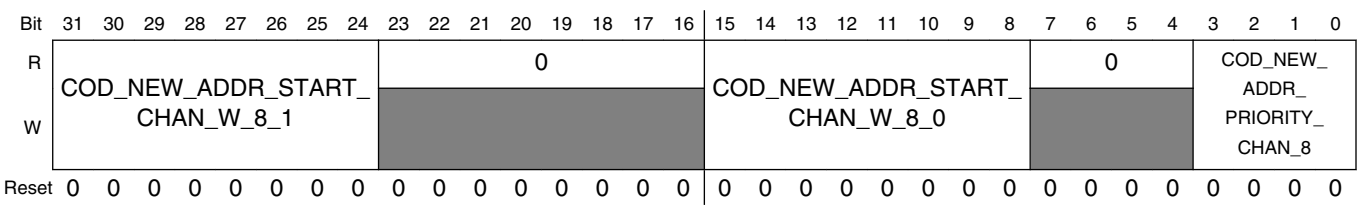


IPU_DC_WR_CH_CONF2_8 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 NEW_ADDR_SPACE_SA_8	Channel #8 is used for ARM platform direct access to the display. This field defines the base address of the second region accessible on the display

45.51.267 DC Routine Link Register 1 Channel 8 (IPU_DC_RL1_CH_8)

Address: IPU_DC_RL1_CH_8 is 1E00_0000h base + 5809Ch offset = 1E05_809Ch



IPU_DC_RL1_CH_8 field descriptions

Field	Description
31–24 COD_NEW_ADDR_START_CHAN_W_8_1	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #8, second region)
23–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 COD_NEW_ADDR_START_CHAN_W_8_0	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #8, first region)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_RL1_CH_8 field descriptions (continued)

Field	Description
3–0 COD_NEW_ ADDR_ PRIORITY_ CHAN_8	This field defines the priority of the new address event (associated with channel #8, both regions) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.268 DC Routine Link Register 2 Channel 8 (IPU_DC_RL2_CH_8)

Address: IPU_DC_RL2_CH_8 is 1E00_0000h base + 580A0h offset = 1E05_80A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_CHAN_START_CHAN_W_8_1								0								COD_NEW_CHAN_START_CHAN_W_8_0								0				COD_NEW_CHAN_PRIORITY_CHAN_8			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL2_CH_8 field descriptions

Field	Description
31–24 COD_NEW_ CHAN_START_ CHAN_W_8_1	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #8, second region)
23–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 COD_NEW_ CHAN_START_ CHAN_W_8_0	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #8, first region)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NEW_ CHAN_ PRIORITY_ CHAN_8	This field defines the priority of the new address event (associated with channel #8, both regions) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest)

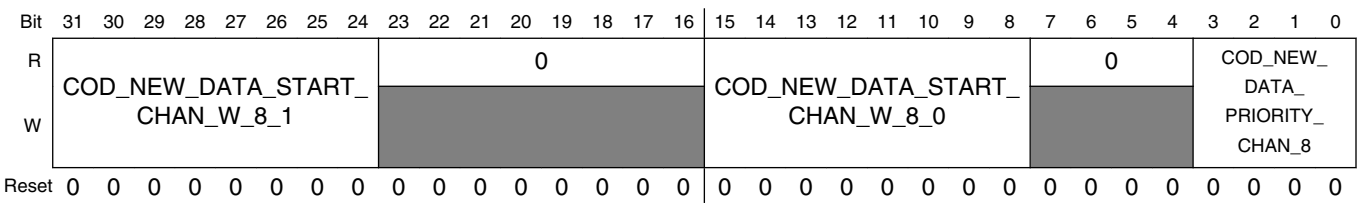
Table continues on the next page...

IPU_DC_RL2_CH_8 field descriptions (continued)

Field	Description
	1110 Reserved 1111 Reserved

45.51.269 DC Routine Link Register 3 Channel 8 (IPU_DC_RL3_CH_8)

Address: IPU_DC_RL3_CH_8 is 1E00_0000h base + 580A4h offset = 1E05_80A4h



IPU_DC_RL3_CH_8 field descriptions

Field	Description
31–24 COD_NEW_DATA_START_CHAN_W_8_1	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #8,second region)
23–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 COD_NEW_DATA_START_CHAN_W_8_0	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #8, first region)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NEW_DATA_PRIORITY_CHAN_8	This field defines the priority of the new address event (associated with channel #8, both regions) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.270 DC Routine Link Register 4 Channel 8 (IPU_DC_RL4_CH_8)

Address: IPU_DC_RL4_CH_8 is 1E00_0000h base + 580A8h offset = 1E05_80A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_ADDR_START_CHAN_R_8_1								0								COD_NEW_ADDR_START_CHAN_R_8_0								0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL4_CH_8 field descriptions

Field	Description
31–24 COD_NEW_ADDR_START_CHAN_R_8_1	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #8, second region)
23–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 COD_NEW_ADDR_START_CHAN_R_8_0	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #8, first region)
7–0 Reserved	This read-only field is reserved and always has the value zero. Reserved.

45.51.271 DC Routine Link Register 5 Channel 8 (IPU_DC_RL5_CH_8)

Address: IPU_DC_RL5_CH_8 is 1E00_0000h base + 580ACh offset = 1E05_80ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_CHAN_START_CHAN_R_8_1								0								COD_NEW_CHAN_START_CHAN_R_8_0								0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL5_CH_8 field descriptions

Field	Description
31–24 COD_NEW_CHAN_START_CHAN_R_8_1	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #8, second region)
23–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_RL5_CH_8 field descriptions (continued)

Field	Description
15–8 COD_NEW_ CHAN_START_ CHAN_R_8_0	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #8, first region)
7–0 Reserved	This read-only field is reserved and always has the value zero. Reserved.

45.51.272 DC Routine Link Register 6 Channel 8 (IPU_DC_RL6_CH_8)

Address: IPU_DC_RL6_CH_8 is 1E00_0000h base + 580B0h offset = 1E05_80B0h

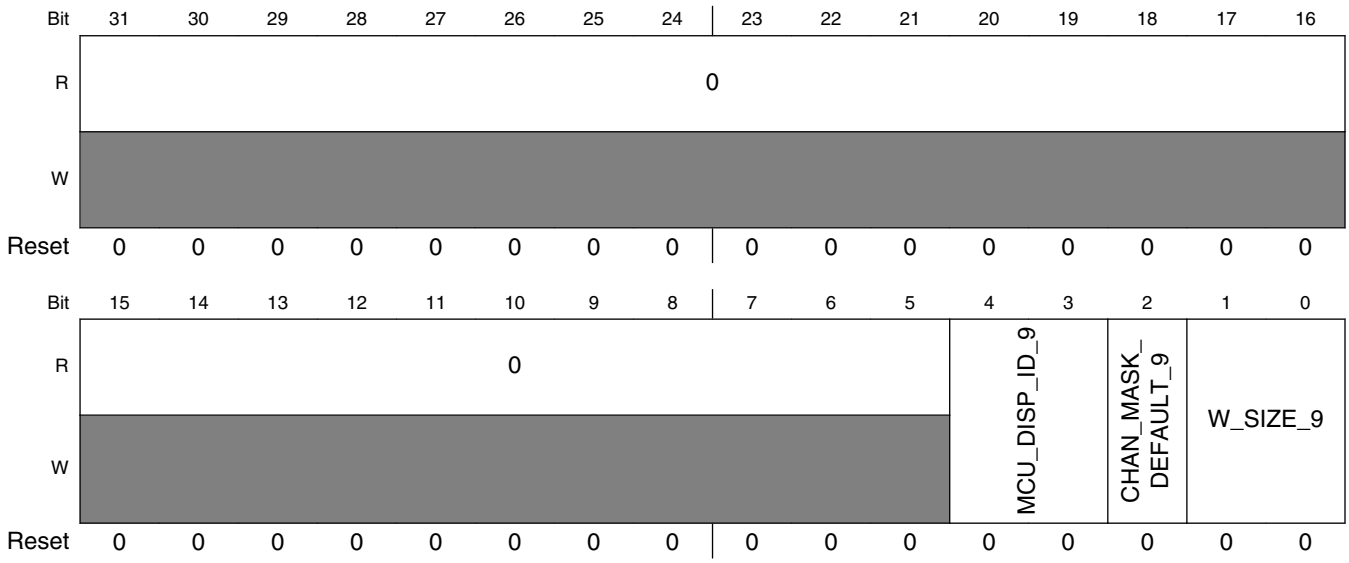
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_DATA_START_CHAN_R_8_1								0								COD_NEW_DATA_START_CHAN_R_8_0								0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL6_CH_8 field descriptions

Field	Description
31–24 COD_NEW_ DATA_START_ CHAN_R_8_1	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #8, second region)
23–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 COD_NEW_ DATA_START_ CHAN_R_8_0	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #8, first region)
7–0 Reserved	This read-only field is reserved and always has the value zero. Reserved.

45.51.273 DC Write Channel 9 Configuration 1Register (IPU_DC_WR_CH_CONF1_9)

Address: IPU_DC_WR_CH_CONF1_9 is 1E00_0000h base + 580B4h offset = 1E05_80B4h

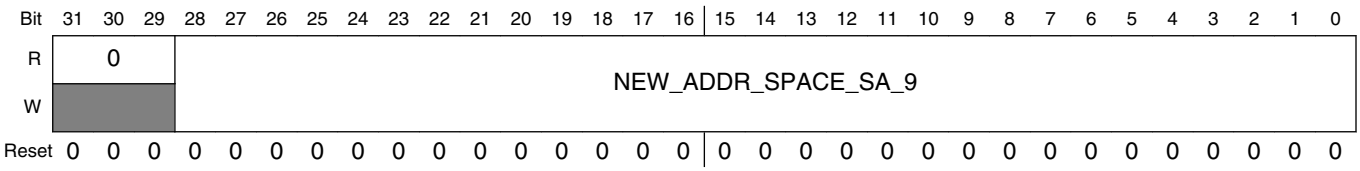


IPU_DC_WR_CH_CONF1_9 field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved.
4–3 MCU_DISP_ID_9	The field defines which one of the 4 displays is associated with channel #9. 00 display #0 01 display #1 10 display #2 11 display #3
2 CHAN_MASK_DEFAULT_9	Event mask bit for channel #9 When more then one event is used during a flow (EOF, EOL, NL, NF, EOFIELD, etc.) masks all the event besides the event that is defined as the highest priority event 1 All the events are used - no mask 0 Only the highest priority event is used, the rest are masked
1–0 W_SIZE_9	Word Size associated with channel #9 The data coming from the IDMAC is 32bit wide. This field defines the size of the word used by the DC 00 8 bits are used - a 32 bit words includes 4X8bit valid words. 01 16 LSB bits - a 32 bit words includes 2 X16bit valid words. 10 24 MSB bits are used (RGB) - 8 LSB bits are ignored by the DC 11 32 bits are used

45.51.274 DC Write Channel 9Configuration 2Register (IPU_DC_WR_CH_CONF2_9)

Address: IPU_DC_WR_CH_CONF2_9 is 1E00_0000h base + 580B8h offset = 1E05_80B8h

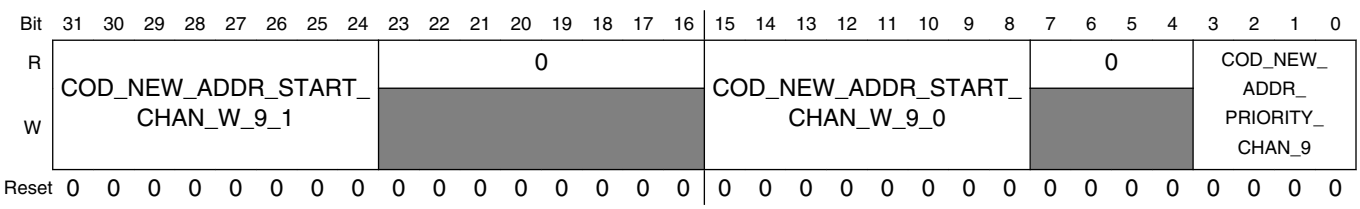


IPU_DC_WR_CH_CONF2_9 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 NEW_ADDR_SPACE_SA_9	Channel #8 is used for ARM platform direct access to the display. This field defines the base address of the second region accessible on the display

45.51.275 DC Routine Link Register 1 Channel 9 (IPU_DC_RL1_CH_9)

Address: IPU_DC_RL1_CH_9 is 1E00_0000h base + 580BCh offset = 1E05_80BCh



IPU_DC_RL1_CH_9 field descriptions

Field	Description
31–24 COD_NEW_ADDR_START_CHAN_W_9_1	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #9, second region)
23–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 COD_NEW_ADDR_START_CHAN_W_9_0	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #9, first region)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.

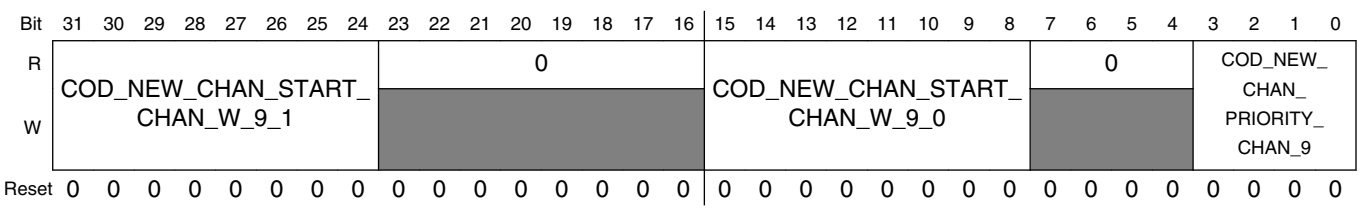
Table continues on the next page...

IPU_DC_RL1_CH_9 field descriptions (continued)

Field	Description
3–0 COD_NEW_ ADDR_ PRIORITY_ CHAN_9	This field defines the priority of the new address event (associated with channel #9, both regions) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.276 DC Routine Link Register 2 Channel 9 (IPU_DC_RL2_CH_9)

Address: IPU_DC_RL2_CH_9 is 1E00_0000h base + 580C0h offset = 1E05_80C0h



IPU_DC_RL2_CH_9 field descriptions

Field	Description
31–24 COD_NEW_ CHAN_START_ CHAN_W_9_1	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #9, second region)
23–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 COD_NEW_ CHAN_START_ CHAN_W_9_0	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #9, first region)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NEW_ CHAN_ PRIORITY_ CHAN_9	This field defines the priority of the new address event (associated with channel #9, both regions) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest)

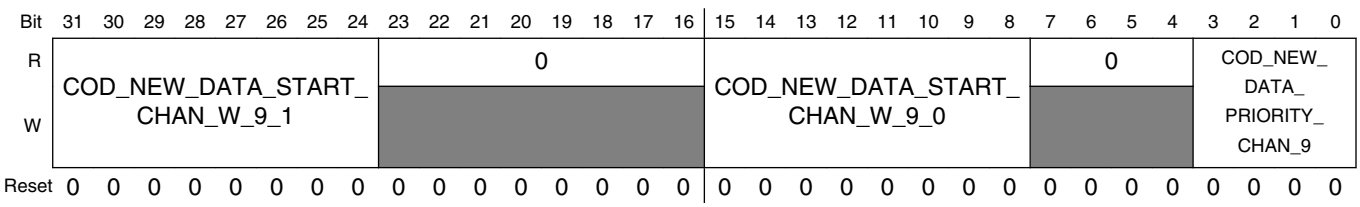
Table continues on the next page...

IPU_DC_RL2_CH_9 field descriptions (continued)

Field	Description
	1110 Reserved 1111 Reserved

45.51.277 DC Routine Link Register 3Channel 9 (IPU_DC_RL3_CH_9)

Address: IPU_DC_RL3_CH_9 is 1E00_0000h base + 580C4h offset = 1E05_80C4h



IPU_DC_RL3_CH_9 field descriptions

Field	Description
31–24 COD_NEW_DATA_START_CHAN_W_9_1	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #9, second region)
23–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 COD_NEW_DATA_START_CHAN_W_9_0	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #9, first region)
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved.
3–0 COD_NEW_DATA_PRIORITY_CHAN_9	This field defines the priority of the new address event (associated with channel #9, both regions) The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved

45.51.278 DC Routine Link Register 4 Channel 9 (IPU_DC_RL4_CH_9)

Address: IPU_DC_RL4_CH_9 is 1E00_0000h base + 580C8h offset = 1E05_80C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_ADDR_START_CHAN_R_9_1								0								COD_NEW_ADDR_START_CHAN_R_9_0								0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL4_CH_9 field descriptions

Field	Description
31–24 COD_NEW_ADDR_START_CHAN_R_9_1	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #9, second region)
23–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 COD_NEW_ADDR_START_CHAN_R_9_0	This field is a pointer to the address within the microcode memory where the routine that handles the new address event resides (associated with channel #9, first region)
7–0 Reserved	This read-only field is reserved and always has the value zero. Reserved.

45.51.279 DC Routine Link Register 5 Channel 9 (IPU_DC_RL5_CH_9)

Address: IPU_DC_RL5_CH_9 is 1E00_0000h base + 580CCh offset = 1E05_80CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_CHAN_START_CHAN_R_9_1								0								COD_NEW_CHAN_START_CHAN_R_9_0								0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL5_CH_9 field descriptions

Field	Description
31–24 COD_NEW_CHAN_START_CHAN_R_9_1	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #9, second region)
23–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_RL5_CH_9 field descriptions (continued)

Field	Description
15–8 COD_NEW_ CHAN_START_ CHAN_R_9_0	This field is a pointer to the address within the microcode memory where the routine that handles the new channel event resides (associated with channel #9, first region)
7–0 Reserved	This read-only field is reserved and always has the value zero. Reserved.

45.51.280 DC Routine Link Register 6 Channel 9 (IPU_DC_RL6_CH_9)

Address: IPU_DC_RL6_CH_9 is 1E00_0000h base + 580D0h offset = 1E05_80D0h

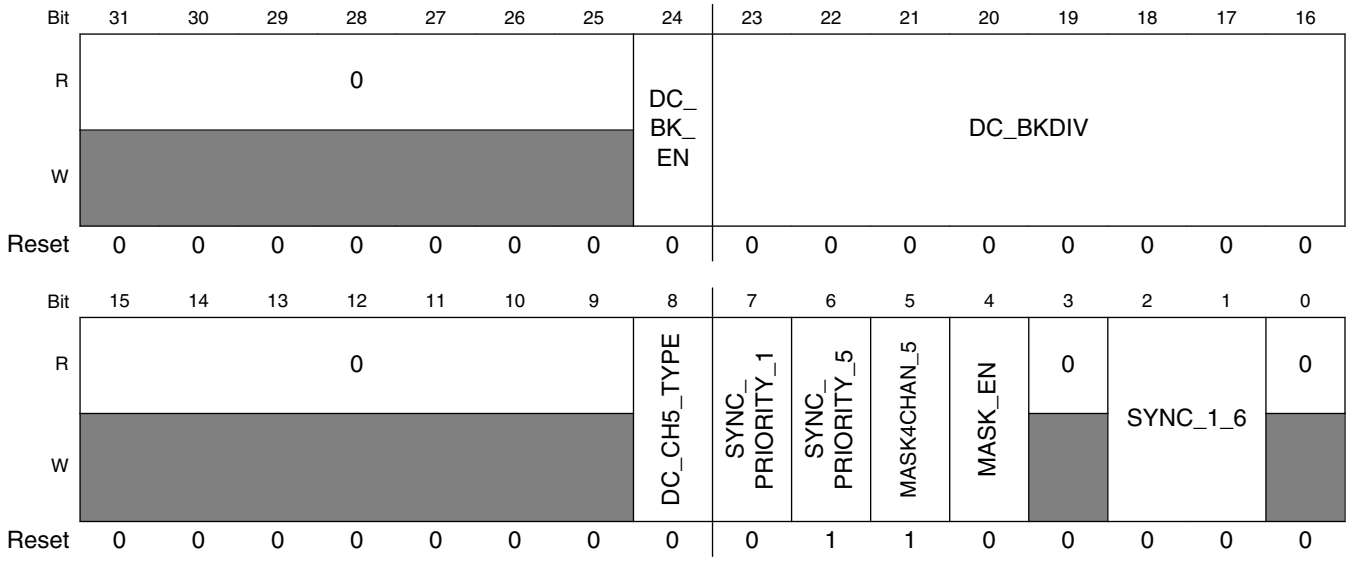
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COD_NEW_DATA_START_CHAN_R_9_1								0								COD_NEW_DATA_START_CHAN_R_9_0								0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_RL6_CH_9 field descriptions

Field	Description
31–24 COD_NEW_ DATA_START_ CHAN_R_9_1	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #9, second region)
23–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 COD_NEW_ DATA_START_ CHAN_R_9_0	This field is a pointer to the address within the microcode memory where the routine that handles the new data event resides (associated with channel #9, first region)
7–0 Reserved	This read-only field is reserved and always has the value zero. Reserved.

45.51.281 DC General Register (IPU_DC_GEN)

Address: IPU_DC_GEN is 1E00_0000h base + 580D4h offset = 1E05_80D4h



IPU_DC_GEN field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24 DC_BK_EN	Cursor blinking enable 1 blinking is enabled 0 blinking is disabled
23–16 DC_BKDIV	Blinking Rate This field defines the blinking rate. The blinking occurs every N-th frame While N is defined by DC_BKDIV
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8 DC_CH5_TYPE	Channel 5 is used for synchronous flow. When this channel is used for accessing asynchronous display that is activated in a synchronous way 1 Enable the asynchronous interface via channel 5 0 normal mode, synchronous flow via channel 5
7 SYNC_PRIORITY_1	When 2 sync flows are running, this bit sets the priority of channel #1. both SYNC_PRIORITY_5 and SYNC_PRIORITY_1 should not have the value of 0 This bit should be 1 high Priority 0 low Priority

Table continues on the next page...

IPU_DC_GEN field descriptions (continued)

Field	Description
6 SYNC_ PRIORITY_5	When 2 sync flows are running, this bit sets the priority of channel #5. both SYNC_PRIORITY_5 and SYNC_PRIORITY_1 should not have the value of 0 This bit should be 1 high priority 0 low Priority
5 MASK4CHAN_5	Sync flow can be associated with a mask channel. Only one sync flow can have a mask. This bit is ignored if MASK_EN is clear 1 mask channel is associated to the sync flow via DP 0 mask channel is associated to the sync flow via DC (without DP)
4 MASK_EN	Enable of the mask channel 1 mask channel is enabled 0 mask channel is disabled
3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–1 SYNC_1_6	This field 00 Channel 1 of the DC handles async flow 01 Illegal 10 Channel 1 of the DC handles sync flow 11 illegal
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

45.51.282 DC Display Configuration 1 Register 0 (IPU_DC_DISP_CONF1_0)

Address: IPU_DC_DISP_CONF1_0 is 1E00_0000h base + 580D8h offset = 1E05_80D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

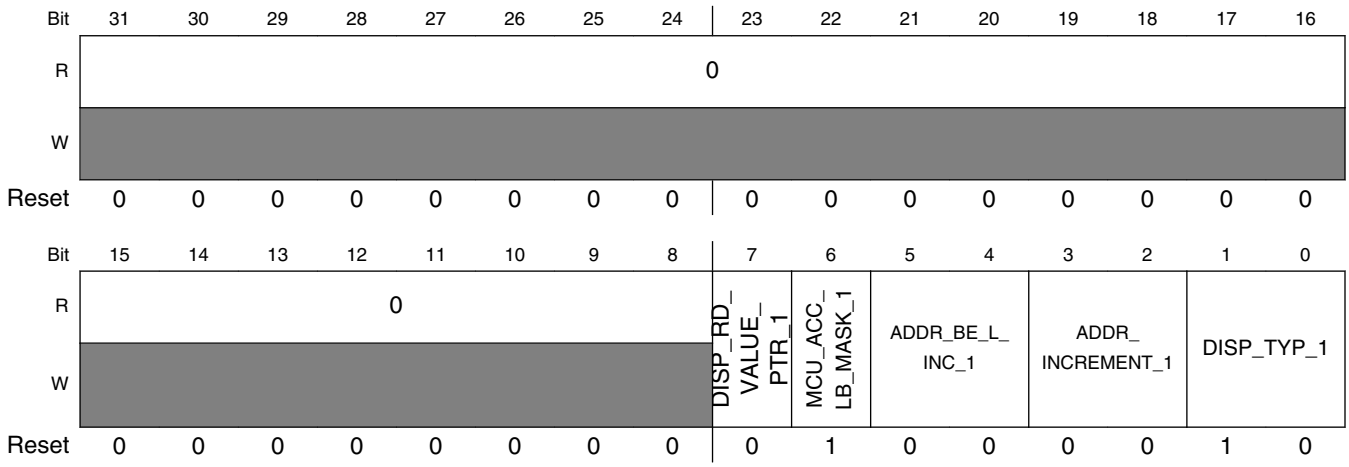
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DISP_RD_VALUE_PTR_0	MCU_ACC_LB_MASK_0	ADDR_BE_L_INC_0		ADDR_INCREMENT_0		DISP_TYP_0	
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0

IPU_DC_DISP_CONF1_0 field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved.
7 DISP_RD_ VALUE_PTR_0	When the display works in wait for status mode. The IPU polls the display and compare the value to the value stored on the status DI_READ_DATA_ACK_VALUE and mask it with DI_READ_DATA_MASK. The DC holds 2 sets of mask and value. This field holds a pointer to the corresponding set of mask and value. 1 DI_READ_DATA_ACK_VALUE_1 & DI_READ_DATA_MASK_1 are used for display 0 0 DI_READ_DATA_ACK_VALUE_0 & DI_READ_DATA_MASK_0 are used for display 0
6 MCU_ACC_LB_ MASK_0	The DC compares between the current access to the a calculated address. The calculated address is the next consecutive address following the last address. This bit defines the comparing mode 1 The 2 addresses are fully compared 0 The 2 addresses are compared, but the ADDR[0] bit of the new address is ignored
5–4 ADDR_BE_L_ INC_0	This bits define the increment mode when the latest access was done with some of the byte enable signals are low, in that case different increment should be done instead of the normal auto increment of the address IF MCU_ACC_LB_MASK_0 is 0 then only 00 and 10 values are allowed. 00 No increment 01 Increment by 1 10 Increment by 2 11 Increment by 3
3–2 ADDR_ INCREMENT_0	This field is the increment step for auto increment mode 00 Increment the address by 1 01 Increment the address by 2 10 Increment the address by 3 11 Increment the address by 4
1–0 DISP_TYP_0	This field defines the type of the display 00 Serial accesses display 01 Reserved 10 parallel display, without byte_enable support 11 parallel display, with byte_enable support

45.51.283 DC Display Configuration 1 Register 1 (IPU_DC_DISP_CONF1_1)

Address: IPU_DC_DISP_CONF1_1 is 1E00_0000h base + 580DCh offset = 1E05_80DCh



IPU_DC_DISP_CONF1_1 field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved.
7 DISP_RD_VALUE_PTR_1	When the display works in wait for status mode. The IPU polls the display and compare the value to the value stored on the status DI_READ_DATA_ACK_VALUE and mask it with DI_READ_DATA_MASK. The DC holds 2 sets of mask and value. This field holds a pointer to the corresponding set of mask and value. 1 DI_READ_DATA_ACK_VALUE_1 & DI_READ_DATA_MASK_1 are used for display 1 0 DI_READ_DATA_ACK_VALUE_0 & DI_READ_DATA_MASK_0 are used for display 1
6 MCU_ACC_LB_MASK_1	The DC compares between the current access to the a calculated address. The calculated address is the next consecutive address following the last address. This bit defines the comparing mode 1 The 2 addresses are fully compared 0 The 2 addresses are compared, but the ADDR[0] bit of the new address is ignored
5–4 ADDR_BE_L_INC_1	This bits define the increment mode when the latest access was done with some of the byte enable signals are low, in that case different increment should be done instead of the normal auto increment of the address IF MCU_ACC_LB_MASK_1 is 0 then only 00 and 10 values are allowed. 00 No increment 01 Increment by 1 10 Increment by 2 11 Increment by 3
3–2 ADDR_INCREMENT_1	This field is the increment step for auto increment mode 00 Increment the address by 1 01 Increment the address by 2

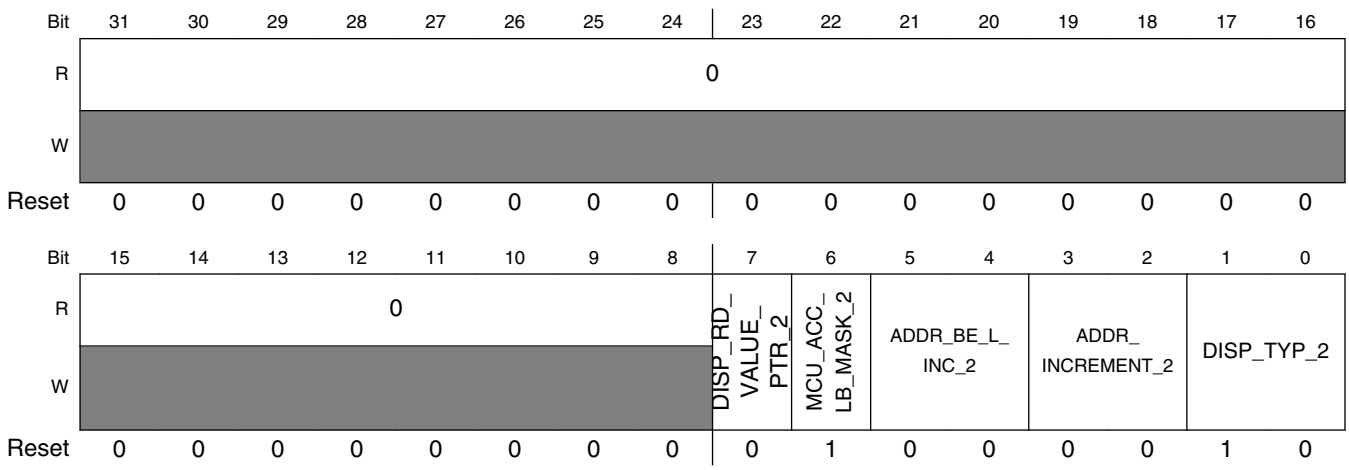
Table continues on the next page...

IPU_DC_DISP_CONF1_1 field descriptions (continued)

Field	Description
	10 Increment the address by 3 11 Increment the address by 4
1–0 DISP_TYP_1	This field defines the type of the display 00 Serial accesses display 01 Reserved 10 parallel display, without byte_enable support 11 parallel display, with byte_enable support

45.51.284 DC Display Configuration 1 Register 2 (IPU_DC_DISP_CONF1_2)

Address: IPU_DC_DISP_CONF1_2 is 1E00_0000h base + 580E0h offset = 1E05_80E0h



IPU_DC_DISP_CONF1_2 field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved.
7 DISP_RD_ VALUE_PTR_2	When the display works in wait for status mode. The IPU polls the display and compare the value to the value stored on the status DI_READ_DATA_ACK_VALUE and mask it with DI_READ_DATA_MASK. The DC holds 2 sets of mask and value. This field holds a pointer to the corresponding set of mask and value. 1 DI_READ_DATA_ACK_VALUE_1 & DI_READ_DATA_MASK_1 are used for display 2 0 DI_READ_DATA_ACK_VALUE_0 & DI_READ_DATA_MASK_0 are used for display 2
6 MCU_ACC_LB_ MASK_2	The DC compares between the current access to the a calculated address. The calculated address is the next consecutive address following the last address. This bit defines the comparing mode 1 The 2 addresses are fully compared 0 The 2 addresses are compared, but the ADDR[0] bit of the new address is ignored

Table continues on the next page...

IPU_DC_DISP_CONF1_2 field descriptions (continued)

Field	Description
5–4 ADDR_BE_L_ INC_2	This bits define the increment mode when the latest access was done with some of the byte enable signals are low, in that case different increment should be done instead of the normal auto increment of the address IF MCU_ACC_LB_MASK_2 is 0 then only 00 and 10 values are allowed. 00 No increment 01 Increment by 1 10 Increment by 2 11 Increment by 3
3–2 ADDR_ INCREMENT_2	This field is the increment step for auto increment mode 00 Increment the address by 1 01 Increment the address by 2 10 Increment the address by 3 11 Increment the address by 4
1–0 DISP_TYP_2	This field defines the type of the display 00 Serial accesses display 01 Reserved 10 parallel display, without byte_enable support 11 parallel display, with byte_enable support

45.51.285 DC Display Configuration 1 Register 3 (IPU_DC_DISP_CONF1_3)

Address: IPU_DC_DISP_CONF1_3 is 1E00_0000h base + 580E4h offset = 1E05_80E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DISP_RD_ VALUE_ PTR_3	MCU_ACC_ LB_MASK_3	ADDR_BE_L_ INC_3		ADDR_ INCREMENT_3		DISP_TYP_3	
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0

IPU_DC_DISP_CONF1_3 field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved.
7 DISP_RD_ VALUE_PTR_3	When the display works in wait for status mode. The IPU polls the display and compare the value to the value stored on the status DI_READ_DATA_ACK_VALUE and mask it with DI_READ_DATA_MASK. The DC holds 2 sets of mask and value. This field holds a pointer to the corresponding set of mask and value. 1 DI_READ_DATA_ACK_VALUE_1 & DI_READ_DATA_MASK_1 are used for display 3 0 DI_READ_DATA_ACK_VALUE_0 & DI_READ_DATA_MASK_0 are used for display 3
6 MCU_ACC_LB_ MASK_3	The DC compares between the current access to the a calculated address. The calculated address is the next consecutive address following the last address. This bit defines the comparing mode 1 The 2 addresses are fully compared 0 The 2 addresses are compared, but the ADDR[0] bit of the new address is ignored
5–4 ADDR_BE_L_ INC_3	This bits define the increment mode when the latest access was done with some of the byte enable signals are low, in that case different increment should be done instead of the normal auto increment of the address IF MCU_ACC_LB_MASK_3 is 0 then only 00 and 10 values are allowed. 00 No increment 01 Increment by 1 10 Increment by 2 11 Increment by 3
3–2 ADDR_ INCREMENT_3	This field is the increment step for auto increment mode 00 Increment the address by 1 01 Increment the address by 2 10 Increment the address by 3 11 Increment the address by 4
1–0 DISP_TYP_3	This field defines the type of the display 00 Serial accesses display 01 Reserved 10 parallel display, without byte_enable support 11 parallel display, with byte_enable support

45.51.286 DC Display Configuration 2 Register 0 (IPU_DC_DISP_CONF2_0)

Address: IPU_DC_DISP_CONF2_0 is 1E00_0000h base + 580E8h offset = 1E05_80E8h

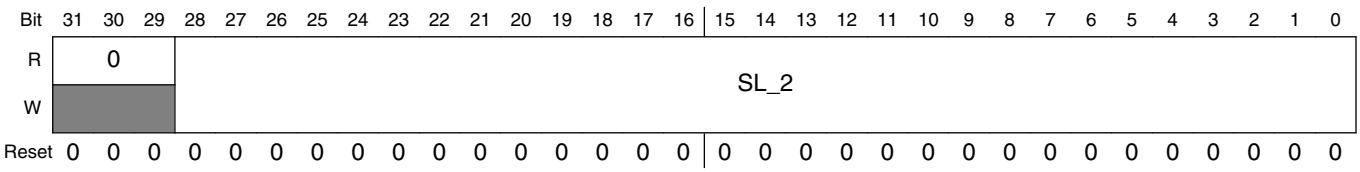
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																SL_0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_DISP_CONF2_0 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 SL_0	Stride line of display 0

45.51.287 DC Display Configuration 2 Register 2 (IPU_DC_DISP_CONF2_2)

Address: IPU_DC_DISP_CONF2_2 is 1E00_0000h base + 580F0h offset = 1E05_80F0h

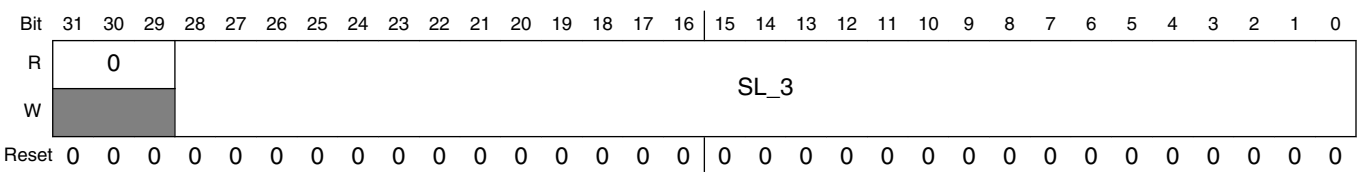


IPU_DC_DISP_CONF2_2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 SL_2	Stride line of display 2

45.51.288 DC Display Configuration 2 Register 3 (IPU_DC_DISP_CONF2_3)

Address: IPU_DC_DISP_CONF2_3 is 1E00_0000h base + 580F4h offset = 1E05_80F4h



IPU_DC_DISP_CONF2_3 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 SL_3	Stride line of display 3

45.51.289 DC DI0Configuration Register 1 (IPU_DC_DI0_CONF_1)

Address: IPU_DC_DI0_CONF_1 is 1E00_0000h base + 580F8h offset = 1E05_80F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DI_READ_DATA_MASK_0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_DI0_CONF_1 field descriptions

Field	Description
31–0 DI_READ_ DATA_MASK_0	This field defines the mask value of the data read from the display.

45.51.290 DC DI0Configuration Register 2 (IPU_DC_DI0_CONF_2)

Address: IPU_DC_DI0_CONF_2 is 1E00_0000h base + 580FCh offset = 1E05_80FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DI_READ_DATA_ACK_VALUE_0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_DI0_CONF_2 field descriptions

Field	Description
31–0 DI_READ_ DATA_ACK_ VALUE_0	This is the expected data to be read from the display. The value reads from the display is anded with the DI_READ_DATA_MASK_0 and compared with the DI_READ_DATA_ACK_VALUE_0. This field is used for the READ_STATUS task of the DC

45.51.291 DC DI1Configuration Register 1 (IPU_DC_DI1_CONF_1)

Address: IPU_DC_DI1_CONF_1 is 1E00_0000h base + 58100h offset = 1E05_8100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DI_READ_DATA_MASK_1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_DI1_CONF_1 field descriptions

Field	Description
31–0 DI_READ_ DATA_MASK_1	This field defines the mask value of the data read from the display.

45.51.292 DC DI1Configuration Register 2 (IPU_DC_DI1_CONF_2)

Address: IPU_DC_DI1_CONF_2 is 1E00_0000h base + 58104h offset = 1E05_8104h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DI_READ_DATA_ACK_VALUE_1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_DI1_CONF_2 field descriptions

Field	Description
31–0 DI_READ_ DATA_ACK_ VALUE_1	This is the expected data to be read from the display. The value reads from the display is anded with the DI_READ_DATA_MASK_1 and compared with the DI_READ_DATA_ACK_VALUE_1 This field is used for the READ_STATUS task of the DC

45.51.293 DC Mapping Configuration Register 0 (IPU_DC_MAP_CONF_0)

Address: IPU_DC_MAP_CONF_0 is 1E00_0000h base + 58108h offset = 1E05_8108h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_ PNTR_BYTE2_1					MAPPING_ PNTR_BYTE1_1					MAPPING_ PNTR_BYTE0_1					0	MAPPING_ PNTR_BYTE2_0					MAPPING_ PNTR_BYTE1_0					MAPPING_ PNTR_BYTE0_0				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_0 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–26 MAPPING_ PNTR_BYTE2_1	Mapping pointer #1 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_1	Mapping pointer #1 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1

Table continues on the next page...

IPU_DC_MAP_CONF_0 field descriptions (continued)

Field	Description
20–16 MAPPING_ PNTR_BYTE0_1	Mapping pointer #1 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–10 MAPPING_ PNTR_BYTE2_0	Mapping pointer #0 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_0	Mapping pointer #0 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
4–0 MAPPING_ PNTR_BYTE0_0	Mapping pointer #0 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

45.51.294 DC Mapping Configuration Register 1 (IPU_DC_MAP_CONF_1)

Address: IPU_DC_MAP_CONF_1 is 1E00_0000h base + 5810Ch offset = 1E05_810Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_ PNTR_BYTE2_3					MAPPING_ PNTR_BYTE1_3					MAPPING_ PNTR_BYTE0_3					0	MAPPING_ PNTR_BYTE2_2					MAPPING_ PNTR_BYTE1_2					MAPPING_ PNTR_BYTE0_2				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–26 MAPPING_ PNTR_BYTE2_3	Mapping pointer #3 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_3	Mapping pointer #3 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_3	Mapping pointer #3 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_MAP_CONF_1 field descriptions (continued)

Field	Description
14–10 MAPPING_ PNTR_BYTE2_2	Mapping pointer #1 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_2	Mapping pointer #1 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
4–0 MAPPING_ PNTR_BYTE0_2	Mapping pointer #1 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

45.51.295 DC Mapping Configuration Register 2 (IPU_DC_MAP_CONF_2)

Address: IPU_DC_MAP_CONF_2 is 1E00_0000h base + 58110h offset = 1E05_8110h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_ PNTR_BYTE2_5					MAPPING_ PNTR_BYTE1_5					MAPPING_ PNTR_BYTE0_5					0	MAPPING_ PNTR_BYTE2_4					MAPPING_ PNTR_BYTE1_4					MAPPING_ PNTR_BYTE0_4				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–26 MAPPING_ PNTR_BYTE2_5	Mapping pointer #5 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_5	Mapping pointer #5 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_5	Mapping pointer #5 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–10 MAPPING_ PNTR_BYTE2_4	Mapping pointer #4 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_4	Mapping pointer #4 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1

Table continues on the next page...

IPU_DC_MAP_CONF_2 field descriptions (continued)

Field	Description
4–0 MAPPING_ PNTR_BYTE0_4	Mapping pointer #4 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

45.51.296 DC Mapping Configuration Register 3 (IPU_DC_MAP_CONF_3)

Address: IPU_DC_MAP_CONF_3 is 1E00_0000h base + 58114h offset = 1E05_8114h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_ PNTR_BYTE2_7					MAPPING_ PNTR_BYTE1_7					MAPPING_ PNTR_BYTE0_7					0	MAPPING_ PNTR_BYTE2_6					MAPPING_ PNTR_BYTE1_6					MAPPING_ PNTR_BYTE0_6				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_3 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–26 MAPPING_ PNTR_BYTE2_7	Mapping pointer #7 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_7	Mapping pointer #7 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_7	Mapping pointer #7 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–10 MAPPING_ PNTR_BYTE2_6	Mapping pointer #6 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_6	Mapping pointer #6 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
4–0 MAPPING_ PNTR_BYTE0_6	Mapping pointer #6 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

45.51.297 DC Mapping Configuration Register 4 (IPU_DC_MAP_CONF_4)

Address: IPU_DC_MAP_CONF_4 is 1E00_0000h base + 58118h offset = 1E05_8118h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_ PNTR_BYTE2_9					MAPPING_ PNTR_BYTE1_1					MAPPING_ PNTR_BYTE0_9					0	MAPPING_ PNTR_BYTE2_8					MAPPING_ PNTR_BYTE1_8					MAPPING_ PNTR_BYTE0_8				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_4 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–26 MAPPING_ PNTR_BYTE2_9	Mapping pointer #9 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_1	Mapping pointer #9 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_9	Mapping pointer #9 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–10 MAPPING_ PNTR_BYTE2_8	Mapping pointer #8 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_8	Mapping pointer #8 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
4–0 MAPPING_ PNTR_BYTE0_8	Mapping pointer #8 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

45.51.298 DC Mapping Configuration Register 5 (IPU_DC_MAP_CONF_5)

Address: IPU_DC_MAP_CONF_5 is 1E00_0000h base + 5811Ch offset = 1E05_811Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_ PNTR_BYTE2_ 11					MAPPING_ PNTR_BYTE1_ 11					MAPPING_ PNTR_BYTE0_ 11					0	MAPPING_ PNTR_BYTE2_ 10					MAPPING_ PNTR_BYTE1_ 10					MAPPING_ PNTR_BYTE0_ 10				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_5 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–26 MAPPING_ PNTR_BYTE2_ 11	Mapping pointer #11 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 11	Mapping pointer #11 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 11	Mapping pointer #11 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–10 MAPPING_ PNTR_BYTE2_ 10	Mapping pointer #10 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 10	Mapping pointer #10 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
4–0 MAPPING_ PNTR_BYTE0_ 10	Mapping pointer #10 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

45.51.299 DC Mapping Configuration Register 6 (IPU_DC_MAP_CONF_6)

Address: IPU_DC_MAP_CONF_6 is 1E00_0000h base + 58120h offset = 1E05_8120h

Bit	31	30–26	25–21	20–16	15	14–10	9–5	4–0
R	0	MAPPING_PNTR_BYTE2_13	MAPPING_PNTR_BYTE1_13	MAPPING_PNTR_BYTE0_13	0	MAPPING_PNTR_BYTE2_12	MAPPING_PNTR_BYTE1_12	MAPPING_PNTR_BYTE0_12
W								
Reset	0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0

IPU_DC_MAP_CONF_6 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_MAP_CONF_6 field descriptions (continued)

Field	Description
30–26 MAPPING_ PNTR_BYTE2_ 13	Mapping pointer #13 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 13	Mapping pointer #13 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 13	Mapping pointer #13 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–10 MAPPING_ PNTR_BYTE2_ 12	Mapping pointer #12 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 12	Mapping pointer #12 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
4–0 MAPPING_ PNTR_BYTE0_ 12	Mapping pointer #12 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

45.51.300 DC Mapping Configuration Register 7 (IPU_DC_MAP_CONF_7)

Address: IPU_DC_MAP_CONF_7 is 1E00_0000h base + 58124h offset = 1E05_8124h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_ PNTR_BYTE2_ 15						MAPPING_ PNTR_BYTE1_ 15					MAPPING_ PNTR_BYTE0_ 15				0	MAPPING_ PNTR_BYTE2_ 14						MAPPING_ PNTR_BYTE1_ 14					MAPPING_ PNTR_BYTE0_ 14			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_7 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_MAP_CONF_7 field descriptions (continued)

Field	Description
30–26 MAPPING_ PNTR_BYTE2_ 15	Mapping pointer #15 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 15	Mapping pointer #15 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 15	Mapping pointer #15 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–10 MAPPING_ PNTR_BYTE2_ 14	Mapping pointer #14 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 14	Mapping pointer #14 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
4–0 MAPPING_ PNTR_BYTE0_ 14	Mapping pointer #14 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

45.51.301 DC Mapping Configuration Register 8 (IPU_DC_MAP_CONF_8)

Address: IPU_DC_MAP_CONF_8 is 1E00_0000h base + 58128h offset = 1E05_8128h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_ PNTR_BYTE2_ 17						MAPPING_ PNTR_BYTE1_ 17					MAPPING_ PNTR_BYTE0_ 17				0	MAPPING_ PNTR_BYTE2_ 16						MAPPING_ PNTR_BYTE1_ 16					MAPPING_ PNTR_BYTE0_ 16			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_8 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_MAP_CONF_8 field descriptions (continued)

Field	Description
30–26 MAPPING_ PNTR_BYTE2_ 17	Mapping pointer #17 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 17	Mapping pointer #17 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 17	Mapping pointer #17 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–10 MAPPING_ PNTR_BYTE2_ 16	Mapping pointer #16 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 16	Mapping pointer #16 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
4–0 MAPPING_ PNTR_BYTE0_ 16	Mapping pointer #16 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

45.51.302 DC Mapping Configuration Register 9 (IPU_DC_MAP_CONF_9)

Address: IPU_DC_MAP_CONF_9 is 1E00_0000h base + 5812Ch offset = 1E05_812Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_ PNTR_BYTE2_ 19						MAPPING_ PNTR_BYTE1_ 19					MAPPING_ PNTR_BYTE0_ 19				0	MAPPING_ PNTR_BYTE2_ 18						MAPPING_ PNTR_BYTE1_ 18					MAPPING_ PNTR_BYTE0_ 18			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_9 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_MAP_CONF_9 field descriptions (continued)

Field	Description
30–26 MAPPING_ PNTR_BYTE2_ 19	Mapping pointer #19 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 19	Mapping pointer #19 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 19	Mapping pointer #19 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–10 MAPPING_ PNTR_BYTE2_ 18	Mapping pointer #18 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 18	Mapping pointer #18 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
4–0 MAPPING_ PNTR_BYTE0_ 18	Mapping pointer #18 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

45.51.303 DC Mapping Configuration Register 10 (IPU_DC_MAP_CONF_10)

Address: IPU_DC_MAP_CONF_10 is 1E00_0000h base + 58130h offset = 1E05_8130h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_ PNTR_BYTE2_ 21					MAPPING_ PNTR_BYTE1_ 21					MAPPING_ PNTR_BYTE0_ 21					0	MAPPING_ PNTR_BYTE2_ 20					MAPPING_ PNTR_BYTE1_ 20					MAPPING_ PNTR_BYTE0_ 20				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_10 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_MAP_CONF_10 field descriptions (continued)

Field	Description
30–26 MAPPING_ PNTR_BYTE2_ 21	Mapping pointer #21 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 21	Mapping pointer #21 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 21	Mapping pointer #21 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–10 MAPPING_ PNTR_BYTE2_ 20	Mapping pointer #20 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 20	Mapping pointer #20 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
4–0 MAPPING_ PNTR_BYTE0_ 20	Mapping pointer #20 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

45.51.304 DC Mapping Configuration Register 11 (IPU_DC_MAP_CONF_11)

Address: IPU_DC_MAP_CONF_11 is 1E00_0000h base + 58134h offset = 1E05_8134h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_ PNTR_BYTE2_ 23							MAPPING_ PNTR_BYTE1_ 23								MAPPING_ PNTR_BYTE0_ 23						0	MAPPING_ PNTR_BYTE2_ 22									MAPPING_ PNTR_BYTE1_ 22									MAPPING_ PNTR_BYTE0_ 22					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_11 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_MAP_CONF_11 field descriptions (continued)

Field	Description
30–26 MAPPING_ PNTR_BYTE2_ 23	Mapping pointer #23 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 23	Mapping pointer #23 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 23	Mapping pointer #23 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–10 MAPPING_ PNTR_BYTE2_ 22	Mapping pointer #22 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 22	Mapping pointer #22 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
4–0 MAPPING_ PNTR_BYTE0_ 22	Mapping pointer #22 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

45.51.305 DC Mapping Configuration Register 12 (IPU_DC_MAP_CONF_12)

Address: IPU_DC_MAP_CONF_12 is 1E00_0000h base + 58138h offset = 1E05_8138h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_ PNTR_BYTE2_ 25						MAPPING_ PNTR_BYTE1_ 25						MAPPING_ PNTR_BYTE0_ 25			0	MAPPING_ PNTR_BYTE2_ 24						MAPPING_ PNTR_BYTE1_ 24						MAPPING_ PNTR_BYTE0_ 24			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_12 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_MAP_CONF_12 field descriptions (continued)

Field	Description
30–26 MAPPING_ PNTR_BYTE2_ 25	Mapping pointer #25 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 25	Mapping pointer #25 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 25	Mapping pointer #25 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–10 MAPPING_ PNTR_BYTE2_ 24	Mapping pointer #24 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 24	Mapping pointer #24 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
4–0 MAPPING_ PNTR_BYTE0_ 24	Mapping pointer #24 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

45.51.306 DC Mapping Configuration Register 13 (IPU_DC_MAP_CONF_13)

Address: IPU_DC_MAP_CONF_13 is 1E00_0000h base + 5813Ch offset = 1E05_813Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_ PNTR_BYTE2_ 27					MAPPING_ PNTR_BYTE1_ 27					MAPPING_ PNTR_BYTE0_ 27					0	MAPPING_ PNTR_BYTE2_ 26					MAPPING_ PNTR_BYTE1_ 26					MAPPING_ PNTR_BYTE0_ 26				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_13 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_MAP_CONF_13 field descriptions (continued)

Field	Description
30–26 MAPPING_ PNTR_BYTE2_ 27	Mapping pointer #27 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 27	Mapping pointer #27 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 27	Mapping pointer #27 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–10 MAPPING_ PNTR_BYTE2_ 26	Mapping pointer #26 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 26	Mapping pointer #26 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
4–0 MAPPING_ PNTR_BYTE0_ 26	Mapping pointer #26 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

45.51.307 DC Mapping Configuration Register 14 (IPU_DC_MAP_CONF_14)

Address: IPU_DC_MAP_CONF_14 is 1E00_0000h base + 58140h offset = 1E05_8140h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MAPPING_ PNTR_BYTE2_ 29					MAPPING_ PNTR_BYTE1_ 29					MAPPING_ PNTR_BYTE0_ 29					0	MAPPING_ PNTR_BYTE2_ 28f					MAPPING_ PNTR_BYTE1_ 28					MAPPING_ PNTR_BYTE2_ 28				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_14 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.

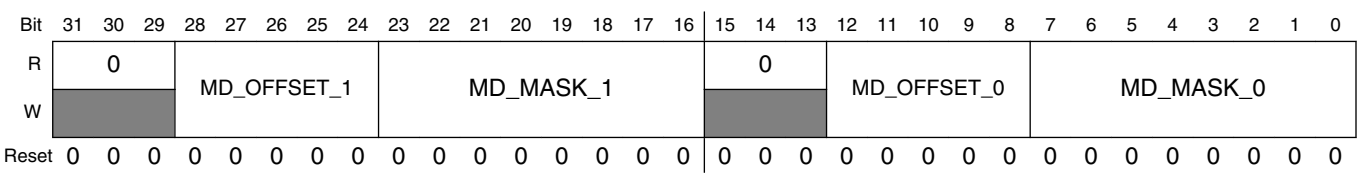
Table continues on the next page...

IPU_DC_MAP_CONF_14 field descriptions (continued)

Field	Description
30–26 MAPPING_ PNTR_BYTE2_ 29	Mapping pointer #29 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
25–21 MAPPING_ PNTR_BYTE1_ 29	Mapping pointer #29 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
20–16 MAPPING_ PNTR_BYTE0_ 29	Mapping pointer #29 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–10 MAPPING_ PNTR_BYTE2_ 28f	Mapping pointer #28 for Byte 2 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #2
9–5 MAPPING_ PNTR_BYTE1_ 28	Mapping pointer #28 for Byte 1 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #1
4–0 MAPPING_ PNTR_BYTE2_ 28	Mapping pointer #28 for Byte 0 This field is a pointer to the set of MD_OFFSET_<i> and MD_MASK_<i> used for mapping byte #0

45.51.308 DC Mapping Configuration Register 15 (IPU_DC_MAP_CONF_15)

Address: IPU_DC_MAP_CONF_15 is 1E00_0000h base + 58144h offset = 1E05_8144h



IPU_DC_MAP_CONF_15 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–24 MD_OFFSET_1	Mapping unit's offset parameter #1 This field defines the offset parameter #1 within the 24bit word coming from the DC.

Table continues on the next page...

IPU_DC_MAP_CONF_15 field descriptions (continued)

Field	Description
23–16 MD_MASK_1	Mapping unit's mask value #1 This field defines the mask value #1 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved.
12–8 MD_OFFSET_0	Mapping unit's offset parameter #0 This field defines the offset parameter #0 within the 24bit word coming from the DC.
7–0 MD_MASK_0	Mapping unit's mask value #0 This field defines the mask value #0 within the 8bit word coming from the DC.

45.51.309 DC Mapping Configuration Register 16 (IPU_DC_MAP_CONF_16)

Address: IPU_DC_MAP_CONF_16 is 1E00_0000h base + 58148h offset = 1E05_8148h

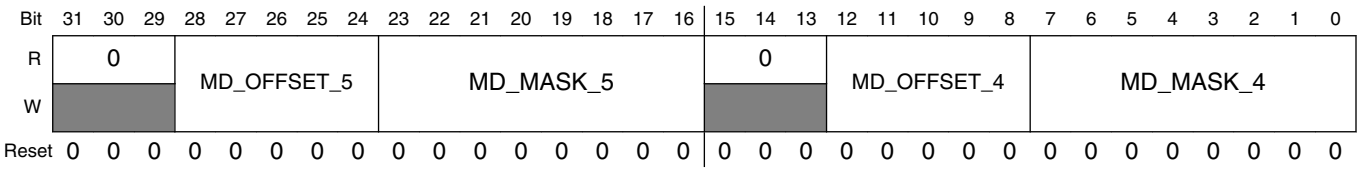
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			MD_OFFSET_3					MD_MASK_3								0			MD_OFFSET_2					MD_MASK_0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_16 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–24 MD_OFFSET_3	Mapping unit's offset parameter #3 This field defines the offset parameter #3 within the 24bit word coming from the DC.
23–16 MD_MASK_3	Mapping unit's mask value #3 This field defines the mask value #3 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved.
12–8 MD_OFFSET_2	Mapping unit's offset parameter #2 This field defines the offset parameter #2 within the 24bit word coming from the DC.
7–0 MD_MASK_0	Mapping unit's mask value #2 This field defines the mask value #2 within the 8bit word coming from the DC.

45.51.310 DC Mapping Configuration Register 17 (IPU_DC_MAP_CONF_17)

Address: IPU_DC_MAP_CONF_17 is 1E00_0000h base + 5814Ch offset = 1E05_814Ch

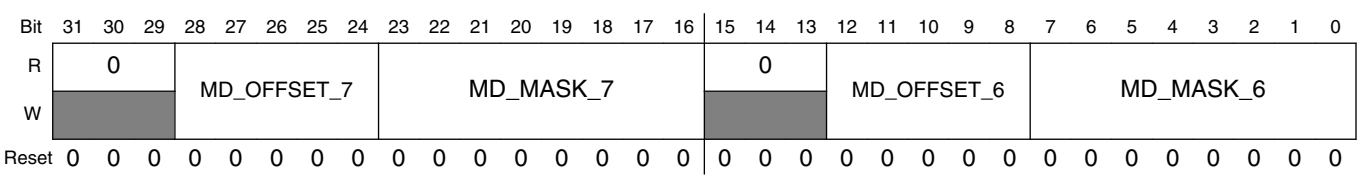


IPU_DC_MAP_CONF_17 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–24 MD_OFFSET_5	Mapping unit's offset parameter #5 This field defines the offset parameter #5 within the 24bit word coming from the DC.
23–16 MD_MASK_5	Mapping unit's mask value #5 This field defines the mask value #5 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved.
12–8 MD_OFFSET_4	Mapping unit's offset parameter #4 This field defines the offset parameter #4 within the 24bit word coming from the DC.
7–0 MD_MASK_4	Mapping unit's mask value #4 This field defines the mask value #4 within the 8bit word coming from the DC.

45.51.311 DC Mapping Configuration Register 18 (IPU_DC_MAP_CONF_18)

Address: IPU_DC_MAP_CONF_18 is 1E00_0000h base + 58150h offset = 1E05_8150h



IPU_DC_MAP_CONF_18 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_MAP_CONF_18 field descriptions (continued)

Field	Description
28–24 MD_OFFSET_7	Mapping unit's offset parameter #7 This field defines the offset parameter #7 within the 24bit word coming from the DC.
23–16 MD_MASK_7	Mapping unit's mask value #7 This field defines the mask value #7 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved.
12–8 MD_OFFSET_6	Mapping unit's offset parameter #6 This field defines the offset parameter #6 within the 24bit word coming from the DC.
7–0 MD_MASK_6	Mapping unit's mask value #6 This field defines the mask value #6 within the 8bit word coming from the DC.

45.51.312 DC Mapping Configuration Register 19 (IPU_DC_MAP_CONF_19)

Address: IPU_DC_MAP_CONF_19 is 1E00_0000h base + 58154h offset = 1E05_8154h

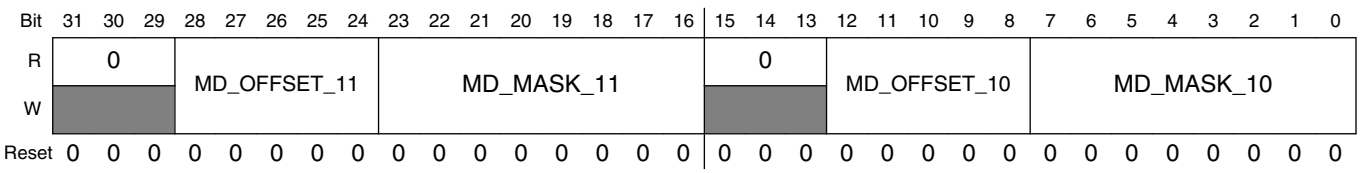
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			MD_OFFSET_9					MD_MASK_9								0			MD_OFFSET_8					MD_MASK_8							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_19 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–24 MD_OFFSET_9	Mapping unit's offset parameter #9 This field defines the offset parameter #9 within the 24bit word coming from the DC.
23–16 MD_MASK_9	Mapping unit's mask value #9 This field defines the mask value #9 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved.
12–8 MD_OFFSET_8	Mapping unit's offset parameter #8 This field defines the offset parameter #8 within the 24bit word coming from the DC.
7–0 MD_MASK_8	Mapping unit's mask value #8 This field defines the mask value #8 within the 8bit word coming from the DC.

45.51.313 DC Mapping Configuration Register 20 (IPU_DC_MAP_CONF_20)

Address: IPU_DC_MAP_CONF_20 is 1E00_0000h base + 58158h offset = 1E05_8158h

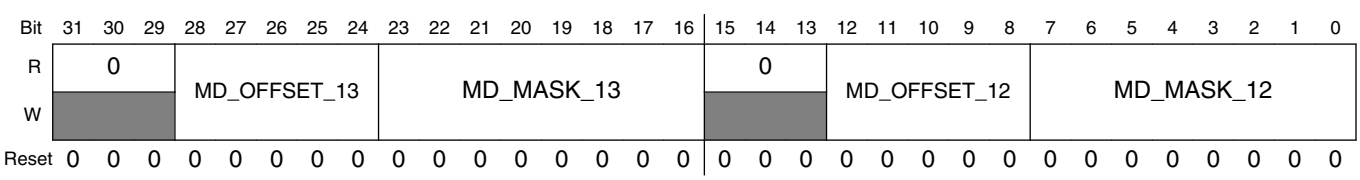


IPU_DC_MAP_CONF_20 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–24 MD_OFFSET_11	Mapping unit's offset parameter #11 This field defines the offset parameter #11 within the 24bit word coming from the DC.
23–16 MD_MASK_11	Mapping unit's mask value #11 This field defines the mask value #11 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved.
12–8 MD_OFFSET_10	Mapping unit's offset parameter #10 This field defines the offset parameter #10 within the 24bit word coming from the DC.
7–0 MD_MASK_10	Mapping unit's mask value #10 This field defines the mask value #10 within the 8bit word coming from the DC.

45.51.314 DC Mapping Configuration Register 21 (IPU_DC_MAP_CONF_21)

Address: IPU_DC_MAP_CONF_21 is 1E00_0000h base + 5815Ch offset = 1E05_815Ch



IPU_DC_MAP_CONF_21 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.

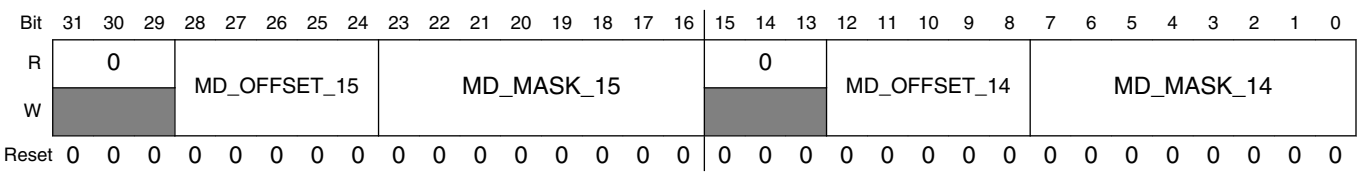
Table continues on the next page...

IPU_DC_MAP_CONF_21 field descriptions (continued)

Field	Description
28–24 MD_OFFSET_13	Mapping unit's offset parameter #13 This field defines the offset parameter #13 within the 24bit word coming from the DC.
23–16 MD_MASK_13	Mapping unit's mask value #13 This field defines the mask value #13 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved.
12–8 MD_OFFSET_12	Mapping unit's offset parameter #12 This field defines the offset parameter #12 within the 24bit word coming from the DC.
7–0 MD_MASK_12	Mapping unit's mask value #12 This field defines the mask value #12 within the 8bit word coming from the DC.

45.51.315 DC Mapping Configuration Register 22 (IPU_DC_MAP_CONF_22)

Address: IPU_DC_MAP_CONF_22 is 1E00_0000h base + 58160h offset = 1E05_8160h

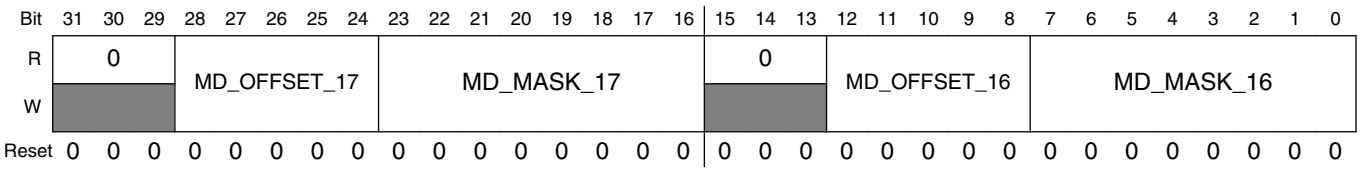


IPU_DC_MAP_CONF_22 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–24 MD_OFFSET_15	Mapping unit's offset parameter #15 This field defines the offset parameter #15 within the 24bit word coming from the DC.
23–16 MD_MASK_15	Mapping unit's mask value #15 This field defines the mask value #15 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved.
12–8 MD_OFFSET_14	Mapping unit's offset parameter #14 This field defines the offset parameter #14 within the 24bit word coming from the DC.
7–0 MD_MASK_14	Mapping unit's mask value #14 This field defines the mask value #14 within the 8bit word coming from the DC.

45.51.316 DC Mapping Configuration Register 23 (IPU_DC_MAP_CONF_23)

Address: IPU_DC_MAP_CONF_23 is 1E00_0000h base + 58164h offset = 1E05_8164h

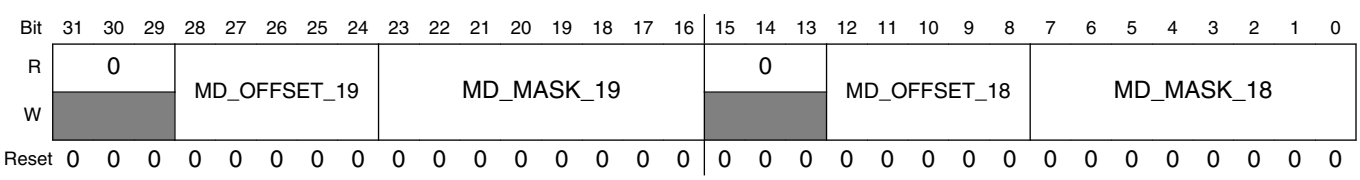


IPU_DC_MAP_CONF_23 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–24 MD_OFFSET_17	Mapping unit's offset parameter #17 This field defines the offset parameter #17 within the 24bit word coming from the DC.
23–16 MD_MASK_17	Mapping unit's mask value #17 This field defines the mask value #17 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved.
12–8 MD_OFFSET_16	Mapping unit's offset parameter #16 This field defines the offset parameter #16 within the 24bit word coming from the DC.
7–0 MD_MASK_16	Mapping unit's mask value #16 This field defines the mask value #16 within the 8bit word coming from the DC.

45.51.317 DC Mapping Configuration Register 24 (IPU_DC_MAP_CONF_24)

Address: IPU_DC_MAP_CONF_24 is 1E00_0000h base + 58168h offset = 1E05_8168h



IPU_DC_MAP_CONF_24 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_MAP_CONF_24 field descriptions (continued)

Field	Description
28–24 MD_OFFSET_19	Mapping unit's offset parameter #19 This field defines the offset parameter #19 within the 24bit word coming from the DC.
23–16 MD_MASK_19	Mapping unit's mask value #19 This field defines the mask value #19 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved.
12–8 MD_OFFSET_18	Mapping unit's offset parameter #18 This field defines the offset parameter #18 within the 24bit word coming from the DC.
7–0 MD_MASK_18	Mapping unit's mask value #18 This field defines the mask value #18 within the 8bit word coming from the DC.

45.51.318 DC Mapping Configuration Register 25 (IPU_DC_MAP_CONF_25)

Address: IPU_DC_MAP_CONF_25 is 1E00_0000h base + 5816Ch offset = 1E05_816Ch

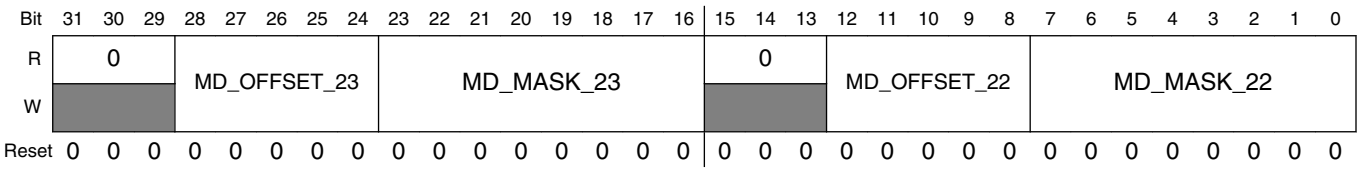
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			MD_OFFSET_21					MD_MASK_21								0			MD_OFFSET_20					MD_MASK_20							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_MAP_CONF_25 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–24 MD_OFFSET_21	Mapping unit's offset parameter #21 This field defines the offset parameter #21 within the 24bit word coming from the DC.
23–16 MD_MASK_21	Mapping unit's mask value #21 This field defines the mask value #21 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved.
12–8 MD_OFFSET_20	Mapping unit's offset parameter #20 This field defines the offset parameter #20 within the 24bit word coming from the DC.
7–0 MD_MASK_20	Mapping unit's mask value #20 This field defines the mask value #20 within the 8bit word coming from the DC.

45.51.319 DC Mapping Configuration Register 26 (IPU_DC_MAP_CONF_26)

Address: IPU_DC_MAP_CONF_26 is 1E00_0000h base + 58170h offset = 1E05_8170h

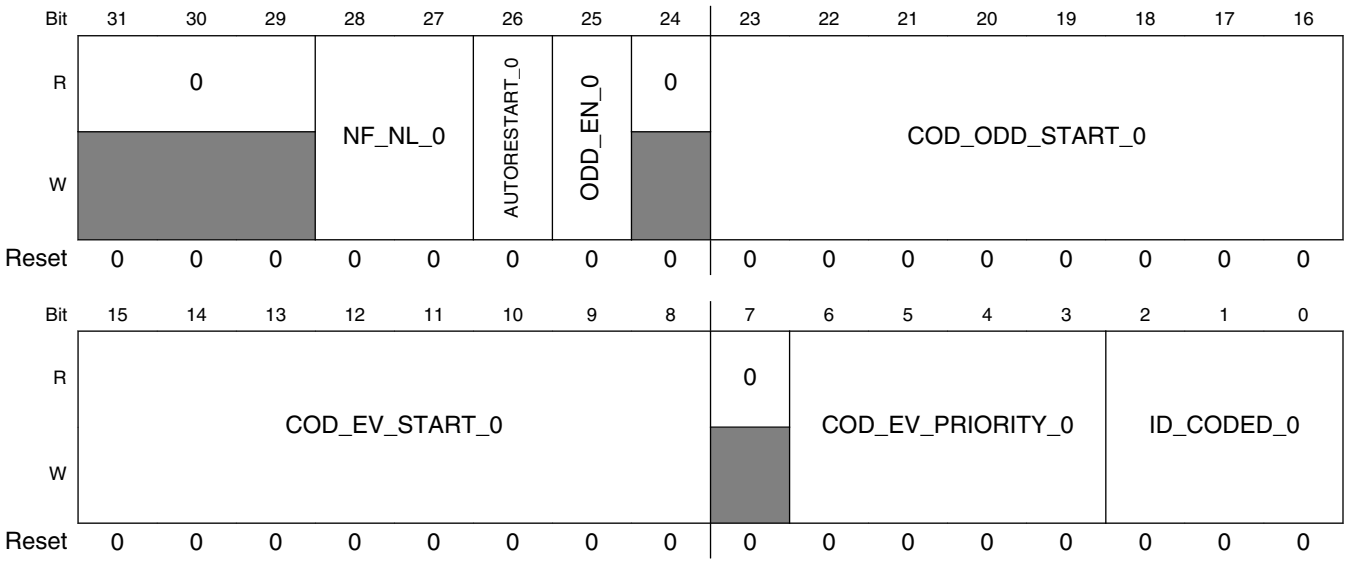


IPU_DC_MAP_CONF_26 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–24 MD_OFFSET_23	Mapping unit's offset parameter #23 This field defines the offset parameter #23 within the 24bit word coming from the DC.
23–16 MD_MASK_23	Mapping unit's mask value #23 This field defines the mask value #23 within the 8bit word coming from the DC.
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved.
12–8 MD_OFFSET_22	Mapping unit's offset parameter #22 This field defines the offset parameter #22 within the 24bit word coming from the DC.
7–0 MD_MASK_22	Mapping unit's mask value #22 This field defines the mask value #22 within the 8bit word coming from the DC.

45.51.320 DC User General Data Event 0 Register 0 (IPU_DC_UGDE0_0)

Address: IPU_DC_UGDE0_0 is 1E00_0000h base + 58174h offset = 1E05_8174h



IPU_DC_UGDE0_0 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–27 NF_NL_0	the user may attach his general event #0 to New-line New-Frame and New-field events. One of these event triggers the user's general event #0's counter. The actual internal trigger is the pixel following the occurrence of the selected event. 00 New Line 01 New Frame 10 New Field 11 Reserved
26 AUTORESTART_0	User's general event #0 auto restart mode 0 disable 1 User's general event #0's counter is automatically restarted.
25 ODD_EN_0	The user's general event #0 may be split into 2 internal signals. One one mode all the events are sent on one signal, on the second mode odd events are sent over one signal while even events are sent over the other signal (ODD_MODE) 1 ODD_MODE is enabled 0 ODD_MODE is disabled
24 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_UGDE0_0 field descriptions (continued)

Field	Description
23–16 COD_ODD_ START_0	This field holds a pointer in the microcode holding the routine to be performed following the user general event #0. When ODD_MODE is enabled, only the odd events will use this pointer When ODD_MODE is disabled this field is ignored
15–8 COD_EV_ START_0	This field holds a pointer in the microcode holding the routine to be performed following the user general event #0. When ODD_MODE is enabled, only the even events will use this pointer When ODD_MODE is disabled, all the events will use this pointer
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–3 COD_EV_ PRIORITY_0	This field defines the priority of the user general event #0 The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
2–0 ID_CODED_0	This field defines the number of DC channel number that user's general event #0 will be associated to. 000 DC channel_0 001 DC channel_1 010 DC channel_2 011 DC channel_5 (DP_SYNC) 100 DC channel_6 (DP ASYNC) 101 Reserved 110 Reserved. 111 Reserved.

45.51.321 DC User General Data Event 0 Register 1 (IPU_DC_UGDE0_1)

Address: IPU_DC_UGDE0_1 is 1E00_0000h base + 58178h offset = 1E05_8178h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																STEP_0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE0_1 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 STEP_0	This field holds the pre defined value that the counter counts too.

45.51.322 DC User General Data Event 0 Register2 (IPU_DC_UGDE0_2)

Address: IPU_DC_UGDE0_2 is 1E00_0000h base + 5817Ch offset = 1E05_817Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																OFFSET_DT_0																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE0_2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 OFFSET_DT_0	This field defines the offset value from which the counter of user general event #0 will start counting from

45.51.323 DC User General Data Event 0 Register 3 (IPU_DC_UGDE0_3)

Address: IPU_DC_UGDE0_3 is 1E00_0000h base + 58180h offset = 1E05_8180h

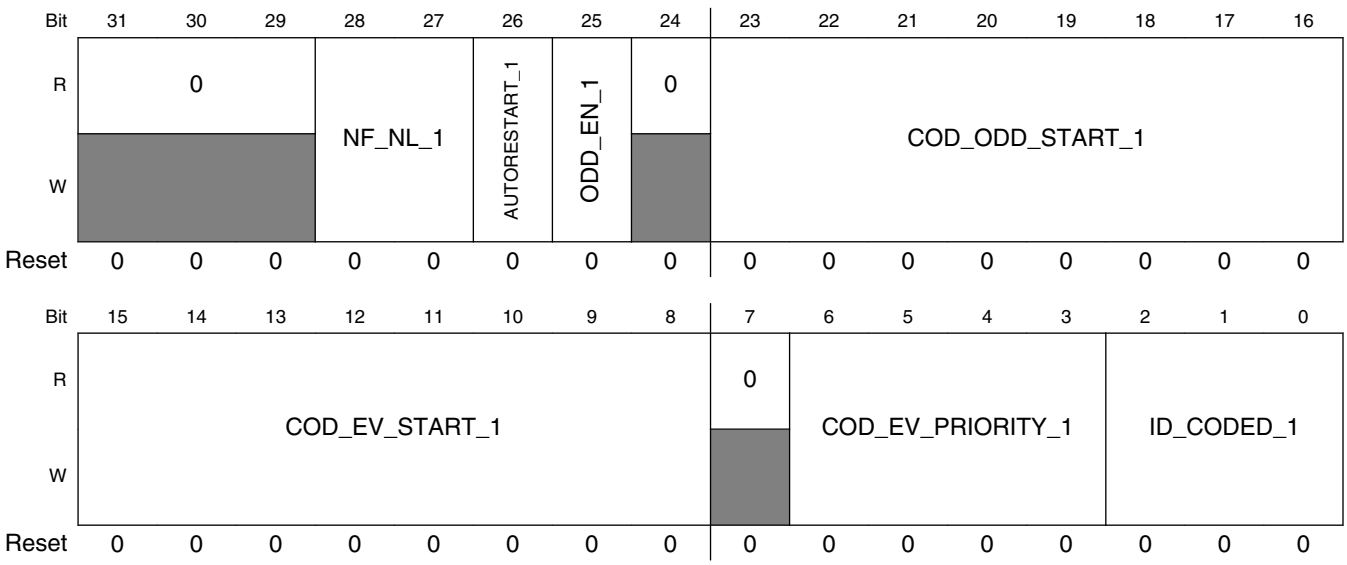
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																STEP_REPEAT_0																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE0_3 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 STEP_ REPEAT_0	When auto reload mode is disabled this field defines the number of events that will be generated by the user general event #0 mechanism

45.51.324 DC User General Data Event 1Register0 (IPU_DC_UGDE1_0)

Address: IPU_DC_UGDE1_0 is 1E00_0000h base + 58184h offset = 1E05_8184h



IPU_DC_UGDE1_0 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–27 NF_NL_1	the user may attach his general event #1 to New-line New-Frame and New-field events. One of these event triggers the user's general event #1's counter. The actual internal trigger is the pixel following the occurrence of the selected event. 00 New Line 01 New Frame 10 New Field 11 Reserved
26 AUTORESTART_1	User's general event #1 auto restart mode 0 disable 1 User's general event #1's counter is automatically restarted.
25 ODD_EN_1	The user's general event #1 may be split into 2 internal signals. One one mode all the events are sent on one signal, on the second mode odd events are sent over one signal while even events are sent over the other signal (ODD_MODE) 1 ODD_MODE is enabled 0 ODD_MODE is disabled
24 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_UGDE1_0 field descriptions (continued)

Field	Description
23–16 COD_ODD_ START_1	This field holds a pointer in the microcode holding the routine to be performed following the user general event #1. When ODD_MODE is enabled, only the odd events will use this pointer When ODD_MODE is disabled this field is ignored
15–8 COD_EV_ START_1	This field holds a pointer in the microcode holding the routine to be performed following the user general event #1. When ODD_MODE is enabled, only the even events will use this pointer When ODD_MODE is disabled, all the events will use this pointer
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–3 COD_EV_ PRIORITY_1	This field defines the priority of the user general event #1 The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
2–0 ID_CODED_1	This field defines the number of DC channel number that user's general event #1 will be associated to 000 DC channel_0 001 DC channel_1 010 DC channel_2 011 DC channel_5 (DP_SYNC) 100 DC channel_6 (DP ASYNC) 101 Reserved 110 Reserved 111 Reserved

45.51.325 DC User General Data Event 1 Register 1 (IPU_DC_UGDE1_1)

Address: IPU_DC_UGDE1_1 is 1E00_0000h base + 58188h offset = 1E05_8188h

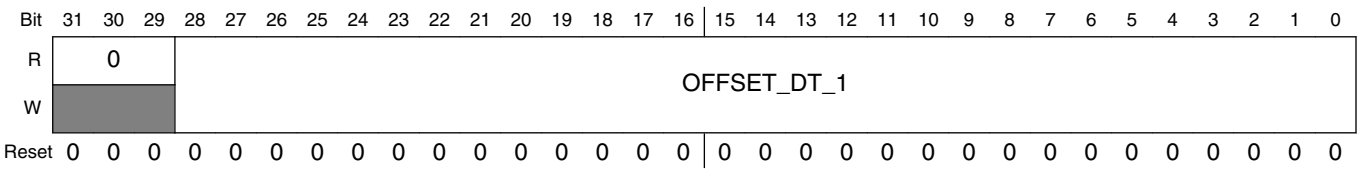
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																STEP_1															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE1_1 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 STEP_1	This field hold the pre defined value that the counter counts too

45.51.326 DC User General Data Event 1Register 2 (IPU_DC_UGDE1_2)

Address: IPU_DC_UGDE1_2 is 1E00_0000h base + 5818Ch offset = 1E05_818Ch

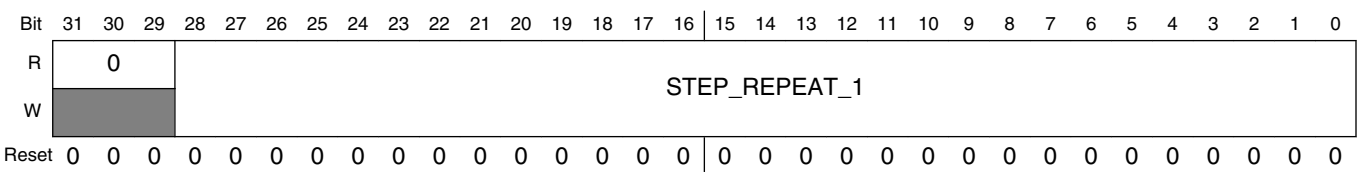


IPU_DC_UGDE1_2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 OFFSET_DT_1	This field defines the offset value from which the counter of user general event #1 will start counting from

45.51.327 DC User General Data Event 1Register 3 (IPU_DC_UGDE1_3)

Address: IPU_DC_UGDE1_3 is 1E00_0000h base + 58190h offset = 1E05_8190h

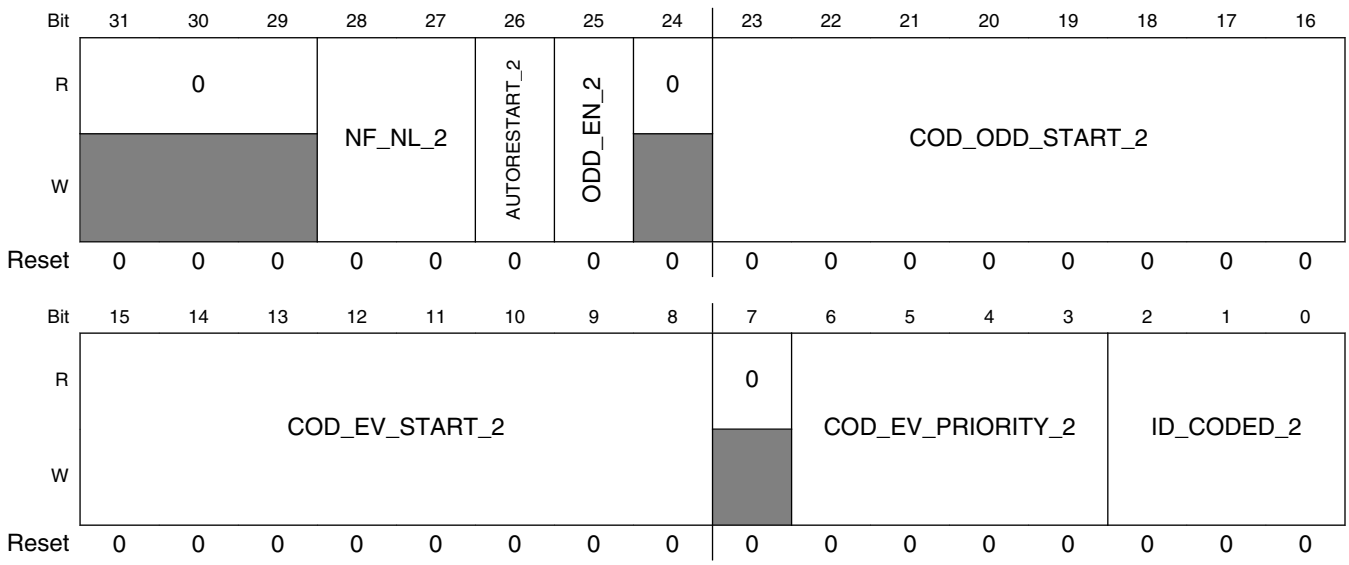


IPU_DC_UGDE1_3 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 STEP_ REPEAT_1	When auto reload mode is disabled this field defines the number of events that will be generated by the user general event #1 mechanism

45.51.328 DC User General Data Event 2 Register 0 (IPU_DC_UGDE2_0)

Address: IPU_DC_UGDE2_0 is 1E00_0000h base + 58194h offset = 1E05_8194h



IPU_DC_UGDE2_0 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–27 NF_NL_2	the user may attach his general event #2 to New-line New-Frame and New-field events. One of these event triggers the user's general event #2's counter. The actual internal trigger is the pixel following the occurrence of the selected event. 00 New Line 01 New Frame 10 New Field 11 Reserved
26 AUTORESTART_2	User's general event #2 auto restart mode 0 disable 1 User's general event #2's counter is automatically restarted.
25 ODD_EN_2	The user's general event #2 may be split into 2 internal signals. One one mode all the events are sent on one signal, on the second mode odd events are sent over one signal while even events are sent over the other signal (ODD_MODE) 1 ODD_MODE is enabled 0 ODD_MODE is disabled
24 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_UGDE2_0 field descriptions (continued)

Field	Description
23–16 COD_ODD_ START_2	This field holds a pointer in the microcode holding the routine to be performed following the user general event #2 When ODD_MODE is enabled, only the odd events will use this pointer When ODD_MODE is disabled this field is ignored
15–8 COD_EV_ START_2	This field holds a pointer in the microcode holding the routine to be performed following the user general event #2. When ODD_MODE is enabled, only the even events will use this pointer When ODD_MODE is disabled, all the events will use this pointer
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–3 COD_EV_ PRIORITY_2	This field defines the priority of the user general event #2 The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0000 disable 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
2–0 ID_CODED_2	This field defines the number of DC channel number that user's general event #2 will be associated to 000 DC channel_0 001 DC channel_1 010 DC channel_2 011 DC channel_5 (DP_SYNC) 100 DC channel_6 (DP ASYNC) 101 Reserved 110 Reserved 111 Reserved

45.51.329 DC User General Data Event 2 Register 1 (IPU_DC_UGDE2_1)

Address: IPU_DC_UGDE2_1 is 1E00_0000h base + 58198h offset = 1E05_8198h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																STEP_2															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE2_1 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 STEP_2	This field hold the pre defined value that the counter counts too

45.51.330 DC User General Data Event 2Register 2 (IPU_DC_UGDE2_2)

Address: IPU_DC_UGDE2_2 is 1E00_0000h base + 5819Ch offset = 1E05_819Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																OFFSET_DT_2																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE2_2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 OFFSET_DT_2	This field defines the offset value from which the counter of user general event #2 will start counting from

45.51.331 DC User General Data Event 2Register 3 (IPU_DC_UGDE2_3)

Address: IPU_DC_UGDE2_3 is 1E00_0000h base + 581A0h offset = 1E05_81A0h

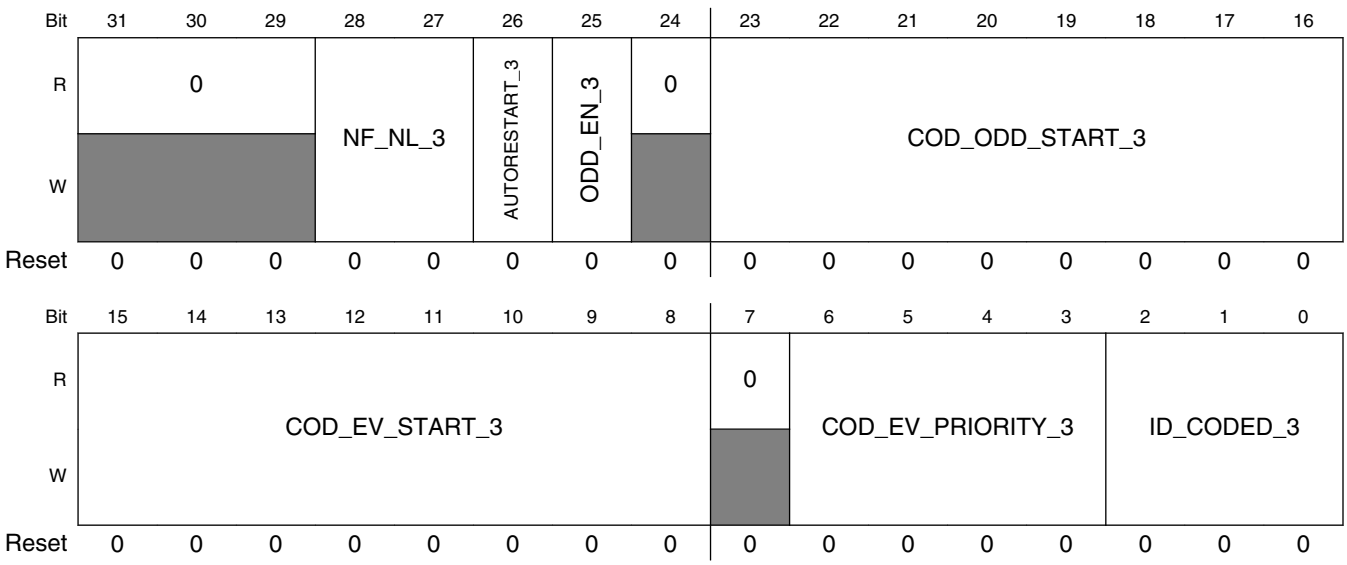
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																STEP_REPEAT_2																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE2_3 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 STEP_ REPEAT_2	When auto reload mode is disabled this field defines the number of events that will be generated by the user general event #2 mechanism

45.51.332 DC User General Data Event 3Register 0 (IPU_DC_UGDE3_0)

Address: IPU_DC_UGDE3_0 is 1E00_0000h base + 581A4h offset = 1E05_81A4h



IPU_DC_UGDE3_0 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–27 NF_NL_3	the user may attach his general event #3 to New-line New-Frame and New-field events. One of these event triggers the user's general event #3's counter. The actual internal trigger is the pixel following the occurrence of the selected event. 00 New Line 01 New Frame 10 New Field 11 Reserved
26 AUTORESTART_3	User's general event #3 auto restart mode 0 disable 1 User's general event #3's counter is automatically restarted.
25 ODD_EN_3	The user's general event #3 may be split into 2 internal signals. One one mode all the events are sent on one signal, on the second mode odd events are sent over one signal while even events are sent over the other signal (ODD_MODE) 1 ODD_MODE is enabled 0 ODD_MODE is disabled
24 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DC_UGDE3_0 field descriptions (continued)

Field	Description
23–16 COD_ODD_ START_3	This field holds a pointer in the microcode holding the routine to be performed following the user general event #3. When ODD_MODE is enabled, only the odd events will use this pointer When ODD_MODE is disabled this field is ignored
15–8 COD_EV_ START_3	This field holds a pointer in the microcode holding the routine to be performed following the user general event #3. When ODD_MODE is enabled, only the even events will use this pointer When ODD_MODE is disabled, all the events will use this pointer
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–3 COD_EV_ PRIORITY_3	This field defines the priority of the user general event #3 0000 disable The priority between the events should be set to a unique value. i.e. two events must not have the same priority (except 0000 - disable) 0001 Priority #1 (lowest) 0010 Priority #2 1101 Priority #13 (highest) 1110 Reserved 1111 Reserved
2–0 ID_CODED_3	This field defines the number of DC channel number that user's general event #3 will be associated to 000 DC channel_0 001 DC channel_1 010 DC channel_2 011 DC channel_5 (DP_SYNC) 100 DC channel_6 (DP ASYNC) 101 Reserved 110 Reserved 111 Reserved.

45.51.333 DC User General Data Event 3Register 1 (IPU_DC_UGDE3_1)

Address: IPU_DC_UGDE3_1 is 1E00_0000h base + 581A8h offset = 1E05_81A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																STEP_3															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE3_1 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 STEP_3	This field hold the pre defined value that the counter counts too

45.51.334 DC User General Data Event 3Register 2 (IPU_DC_UGDE3_2)

Address: IPU_DC_UGDE3_2 is 1E00_0000h base + 581ACh offset = 1E05_81ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																OFFSET_DT_3																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE3_2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 OFFSET_DT_3	This field defines the offset value from which the counter of user general event #3 will start counting from

45.51.335 DC User General Data Event 3Register 2 (IPU_DC_UGDE3_3)

Address: IPU_DC_UGDE3_3 is 1E00_0000h base + 581B0h offset = 1E05_81B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																STEP_REPEAT_3																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_UGDE3_3 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 STEP_ REPEAT_3	When auto reload mode is disabled this field defines the number of events that will be generated by the user general event #3 mechanism

45.51.336 DC Low Level Access Control Register 0 (IPU_DC_LLA0)

Address: IPU_DC_LLA0 is 1E00_0000h base + 581B4h offset = 1E05_81B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MCU_RS_3_0								MCU_RS_2_0								MCU_RS_1_0								MCU_RS_0_0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_LLA0 field descriptions

Field	Description
31–24 MCU_RS_3_0	This field holds a pointer in the microcode handling the RS_3 routine for the display defined at DISP_ID_8, when in Low level access mode,
23–16 MCU_RS_2_0	This field holds a pointer in the microcode handling the RS_2 routine for the display defined at DISP_ID_8, when in Low level access mode,
15–8 MCU_RS_1_0	This field holds a pointer in the microcode handling the RS_1 routine for the display defined at DISP_ID_8, when in Low level access mode,
7–0 MCU_RS_0_0	This field holds a pointer in the microcode handling the RS_0 routine for the display defined at DISP_ID_8, when in Low level access mode,

45.51.337 DC Low Level Access Control Register 1 (IPU_DC_LLA1)

Address: IPU_DC_LLA1 is 1E00_0000h base + 581B8h offset = 1E05_81B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MCU_RS_3_1								MCU_RS_2_1								MCU_RS_1_1								MCU_RS_0_1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_LLA1 field descriptions

Field	Description
31–24 MCU_RS_3_1	This field holds a pointer in the microcode handling the RS_3 routine for the display defined at DISP_ID_9, when in Low level access mode,
23–16 MCU_RS_2_1	This field holds a pointer in the microcode handling the RS_2 routine for the display defined at DISP_ID_9, when in Low level access mode,
15–8 MCU_RS_1_1	This field holds a pointer in the microcode handling the RS_1 routine for the display defined at DISP_ID_9, when in Low level access mode,
7–0 MCU_RS_0_1	This field holds a pointer in the microcode handling the RS_0 routine for the display defined at DISP_ID_9, when in Low level access mode,

45.51.338 DC Read Low Level Read Access Control Register 0 (IPU_DC_R_LLA0)

Address: IPU_DC_R_LLA0 is 1E00_0000h base + 581BCh offset = 1E05_81BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MCU_RS_3_0								MCU_RS_2_0								MCU_RS_R_1_0								MCU_RS_R_0_0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_R_LLA0 field descriptions

Field	Description
31–24 MCU_RS_3_0	This field holds a pointer in the microcode handling the RS_3 routine for the display defined at DISP_ID_8, when in Read Low level access mode,
23–16 MCU_RS_2_0	This field holds a pointer in the microcode handling the RS_2 routine for the display defined at DISP_ID_8, when in Read Low level access mode,
15–8 MCU_RS_R_1_0	This field holds a pointer in the microcode handling the RS_1 routine for the display defined at DISP_ID_8, when in Read Low level access mode,
7–0 MCU_RS_R_0_0	This field holds a pointer in the microcode handling the RS_0 routine for the display defined at DISP_ID_8, when in Read Low level access mode,

45.51.339 DC Read Low Level Read Access Control Register1 (IPU_DC_R_LLA1)

Address: IPU_DC_R_LLA1 is 1E00_0000h base + 581C0h offset = 1E05_81C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MCU_RS_R_3_1								MCU_RS_R_2_1								MCU_RS_R_1_1								MCU_RS_R_0_1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_R_LLA1 field descriptions

Field	Description
31–24 MCU_RS_R_3_1	This field holds a pointer in the microcode handling the RS_3 routine for the display defined at DISP_ID_9, when in Read Low level access mode,
23–16 MCU_RS_R_2_1	This field holds a pointer in the microcode handling the RS_2 routine for the display defined at DISP_ID_9, when in Read Low level access mode,

Table continues on the next page...

IPU_DC_R_LLA1 field descriptions (continued)

Field	Description
15–8 MCU_RS_R_1_1	This field holds a pointer in the microcode handling the RS_1 routine for the display defined at DISP_ID_9, when in Read Low level access mode,
7–0 MCU_RS_R_0_1	This field holds a pointer in the microcode handling the RS_0 routine for the display defined at DISP_ID_9, when in Read Low level access mode,

45.51.340 DC Write Channel 5 Configuration Register (IPU_DC_WR_CH_ADDR_5_ALT)

Address: IPU_DC_WR_CH_ADDR_5_ALT is 1E00_0000h base + 581C4h offset = 1E05_81C4h

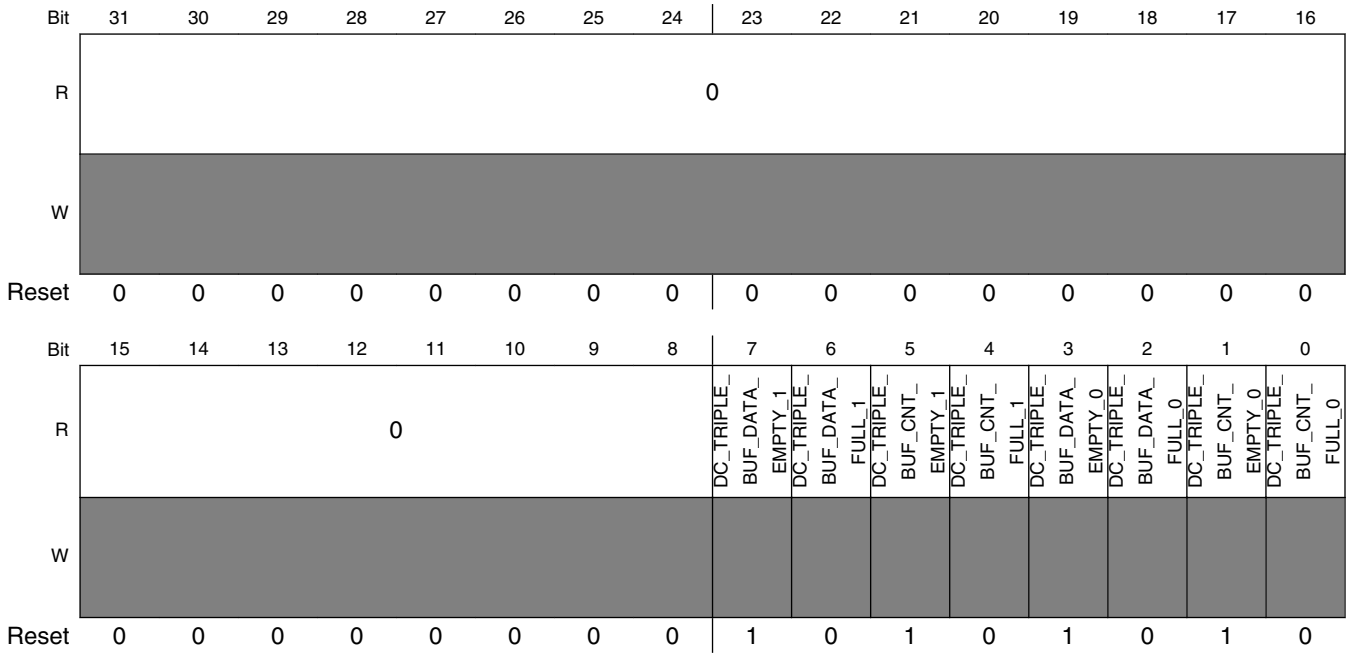
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			ST_ADDR_5_ALT																												
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_WR_CH_ADDR_5_ALT field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 ST_ADDR_5_ALT	This field defines the start address within the display's memory space where the write transactions will be done to for channel #5, when alternate flow is performed via channel #5

45.51.341 DC Status Register (IPU_DC_STAT)

Address: IPU_DC_STAT is 1E00_0000h base + 581C8h offset = 1E05_81C8h



IPU_DC_STAT field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved.
7 DC_TRIPLE_BUF_DATA_EMPTY_1	This bit indicates a FIFO full state on the DC FIFO accessing DI1 when write to the display flow is used
6 DC_TRIPLE_BUF_DATA_FULL_1	This bit indicates a FIFO empty state on the DC FIFO accessing DI1 when read from the display flow is used
5 DC_TRIPLE_BUF_CNT_EMPTY_1	This bit indicates a FIFO empty state on the DC FIFO accessing DI1 when write to the display flow is used
4 DC_TRIPLE_BUF_CNT_FULL_1	This bit indicates a FIFO full state on the DC FIFO accessing DI1 when write to the display flow is used
3 DC_TRIPLE_BUF_DATA_EMPTY_0	This bit indicates a FIFO empty state on the DC FIFO accessing DI0 when read from the display flow is used

Table continues on the next page...

IPU_DC_STAT field descriptions (continued)

Field	Description
2 DC_TRIPLE_BUF_DATA_FULL_0	This bit indicates a FIFO full state on the DC FIFO accessing DI0 when read from the display flow is used
1 DC_TRIPLE_BUF_CNT_EMPTY_0	This bit indicates a FIFO empty state on the DC FIFO accessing DI0 when write to the display flow is used
0 DC_TRIPLE_BUF_CNT_FULL_0	This bit indicates a FIFO full state on the DC FIFO accessing DI0 when write to the display flow is used

45.51.342 DC Display Configuration 2 Register 1 (IPU_DC_DISP_CONF2_1)

Address: IPU_DC_DISP_CONF2_1 is 1E00_0000h base + 590ECh offset = 1E05_90ECh

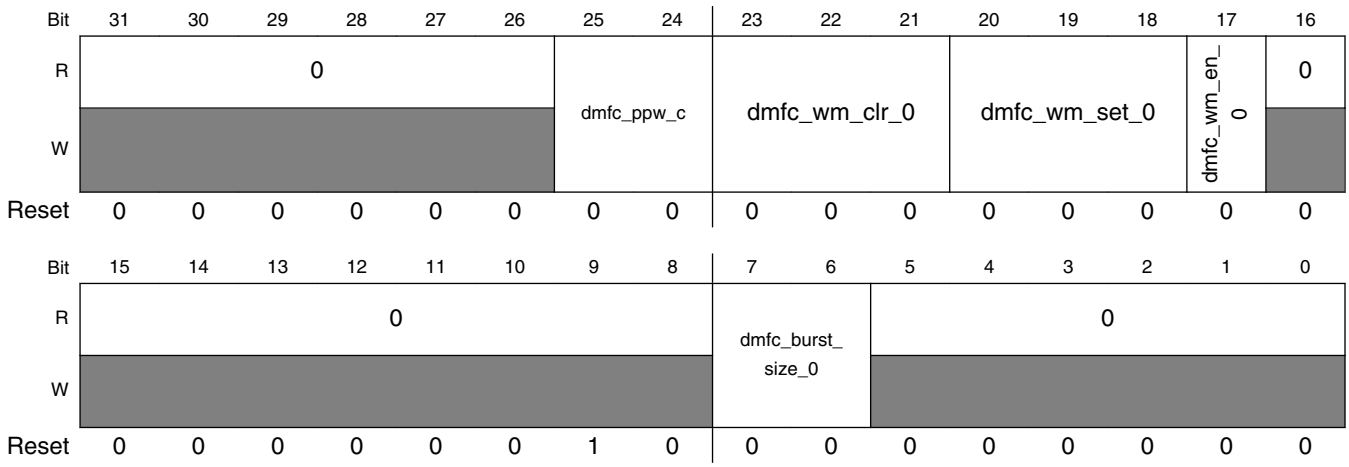
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																SL_1																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DC_DISP_CONF2_1 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–0 SL_1	Stride line of display 1

45.51.343 DMFC Read Channel Register (IPU_DMFC_RD_CHAN)

Address: IPU_DMFC_RD_CHAN is 1E00_0000h base + 60000h offset = 1E06_0000h



IPU_DMFC_RD_CHAN field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value zero. Reserved.
25–24 dmfc_ppw_c	Pixel Per Word coded. This field defines the size of the read data from the display. 00 8 bit per pixel 01 16 bit per pixel 10 24 (rgb) bit per pixel or 32 bit per pixel 11 Reserved
23–21 dmfc_wm_clr_0	Watermark Clear This field defines the watermark's level of the DMFC read FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of free bursts at the FIFO (dmfc_wm_clr_0 > dmfc_wm_set_0)
20–18 dmfc_wm_set_0	Watermark Set This field defines the watermark's level of the DMFC read FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of free bursts at the FIFO (dmfc_wm_clr_0 > dmfc_wm_set_0)
17 dmfc_wm_en_0	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
16–8 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

IPU_DMFC_RD_CHAN field descriptions (continued)

Field	Description
7–6 dmfc_burst_size_0	Read burst Size This field defines the burst size of the DMFC's read accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, going to the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
5–0 Reserved	This read-only field is reserved and always has the value zero. Reserved.

45.51.344 DMFC Write Channel Register (IPU_DMFC_WR_CHAN)

Address: IPU_DMFC_WR_CHAN is 1E00_0000h base + 60004h offset = 1E06_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	dmfc_burst_size_2c		dmfc_fifo_size_2c			dmfc_st_addr_2c			dmfc_burst_size_1c		dmfc_fifo_size_1c			dmfc_st_addr_1c		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	dmfc_burst_size_2		dmfc_fifo_size_2			dmfc_st_addr_2			dmfc_burst_size_1		dmfc_fifo_size_1			dmfc_st_addr_1		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DMFC_WR_CHAN field descriptions

Field	Description
31–30 dmfc_burst_size_2c	Burst size of IDMAC's channel 43 This field defines the burst size of the IDMAC's channel 43 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
29–27 dmfc_fifo_size_2c	DMFC FIFO size for IDMAC's channel 43 This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 43 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel

Table continues on the next page...

IPU_DMFC_WR_CHAN field descriptions (continued)

Field	Description
	101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
26–24 dmfc_st_addr_2c	DMFC Start Address for IDMAC's channel 43 This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 43. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7
23–22 dmfc_burst_ size_1c	Burst size of IDMAC's channel 42 This field defines the burst size of the IDMAC's channel 42 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
21–19 dmfc_fifo_size_ 1c	DMFC FIFO size for IDMAC's channel 42 This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 42 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
18–16 dmfc_st_addr_1c	DMFC Start Address for IDMAC's channel 42 This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 42. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7
15–14 dmfc_burst_ size_2	Burst size of IDMAC's channel 41 This field defines the burst size of the IDMAC's channel 41 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC)

Table continues on the next page...

IPU_DMFC_WR_CHAN field descriptions (continued)

Field	Description
	01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
13–11 dmfc_fifo_size_2	DMFC FIFO size for IDMAC's channel 41 This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 41 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
10–8 dmfc_st_addr_2	DMFC Start Address for IDMAC's channel 41 This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 41. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7
7–6 dmfc_burst_ size_1	Burst size of IDMAC's channel 28 This field defines the burst size of the IDMAC's channel 28 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 1hbit
5–3 dmfc_fifo_size_1	DMFC FIFO size for IDMAC's channel 28 This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 28 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
2–0 dmfc_st_addr_1	DMFC Start Address for IDMAC's channel 28 This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 28. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment

Table continues on the next page...

IPU_DMFC_WR_CHAN field descriptions (continued)

Field	Description
	000 Segment 0 001 Segment 1 111 Segment 7

45.51.345 DMFC Write Channel Definition Register (IPU_DMFC_WR_CHAN_DEF)

Address: IPU_DMFC_WR_CHAN_DEF is 1E00_0000h base + 60008h offset = 1E06_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	dmfc_wm_clr_2c			dmfc_wm_set_2c			dmfc_wm_en_2c	0	dmfc_wm_clr_1c			dmfc_wm_set_1c			dmfc_wm_en_1c	0
Reset	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	dmfc_wm_clr_2			dmfc_wm_set_2			dmfc_wm_en_2	0	dmfc_wm_clr_1			dmfc_wm_set_1			dmfc_wm_en_1	0
Reset	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

IPU_DMFC_WR_CHAN_DEF field descriptions

Field	Description
31–29 dmfc_wm_clr_2c	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
28–26 dmfc_wm_set_2c	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
25 dmfc_wm_en_2c	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23–21 dmfc_wm_clr_1c	Watermark Clear

Table continues on the next page...

IPU_DMFC_WR_CHAN_DEF field descriptions (continued)

Field	Description
	This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
20–18 dmfc_wm_set_1c	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
17 dmfc_wm_en_1c	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–13 dmfc_wm_clr_2	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
12–10 dmfc_wm_set_2	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
9 dmfc_wm_en_2	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–5 dmfc_wm_clr_1	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
4–2 dmfc_wm_set_1	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
1 dmfc_wm_en_1	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

45.51.346 DMFC Display Processor Channel Register (IPU_DMFC_DP_CHAN)

Address: IPU_DMFC_DP_CHAN is 1E00_0000h base + 6000Ch offset = 1E06_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	dmfc_burst_size_6f		dmfc_fifo_size_6f			dmfc_st_addr_6f			dmfc_burst_size_6b		dmfc_fifo_size_6b			dmfc_st_addr_6b		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	dmfc_burst_size_5f		dmfc_fifo_size_5f			dmfc_st_addr_5f			dmfc_burst_size_5b		dmfc_fifo_size_5b			dmfc_st_addr_5b		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DMFC_DP_CHAN field descriptions

Field	Description
31–30 dmfc_burst_size_6f	Burst size of IDMAC's channel 29 This field defines the burst size of the IDMAC's channel 29 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
29–27 dmfc_fifo_size_6f	DMFC FIFO size for IDMAC's channel 29 This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 29 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
26–24 dmfc_st_addr_6f	DMFC Start Address for IDMAC's channel 29 This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 29. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7

Table continues on the next page...

IPU_DMFC_DP_CHAN field descriptions (continued)

Field	Description
23–22 dmfc_burst_size_6b	Burst size of IDMAC's channel 24 This field defines the burst size of the IDMAC's channel 24 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
21–19 dmfc_fifo_size_6b	DMFC FIFO size for IDMAC's channel 24 This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 24 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
18–16 dmfc_st_addr_6b	DMFC Start Address for IDMAC's channel 24 This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 24. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7
15–14 dmfc_burst_size_5f	Burst size of IDMAC's channel 27 This field defines the burst size of the IDMAC's channel 27 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
13–11 dmfc_fifo_size_5f	DMFC FIFO size for IDMAC's channel 27 This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 27 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel

Table continues on the next page...

IPU_DMFC_DP_CHAN field descriptions (continued)

Field	Description
	110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
10–8 dmfc_st_addr_5f	DMFC Start Address for IDMAC's channel 27 This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 27. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7
7–6 dmfc_burst_ size_5b	Burst size of IDMAC's channel 23 This field defines the burst size of the IDMAC's channel 23 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
5–3 dmfc_fifo_size_ 5b	DMFC FIFO size for IDMAC's channel 23 This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 23 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
2–0 dmfc_st_addr_ 5b	DMFC Start Address for IDMAC's channel 23 This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 23. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7

45.51.347 DMFC Display Processor Channel Definition Register (IPU_DMFC_DP_CHAN_DEF)

Address: IPU_DMFC_DP_CHAN_DEF is 1E00_0000h base + 60010h offset = 1E06_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	dmfc_wm_clr_6f			dmfc_wm_set_6f			dmfc_wm_en_6f	0	dmfc_wm_clr_6b			dmfc_wm_set_6b			dmfc_wm_en_6b	0
W																
Reset	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	dmfc_wm_clr_5f			dmfc_wm_set_5f			dmfc_wm_en_5f	0	dmfc_wm_clr_5b			dmfc_wm_set_5b			dmfc_wm_en_5b	0
W																
Reset	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

IPU_DMFC_DP_CHAN_DEF field descriptions

Field	Description
31–29 dmfc_wm_clr_6f	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
28–26 dmfc_wm_set_6f	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
25 dmfc_wm_en_6f	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23–21 dmfc_wm_clr_6b	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
20–18 dmfc_wm_set_6b	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set

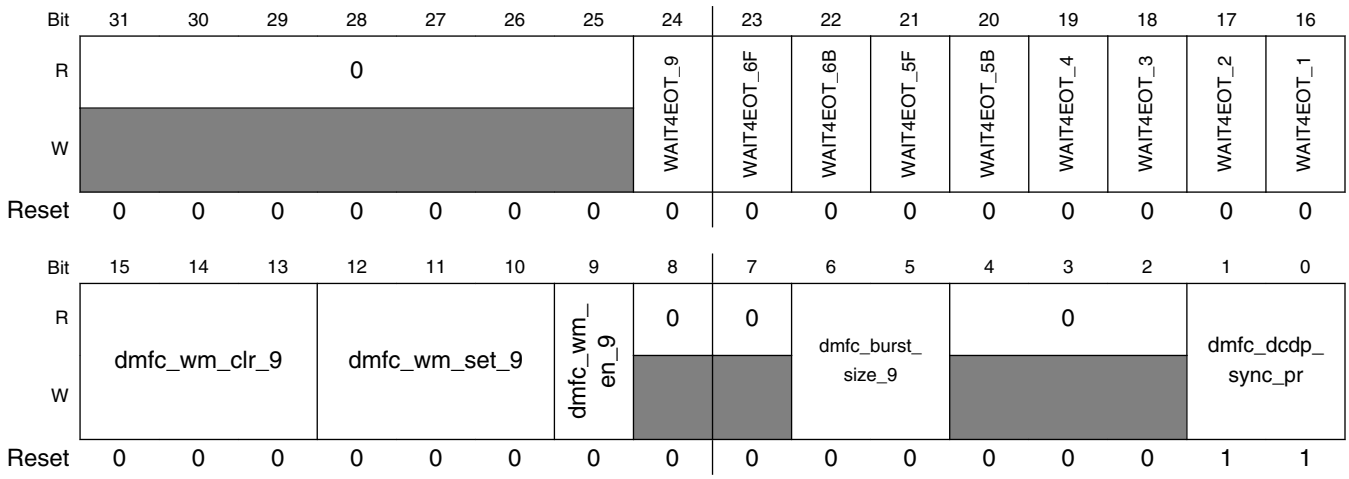
Table continues on the next page...

IPU_DMFC_DP_CHAN_DEF field descriptions (continued)

Field	Description
17 dmfc_wm_en_6b	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–13 dmfc_wm_clr_5f	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
12–10 dmfc_wm_set_5f	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
9 dmfc_wm_en_5f	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–5 dmfc_wm_clr_5b	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
4–2 dmfc_wm_set_ 5b	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
1 dmfc_wm_en_5b	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

45.51.348 DMFC General 1 Register (IPU_DMFC_GENERAL_1)

Address: IPU_DMFC_GENERAL_1 is 1E00_0000h base + 60014h offset = 1E06_0014h



IPU_DMFC_GENERAL_1 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24 WAIT4EOT_9	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #9 is in wait4eot mode 0 FIFO #9 is in normal mode
23 WAIT4EOT_6F	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #6F is in wait4eot mode 0 FIFO #6F is in normal mode
22 WAIT4EOT_6B	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #6B is in wait4eot mode 0 FIFO #6B is in normal mode
21 WAIT4EOT_5F	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO.

Table continues on the next page...

IPU_DMFC_GENERAL_1 field descriptions (continued)

Field	Description
	When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #5F is in wait4eot mode 0 FIFO #5F is in normal mode
20 WAIT4EOT_5B	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #5B is in wait4eot mode 0 FIFO #5B is in normal mode
19 WAIT4EOT_4	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #4 is in wait4eot mode 0 FIFO #4 is in normal mode
18 WAIT4EOT_3	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #3 is in wait4eot mode 0 FIFO #3 is in normal mode
17 WAIT4EOT_2	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #2 is in wait4eot mode 0 FIFO #2 is in normal mode
16 WAIT4EOT_1	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #1 is in wait4eot mode 0 FIFO #1 is in normal mode
15–13 dmfc_wm_clr_9	Watermark Clear This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)

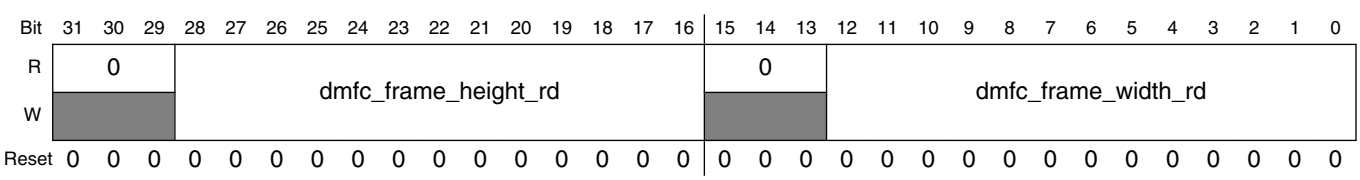
Table continues on the next page...

IPU_DMFC_GENERAL_1 field descriptions (continued)

Field	Description
12–10 dmfc_wm_set_9	Watermark Set This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
9 dmfc_wm_en_9	Watermark enable. This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–5 dmfc_burst_size_9	Burst size of IDMAC's channel 44 This field defines the burst size of the IDMAC's channel 44 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. This channel is targeted for MASK - the FIFO size is always 32X128; The base address is always the upper half of the 8th segment 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
4–2 Reserved	This read-only field is reserved and always has the value zero. Reserved.
1–0 dmfc_dcdp_sync_pr	DMFC's memory access priority settings for simultaneous synchronous flows from DC & DP 00 Forbidden - should not be used. 01 DC has higher priority over DP 10 DP has higher priority over DC 11 Round Robin

45.51.349 DMFC General 2 Register (IPU_DMFC_GENERAL_2)

Address: IPU_DMFC_GENERAL_2 is 1E00_0000h base + 60018h offset = 1E06_0018h



IPU_DMFC_GENERAL_2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–16 dmfc_frame_height_rd	Frame height for read channel from the display to the IDMAC; Units are pixels
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved.
12–0 dmfc_frame_width_rd	Frame width for read channel from the display to the IDMAC; Units are pixels

45.51.350 DMFC IC Interface Control Register (IPU_DMFC_IC_CTRL)

Address: IPU_DMFC_IC_CTRL is 1E00_0000h base + 6001Ch offset = 1E06_001Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	dmfc_ic_frame_height_rd													dmfc_ic_frame_width_rd[-3:10]		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	dmfc_ic_frame_width_rd[9:0]										dmfc_ic_ppw_c		0	dmfc_ic_in_port		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

IPU_DMFC_IC_CTRL field descriptions

Field	Description
31–19 dmfc_ic_frame_height_rd	Frame's height for the channel coming from IC. Units are lines
18–6 dmfc_ic_frame_width_rd	Frame's width for the channel coming from IC. Units are pixels
5–4 dmfc_ic_ppw_c	Pixel Per Word coded from IC. This field defines the size of the data coming from the IC. 00 8 bit per pixel 01 16 bit per pixel 10 24 bit per pixel 11 Reserved
3 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

IPU_DMFC_IC_CTRL field descriptions (continued)

Field	Description
2–0 dmfc_ic_in_port	DMFC input port When data is coming from the IC, the IC channel replaces one of the IDMAC's channels connected to the DMFC. This field defines which IDMAC's channel is replaced by the IC channel. 000 CH28 001 CH41 010 Reserved, IC channel is disabled 011 Reserved, IC channel is disabled 100 CH23 101 CH27 110 CH24 111 CH29

45.51.351 DMFC Write Channel Alternate Register (IPU_DMFC_WR_CHAN_ALT)

Address: IPU_DMFC_WR_CHAN_ALT is 1E00_0000h base + 60020h offset = 1E06_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	dmfc_burst_size_2_alt		dmfc_fifo_size_2_alt			dmfc_st_addr_2_alt			0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DMFC_WR_CHAN_ALT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–14 dmfc_burst_size_2_alt	Burst size of IDMAC's channel 41 (for alternate flow) This field defines the burst size of the IDMAC's channel 41 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
13–11 dmfc_fifo_size_2_alt	DMFC FIFO size for IDMAC's channel 41 (for alternate flow) This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 41

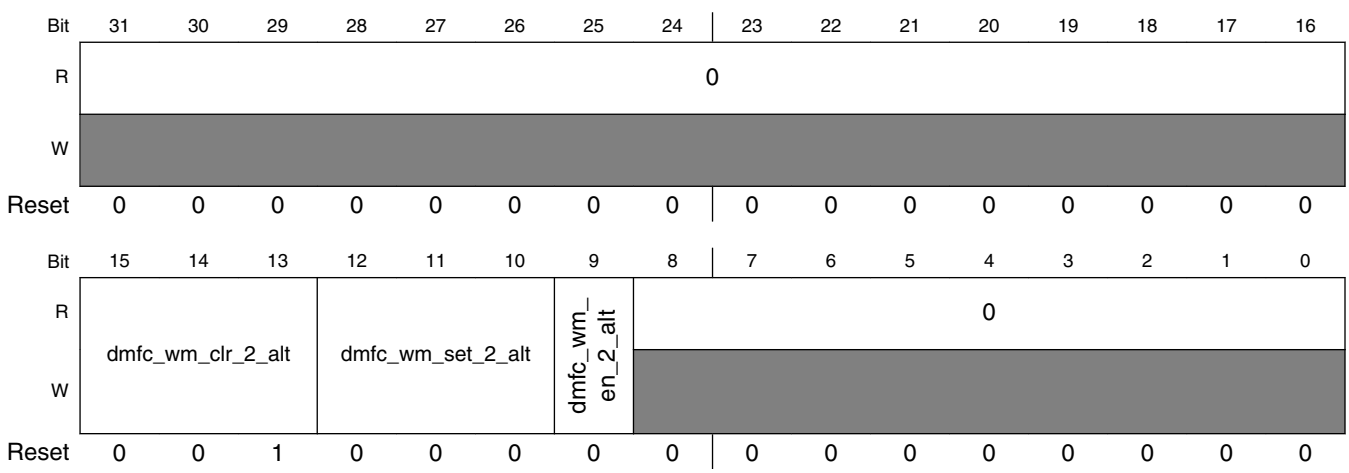
Table continues on the next page...

IPU_DMFC_WR_CHAN_ALT field descriptions (continued)

Field	Description
	000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
10–8 dmfc_st_addr_2_alt	DMFC Start Address for IDMAC's channel 41 (for alternate flow) This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 41. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7
7–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

45.51.352 DMFC Write Channel Definition Alternate Register (IPU_DMFC_WR_CHAN_DEF_ALT)

Address: IPU_DMFC_WR_CHAN_DEF_ALT is 1E00_0000h base + 60024h offset = 1E06_0024h



IPU_DMFC_WR_CHAN_DEF_ALT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved

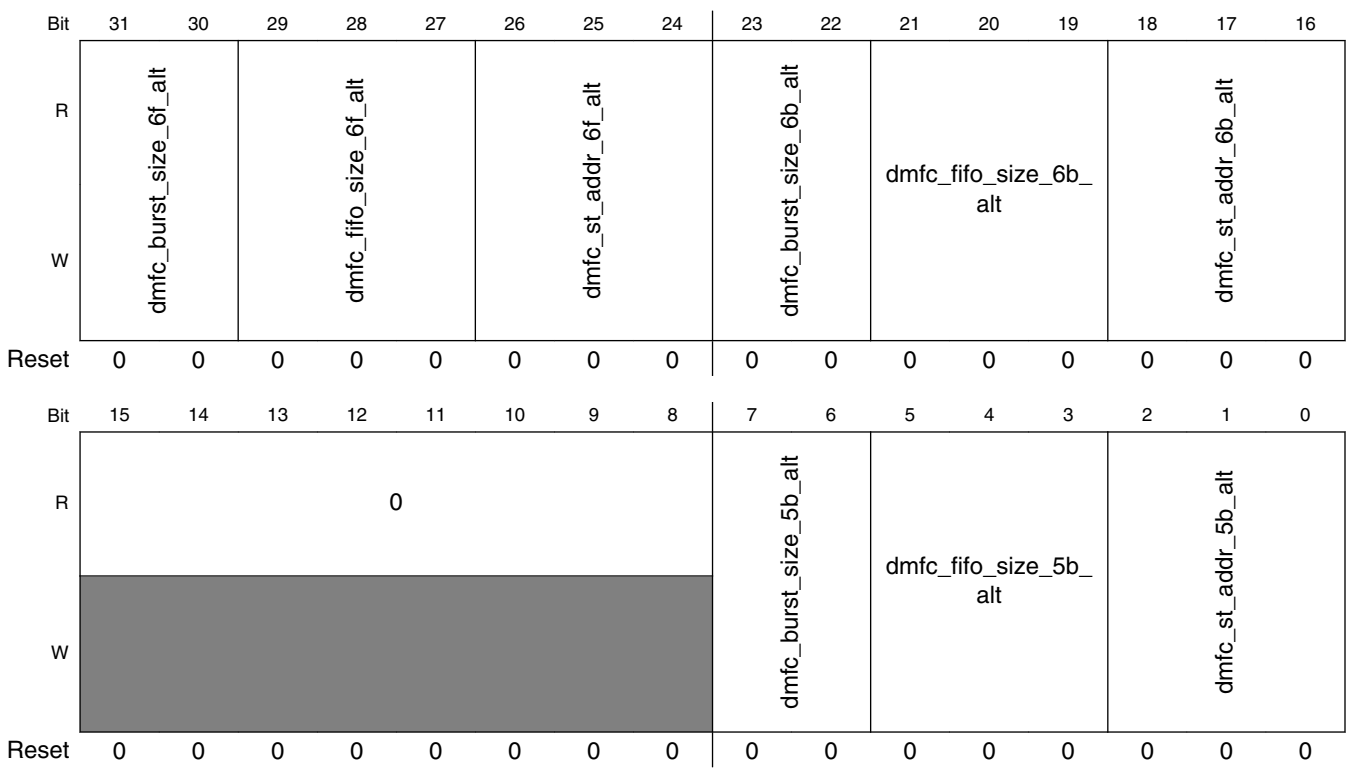
Table continues on the next page...

IPU_DMFC_WR_CHAN_DEF_ALT field descriptions (continued)

Field	Description
15–13 dmfc_wm_clr_2_alt	Watermark Clear (for alternate flow) This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
12–10 dmfc_wm_set_2_alt	Watermark Set (for alternate flow) This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
9 dmfc_wm_en_2_alt	Watermark enable. (for alternate flow) This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
8–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

45.51.353 DMFC MFC Display Processor Channel Alternate Register (IPU_DMFC_DP_CHAN_ALT)

Address: IPU_DMFC_DP_CHAN_ALT is 1E00_0000h base + 60028h offset = 1E06_0028h



IPU_DMFC_DP_CHAN_ALT field descriptions

Field	Description
31–30 dmfc_burst_size_6f_alt	Burst size of IDMAC's channel 29 (for alternate flow) This field defines the burst size of the IDMAC's channel 29 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
29–27 dmfc_fifo_size_6f_alt	DMFC FIFO size for IDMAC's channel 29 (for alternate flow) This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 29 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
26–24 dmfc_st_addr_6f_alt	DMFC Start Address for IDMAC's channel 29 (for alternate flow) This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 29. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7
23–22 dmfc_burst_size_6b_alt	Burst size of IDMAC's channel 24 (for alternate flow) This field defines the burst size of the IDMAC's channel 24 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
21–19 dmfc_fifo_size_6b_alt	DMFC FIFO size for IDMAC's channel 24 (for alternate flow) This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 24 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel

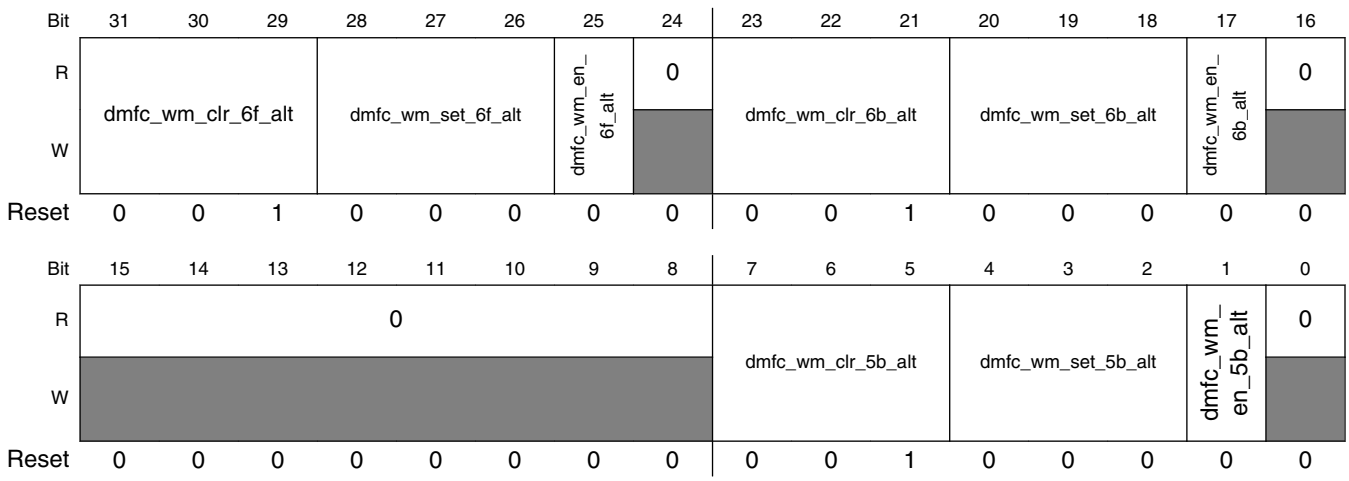
Table continues on the next page...

IPU_DMFC_DP_CHAN_ALT field descriptions (continued)

Field	Description
	110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
18–16 dmfc_st_addr_6b_alt	DMFC Start Address for IDMAC's channel 24 (for alternate flow) This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 24. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7
15–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–6 dmfc_burst_size_5b_alt	Burst size of IDMAC's channel 23 (for alternate flow) This field defines the burst size of the IDMAC's channel 23 write accesses. This settings must match the settings in the corresponding IDMAC channel's settings. 00 32 words of 128 bit (4 pixels of 32 bit each, coming from the IDMAC) 01 16 words of 128 bit 10 8 words of 128 bit 11 4 words of 128 bit
5–3 dmfc_fifo_size_5b_alt	DMFC FIFO size for IDMAC's channel 23 (for alternate flow) This field defines the FIFO partition for the DMFC channel connected to IDMAC's channel 23 000 All (512X128 words) the DMFC's FIFO is allocated to this channel 001 256X128 words are allocated to this channel 010 128X128 words are allocated to this channel 011 64X128 words are allocated to this channel 100 32X128 words are allocated to this channel 101 16X128 words are allocated to this channel 110 8X128 words are allocated to this channel 111 4X128 words are allocated to this channel
2–0 dmfc_st_addr_5b_alt	DMFC Start Address for IDMAC's channel 23 (for alternate flow) This field defines the base address at the DMFC's FIFO of the partition allocated to the channel connected to IDMAC's channel 23. The FIFO is partitioned to 8 equal segments. Each segment is 64X128 words The value of this field is the number of the segment 000 Segment 0 001 Segment 1 111 Segment 7

45.51.354 DMFC Display Channel Definition Alternate Register (IPU_DMFC_DP_CHAN_DEF_ALT)

Address: IPU_DMFC_DP_CHAN_DEF_ALT is 1E00_0000h base + 6002Ch offset = 1E06_002Ch



IPU_DMFC_DP_CHAN_DEF_ALT field descriptions

Field	Description
31–29 dmfc_wm_clr_6f_alt	Watermark Clear (for alternate flow) This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
28–26 dmfc_wm_set_6f_alt	Watermark Set (for alternate flow) This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
25 dmfc_wm_en_6f_alt	Watermark enable. (for alternate flow) This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23–21 dmfc_wm_clr_6b_alt	Watermark Clear (for alternate flow) This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
20–18 dmfc_wm_set_6b_alt	Watermark Set (for alternate flow) This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set

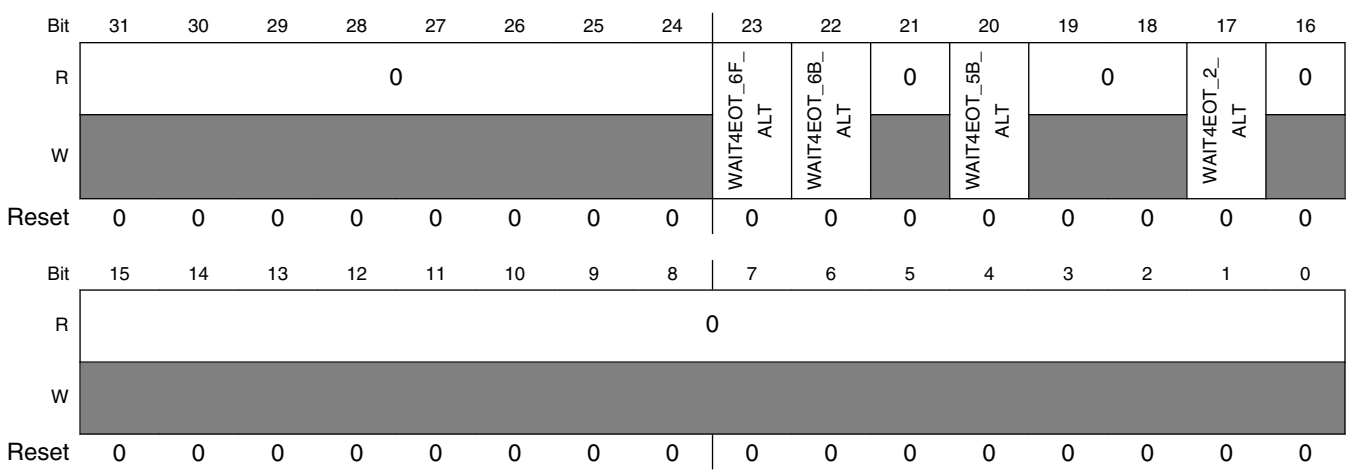
Table continues on the next page...

IPU_DMFC_DP_CHAN_DEF_ALT field descriptions (continued)

Field	Description
17 dmfc_wm_en_6b_alt	Watermark enable. (for alternate flow) This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
16–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–5 dmfc_wm_clr_5b_alt	Watermark Clear (for alternate flow) This field defines the watermark's level of the DMFC's write FIFO. Crossing this level will clear the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set)
4–2 dmfc_wm_set_5b_alt	Watermark Set (for alternate flow) This field defines the watermark's level of the DMFC write FIFO. Crossing this level will send the watermark signal to the IDMAC. The WM level is the amount of occupied bursts at the FIFO (dmfc_wm_clr > dmfc_wm_set
1 dmfc_wm_en_5b_alt	Watermark enable. (for alternate flow) This bit enables the watermark feature of the FIFO 1 WM feature is enabled 0 WM feature is disabled
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

45.51.355 DMFC General 1 Alternate Register (IPU_DMFC_GENERAL1_ALT)

Address: IPU_DMFC_GENERAL1_ALT is 1E00_0000h base + 60030h offset = 1E06_0030h



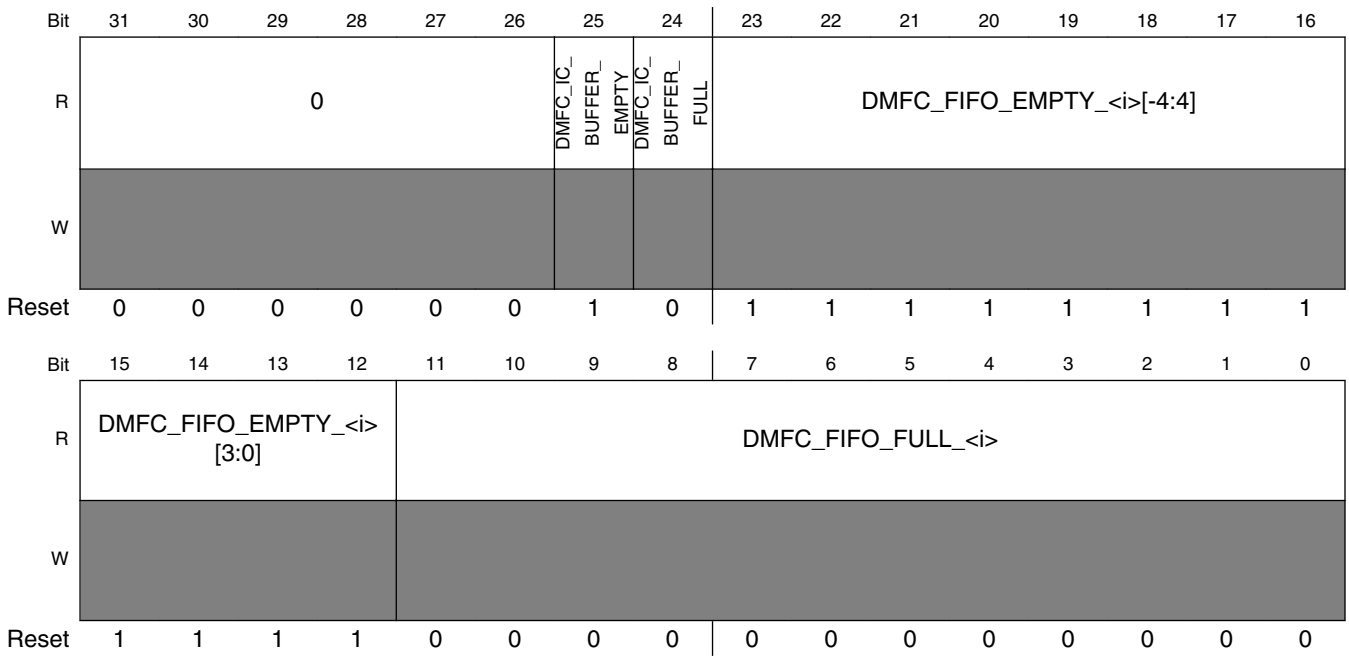
IPU_DMFC_GENERAL1_ALT field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved.
23 WAIT4EOT_6F_ALT	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #6F is in wait4eot mode (for alternate flow) 0 FIFO #6F is in normal mode (for alternate flow)
22 WAIT4EOT_6B_ALT	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #6B is in wait4eot mode (for alternate flow) 0 FIFO #6B is in normal mode (for alternate flow)
21 Reserved	This read-only field is reserved and always has the value zero. Reserved
20 WAIT4EOT_5B_ALT	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #5B is in wait4eot mode (for alternate flow) 0 FIFO #5B is in normal mode (for alternate flow)
19–18 Reserved	This read-only field is reserved and always has the value zero. Reserved
17 WAIT4EOT_2_ALT	In normal operation the DMFC sends requests to the IDMAC whenever there is room in the FIFO. When this bit is set the DMFC sends the request only after the current transfer is terminated, (wait4eot mode) When the FIFO is larger then the size of the line, the user should work in wait4eot mode. 1 FIFO #2 is in wait4eot mode (for alternate flow) 0 FIFO #2 is in normal mode (for alternate flow)
16–0 Reserved	This read-only field is reserved and always has the value zero. Reserved.

45.51.356 DMFC Status Register (IPU_DMFC_STAT)

This register contains DMFC's status bits. All the bits in this register are read-only.

Address: IPU_DMFC_STAT is 1E00_0000h base + 60034h offset = 1E06_0034h



IPU_DMFC_STAT field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value zero. Reserved
25 DMFC_IC_BUFFER_EMPTY	This bit indicates on a IC FIFO, inside the DMFC, empty condition. 0 IC FIFO not empty 1 IC FIFO is empty
24 DMFC_IC_BUFFER_FULL	This bit indicates on a IC FIFO, inside the DMFC, full condition. 0 IC FIFO not full 1 IC FIFO is full
23–12 DMFC_FIFO_EMPTY_<i>	This bit indicates on a DMFC FIFO#<i> empty condition. Mapping of these bit to an actual FIFO number is as follows: bit 0 => 0 bit 1=> 1 bit 2 => 2 bit 3 =>1c bit 4 => 2c bit 5 => 5b bit 6 => 5f bit 7 => 6b bit 8 => 6f bit 9 => 9 bit 10 => 10 (ARM platform access) bit 11 => 11 (ARM platform access) 0 FIFO #<i> is not empty 1 FIFO #<i> is empty
11–0 DMFC_FIFO_FULL_<i>	This bit indicates on a DMFC FIFO#<i> full condition.

Table continues on the next page...

IPU_DMFC_STAT field descriptions (continued)

Field	Description
	Mapping of these bit to an actual FIFO number is as follows: bit 0 => 0 bit 1=> 1 bit 2 => 2 bit 3 =>1c bit 4 => 2c bit 5 => 5b bit 6 => 5f bit 7 => 6b bit 8 => 6f bit 9 => 9 bit 10 => 10 (ARM platform access) bit 11 => 11 (ARM platform access) 0 FIFO #<i> is not full 1 FIFO #<i> is full

45.51.357 VDI Field Size Register (IPU_VDI_FSIZE)

The register used to control size of VDIC input fields.

Address: IPU_VDI_FSIZE is 1E00_0000h base + 68000h offset = 1E06_8000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					VDI_FHEIGHT											0					VDI_FWIDTH										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_VDI_FSIZE field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–16 VDI_FHEIGHT	Frame height The value to be written to this register is the frame's height minus 1. The frame height should not be smaller than 16. When VDI_CMB_EN bit is clear: • The frame height should not be greater than 1080. • The frame's height must be even (which means that both fields have the same height) • The frame's height in 4:2:0 format, must be multiple of 4 (which means that both chroma fields have the same height) When VDI_CMB_EN bit is set: • The frame height should not be greater than 1200.
15–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–0 VDI_FWIDTH	Frame width. The value to be written to this register is the frame's width minus 1. The Frame width should not be smaller than 16. The width must be even. When VDI_CMB_EN bit is clear • The Frame width should not be greater than 720968.

Table continues on the next page...

IPU_VDI_FSIZE field descriptions (continued)

Field	Description
	When VDI_CMB_EN bit is set: • The Frame width should not be greater than 1920.

45.51.358 VDI Control Register (IPU_VDI_C)

The register used to control modes of operations of VDIC module.

Address: IPU_VDI_C is 1E00_0000h base + 68004h offset = 1E06_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-	-	0		VDI_VWM3_CLR				VDI_VWM3_SET		VDI_VWM1_CLR			VDI_VWM1_SET		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VDI_BURST_SIZE3				VDI_BURST_SIZE2				VDI_BURST_SIZE1				VDI_MOT_SEL		VDI_CH_422	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_VDI_C field descriptions

Field	Description
31 -	VDIC top filed (automatic) This defines what would be the top field to be processed when the data is coming from the CSI 0 top field is field 0 1 top field is field 1
30 -	VDIC top filed (manual) This defines what would be the next top field to be processed when the data is coming from the memory 0 top field is field 0 1 top field is field 1
29–28 Reserved	This read-only field is reserved and always has the value zero. Reserved
27–25 VDI_VWM3_CLR	VDIC WaterMark "clear" level for channel 3. 0 clear watermark level when FIFO3 is full on 1/8 of their size.

Table continues on the next page...

IPU_VDI_C field descriptions (continued)

Field	Description
	1 clear watermark level when FIFO3 is full on 2/8 of their size. 7 clear watermark level when FIFO3 is full.
24–22 VDI_VWM3_ SET	VDIC WaterMark "set" level for channel 3. 0 set watermark level when FIFO3 is full on 1/8 of their size. 1 set watermark level when FIFO3 is full on 2/8 of their size. 7 set watermark level when FIFO3 is full.
21–19 VDI_VWM1_ CLR	VDIC WaterMark "clear" level for channel 1 or channel 4 (channels 1 and 4 are not working simultaneously). 0 clear watermark level when FIFO1 is full on 1/8 of their size. 1 clear watermark level when FIFO1 is full on 2/8 of their size. 7 clear watermark level when FIFO1 is full.
18–16 VDI_VWM1_ SET	VDIC WaterMark "set" level for channel 1 or channel 2 (channels 1 and 4 are not working simultaneously). 0 set watermark level when FIFO1 is full on 1/8 of their size. 1 set watermark level when FIFO1 is full on 2/8 of their size. 7 set watermark level when FIFO1 is full.
15–12 VDI_BURST_ SIZE3	Burst Size for channel 3. The VDIC's burst size is restrict by the IDMAC's restriction - This value must match the IDMAC settings and follow the IDMAC's restrictions 0 Burst size is 1 access. 1 Burst size is 2 accesses. 15 Burst size is 16 accesses.
11–8 VDI_BURST_ SIZE2	Burst Size for channel 2. The VDIC's burst size is restrict by the IDMAC's restriction - This value must match the IDMAC settings and follow the IDMAC's restrictions 0 Burst size is 1 access. 1 Burst size is 2 accesses. 15 Burst size is 16 accesses.
7–4 VDI_BURST_ SIZE1	Burst Size for channels 1 or 4 (channels 1 and 4 are not working simultaneously). The VDIC's burst size is restrict by the IDMAC's restriction - This value must match the IDMAC settings and follow the IDMAC's restrictions 0 Burst size is 1 access. 1 Burst size is 2 accesses. 15 Burst size is 16 accesses.
3–2 VDI_MOT_SEL	Motion select. 0 Motion determined by ROM "1" (shared toward medium/high motion). 1 Motion determined by ROM "2" (This option will not work well for high motion). 2 Full motion, only vertical filter is used 3 Forbidden.

Table continues on the next page...

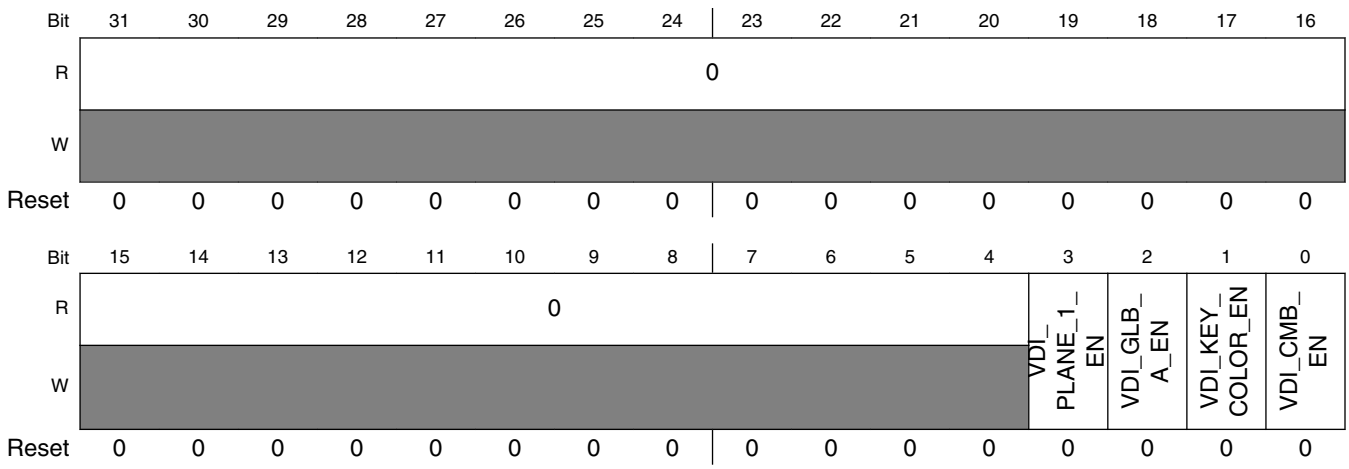
IPU_VDI_C field descriptions (continued)

Field	Description
1 VDI_CH_422	Chroma format at input and output of VDIC. 0 Chroma format is 420. 1 Chroma format is 422.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

45.51.359 VDI Control Register 2 (IPU_VDI_C2_)

The register used to control modes of operations of VDIC module.

Address: IPU_VDI_C2_ is 1E00_0000h base + 68008h offset = 1E06_8008h



IPU_VDI_C2_ field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value zero. Reserved
3 VDI_PLANE_1_EN	Plane 1 enable 0 plane #1 is disabled 1 plane #1 is enabled
2 VDI_GLB_A_EN	Global alpha enable 0 Alpha is local 1 Alpha is global
1 VDI_KEY_COLOR_EN	Key Color Enable 0 Key Color disabled. 1 Key color enabled

Table continues on the next page...

IPU_VDI_C2_ field descriptions (continued)

Field	Description
0 VDI_CMB_EN	Combining enable 0 Combining disabled. The VDIC works in de-interlacing mode 1 Combining enabled. The de-interlacing mode is not functional

45.51.360 VDI Combining Parameters Register 1 (IPU_VDI_CMDP_1)

The register holds combining paramemters.

Address: IPU_VDI_CMDP_1 is 1E00_0000h base + 6800Ch offset = 1E06_800Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VDI_ALPHA								VDI_KEY_COLOR_R								VDI_KEY_COLOR_G								VDI_KEY_COLOR_B							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_VDI_CMDP_1 field descriptions

Field	Description
31–24 VDI_ALPHA	Global Alpha Actual value of the alpha is VDI_ALPHA + VDI_ALPHA[7]
23–16 VDI_KEY_ COLOR_R	Red component of Key Color
15–8 VDI_KEY_ COLOR_G	Green component of Key Color
7–0 VDI_KEY_ COLOR_B	Blue component of Key Color

45.51.361 VDI Combining Parameters Register 2 (IPU_VDI_CMDP_2)

The register holds combining paramemters.

Address: IPU_VDI_CMDP_2 is 1E00_0000h base + 68010h offset = 1E06_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								VDI_KEY_COLOR_R								VDI_KEY_COLOR_G								VDI_KEY_COLOR_B							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

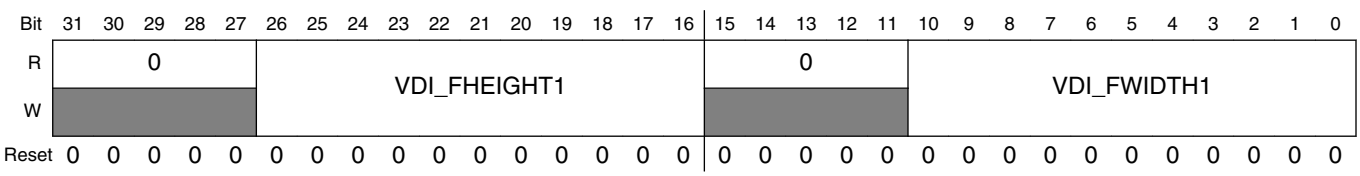
IPU_VDI_CMDP_2 field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23–16 VDI_KEY_ COLOR_R	Red component of background Color
15–8 VDI_KEY_ COLOR_G	Green component of background Color
7–0 VDI_KEY_ COLOR_B	Blue component of background Color

45.51.362 VDI Plane Size Register 1 (IPU_VDI_PS_1)

The register holds the plane size's paramemters.

Address: IPU_VDI_PS_1 is 1E00_0000h base + 68014h offset = 1E06_8014h



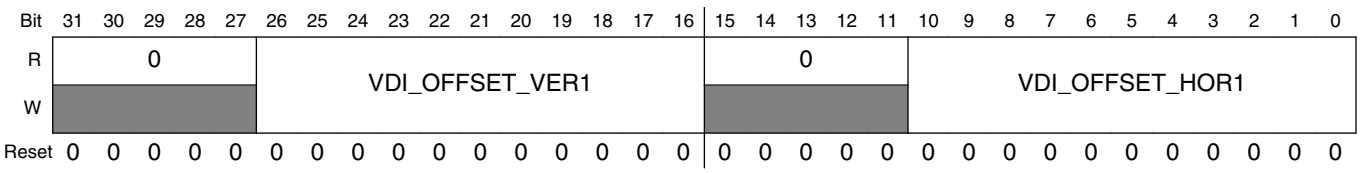
IPU_VDI_PS_1 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26–16 VDI_FHEIGHT1	Plane 1 height The value to be written to this register is the plane's height minus 1. The Plane height should not be smaller than 16. The Plane height should not be greater than 1200. The Plane's height must be even
15–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–0 VDI_FWIDTH1	Plane 1 width. The value to be written to this register is the plane's width minus 1. The Plane width should not be smaller than 16. The Plane width should not be greater than 1920. The width must be even.

45.51.363 VDI Plane Size Register 2 (IPU_VDI_PS_2)

The register holds the plane's offset paramemters.

Address: IPU_VDI_PS_2 is 1E00_0000h base + 68018h offset = 1E06_8018h



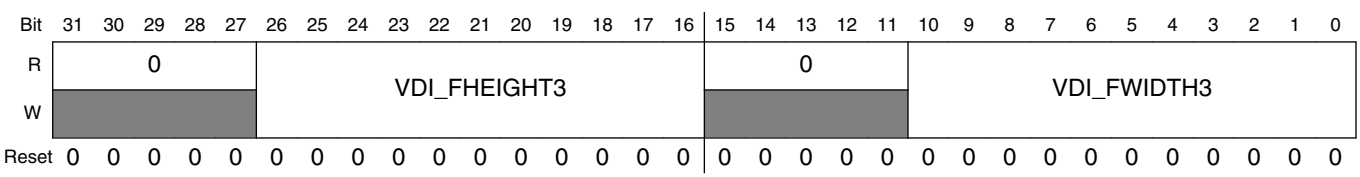
IPU_VDI_PS_2 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26–16 VDI_OFFSET_VER1	Vertical offset of plane 1
15–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–0 VDI_OFFSET_HOR1	Horizontal offset of plane 1

45.51.364 VDI Plane Size Register 3 (IPU_VDI_PS_3)

The register holds the plane size's paramemters.

Address: IPU_VDI_PS_3 is 1E00_0000h base + 6801Ch offset = 1E06_801Ch



IPU_VDI_PS_3 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26–16 VDI_FHEIGHT3	Plane 3 height The value to be written to this register is the plane's height minus 1. The Plane height should not be smaller than 16. The Plane height should not be greater than 1200.

Table continues on the next page...

IPU_VDI_PS_3 field descriptions (continued)

Field	Description
	The Plane's height must be even
15–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–0 VDI_FWIDTH3	Plane 3 width. The value to be written to this register is the plane's width minus 1. The Plane width should not be smaller than 16. The Plane width should not be greater than 1920. The width must be even.

45.51.365 VDI Plane Size Register 4 (IPU_VDI_PS_4)

The register holds the plane's offset paramemters.

Address: IPU_VDI_PS_4 is 1E00_0000h base + 68020h offset = 1E06_8020h

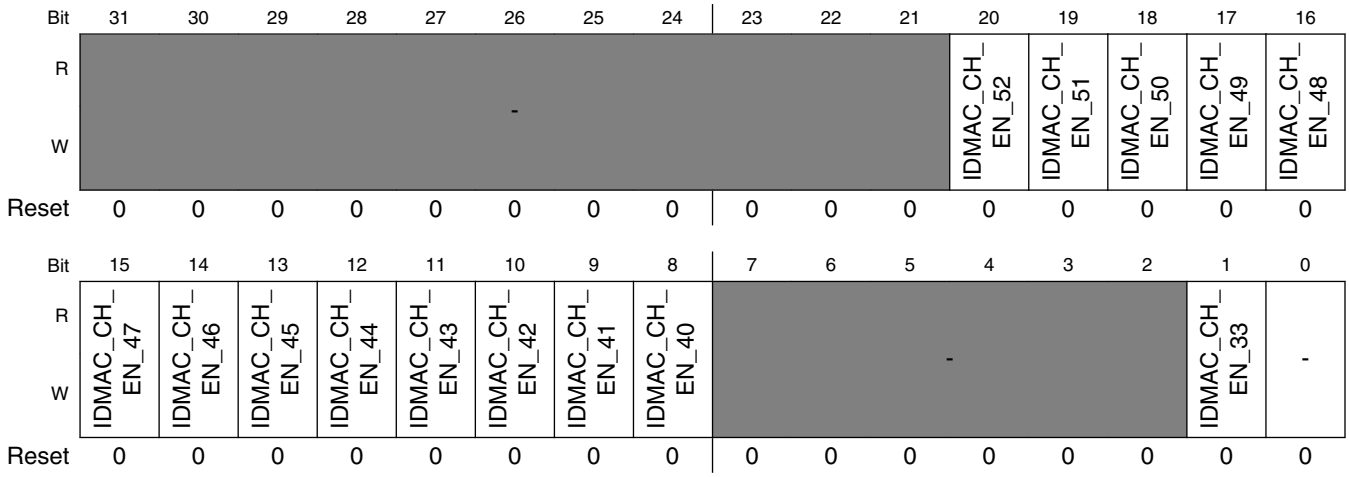
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					VDI_OFFSET_VER3											0					VDI_OFFSET_HOR3										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_VDI_PS_4 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26–16 VDI_OFFSET_ VER3	Vertical offset of plane 3
15–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–0 VDI_OFFSET_ HOR3	Horizontal offset of plane 3

45.51.366 IDMAC Channel Enable 2 Register (IPU_IDMAC_CH_EN_2)

Address: IPU_IDMAC_CH_EN_2 is 1E00_0000h base + 80008h offset = 1E08_0008h



IPU_IDMAC_CH_EN_2 field descriptions

Field	Description
31–21 -	Reserved.
20 IDMAC_CH_EN_52	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
19 IDMAC_CH_EN_51	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
18 IDMAC_CH_EN_50	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
17 IDMAC_CH_EN_49	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
16 IDMAC_CH_EN_48	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
15 IDMAC_CH_EN_47	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled

Table continues on the next page...

IPU_IDMAC_CH_EN_2 field descriptions (continued)

Field	Description
14 IDMAC_CH_ EN_46	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
13 IDMAC_CH_ EN_45	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
12 IDMAC_CH_ EN_44	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
11 IDMAC_CH_ EN_43	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
10 IDMAC_CH_ EN_42	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
9 IDMAC_CH_ EN_41	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
8 IDMAC_CH_ EN_40	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
7–2 -	Reserved.
1 IDMAC_CH_ EN_33	IDMAC Channel enable bit [i] 0 IDMAC channel is disabled 1 IDMAC channel is enabled
0 -	Reserved.

45.51.367 IDMAC Channel Lock Enable 2Register (IPU_IDMAC_LOCK_EN_2)

Address: IPU_IDMAC_LOCK_EN_2 is 1E00_0000h base + 80248h offset = 1E08_0248h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	-															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-				IDMAC_LOCK_50		IDMAC_LOCK_49		IDMAC_LOCK_48		IDMAC_LOCK_47		IDMAC_LOCK_46		IDMAC_LOCK_45	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_IDMAC_LOCK_EN_2 field descriptions

Field	Description
31–12 -	Reserved
11–10 IDMAC_LOCK_50	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
9–8 IDMAC_LOCK_49	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
7–6 IDMAC_LOCK_48	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
5–4 IDMAC_LOCK_47	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.

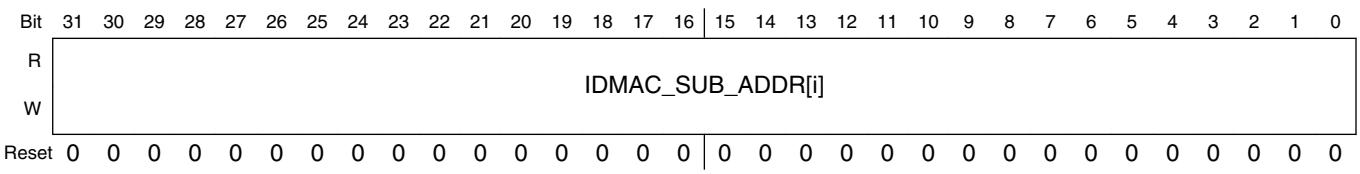
Table continues on the next page...

IPU_IDMAC_LOCK_EN_2 field descriptions (continued)

Field	Description
3–2 IDMAC_LOCK_46	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.
1–0 IDMAC_LOCK_45	IDMAC lock bits for channel [i] 00 The lock feature is disabled. The IDMAC will generate one AXI burst upon the assertion of the DMA request. 01 The IDMAC will generate two AXI bursts upon the assertion of the DMA request. 10 The IDMAC will generate four AXI bursts upon the assertion of the DMA request. 11 The IDMAC will generate eight AXI bursts upon the assertion of the DMA request.

45.51.368 IDMAC Channel Alternate Address 0 Register (IPU_IDMAC_SUB_ADDR_0)

Address: IPU_IDMAC_SUB_ADDR_0 is 1E00_0000h base + 8028Ch offset = 1E08_028Ch

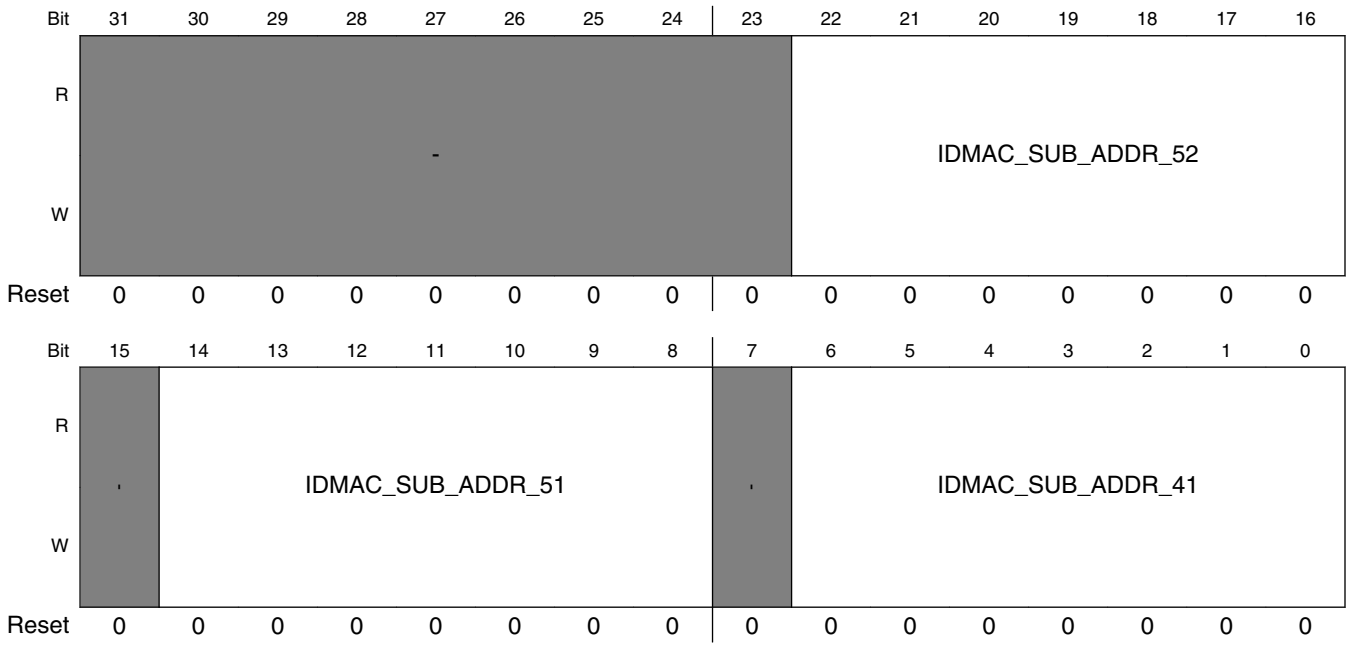


IPU_IDMAC_SUB_ADDR_0 field descriptions

Field	Description
31–0 IDMAC_SUB_ADDR[i]	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR[i]. The user must set each IDMAC_SUB_ADDR[i] field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.

45.51.369 IDMAC Channel Alternate Address 2 Register (IPU_IDMAC_SUB_ADDR_2)

Address: IPU_IDMAC_SUB_ADDR_2 is 1E00_0000h base + 80304h offset = 1E08_0304h

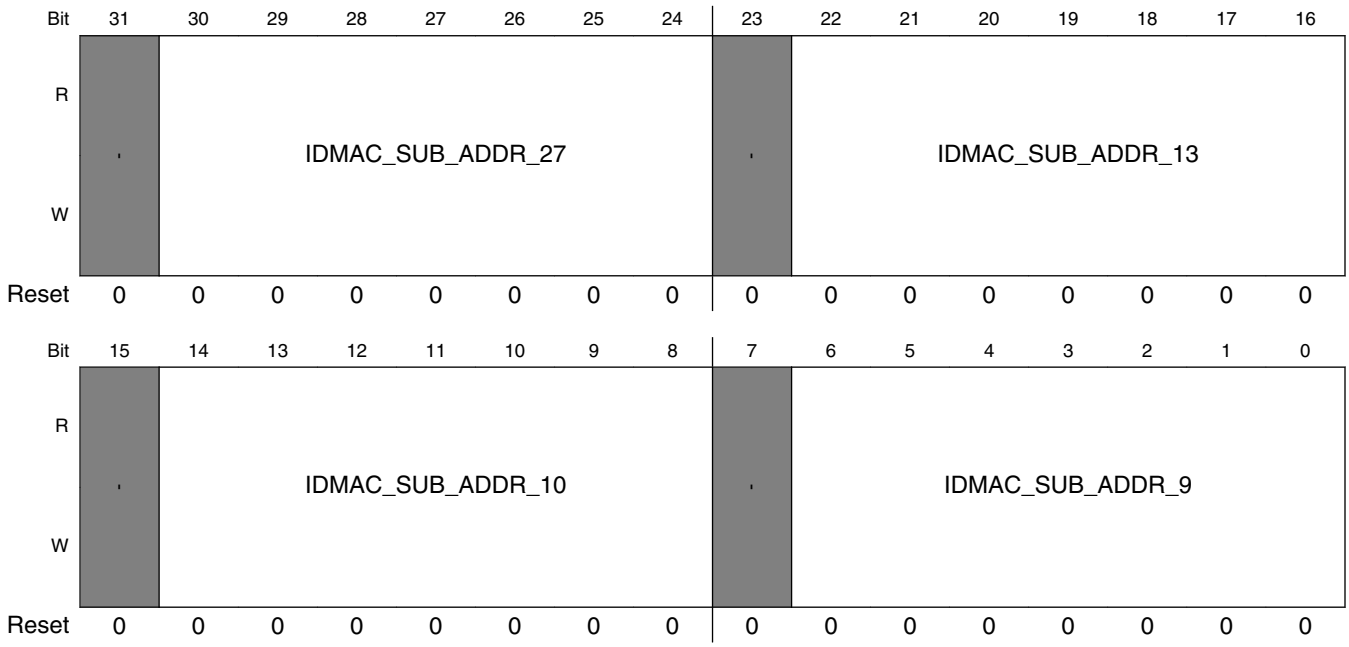


IPU_IDMAC_SUB_ADDR_2 field descriptions

Field	Description
31–23 -	Reserved.
22–16 IDMAC_SUB_ADDR_52	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR[i]. The user must set each IDMAC_SUB_ADDR[i] field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
15 -	Reserved.
14–8 IDMAC_SUB_ADDR_51	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR[i]. The user must set each IDMAC_SUB_ADDR[i] field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
7 -	Reserved.
6–0 IDMAC_SUB_ADDR_41	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR[i]. The user must set each IDMAC_SUB_ADDR[i] field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.

45.51.370 IDMAC Channel Alternate Address 3 Register (IPU_IDMAC_SUB_ADDR_3)

Address: IPU_IDMAC_SUB_ADDR_3 is 1E00_0000h base + 80308h offset = 1E08_0308h

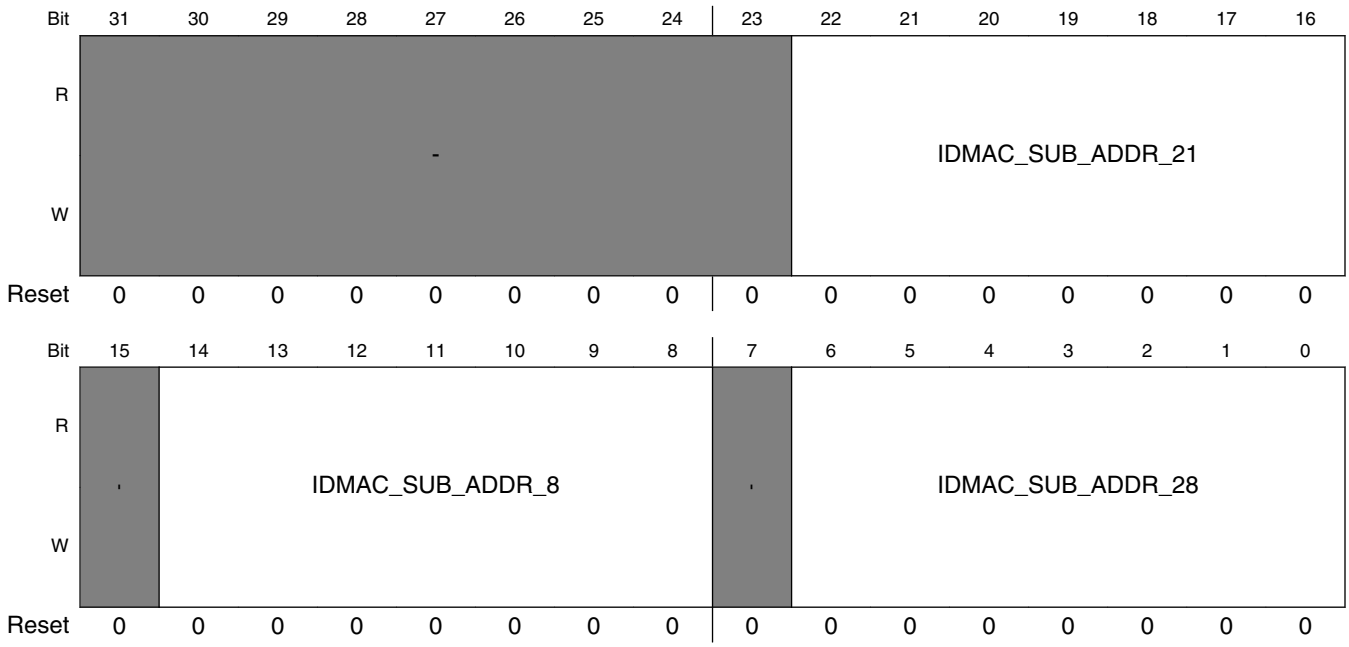


IPU_IDMAC_SUB_ADDR_3 field descriptions

Field	Description
31 -	Reserved.
30–24 IDMAC_SUB_ADDR_27	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR[i]. The user must set each IDMAC_SUB_ADDR[i] field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
23 -	Reserved.
22–16 IDMAC_SUB_ADDR_13	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR[i]. The user must set each IDMAC_SUB_ADDR[i] field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
15 -	Reserved.
14–8 IDMAC_SUB_ADDR_10	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR[i]. The user must set each IDMAC_SUB_ADDR[i] field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
7 -	Reserved.
6–0 IDMAC_SUB_ADDR_9	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR[i]. The user must set each IDMAC_SUB_ADDR[i] field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.

45.51.371 IDMAC Channel Alternate Address 4 Register (IPU_IDMAC_SUB_ADDR_4)

Address: IPU_IDMAC_SUB_ADDR_4 is 1E00_0000h base + 8030Ch offset = 1E08_030Ch

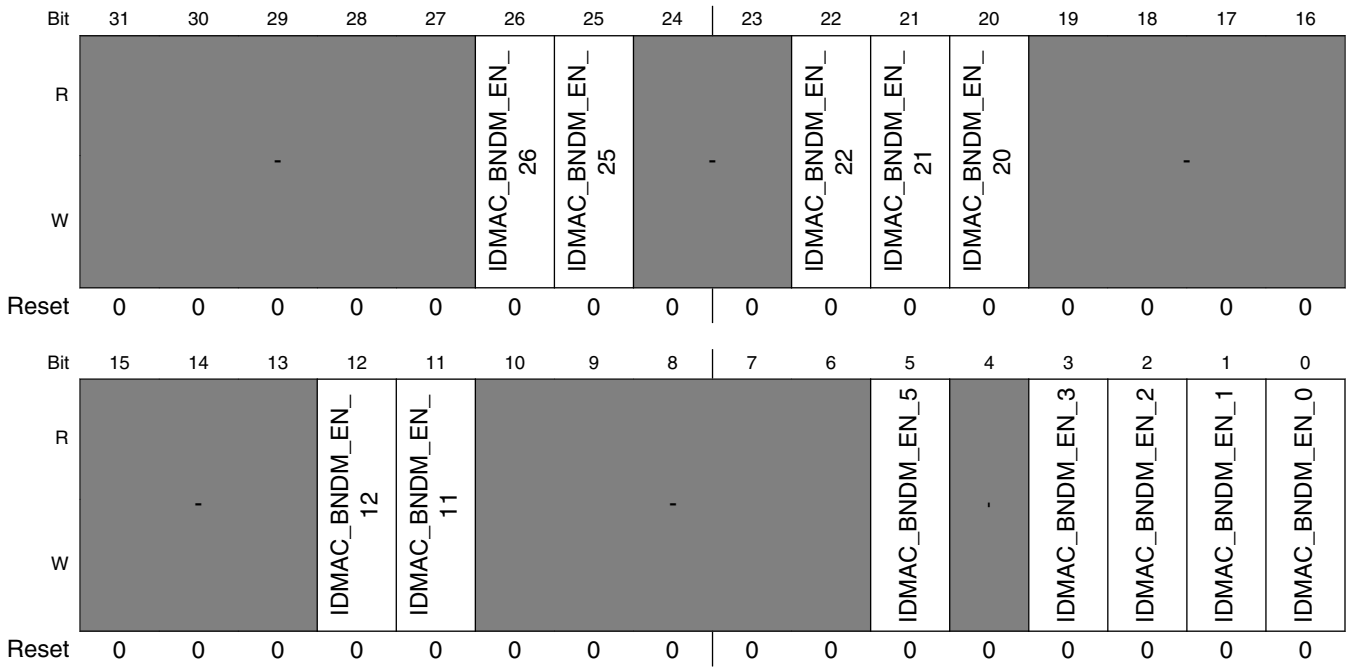


IPU_IDMAC_SUB_ADDR_4 field descriptions

Field	Description
31–23 -	Reserved.
22–16 IDMAC_SUB_ADDR_21	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR[i]. The user must set each IDMAC_SUB_ADDR[i] field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
15 -	Reserved.
14–8 IDMAC_SUB_ADDR_8	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR[i]. The user must set each IDMAC_SUB_ADDR[i] field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
7 -	Reserved.
6–0 IDMAC_SUB_ADDR_28	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR[i]. The user must set each IDMAC_SUB_ADDR[i] field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.

45.51.372 IDMAC Band Mode Enable 1 Register (IPU_IDMAC_BNDM_EN_1)

Address: IPU_IDMAC_BNDM_EN_1 is 1E00_0000h base + 80440h offset = 1E08_0440h



IPU_IDMAC_BNDM_EN_1 field descriptions

Field	Description
31–27 -	Reserved.
26 IDMAC_BNDM_EN_26	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
25 IDMAC_BNDM_EN_25	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
24–23 -	Reserved.

Table continues on the next page...

IPU_IDMAC_BNDM_EN_1 field descriptions (continued)

Field	Description
22 IDMAC_BNDM_EN_22	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
21 IDMAC_BNDM_EN_21	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
20 IDMAC_BNDM_EN_20	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
19–13 -	Reserved.
12 IDMAC_BNDM_EN_12	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
11 IDMAC_BNDM_EN_11	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
10–6 -	Reserved.
5 IDMAC_BNDM_EN_5	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration.

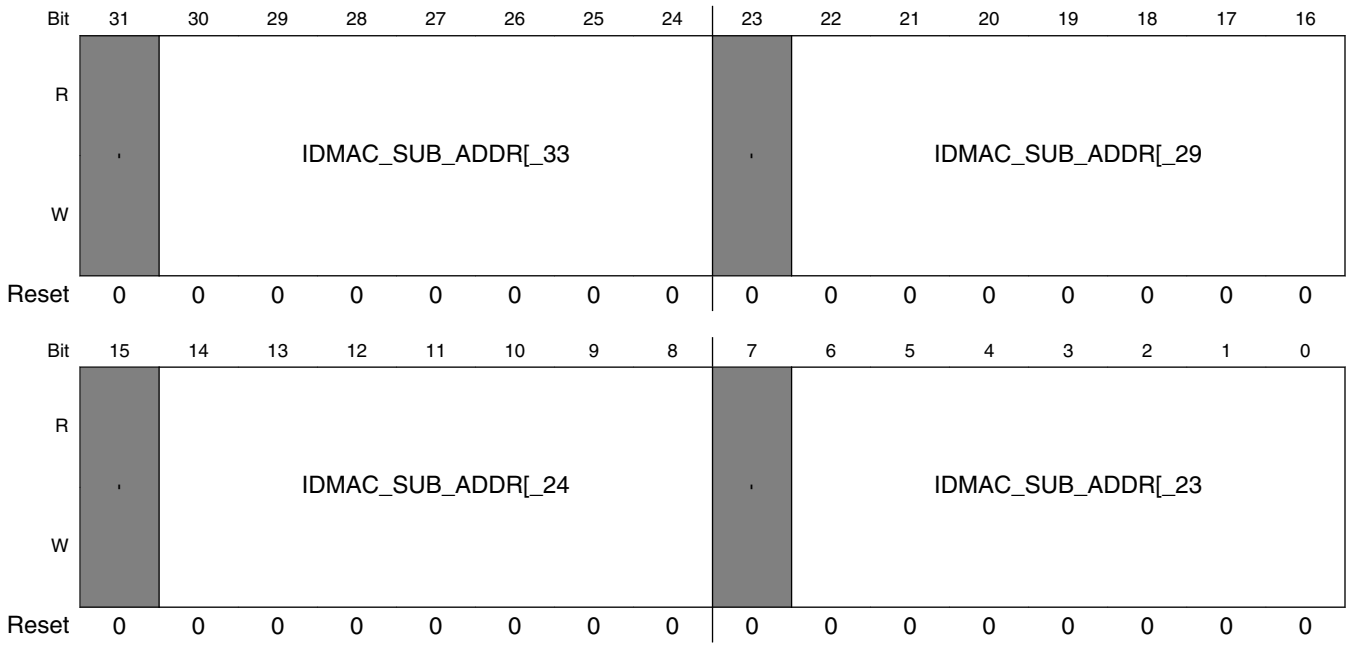
Table continues on the next page...

IPU_IDMAC_BNDM_EN_1 field descriptions (continued)

Field	Description
	0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
4 -	Reserved.
3 IDMAC_BNDM_EN_3	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
2 IDMAC_BNDM_EN_2	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
1 IDMAC_BNDM_EN_1	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
0 IDMAC_BNDM_EN_0	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode

45.51.373 IDMAC Channel Alternate Address 1 Register (IPU_IDMAC_SUB_ADDR_1)

Address: IPU_IDMAC_SUB_ADDR_1 is 1E00_0000h base + 802C30h offset = 1E80_2C30h



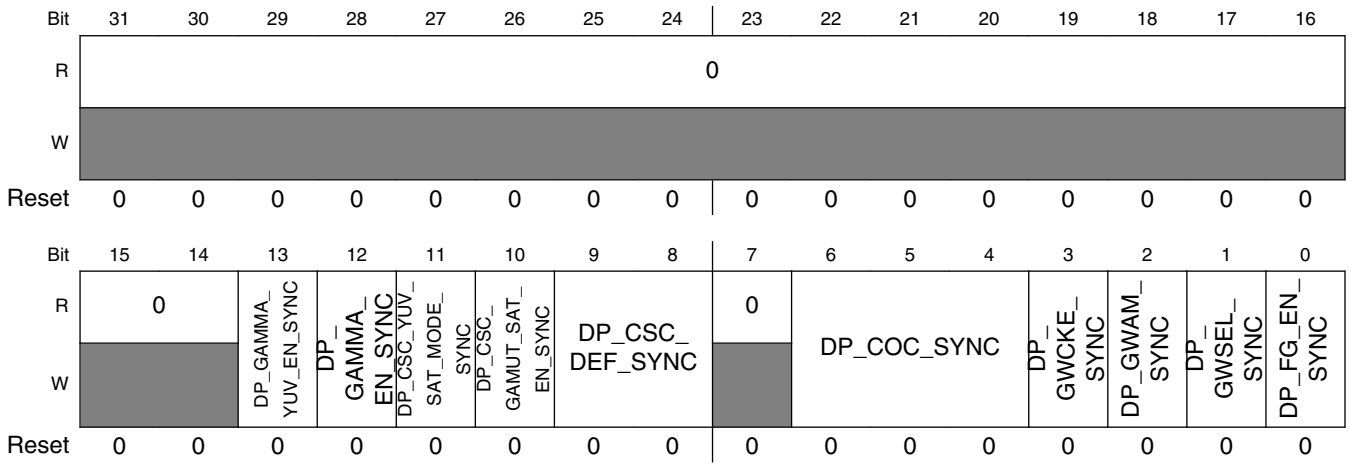
IPU_IDMAC_SUB_ADDR_1 field descriptions

Field	Description
31 -	Reserved.
30–24 IDMAC_SUB_ADDR[_33	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR[i]. The user must set each IDMAC_SUB_ADDR[i] field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
23 -	Reserved.
22–16 IDMAC_SUB_ADDR[_29	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR[i]. The user must set each IDMAC_SUB_ADDR[i] field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
15 -	Reserved.
14–8 IDMAC_SUB_ADDR[_24	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR[i]. The user must set each IDMAC_SUB_ADDR[i] field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.
7 -	Reserved.
6–0 IDMAC_SUB_ADDR[_23	The CPMEM alternative entry [i] holds the parameters of the channel that is number appears in IDMAC_SUB_ADDR[i]. The user must set each IDMAC_SUB_ADDR[i] field to a unique channel. Alternative CPMEM entry is relevant only for channels supporting alternate flows.

45.51.374 DP Common Configuration Sync Flow Register (IPU_DP_COM_CONF_SYNC)

This register contains common configuration parameters for the DP.

Address: IPU_DP_COM_CONF_SYNC is 1E00_0000h base + 1040000h offset = 1F04_0000h



IPU_DP_COM_CONF_SYNC field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 DP_GAMMA_YUV_EN_SYNC	GAMMA's YUV mode enable for sync flow 0 YUV mode is OFF. 1 YUV mode is ON.
12 DP_GAMMA_EN_SYNC	GAMMA_EN - Gamma correction block enable bit 0 disable 1 enable
11 DP_CSC_YUV_SAT_MODE_SYNC	CSC_YUV_SAT_MODE YUV saturation mode for color space conversion 0 Y/U/V range 0 -255 1 Y range 16-235, U/V range 16-240
10 DP_CSC_GAMUT_SAT_EN_SYNC	CSC_GAMUT_SAT_EN Indicate if GAMUT saturation is enabled 0 disable GAMUT mapping 1 enable GAMUT mapping
9–8 DP_CSC_DEF_SYNC	CSC_DEF Enable or disable Color Space Conversion. 00 CSC disable

Table continues on the next page...

IPU_DP_COM_CONF_SYNC field descriptions (continued)

Field	Description
	01 CSC enable after combining 10 CSC enable before combining on BG channel 11 CSC enable before combining on FG channel
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–4 DP_COC_SYNC	COC - Cursor Operation Control Controls the format of the cursor and the type of arithmetic operations 000 Transparent, cursor is disabled. 001 Full cursor. 010 Reversed cursor. 011 AND between full plane and cursor. 100 Reserved 101 OR between full plane and cursor. 110 XOR between full plane and cursor. 111 Reserved.
3 DP_GWCKE_ SYNC	GWCKE - Graphic Window Color Keying Enable Enable or disable graphic window color keying. 1 Enable color keying of graphic window 0 Disable color keying of graphic window
2 DP_GWAM_ SYNC	GWAM - Graphic Window Alpha Mode Select the use of Alpha to be global or local. 1 Global Alpha. 0 Local Alpha.
1 DP_GWSEL_ SYNC	GWSEL - Graphic Window Select Select graphic window to be on partial plane or full plane. 1 Graphic window is partial plane. 0 Graphic window is full plane.
0 DP_FG_EN_ SYNC	FG_EN - partial plane Enable. This bit enables the partial plane channel. 1 partial plane channel is enabled. 0 partial plane channel is disabled.

45.51.375 DP Graphic Window Control Sync Flow Register (IPU_DP_Graph_Wind_CTRL_SYNC)

This register contains common configuration parameters for the DP.

Address: IPU_DP_Graph_Wind_CTRL_SYNC is 1E00_0000h base + 1040004h offset = 1F04_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DP_GWAV_SYNC								DP_GWCKR_SYNC								DP_GWCKG_SYNC								DP_GWCKB_SYNC							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_Graph_Wind_CTRL_SYNC field descriptions

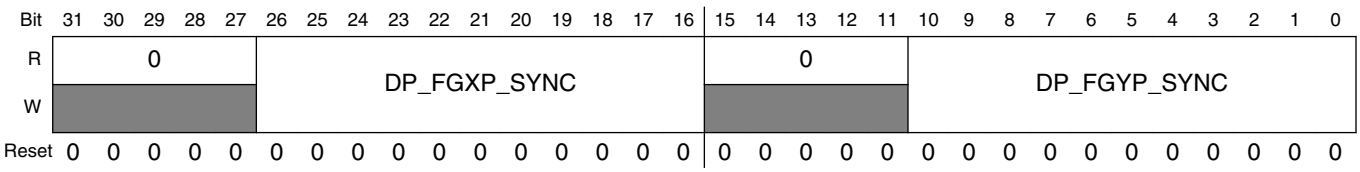
Field	Description
31–24 DP_GWAV_SYNC	GWAV - Graphic Window Alpha Value Defines the alpha value of graphic window used for alpha blending between graphic window and full plane. The Value the number that writing to GWAV register. The Actual Value the number, that insert to calculation in Combining Module. If Value = < 0.5- 1/256 (01111111) then Actual Value = Value. If Value >= 0.5 (10000000), then Actual Value = Value + 1/256. 00000000 Actual value is 00000000; Graphic window totally opaque i.e. overlay on LCD screen 01111111 Actual value is 01111111; 10000000 Actual value is 10000001 10000001 Actual value is 10000010 11111110 Actual value is 11111111 11111111 Actual value is 100000000;Graphic window totally transparent i.e. not displayed on LCD screen
23–16 DP_GWCKR_SYNC	GWCKR - Graphic Window Color Keying Red Component Defines the red component of graphic window color keying. 00000000 No red 11111111 Full red
15–8 DP_GWCKG_SYNC	GWCKG - Graphic Window Color Keying Green Component Defines the green component of graphic window color keying. 00000000 No Green 11111111 Full Green
7–0 DP_GWCKB_SYNC	GWCKB - Graphic Window Color Keying Blue Component Defines the blue component of graphic window color keying. 00000000 No blue 11111111 Full blue

45.51.376 DP Partial Plane Window Position Sync Flow Register (IPU_DP_FG_POS_SYNC)

The DP partial plane Window Position Register is used to determine the starting position of the partial plane window relative to BG window position.

This Register is used for the synchronous flow only.

Address: IPU_DP_FG_POS_SYNC is 1E00_0000h base + 1040008h offset = 1F04_0008h



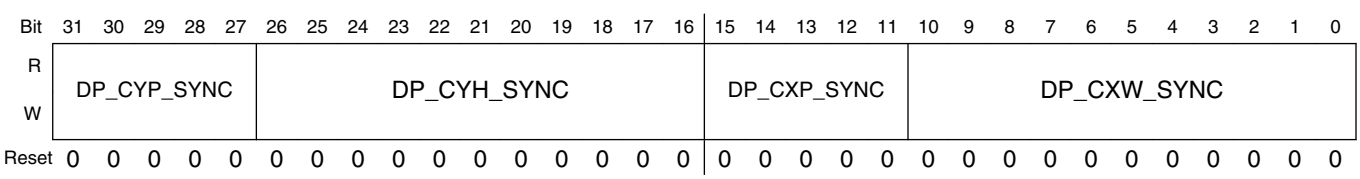
IPU_DP_FG_POS_SYNC field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–16 DP_FGXP_SYNC	FGXP partial plane Window X Position. partial plane Window X Position - Specifies the number of pixels between the start of full plane window X position and the beginning of the first data of new line.
15–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–0 DP_FGYP_SYNC	FGYP partial plane window Y position partial plane Window Y Position - Specifies the number of lines between the start of full plane windows Y position and the beginning of the first data.

45.51.377 DP Cursor Position and Size Sync Flow Register (IPU_DP_CUR_POS_SYNC)

The LCD Cursor Position Register is used to determine the starting position of the cursor relative to BG windows position.

Address: IPU_DP_CUR_POS_SYNC is 1E00_0000h base + 104000Ch offset = 1F04_000Ch



IPU_DP_CUR_POS_SYNC field descriptions

Field	Description
31–27 DP_CYP_SYNC	CYP - Cursor Y Position Represents the cursors vertical starting position Y in pixel count (from 0 to CYH).Live View Resolution Mode.
26–16 DP_CYH_SYNC	CYH - Cursor Height Specifies the height of the hardware cursor in pixels.
15–11 DP_CXP_SYNC	CXP - Cursor X Position Represents the cursors horizontal starting position X in pixel count (from 0 to CXW).
10–0 DP_CXW_SYNC	CXW - Cursor Width. Specifies the width of the hardware cursor in pixels.

45.51.378 DP Color Cursor Mapping Sync Flow Register (IPU_DP_CUR_MAP_SYNC)

The LCD Color Cursor Mapping Register defines the color of the cursor in passive or TFT color modes.

Address: IPU_DP_CUR_MAP_SYNC is 1E00_0000h base + 1040010h offset = 1F04_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DP_CUR_COL_B_SYNC								DP_CUR_COL_G_SYNC								DP_CUR_COL_R_SYNC							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_CUR_MAP_SYNC field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23–16 DP_CUR_COL_B_SYNC	CUR_COL_B - Cursor Blue Field Defines the Blue component of the cursor color in color mode 00000000 No Blue. 11111111 Full Blue.
15–8 DP_CUR_COL_G_SYNC	CUR_COL_G - Cursor Green Field Defines the Green component of the cursor color in color mode 00000000 No Green. 11111111 Full Green.
7–0 DP_CUR_COL_R_SYNC	CUR_COL_B - Cursor Red Field Defines the Red component of the cursor color in color mode

Table continues on the next page...

IPU_DP_CUR_MAP_SYNC field descriptions (continued)

Field	Description
	00000000 No Red. 11111111 Full Red.

45.51.379 DP Gamma Constants Sync Flow Register i (IPU_DP_GAMMA_C_SYNC_i)

This registers contains CONSTANTi parameters used for gamma correction inside the display processor (DP).

Address: IPU_DP_GAMMA_C_SYNC_i is 1E00_0000h base + 1040014h offset = 1F04_0014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							DP_GAMMA_C_SYNC_<2*i+1>									0								DP_GAMMA_C_SYNC_<2*i>								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_GAMMA_C_SYNC_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 DP_GAMMA_C_SYNC_<2*i+1>	CONSTANTi+1 parameter of Gamma Correction.
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 DP_GAMMA_C_SYNC_<2*i>	CONSTANTi parameter of Gamma Correction.

45.51.380 DP Gamma Correction Slope Sync Flow Register i (IPU_DP_GAMMA_S_SYNC_i)

This registers contains SLOPEi parameters used for Gamma Correction inside the display processor (DP).

Address: IPU_DP_GAMMA_S_SYNC_i is 1E00_0000h base + 1040034h offset = 1F04_0034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DP_GAMMA_S_SYNC_<4*i+3>								DP_GAMMA_S_SYNC_<4*i+2>								DP_GAMMA_S_SYNC_<4*i+1>								DP_GAMMA_S_SYNC_<4*i>								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_GAMMA_S_SYNC_i field descriptions

Field	Description
31–24 DP_GAMMA_S_SYNC_<4*i+3>	SLOPE<4*i+3> parameter of Gamma Correction.
23–16 DP_GAMMA_S_SYNC_<4*i+2>	SLOPE<4*i+2> parameter of Gamma Correction.
15–8 DP_GAMMA_S_SYNC_<4*i+1>	SLOPE<4*i+1> parameter of Gamma Correction.
7–0 DP_GAMMA_S_SYNC_<4*i>	SLOPE<4*i> parameter of Gamma Correction.

45.51.381 DP Color Space Conversion Control Sync Flow Registers (IPU_DP_CSCA_SYNC_i)

Address: IPU_DP_CSCA_SYNC_i is 1E00_0000h base + 1040044h offset = 1F04_0044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						DP_CSC_A_SYNC<2*i+1>										0						DP_CSC_A_SYNC<2*i>							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_CSCA_SYNC_i field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value zero. Reserved.
25–16 DP_CSC_A_SYNC<2*i+1>	A<2*i+1> parameter of color conversion
15–10 Reserved	This read-only field is reserved and always has the value zero. Reserved.
9–0 DP_CSC_A_SYNC<2*i>	A<2*i> parameter of color conversion.

45.51.382 DP Color Conversion Control Sync Flow Register 0 (IPU_DP_SCS_SYNC_0)

Address: IPU_DP_SCS_SYNC_0 is 1E00_0000h base + 1040054h offset = 1F04_0054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DP_CSC_S0_SYNC		DP_CSC_B0_SYNC														0						DP_CSC_A8_SYNC									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_SCS_SYNC_0 field descriptions

Field	Description
31–30 DP_CSC_S0_SYNC	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
29–16 DP_CSC_B0_SYNC	B0 parameter of color conversion.
15–10 Reserved	This read-only field is reserved and always has the value zero. Reserved.
9–0 DP_CSC_A8_SYNC	A9 parameter of color conversion.

45.51.383 DP Color Conversion Control Sync Flow Register 1 (IPU_DP_SCS_SYNC_1)

Address: IPU_DP_SCS_SYNC_1 is 1E00_0000h base + 1040058h offset = 1F04_0058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DP_CSC_S2_SYNC		DP_CSC_B2_SYNC														DP_CSC_S1_SYNC		DP_CSC_B1_SYNC													
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_SCS_SYNC_1 field descriptions

Field	Description
31–30 DP_CSC_S2_ SYNC	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
29–16 DP_CSC_B2_ SYNC	B0 parameter of color conversion.
15–14 DP_CSC_S1_ SYNC	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
13–0 DP_CSC_B1_ SYNC	B0 parameter of color conversion.

45.51.384 DP Cursor Position and Size Alternate Register (IPU_DP_CUR_POS_ALT)

The LCD Cursor Position Register is used to determine the starting position of the cursor relative to BG windows position for the alternative flow.

Address: IPU_DP_CUR_POS_ALT is 1E00_0000h base + 104005Ch offset = 1F04_005Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DP_CYP_SYNC_ALT					DP_CYH_SYNC_ALT											DP_CXP_SYNC_ALT					DP_CXW_SYNC_ALT										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_CUR_POS_ALT field descriptions

Field	Description
31–27 DP_CYP_ SYNC_ALT	CYP_ALT - Cursor Y Position Represents the cursors vertical starting position Y in pixel count (from 0 to CYH).Live View Resolution Mode for the alternative flow.
26–16 DP_CYH_ SYNC_ALT	CYH_ALT - Cursor Height Specifies the height of the hardware cursor in pixels.
15–11 DP_CXP_ SYNC_ALT	CXP_ALT - Cursor X Position

Table continues on the next page...

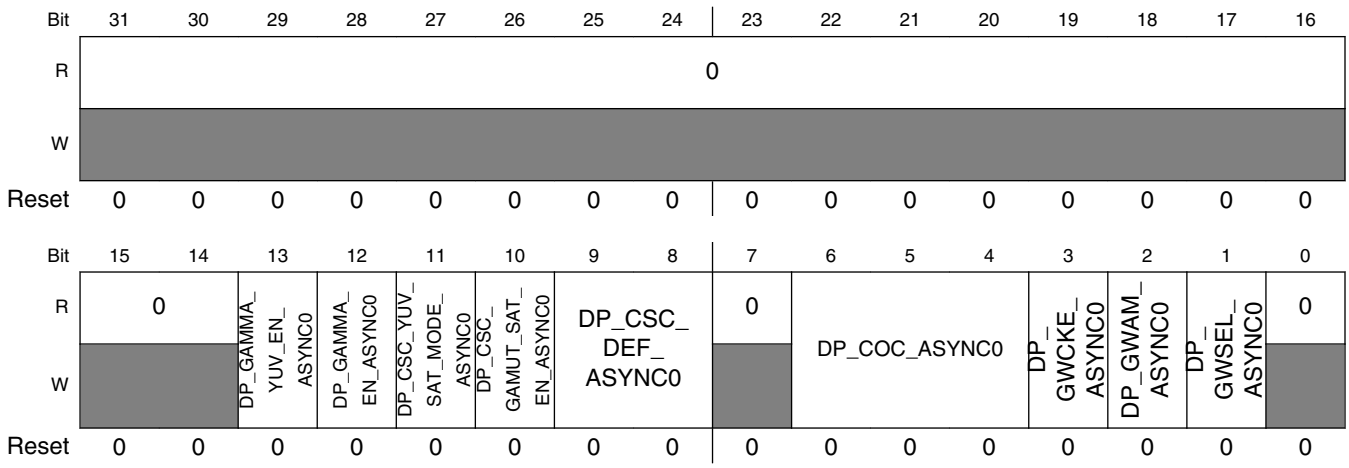
IPU_DP_CUR_POS_ALT field descriptions (continued)

Field	Description
	Represents the cursors horizontal starting position X in pixel count (from 0 to CXW) for the alternative flow.
10–0 DP_CXW_ SYNC_ALT	CXW_ALT - Cursor Width. Specifies the width of the hardware cursor in pixels for the alternative flow.

45.51.385 DP Common Configuration Async 0 Flow Register (IPU_DP_COM_CONF_ASYNC0)

This register contains common configuration parameters for the DP.

Address: IPU_DP_COM_CONF_ASYNC0 is 1E00_0000h base + 1040060h offset = 1F04_0060h



IPU_DP_COM_CONF_ASYNC0 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 DP_GAMMA_ YUV_EN_ ASYNC0	GAMMA's YUV mode enable for async flow 0 0 YUV mode is OFF. 1 YUV mode is ON.
12 DP_GAMMA_ EN_ASYNC0	GAMMA_EN - Gamma correction block enable bit 0 disable 1 enable
11 DP_CSC_YUV_ SAT_MODE_ ASYNC0	CSC_YUV_SAT_MODE YUV saturation mode for color space conversion 0 Y/U/V range 0 -255 1 Y range 16-235, U/V range 16-240

Table continues on the next page...

IPU_DP_COM_CONF_ASYNC0 field descriptions (continued)

Field	Description
10 DP_CSC_ GAMUT_SAT_ EN_ASYNC0	CSC_GAMUT_SAT_EN Indicate if GAMUT saturation is enabled 0 disable GAMUT mapping 1 enable GAMUT mapping
9–8 DP_CSC_DEF_ ASYNC0	CSC_DEF Enable or disable Color Space Conversion. 00 CSC disable 01 CSC enable after combining 10 CSC enable before combining on BG channel 11 CSC enable before combining on FG channel
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–4 DP_COC_ ASYNC0	COC - Cursor Operation Control Controls the format of the cursor and the type of arithmetic operations 000 Transparent, cursor is disabled. 001 Full cursor. 010 Reversed cursor. 011 AND between full plane and cursor. 100 Reserved 101 OR between full plane and cursor. 110 XOR between full plane and cursor. 111 Reserved
3 DP_GWCKE_ ASYNC0	GWCKE - Graphic Window Color Keying Enable Enable or disable graphic window color keying. 1 Enable color keying of graphic window 0 Disable color keying of graphic window
2 DP_GWAM_ ASYNC0	GWAM - Graphic Window Alpha Mode Select the use of Alpha to be global or local. 1 Global Alpha. 0 Local Alpha.
1 DP_GWSEL_ ASYNC0	GWSEL - Graphic Window Select Select graphic window to be on partial plane or full plane. 1 Graphic window is partial plane. 0 Graphic window is full plane.5
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

45.51.386 DP Graphic Window Control Async 0 Flow Register (IPU_DP_GRAPH_WIND_CTRL_ASYNC0)

This register contains common configuration parameters for the DP.

Address: IPU_DP_GRAPH_WIND_CTRL_ASYNC0 is 1E00_0000h base + 1040064h offset = 1F04_0064h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DP_GWAV_ASYNC0								DP_GWCKR_ASYNC0								DP_GWCKG_ASYNC0								DP_GWCKB_ASYNC0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_GRAPH_WIND_CTRL_ASYNC0 field descriptions

Field	Description
31–24 DP_GWAV_ ASYNC0	GWAV - Graphic Window Alpha Value Defines the alpha value of graphic window used for alpha blending between graphic window and full plane. The Value the number that writing to GWAV register. The Actual Value the number, that insert to calculation in Combining Module. If Value = < 0.5- 1/256 (01111111) then Actual Value = Value. If Value >= 0.5 (10000000), then Actual Value = Value + 1/256. 00000000 Actual value is 00000000; Graphic window totally opaque i.e. overlay on LCD screen 01111111 Actual value is 01111111; 10000000 Actual value is 10000001 10000001 Actual value is 10000010 11111110 Actual value is 11111111 11111111 Actual value is 100000000;Graphic window totally transparent i.e. not displayed on LCD screen
23–16 DP_GWCKR_ ASYNC0	GWCKR - Graphic Window Color Keying Red Component Defines the red component of graphic window color keying. 00000000 No red 11111111 Full red
15–8 DP_GWCKG_ ASYNC0	GWCKG - Graphic Window Color Keying Green Component Defines the green component of graphic window color keying. 00000000 No Green 11111111 Full Green
7–0 DP_GWCKB_ ASYNC0	GWCKB - Graphic Window Color Keying Blue Component Defines the blue component of graphic window color keying. 00000000 No blue 11111111 Full blue

45.51.387 DP Partial Plane Window Position Async 0 Flow Register (IPU_DP_FG_POS_ASYNC0)

The DP partial plane Window Position Register is used to determine the starting position of the partial plane window relative to BG window position.

This Register is used for the synchronous flow only.

Address: IPU_DP_FG_POS_ASYNC0 is 1E00_0000h base + 1040068h offset = 1F04_0068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					DP_FGXP_ASYNC0											0					DP_FGYP_ASYNC0										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_FG_POS_ASYNC0 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–16 DP_FGXP_ASYNC0	FGXP partial plane Window X Position. partial plane Window X Position - Specifies the number of pixels between the start of full plane window X position and the beginning of the first data of new line.
15–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–0 DP_FGYP_ASYNC0	FGYP partial plane window Y position partial plane Window Y Position - Specifies the number of lines between the start of full plane windows Y position and the beginning of the first data.

45.51.388 DP Cursor Position and Size Async 0 Flow Register (IPU_DP_CUR_POS_ASYNC0)

The LCD Cursor Position Register is used to determine the starting position of the cursor relative to BG windows position.

Address: IPU_DP_CUR_POS_ASYNC0 is 1E00_0000h base + 104006Ch offset = 1F04_006Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DP_CYP_ASYNC0					DP_CYH_ASYNC0											DP_CXP_ASYNC0					DP_CXW_ASYNC0										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_CUR_POS_ASYNC0 field descriptions

Field	Description
31–27 DP_CYP_ ASYNC0	CYP - Cursor Y Position Represents the cursors vertical starting position Y in pixel count (from 0 to CYH).Live View Resolution Mode.
26–16 DP_CYH_ ASYNC0	CYH - Cursor Height Specifies the height of the hardware cursor in pixels.
15–11 DP_CXP_ ASYNC0	CXP - Cursor X Position Represents the cursors horizontal starting position X in pixel count (from 0 to CXW).
10–0 DP_CXW_ ASYNC0	CXW - Cursor Width. Specifies the width of the hardware cursor in pixels.

45.51.389 DP Color Cursor Mapping Async 0 Flow Register (IPU_DP_CUR_MAP_ASYNC0)

The LCD Color Cursor Mapping Register defines the color of the cursor in passive or TFT color modes.

Address: IPU_DP_CUR_MAP_ASYNC0 is 1E00_0000h base + 1040070h offset = 1F04_0070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DP_CUR_COL_B_ASYNC0								DP_CUR_COL_G_ASYNC0								DP_CUR_COL_R_ASYNC0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_CUR_MAP_ASYNC0 field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23–16 DP_CUR_COL_ B_ASYNC0	CUR_COL_B - Cursor Blue Field Defines the Blue component of the cursor color in color mode 00000000 No Blue. 11111111 Full Blue.
15–8 DP_CUR_COL_ G_ASYNC0	CUR_COL_G - Cursor Green Field Defines the Green component of the cursor color in color mode 00000000 No Green. 11111111 Full Green.

Table continues on the next page...

IPU_DP_CUR_MAP_ASYNC0 field descriptions (continued)

Field	Description
7–0 DP_CUR_COL_R_ASYNC0	CUR_COL_R - Cursor Red Field Defines the Red component of the cursor color in color mode 00000000 No Red. 11111111 Full Red.

45.51.390 DP Gamma Constant Async 0 Flow Register i (IPU_DP_GAMMA_C_ASYNC0_i)

This registers contains CONSTANTi parameters used for gamma correction inside the display processor (DP).

Address: IPU_DP_GAMMA_C_ASYNC0_i is 1E00_0000h base + 1040074h offset = 1F04_0074h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DP_GAMMA_C_ASYNC0_<2*i+1>												0							DP_GAMMA_C_ASYNC0_<2*i>									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_GAMMA_C_ASYNC0_i field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved.
27–16 DP_GAMMA_C_ASYNC0_<2*i+1>	CONSTANTi+1 parameter of Gamma Correction.
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 DP_GAMMA_C_ASYNC0_<2*i>	CONSTANTi parameter of Gamma Correction.

45.51.391 DP Gamma Correction Slope Async 0 Flow Register i (IPU_DP_GAMMA_S_ASYNC0_i)

This registers contains SLOPEi parameters used for Gamma Correction inside the display processor (DP).

Address: IPU_DP_GAMMA_S_ASYNC0_i is 1E00_0000h base + 1040094h offset = 1F04_0094h

Bit	31–24	23–16	15–8	7–0
R W	DP_GAMMA_S_ASYNC0_<4*i+3>	DP_GAMMA_S_ASYNC0_<4*i+2>	DP_GAMMA_S_ASYNC0_<4*i+1>	DP_GAMMA_S_ASYNC0_<4*i>
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

IPU_DP_GAMMA_S_ASYNC0_i field descriptions

Field	Description
31–24 DP_GAMMA_S_ASYNC0_<4*i+3>	SLOPE<4*i+3> parameter of Gamma Correction.
23–16 DP_GAMMA_S_ASYNC0_<4*i+2>	SLOPE<4*i+2> parameter of Gamma Correction.
15–8 DP_GAMMA_S_ASYNC0_<4*i+1>	SLOPE<4*i+1> parameter of Gamma Correction.
7–0 DP_GAMMA_S_ASYNC0_<4*i>	SLOPE<4*i> parameter of Gamma Correction.

45.51.392 DP Color Space Conversion Control Async 0 Flow Register i (IPU_DP_CSCA_ASYNC0_i)

Address: IPU_DP_CSCA_ASYNC0_i is 1E00_0000h base + 10400A4h offset = 1F04_00A4h

Bit	31–26	25–16	15–10	9–0
R	0	DP_CSC_A_ASYNC0<2*i+1>	0	DP_CSC_A_ASYNC0<2*i>
W				
Reset	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0

IPU_DP_CSCA_ASYNC0_i field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value zero. Reserved.
25–16 DP_CSC_A_ ASYNC0<2*i+1>	A<2*i+1> parameter of color conversion
15–10 Reserved	This read-only field is reserved and always has the value zero. Reserved.
9–0 DP_CSC_A_ ASYNC0<2*i>	A<2*i> parameter of color conversion.

45.51.393 DP Color Conversion Control Async 0 Flow Register 0 (IPU_DP_CSC_ASYNC0_0)

Address: IPU_DP_CSC_ASYNC0_0 is 1E00_0000h base + 10400B4h offset = 1F04_00B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DP_CSC_S0_ ASYNC0		DP_CSC_B0_ASYNC0													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						DP_CSC_A8_ASYNC0									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_CSC_ASYNC0_0 field descriptions

Field	Description
31–30 DP_CSC_S0_ ASYNC0	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
29–16 DP_CSC_B0_ ASYNC0	B0 parameter of color conversion.
15–10 Reserved	This read-only field is reserved and always has the value zero. Reserved.
9–0 DP_CSC_A8_ ASYNC0	A9 parameter of color conversion.

45.51.394 DP Color Conversion Control Async 1 Flow Register (IPU_DP_CSC_ASYNC_1)

Address: IPU_DP_CSC_ASYNC_1 is 1E00_0000h base + 10400B8h offset = 1F04_00B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DP_CSC_S2_ASYNC0		DP_CSC_B2_ASYNC0													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DP_CSC_S1_ASYNC0		DP_CSC_B1_ASYNC0													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

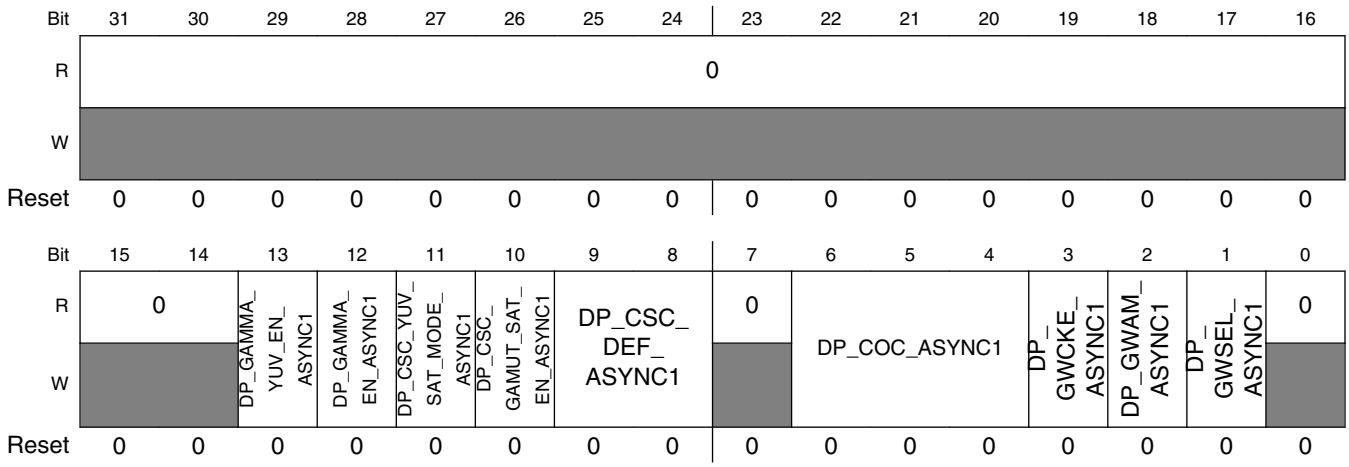
IPU_DP_CSC_ASYNC_1 field descriptions

Field	Description
31–30 DP_CSC_S2_ASYNC0	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
29–16 DP_CSC_B2_ASYNC0	B0 parameter of color conversion.
15–14 DP_CSC_S1_ASYNC0	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
13–0 DP_CSC_B1_ASYNC0	B0 parameter of color conversion.

45.51.395 DP Common Configuration Async 1 Flow Register (IPU_DP_COM_CONF_ASYNC1)

This register contains common configuration parameters for the DP.

Address: IPU_DP_COM_CONF_ASYNC1 is 1E00_0000h base + 10400BCh offset = 1F04_00BCh



IPU_DP_COM_CONF_ASYNC1 field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 DP_GAMMA_YUV_EN_ASYNC1	GAMMA's YUV mode enable for async flow 1 0 YUV mode is OFF. 1 YUV mode is ON.
12 DP_GAMMA_EN_ASYNC1	GAMMA_EN - Gamma correction block enable bit 0 disable 1 enable
11 DP_CSC_YUV_SAT_MODE_ASYNC1	CSC_YUV_SAT_MODE YUV saturation mode for color space conversion 0 Y/U/V range 0 -255 1 Y range 16-235, U/V range 16-240
10 DP_CSC_GAMUT_SAT_EN_ASYNC1	CSC_GAMUT_SAT_EN Indicate if GAMUT saturation is enabled 0 disable GAMUT mapping 1 enable GAMUT mapping
9–8 DP_CSC_DEF_ASYNC1	CSC_DEF Enable or disable Color Space Conversion. 00 CSC disable

Table continues on the next page...

IPU_DP_COM_CONF_ASYNC1 field descriptions (continued)

Field	Description
	01 CSC enable after combining 10 CSC enable before combining on BG channel 11 CSC enable before combining on FG channel
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6–4 DP_COC_ ASYNC1	COC - Cursor Operation Control Controls the format of the cursor and the type of arithmetic operations 000 Transparent, cursor is disabled. 001 Full cursor. 010 Reversed cursor. 011 AND between full plane and cursor. 100 Reserved 101 OR between full plane and cursor. 110 XOR between full plane and cursor. 111 Reserved
3 DP_GWCKE_ ASYNC1	GWCKE - Graphic Window Color Keying Enable Enable or disable graphic window color keying. 1 Enable color keying of graphic window 0 Disable color keying of graphic window
2 DP_GWAM_ ASYNC1	GWAM - Graphic Window Alpha Mode Select the use of Alpha to be global or local. 1 Global Alpha. 0 Local Alpha.
1 DP_GWSEL_ ASYNC1	GWSEL - Graphic Window Select Select graphic window to be on partial plane or full plane. 1 Graphic window is partial plane. 0 Graphic window is full plane.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

45.51.396 DP Graphic Window Control Async 1 Flow Register (IPU_DP_GRAPH_WIND_CTRL_ASYNC1)

This register contains common configuration parameters for the DP.

Address: IPU_DP_GRAPH_WIND_CTRL_ASYNC1 is 1E00_0000h base + 10400C0h offset = 1F04_00C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DP_GWAV_ASYNC1								DP_GWCKR_ASYNC1								DP_GWCKG_ASYNC1								DP_GWCKB_ASYNC1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_GRAPH_WIND_CTRL_ASYNC1 field descriptions

Field	Description
31–24 DP_GWAV_ ASYNC1	GWAV - Graphic Window Alpha Value Defines the alpha value of graphic window used for alpha blending between graphic window and full plane. The Value the number that writing to GWAV register. The Actual Value the number, that insert to calculation in Combining Module. If Value = < 0.5- 1/256 (01111111) then Actual Value = Value. If Value >= 0.5 (10000000), then Actual Value = Value + 1/256. 00000000 Actual value is 00000000; Graphic window totally opaque i.e. overlay on LCD screen 01111111 Actual value is 01111111; 10000000 Actual value is 10000001 10000001 Actual value is 10000010 11111110 Actual value is 11111111 11111111 Actual value is 100000000;Graphic window totally transparent i.e. not displayed on LCD screen
23–16 DP_GWCKR_ ASYNC1	GWCKR - Graphic Window Color Keying Red Component Defines the red component of graphic window color keying. 00000000 No red 11111111 Full red
15–8 DP_GWCKG_ ASYNC1	GWCKG - Graphic Window Color Keying Green Component Defines the green component of graphic window color keying. 00000000 No Green 11111111 Full Green
7–0 DP_GWCKB_ ASYNC1	GWCKB - Graphic Window Color Keying Blue Component Defines the blue component of graphic window color keying. 00000000 No blue 11111111 Full blue

45.51.397 DP Partial Plane Window Position Async 1 Flow Register (IPU_DP_FG_POS_ASYNC1)

The DP partial plane Window Position Register is used to determine the starting position of the partial plane window relative to BG window position.

This Register is used for the synchronous flow only.

Address: IPU_DP_FG_POS_ASYNC1 is 1E00_0000h base + 10400C4h offset = 1F04_00C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					DP_FGXP_ASYNC1											0					DP_FGYP_ASYNC1										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_FG_POS_ASYNC1 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved
26–16 DP_FGXP_ASYNC1	FGXP partial plane Window X Position. partial plane Window X Position - Specifies the number of pixels between the start of full plane window X position and the beginning of the first data of new line.
15–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–0 DP_FGYP_ASYNC1	FGYP partial plane window Y position partial plane Window Y Position - Specifies the number of lines between the start of full plane windows Y position and the beginning of the first data.

45.51.398 DP Cursor Postion and Size Async 1 Flow Register (IPU_DP_CUR_POS_ASYNC1)

The LCD Cursor Position Register is used to determine the starting position of the cursor relative to BG windows position.

Address: IPU_DP_CUR_POS_ASYNC1 is 1E00_0000h base + 10400C8h offset = 1F04_00C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DP_CYP_ASYNC1					DP_CYH_ASYNC1											DP_CXP_ASYNC1					DP_CXW_ASYNC1										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_CUR_POS_ASYNC1 field descriptions

Field	Description
31–27 DP_CYP_ ASYNC1	CYP - Cursor Y Position Represents the cursors vertical starting position Y in pixel count (from 0 to CYH).Live View Resolution Mode.
26–16 DP_CYH_ ASYNC1	CYH - Cursor Height Specifies the height of the hardware cursor in pixels.
15–11 DP_CXP_ ASYNC1	CXP - Cursor X Position Represents the cursors horizontal starting position X in pixel count (from 0 to CXW).
10–0 DP_CXW_ ASYNC1	CXW - Cursor Width. Specifies the width of the hardware cursor in pixels.

45.51.399 DP Color Cursor Mapping Async 1 Flow Register (IPU_DP_CUR_MAP_ASYNC1)

The LCD Color Cursor Mapping Register defines the color of the cursor in passive or TFT color modes.

Address: IPU_DP_CUR_MAP_ASYNC1 is 1E00_0000h base + 10400CCh offset = 1F04_00CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DP_CUR_COL_B_ASYNC1								DP_CUR_COL_G_ASYNC1								DP_CUR_COL_R_ASYNC1							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_CUR_MAP_ASYNC1 field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23–16 DP_CUR_COL_ B_ASYNC1	CUR_COL_B - Cursor Blue Field Defines the Blue component of the cursor color in color mode 00000000 No Blue. 11111111 Full Blue.
15–8 DP_CUR_COL_ G_ASYNC1	CUR_COL_G - Cursor Green Field Defines the Green component of the cursor color in color mode 00000000 No Green. 11111111 Full Green.

Table continues on the next page...

IPU_DP_CUR_MAP_ASYNC1 field descriptions (continued)

Field	Description
7–0 DP_CUR_COL_R_ASYNC1	CUR_COL_R - Cursor Red Field Defines the Red component of the cursor color in color mode 00000000 No Red. 11111111 Full Red.

45.51.400 DP Gamma Constants Async 1 Flow Register i (IPU_DP_GAMMA_C_ASYNC1_i)

This registers contains CONSTANTi parameters used for gamma correction inside the display processor (DP).

Address: IPU_DP_GAMMA_C_ASYNC1_i is 1E00_0000h base + 10400D0h offset = 1F04_00D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							DP_GAMMA_C_ASYNC1_<2*i+1>									0							DP_GAMMA_C_ASYNC1_<2*i>							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_GAMMA_C_ASYNC1_i field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 DP_GAMMA_C_ASYNC1_<2*i+1>	CONSTANTi+1 parameter of Gamma Correction.
15–9 Reserved	This read-only field is reserved and always has the value zero. Reserved.
8–0 DP_GAMMA_C_ASYNC1_<2*i>	CONSTANTi parameter of Gamma Correction.

45.51.401 DP Gamma Correction Slope Async 1 Flow Register i (IPU_DP_GAMMA_S_ASYN1_i)

This registers contains SLOPEi parameters used for Gamma Correction inside the display processor (DP).

Address: IPU_DP_GAMMA_S_ASYN1_i is 1E00_0000h base + 10400F0h offset = 1F04_00F0h

Bit	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
R W	DP_GAMMA_S_ASYNC1_<4*i+3>	DP_GAMMA_S_ASYNC1_<4*i+2>	DP_GAMMA_S_ASYNC1_<4*i+1>	DP_GAMMA_S_ASYNC1_<4*i>
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

IPU_DP_GAMMA_S_ASYN1_i field descriptions

Field	Description
31–24 DP_GAMMA_S_ASYNC1_<4*i+3>	SLOPE<4*i+3> parameter of Gamma Correction.
23–16 DP_GAMMA_S_ASYNC1_<4*i+2>	SLOPE<4*i+2> parameter of Gamma Correction.
15–8 DP_GAMMA_S_ASYNC1_<4*i+1>	SLOPE<4*i+1> parameter of Gamma Correction.
7–0 DP_GAMMA_S_ASYNC1_<4*i>	SLOPE<4*i> parameter of Gamma Correction.

45.51.402 DP Color Space Converstion Control Async 1 Flow Register i (IPU_DP_CSCA_ASYNC1_i)

Address: IPU_DP_CSCA_ASYNC1_i is 1E00_0000h base + 1040100h offset = 1F04_0100h

Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
R	0	DP_CSC_A_ASYNC1<2*i+1>	0	DP_CSC_A_ASYNC1<2*i>
W				
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

IPU_DP_CSCA_ASYNC1_i field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value zero. Reserved.
25–16 DP_CSC_A_ ASYNC1<2*i+1>	A<2*i+1> parameter of color conversion.
15–10 Reserved	This read-only field is reserved and always has the value zero. Reserved.
9–0 DP_CSC_A_ ASYNC1<2*i>	A<2*i> parameter of color conversion.

45.51.403 DP Color Conversion Control Async 1 Flow Register 0 (IPU_DP_CSC_ASYNC1_0)

Address: IPU_DP_CSC_ASYNC1_0 is 1E00_0000h base + 1040110h offset = 1F04_0110h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DP_CSC_S0_ ASYNC1		DP_CSC_B0_ASYNC1													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						DP_CSC_A8_ASYNC1									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_CSC_ASYNC1_0 field descriptions

Field	Description
31–30 DP_CSC_S0_ ASYNC1	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
29–16 DP_CSC_B0_ ASYNC1	B0 parameter of color conversion.
15–10 Reserved	This read-only field is reserved and always has the value zero. Reserved.
9–0 DP_CSC_A8_ ASYNC1	A9 parameter of color conversion.

45.51.404 DP Color Conversion Control Async 1 Flow Register 1 (IPU_DP_CSC_ASYNC1_1)

Address: IPU_DP_CSC_ASYNC1_1 is 1E00_0000h base + 1040114h offset = 1F04_0114h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DP_CSC_S2_ASYNC1		DP_CSC_B2_ASYNC1													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

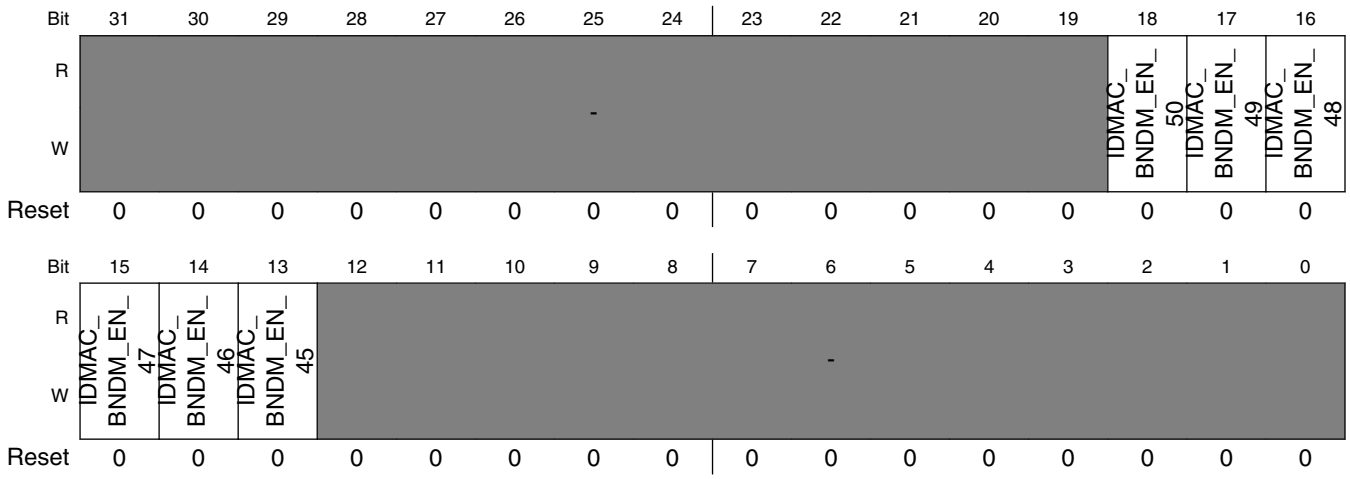
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DP_CSC_S1_ASYNC1		DP_CSC_B1_ASYNC1													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPU_DP_CSC_ASYNC1_1 field descriptions

Field	Description
31–30 DP_CSC_S2_ASYNC1	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
29–16 DP_CSC_B2_ASYNC1	B0 parameter of color conversion.
15–14 DP_CSC_S1_ASYNC1	S0 parameter of color conversion. 00 scale factor of 2 01 scale factor of 1 10 scale factor of 0 11 scale factor of -1
13–0 DP_CSC_B1_ASYNC1	B0 parameter of color conversion.

45.51.405 IDMAC Band Mode Enable 2 Register (IPU_IDMAC_BNDM_EN_2)

Address: IPU_IDMAC_BNDM_EN_2 is 1E00_0000h base + 8038_4444h offset = 9E38_4444h



IPU_IDMAC_BNDM_EN_2 field descriptions

Field	Description
31–19 -	Reserved.
18 IDMAC_BNDM_EN_50	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
17 IDMAC_BNDM_EN_49	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
16 IDMAC_BNDM_EN_48	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode

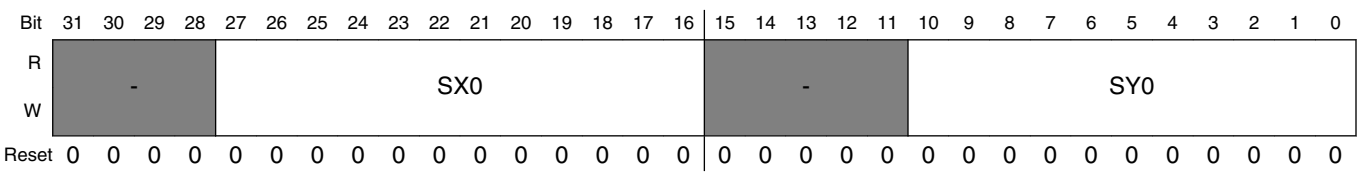
Table continues on the next page...

IPU_IDMAC_BNDM_EN_2 field descriptions (continued)

Field	Description
15 IDMAC_BNDM_EN_47	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
14 IDMAC_BNDM_EN_46	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
13 IDMAC_BNDM_EN_45	IDMAC Band Mode Enable bit [i] This bit controls if the channel currently works in band mode. When alternate flow is running via this channel, both the main flow and the alternate flow should have the same band mode configuration. 0 IDMAC channel [i] is not in band mode 1 IDMAC channel [i] is in band mode
12–0 -	Reserved.

45.51.406 IDMAC Scroll Coordinations Register (IPU_IDMAC_SC_CORD)

Address: IPU_IDMAC_SC_CORD is 1E00_0000h base + 803C_4848h offset = 9E3C_4848h



IPU_IDMAC_SC_CORD field descriptions

Field	Description
31–28 -	Reserved, should be cleared.
27–16 SX0	Scroll X coordination This field indicates the X coordinate of the scroll. This parameter has an affect on continuos scroll mode only. Units are pixels. When the alpha buffer resides in a separate buffer and the alpha is different than 8 BPP this field should be 0.

Table continues on the next page...

IPU_IDMAC_SC_CORD field descriptions (continued)

Field	Description
15–11 -	Reserved, should be cleared.
10–0 SY0	Scroll Y coordination This field indicates the Y coordinate of the scroll. This parameter has an affect on continuos scroll mode only. Units are pixels. When the alpha buffer resides in a separate buffer and the alpha is different than 8 BPP this field should be 0.

Chapter 46
Keypad Port (KPP)

46.1 Overview

The Keypad Port (KPP) is a 16-bit peripheral that can be used as a keypad matrix interface or as general purpose input/output (I/O). The figure below shows the KPP block diagram. The KPP provides interface for the keypad matrix with 2-point contact or 3-point contact keys. The KPP is designed to simplify the software task of scanning a keypad matrix. With appropriate software support, the KPP is capable of detecting, debouncing, and decoding one or multiple keys pressed simultaneously on the keypad.

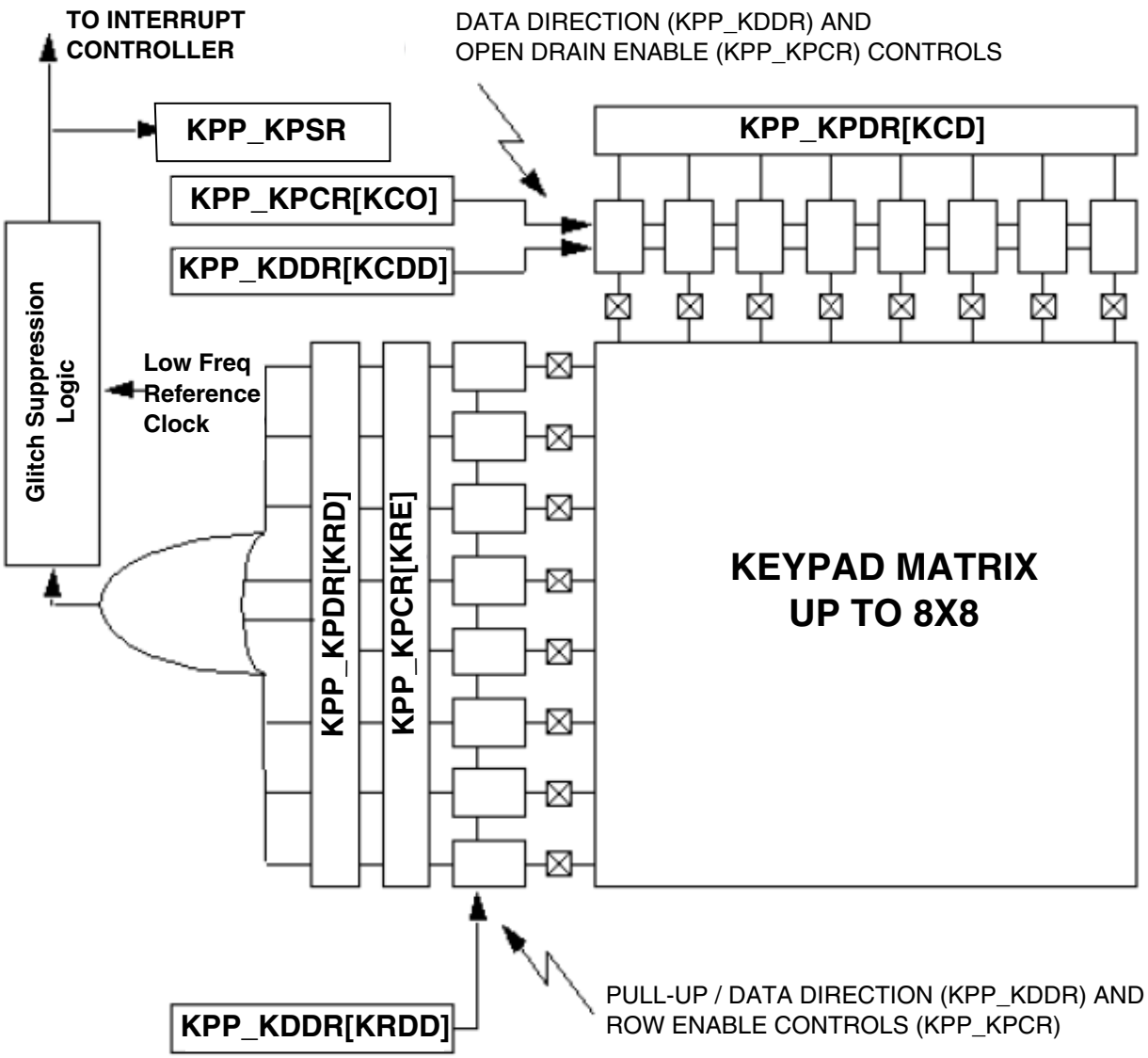


Figure 46-1. KPP Peripheral Block Diagram

46.1.1 Features

The KPP includes these distinctive features:

- Supports up to an 8 x 8 external key pad matrix
- Port pins can be used as general purpose I/O
- Open drain design
- Glitch suppression circuit design
- Multiple-key detection
- Long key-press detection

- Standby key-press detection
- Synchronizer chain clear
- Supports a 2-point and 3-point contact key matrix

46.1.2 Modes and Operations

This block supports the following modes:

- Run Mode-This is the normal functional mode in which the KPP can detect any key press event.
- Low Power Mode-The keypad can detect any key press even in low power modes (when there is no MCU clock).

46.2 External Signals

46.2.1 External Signals Overview

There are 16 pins dedicated to the KPP. Keypads of any configuration up to eight rows and eight columns are supported through the software configuration of the peripheral pins. Any pins not used for the keypad are available as general purpose I/O. The registers are configured such that the pins can be treated as an I/O port up to 16 bits wide.

See the table below for the list of external signals.

Table 46-1. Block Signals

Name	Function	Reset State	Pull up
KPCOL[7:0]	Column input or output pin, from chip	0	Active[1]
KPROW[7:0]	Row input or output pin, from chip	1	Active[2]

1. The corresponding pads are required to be pull-up enabled.
2. The corresponding pads are required to be pull-up enabled.

46.2.1.1 Input Pins

Any of the 16 pins associated with the KPP can be configured as inputs by writing a "0" to the appropriate bits in the KPP_KDDR. Additionally, the least significant 8 bits (ROW inputs) corresponding to KPP_KDDR[KRDD] have internal pull-ups, which are enabled when the pin is used as an input.

46.2.1.2 Output Pins

Any of the 16 pins associated with the KPP can be configured as outputs by writing the appropriate bits in the KPP_KDDR to a "1". Additionally, the 8 most significant bits (15-8) can be designated as open drain outputs by writing a "1" to the appropriate bits in the KPP_KPCR. The lower 8 bits (7-0) are always in "totem pole" style, driven when configured as outputs.

See the table below.

Table 46-2. Keypad Port Column Modes

KPP_KDDR (15:8)	KPP_KPCR (15:8)	Pin Function
0	x	Input
1	0	Totem-Pole Output
1	1	Open-Drain Output

NOTE

Totem pole capability should be provided for column pins. Totem pole configuration helps for a faster discharge of keypad capacitance when all columns need to be quickly brought to a "1" during the scan routine. With this configuration, delay between the scanning of two subsequent columns is reduced.

46.2.1.3 Generation of Transfer Error Signal on Peripheral Bus

If there is an access to an address which is not implemented, then the KPP asserts a transfer error signal on Peripheral Bus.

46.3 Functional Description

The Keypad Port (KPP) is designed to simplify the software task of scanning a keypad matrix. With appropriate software support and matrix organization, the KPP is capable of detecting, debouncing, and decoding one or more keys pressed simultaneously on the keypad.

Logic in the KPP is capable of detecting a key press even while the processor is in one of the low power standby modes provided that a low frequency reference clock is on. The KPP may generate an ARM platform interrupt any time a key press or key release is detected. This interrupt is capable of forcing the processor out of a low power mode.

46.3.1 Keypad Matrix Construction

The KPP is designed to interface to a keypad matrix, which shorts the intersecting row and column lines together whenever a key is depressed. The interface is not optimized for any other switch configuration.

46.3.2 Keypad Port Configuration

The software must initialize the KPP for the size of the keypad matrix. Pins connected to the keypad columns should be configured as open-drain outputs. Pins connected to the keypad rows should be configured as inputs. On-chip, pull-up resistors should be implemented for active keypad rows.

In addition to enabled row inputs in the Keypad Control register, corresponding interrupt (depress or/and release) must also be enabled to generate an interrupt.

Discrete switches that are not part of the matrix may be connected to any unused row inputs. The second terminal of the discrete switch is connected to ground. The hardware detects closures of these switches without the need for software polling.

46.3.3 Keypad Matrix Scanning

Keypad scanning is performed by a software loop that walks a zero across each of the keypad columns, reading the value on the rows at each step. The process is repeated several times in succession, with the results of each pass optionally compared to those

from the previous pass. When several (3 or 4) consecutive scans yield the same key closures, a valid key press has been detected. Software then can decode exactly which switch was depressed and pass the value up to the next higher software layer.

The basic debouncing period, which must be defined in the software routine, may be controlled with an internal timer. The basic period is the period between the scan of two consecutive columns, so the debouncing time between two consecutive scans of the whole matrix shall be the number of columns multiplied by the basic period.

46.3.4 Keypad Standby

There is no need for the ARM platform to continually scan the keypad. Between key presses, the keypad can be left in a state that requires no software intervention until the next key press is detected. To place the keypad in a standby state, software should write all column outputs low. Row inputs are left enabled. At this point, the ARM platform can attend to other tasks or revert to a low power standby mode. The KPP will interrupt the ARM platform if any key is pressed.

Upon receiving a keypad interrupt, the ARM platform should set all the column strobes high, and begin a normal keypad scanning routine to determine which key was pressed. It is important that open-drain drivers be used when scanning to prevent a possible DC path between power and ground through two or more switches.

46.3.5 Glitch Suppression on Keypad Inputs

A glitch suppression circuit qualifies the keypad inputs to prevent noise from inadvertently interrupting the ARM platform. The circuit is a 4-state synchronizer clocked from a low frequency reference clock source.

This clock must continue to run in any low power mode where the keypad is a wake-up source, as the ARM platform interrupt is generated from the synchronized input. An interrupt is not generated until all four synchronizer stages have latched a valid key assertion. This guarantees the filtering out of any noise less than three clock periods in duration of a low frequency reference clock. Noise filtering of the duration between three to four clock periods cannot be guaranteed. The interrupt output is latched in an S-R latch and remains asserted until cleared by the software. The Set input of the latch is rising-edge clocked. See the figure below.

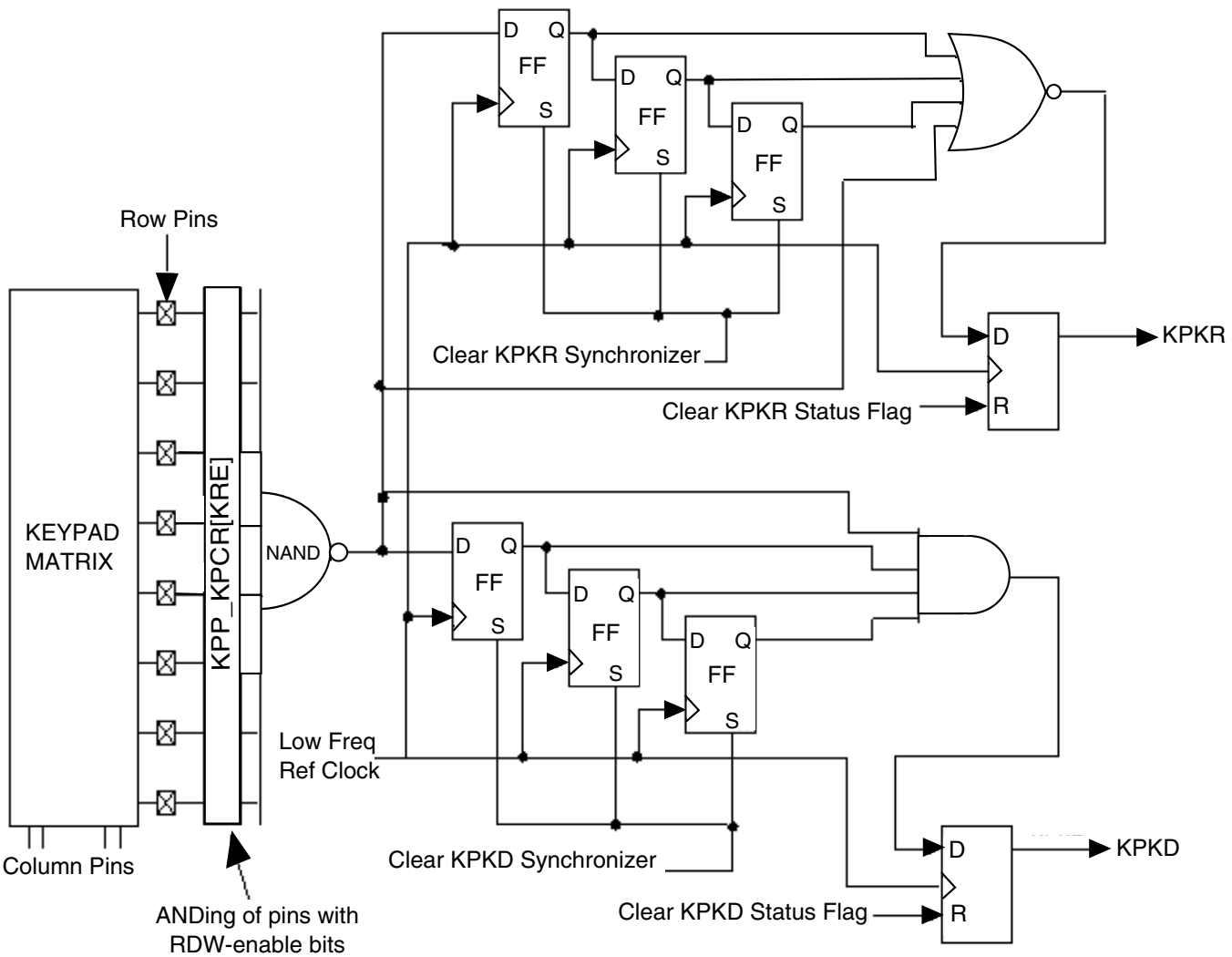


Figure 46-2. Keypad Synchronizer Functional Diagram

46.3.6 Multiple Key Closures

Using the key press and Key release interrupts, the software can detect multiple keys or achieve n key rollover. The key scanning routine can be programmed accordingly.

See the following figures for illustrations of the interfacing of a 2-contact keypad matrix with the KPP controller. With proper enabling of row lines and the performing scan-routine, multiple key presses can be detected. When keys present on the same row are pressed, corresponding row lines (multiple lines) become low when the column is driven low during a scan-routine. By reading the data-register, pressed keys can be detected. Similarly, when keys present on same row line are pressed, the corresponding row line (only one line) becomes low when logic "0" is driven on the column line during a scan-routine.

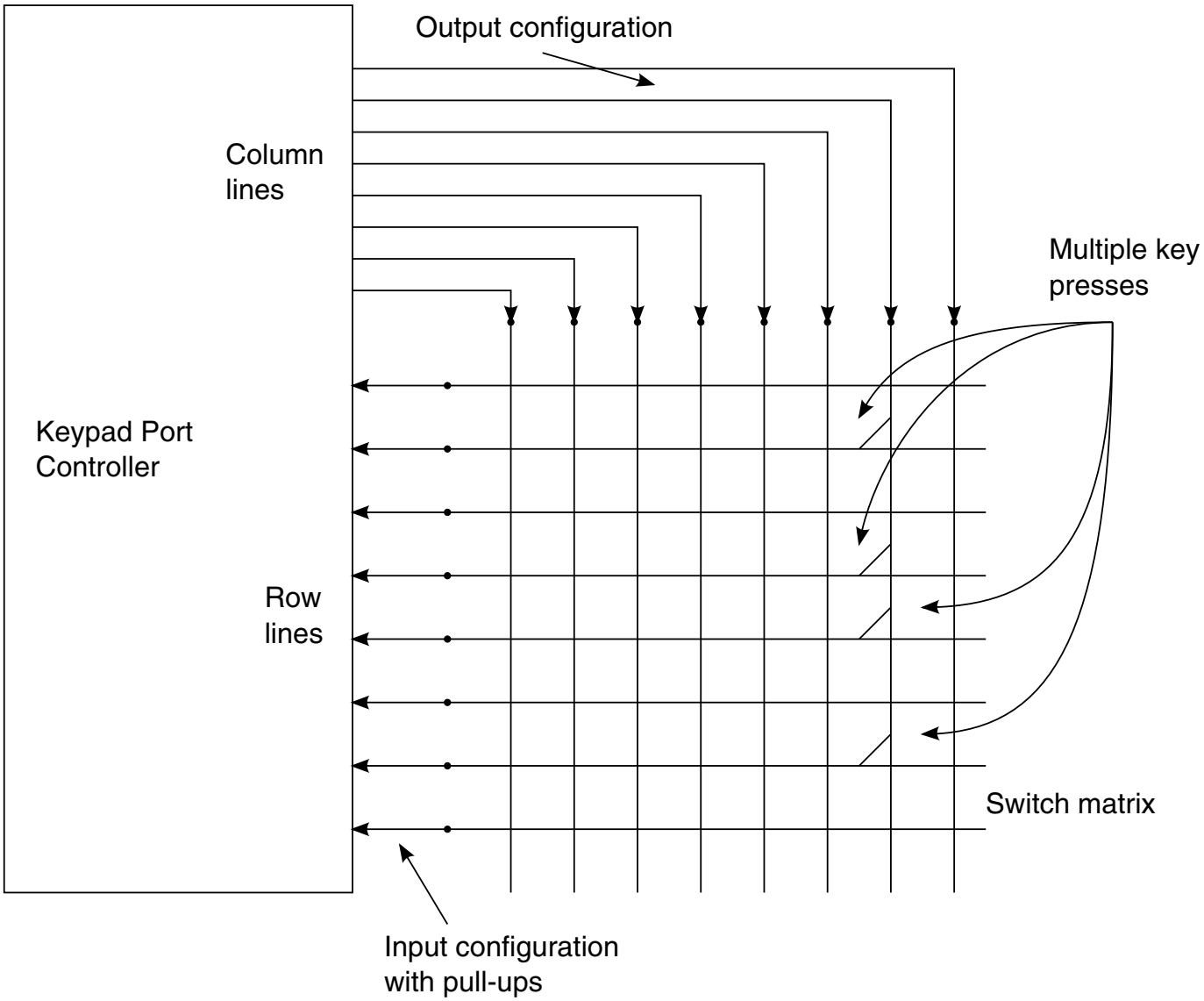


Figure 46-3. Multiple Key Presses on Same Column Line (Simplified View)

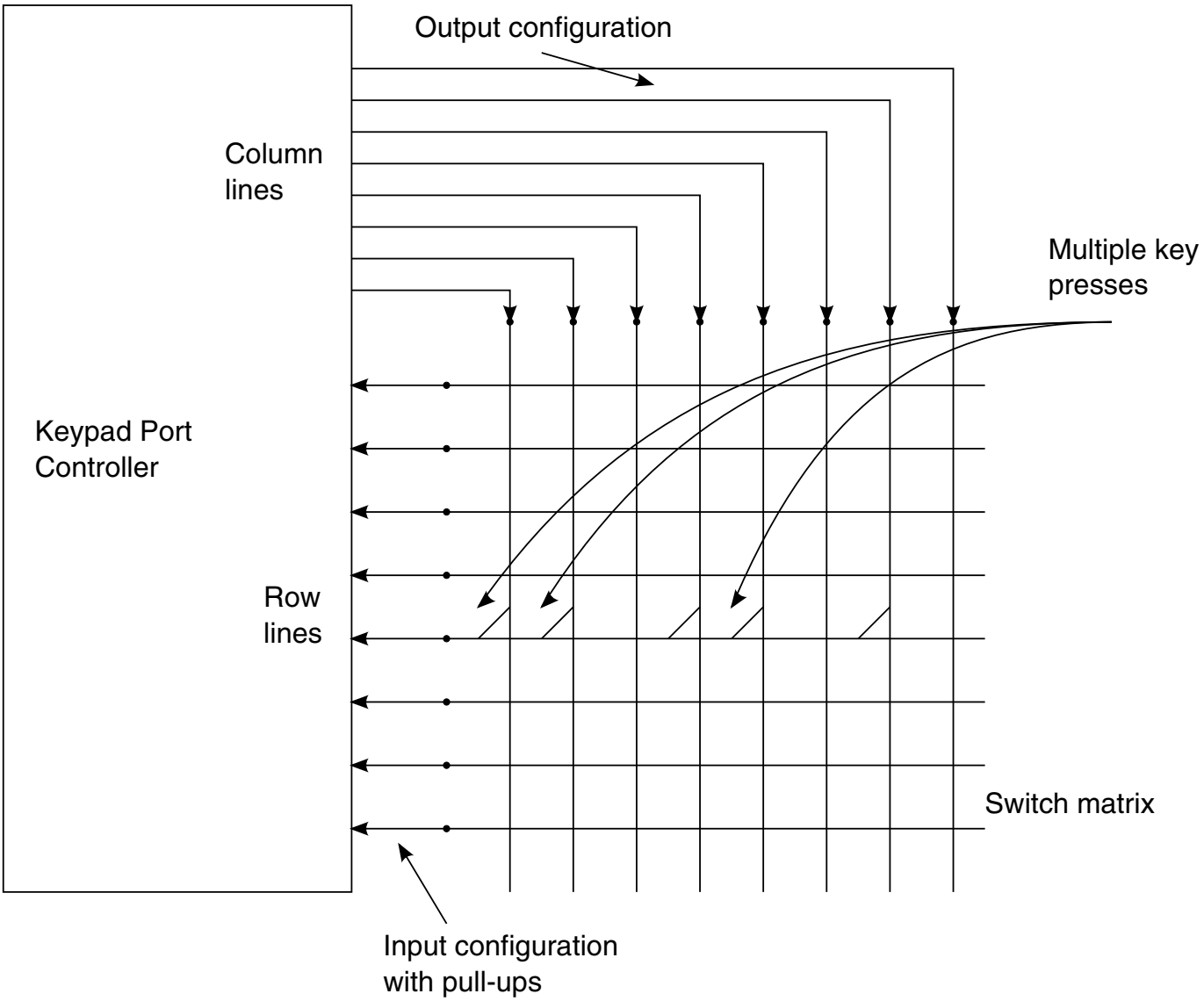


Figure 46-4. Multiple Key Presses on Same Row Line (Simplified View)

NOTE

An n key rollover is a technique with which the system can recognize the order in which keys are pressed.

46.3.6.1 Ghost Key Problem and Correction

The KPP detects if one or multiple keys are pressed or released. In the case where a simple keypad matrix with two-contact switches is used, there is a chance of "ghost" key detection when three or more keys are pressed. This is a limitation imposed by such a keypad matrix.

As can be seen in Figure 46-5, three keys pressed simultaneously can cause a short between the column currently "scanned" by the software and another column. Depending on the location of the third key pressed, a "ghost" key press may be detected.

However, this can be corrected by using a keypad matrix that provides "ghost" key protection. Such a matrix implements a one-way "diode" at all keypad points between rows and columns. This way, the multiple pressing of three keys will not cause a short at a fourth key (see Figure 46-6).

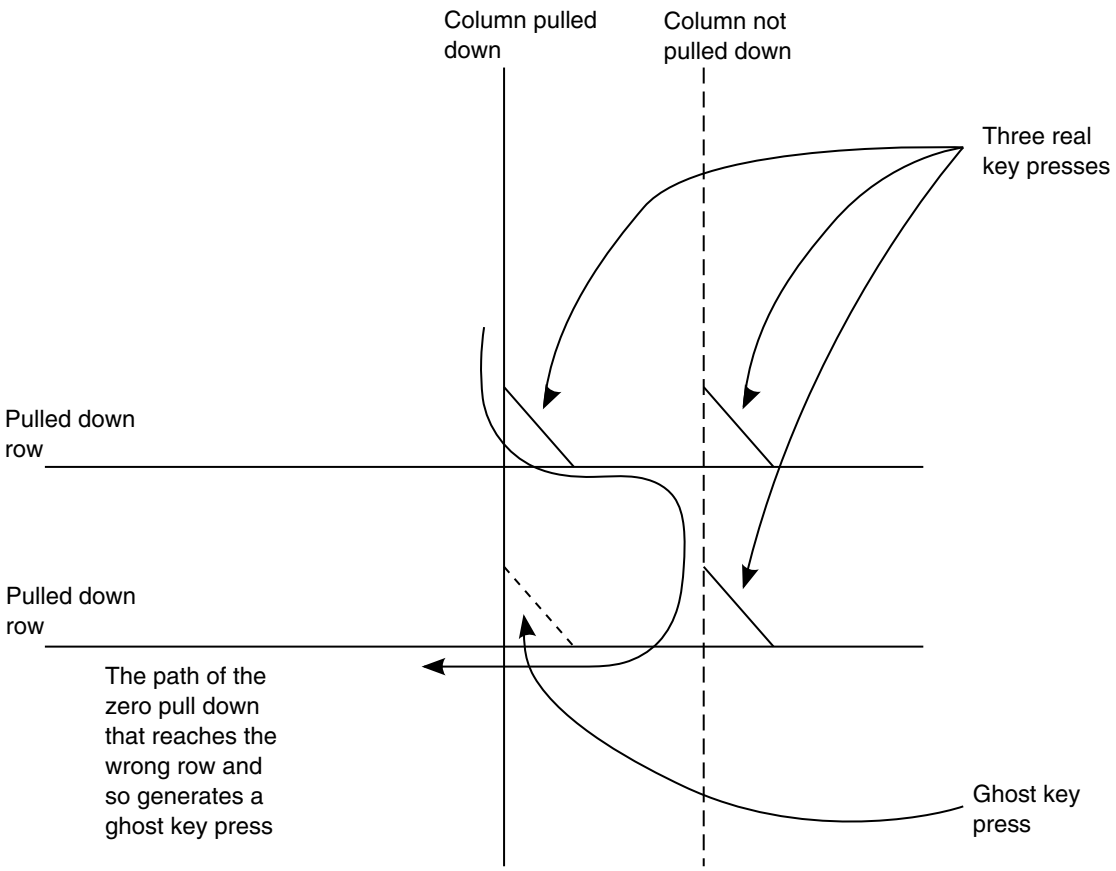


Figure 46-5. Decoding Wrong Three- Key-Presses

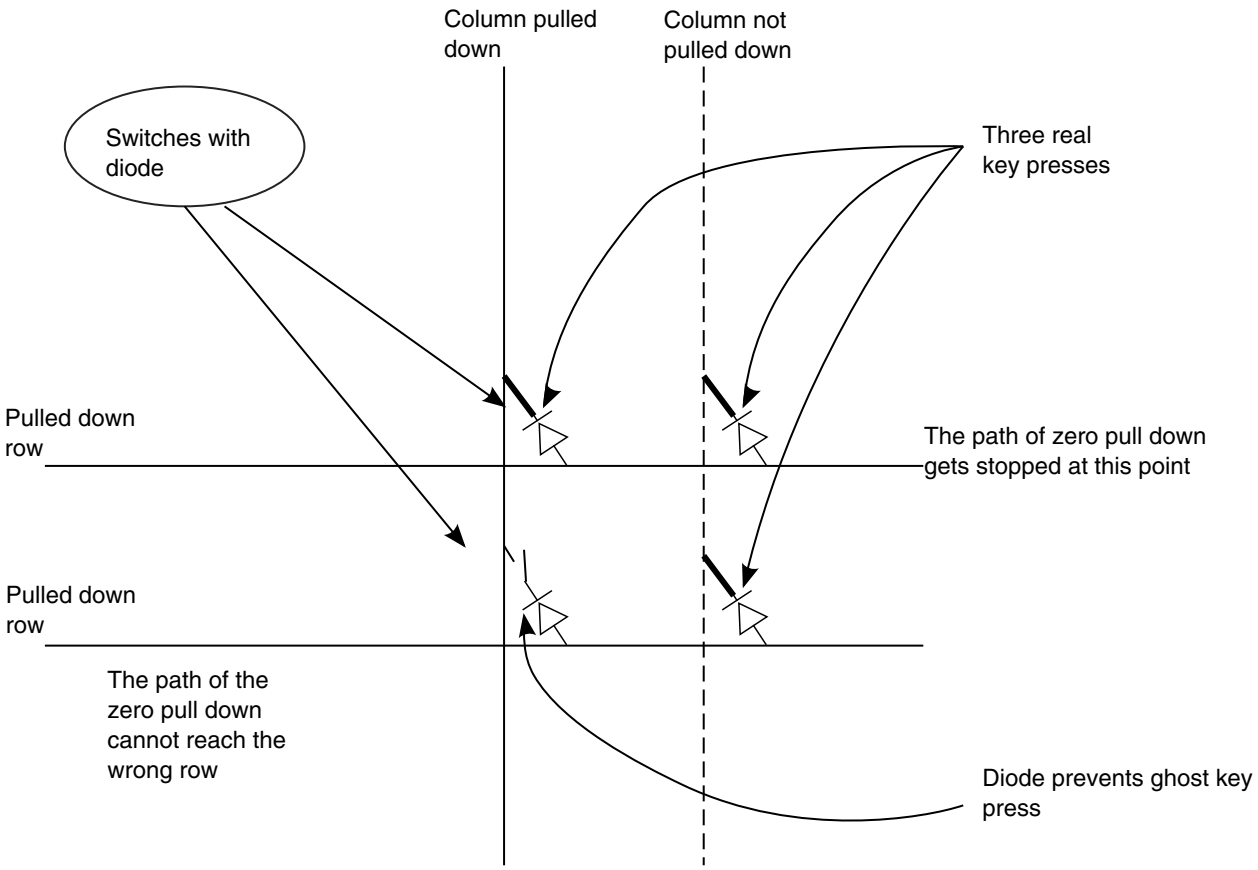


Figure 46-6. Matrix with "Ghost" Key Protections

46.3.7 3-Point Contact Keys Support

The KPP supports interfacing to a matrix consisting of 3-point contact keys. As shown in Figure 46-7, two points of such a key are connected to keypad lines, while a third point is connected to ground (low logic).

The keypad lines should be configured as input and a pull-up should to be present on these lines. When such a key is pressed, corresponding keypad lines go low and an interrupt is generated. There is no need to perform a scanning routine for identification of pressed key as it can be done by reading the keypad data-register. A limitation with such a matrix is that for every key at least one keypad row line should be used.

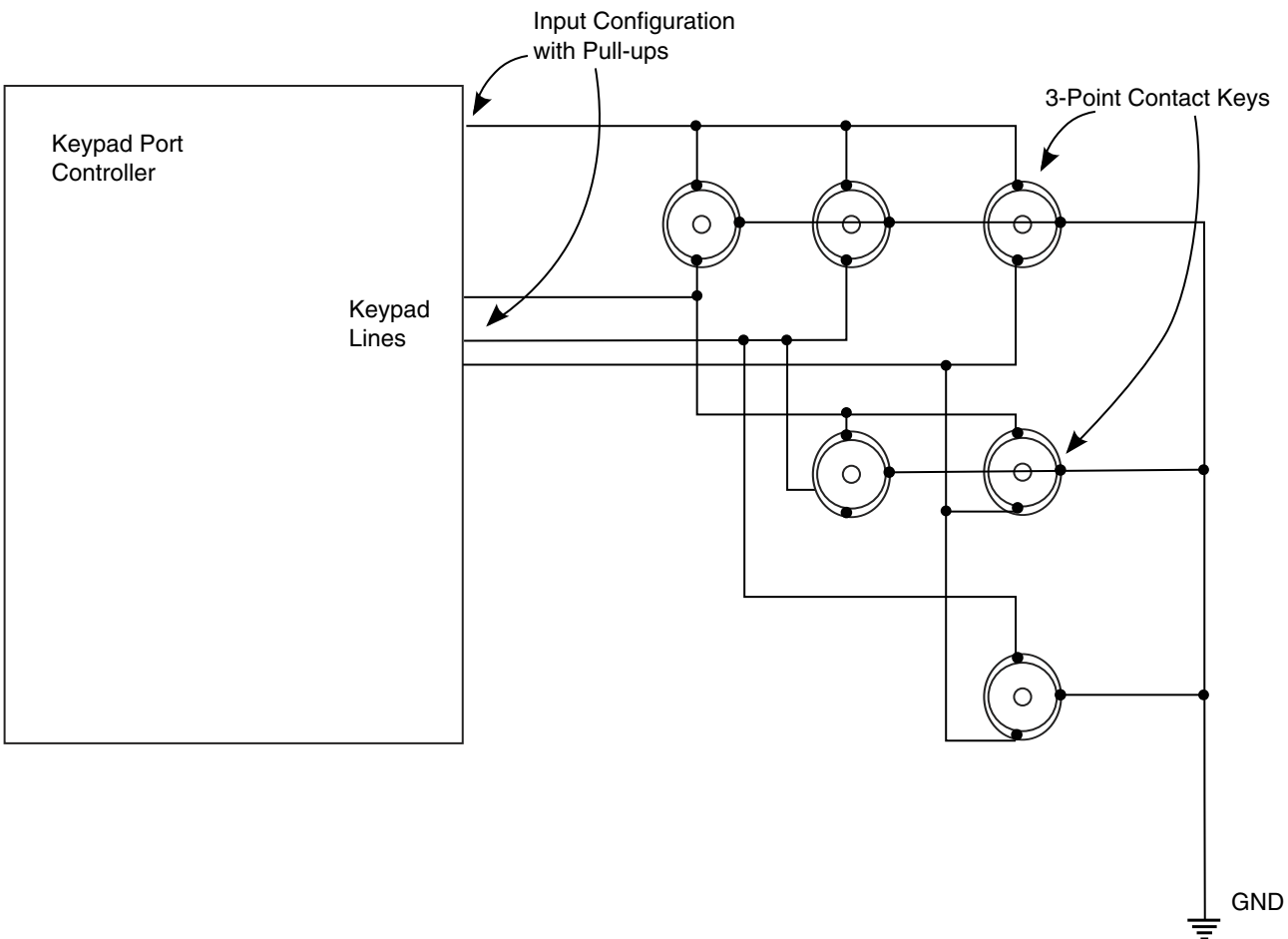


Figure 46-7. KPP Interface with 3-point Contact Key Matrix (Simplified View)

46.4 Initialization/Application Information

46.4.1 Typical Keypad Configuration and Scanning Sequence

Perform the following steps to configure the keypad:

1. Enable the number of rows in the keypad (KPP_KPCR[KRE]).
2. Write 0s to KPP_KPDR[KCD].
3. Configure the keypad columns as open-drain (KPP_KPCR[KCO]).
4. Configure columns as output (KPP_KDDR[KCDD]) and rows as input (KPP_KDDR[KRDD]).
5. Clear the KPKD Status Flag and Synchronizer chain.
6. Set the KDIE control bit, and clear the KRIE control bit (avoid false release events).
7. (The system is now in standby mode, and awaiting a key press.)

46.4.2 Key Press Interrupt Scanning Sequence

Perform the following steps to perform a keypad scanning routine:

1. Disable both (depress and release) keypad interrupts.
2. Write 1s to KPP_KPDR[KCD], setting column data to 1s.
3. Configure columns as totem pole outputs (for quick discharging of keypad capacitance).
4. Configure columns as open-drain.
5. Write a single column to 0, and other columns to 1.
6. Sample row inputs and save data. Multiple key presses can be detected on a single column.
7. Repeat Steps 2-6 for remaining columns.
8. Return all columns to 0 in preparation for standby mode.
9. Clear KPKD and KPKR status bit(s) by writing to a "1"; set the KPKR synchronizer chain by writing a "1" to the KPP_KRSS register; and clear the KPKD synchronizer chain by writing a "1" to the KDSC register.
10. Re-enable the appropriate keypad interrupt(s) so that the KDIE detects a key hold condition, or the KRIE detects a key-release event.

46.4.3 Additional Comments

The order of key press detection can be done in software only. Therefore, the software may need to run the scan routines at very short intervals of time per the application's demands. The reason that such functionality cannot be put in the KPP is that the block is limited by the number of external pins.

For the keys that require a very precise order (such as game keys), individual GPIO pins may be more useful.

46.5 Programmable Registers

The KPP contains four registers.

KPP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53F9_4000	Keypad Control Register (KPP_KPCR)	16	R/W	0000h	46.5.1/ 3427
53F9_4002	Keypad Status Register (KPP_KPSR)	16	R/W	0400h	46.5.2/ 3428
53F9_4004	Keypad Data Direction Register (KPP_KDDR)	16	R/W	0000h	46.5.3/ 3430
53F9_4006	Keypad Data Register (KPP_KPDR)	16	R/W	0000h	46.5.4/ 3430

46.5.1 Keypad Control Register (KPP_KPCR)

The Keypad Control Register determines which of the eight possible column strobes are to be open drain when configured as outputs, and which of the eight row sense lines are considered in generating an interrupt to the core.

It is up to the programmer to ensure that pins being used for functions other than the keypad are properly disabled. The KPP_KPCR register is byte- or half-word-addressable.

Address: KPP_KPCR is 53F9_4000h base + 0h offset = 53F9_4000h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read / Write	KCO								KRE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

KPP_KPCR field descriptions

Field	Description
15–8 KCO	Keypad Column Strobe Open-Drain Enable. Setting a column open-drain enable bit (KCO7-KCO0) disables the pull-up driver on that pin. Clearing the bit allows the pin to drive to the high state. This bit has no effect when the pin is configured as an input.

Table continues on the next page...

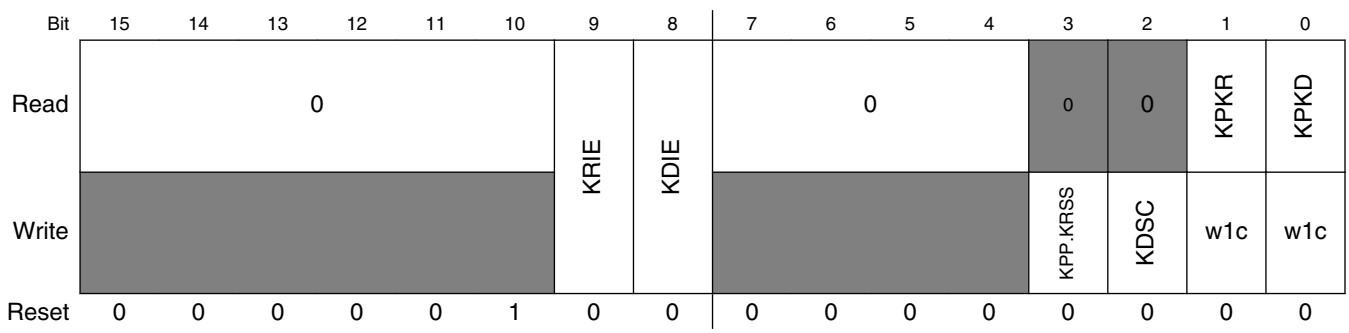
KPP_KPCR field descriptions (continued)

Field	Description
	NOTE: Configuration of external port control logic (for example, IOMUX) should be done properly so that the KPP controls an open-drain enable of the pin. 0 Column strobe output is totem pole drive. 1 Column strobe output is open drain.
7–0 KRE	Keypad Row Enable. Setting a row enable control bit in this register enables the corresponding row line to participate in interrupt generation. Likewise, clearing a bit disables that row from being used to generate an interrupt. This register is cleared by a reset, disabling all rows. The row-enable logic is independent of the programmed direction of the pin. Writing a "0" to the data register of the pins configured as outputs will cause a keypad interrupt to be generated if the row enable associated with that bit is set. 0 Row is not included in the keypad key press detect. 1 Row is included in the keypad key press detect.

46.5.2 Keypad Status Register (KPP_KPSR)

The Keypad Status Register reflects the state of the key press detect circuit. The KPP_KPSR register is byte- or half-word-addressable.

Address: KPP_KPSR is 53F9_4000h base + 2h offset = 53F9_4002h



KPP_KPSR field descriptions

Field	Description
15–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 KRIE	Keypad Release Interrupt Enable. The software should ensure that the interrupt for a Key Release event is masked until it has entered the key pressed state, and vice versa, unless this activity is desired (as might be the case when a repeated interrupt is to be generated). The synchronizer chains are capable of being initialized to detect repeated key presses or releases. If they are not initialized when the corresponding event flag is cleared, false interrupts may be generated for depress (or release) events shorter than the length of the corresponding chain. 0 No interrupt request is generated when KPKR is set. 1 An interrupt request is generated when KPKR is set.

Table continues on the next page...

KPP_KPSR field descriptions (continued)

Field	Description
8 KDIE	Keypad Key Depress Interrupt Enable. Software should ensure that the interrupt for a Key Release event is masked until it has entered the key pressed state, and vice-versa, unless this activity is desired (as might be the case when a repeated interrupt is to be generated). The synchronizer chains are capable of being initialized to detect repeated key presses or releases. If they are not initialized when the corresponding event flag is cleared, false interrupts may be generated for depress (or release) events shorter than the length of the corresponding chain. 0 No interrupt request is generated when KPKD is set. 1 An interrupt request is generated when KPKD is set.
7–4 Reserved	This read-only field is reserved and always has the value zero. Reserved, should be cleared
3 KPP.KRSS	Key Release Synchronizer Set. Self-clear bit. The Key release synchronizer is set by writing a logic one into this bit. Reads return a value of "0". 0 No effect 1 Set bits which sets keypad release synchronizer chain
2 KDSC	Key Depress Synchronizer Clear. Self-clear bit. The Key depress synchronizer is cleared by writing a logic "1" into this bit. Reads return a value of "0". 0 No effect 1 Set bits that clear the keypad depress synchronizer chain
1 KPKR	Keypad Key Release. The keypad key release (KPKR) status bit is set when all enabled rows are detected high after synchronization (the KPKR status bit will be set when cleared by a reset). The KPKR bit may be used to generate a maskable key release interrupt. The key release synchronizer may be set high by software after scanning the keypad to ensure a known state. Due to the logic function of the release and depress synchronizer chains, it is possible to see the re-assertion of a status flag (KPKD or KPKR) if it is cleared by software prior to the system exiting the state it represents. Reset value of register is "0" as long as reset is asserted. However when reset is de-asserted, the value of the register depends upon the external row pins and can become "1". 0 No key release detected 1 All keys have been released
0 KPKD	Keypad Key Depress. The keypad key depress (KPKD) status bit is set when one or more enabled rows are detected low after synchronization. The KPKD status bit remains set until cleared by the software. The KPKD bit may be used to generate a maskable key depress interrupt. If desired, the software may clear the key press synchronizer chain to allow a repeated interrupt to be generated while a key remains pressed. In this case, a new interrupt will be generated after the synchronizer delay (4 cycles of the low frequency reference clock elapses if a key remains pressed. This functionality can be used to detect a long key press. This allows detection of additional key presses of the same key or other keys. Due to the logic function of the release and depress synchronizer chains, it is possible to see the re-assertion of a status flag (KPKD or KPKR) if it is cleared by the software prior to the system exiting the state it represents. 0 No key presses detected 1 A key has been depressed

46.5.3 Keypad Data Direction Register (KPP_KDDR)

The bits in the KPP_KDDR control the direction of the keypad port pins. The upper eight bits in the register affect the pins designated as column strobes, while the lower eight bits affect the row sense pins. Setting any bit in this register configures the corresponding pin as an output. Clearing any bit in this register configures the corresponding port pin as an input. For the Keypad Row DDR, an internal pull-up is enabled if the corresponding bit is clear. This register is cleared by a reset, configuring all pins as inputs. The KPP_KDDR register is byte- or half-word addressable.

NOTE

When a pin is used as row pin for keypad purposes, all corresponding pull-ups should be enabled at the upper level (for example, IOMUX) when the bit in row DDR is cleared.

Address: KPP_KDDR is 53F9_4000h base + 4h offset = 53F9_4004h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read / Write	KCDD								KRDD							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

KPP_KDDR field descriptions

Field	Description
15–8 KCDD	Keypad Column Data Direction Register. Setting a bit configures the corresponding COL*n* pin as an output (where *n* = 7 through 0). 0 COLn pin is configured as an input. 1 COLn pin is configured as an output.
7–0 KRDD	Keypad Row Data Direction. Setting a bit configures the corresponding ROW*n* pin as an output (where *n* = 7 through 0). 0 ROWn pin configured as an input. 1 ROWn pin configured as an output.

46.5.4 Keypad Data Register (KPP_KPDR)

This 16-bit register is used to access the column and row data. Data written to this register is stored in an internal latch, and for each pin configured as an output, the stored data is driven onto the pin. A read of this register returns the value on the pin for those bits configured as inputs. Otherwise, the value read is the value stored in the register.

The KPP_KPDR register is byte- or half-word addressable. This register is not initialized by a reset. Valid data should be written to this register before any bits are configured as outputs.

Address: KPP_KPDR is 53F9_4000h base + 6h offset = 53F9_4006h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	KCD								KRD							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

KPP_KPDR field descriptions

Field	Description
15–8 KCD	Keypad Column Data. A read of these bits returns the value on the pin for those bits configured as inputs. Otherwise, the value read is the value stored in the register. 0 Read/Write "0" from/to column ports 1 Read/Write "1" from/to column ports
7–0 KRD	Keypad Row Data. A read of these bits returns the value on the pin for those bits configured as inputs. Otherwise, the value read is the value stored in the register. 0 Read/Write "0" from/to row ports 1 Read/Write "1" from/to row ports

Chapter 47
LVDS Display Bridge (LDB)

47.1 Introduction

LVDS Display Bridge (LDB) will be used to connect the IPU (Image Processing Unit) to External LVDS Display Interface.

Relevant Standards:

1. ANSI EIA-644-A. Electrical Characteristics of Low Voltage Differential Signaling (LVDS) Interface Circuits.
2. SPWG Notebook Panel Specification (V3.8 from 03/2007) http://www.spwg.org/specifications.htm.
3. PSWG standards (Panel Standardization Working Group) - set of standards for panels using LVDS. All are available fromhttp://www.vesa.org.
4. DISM Standard JEIDA-59-1999

Table 47-1. LDB IP Parametric Table

Name	IPU
Function	Connectivity to displays with LVDS interface
External I/O Pins Those are LVDS I/O pads	LVDS Display port: 2 channels. Each channel consists of: • 1 clock pair • 4 data pairs Each pair contains - LVDS special differential pad (PadP, PadM). total of 20 I/O pads.
SoC Buses	None. Only configuration signals.
Interrupts	None
DMA Requests	None
Number of instantiations	1
Clock sources and range	DI0_CLK, DI1_CLK- Display interface clock: 20-150 MHz DI0_SER_CLK, DI1_SER_CLK - Serializer clock: 140-595 MHz

The purpose of the LDB is to support flow of synchronous RGB data from the IPU to external display devices through LVDS interface. This support covers all aspects of these activities:

- Connectivity to relevant devices - Displays with LVDS receivers.
- Arranging the data as required by the external display receiver and by LVDS display standards.
- Synchronization and control capabilities.

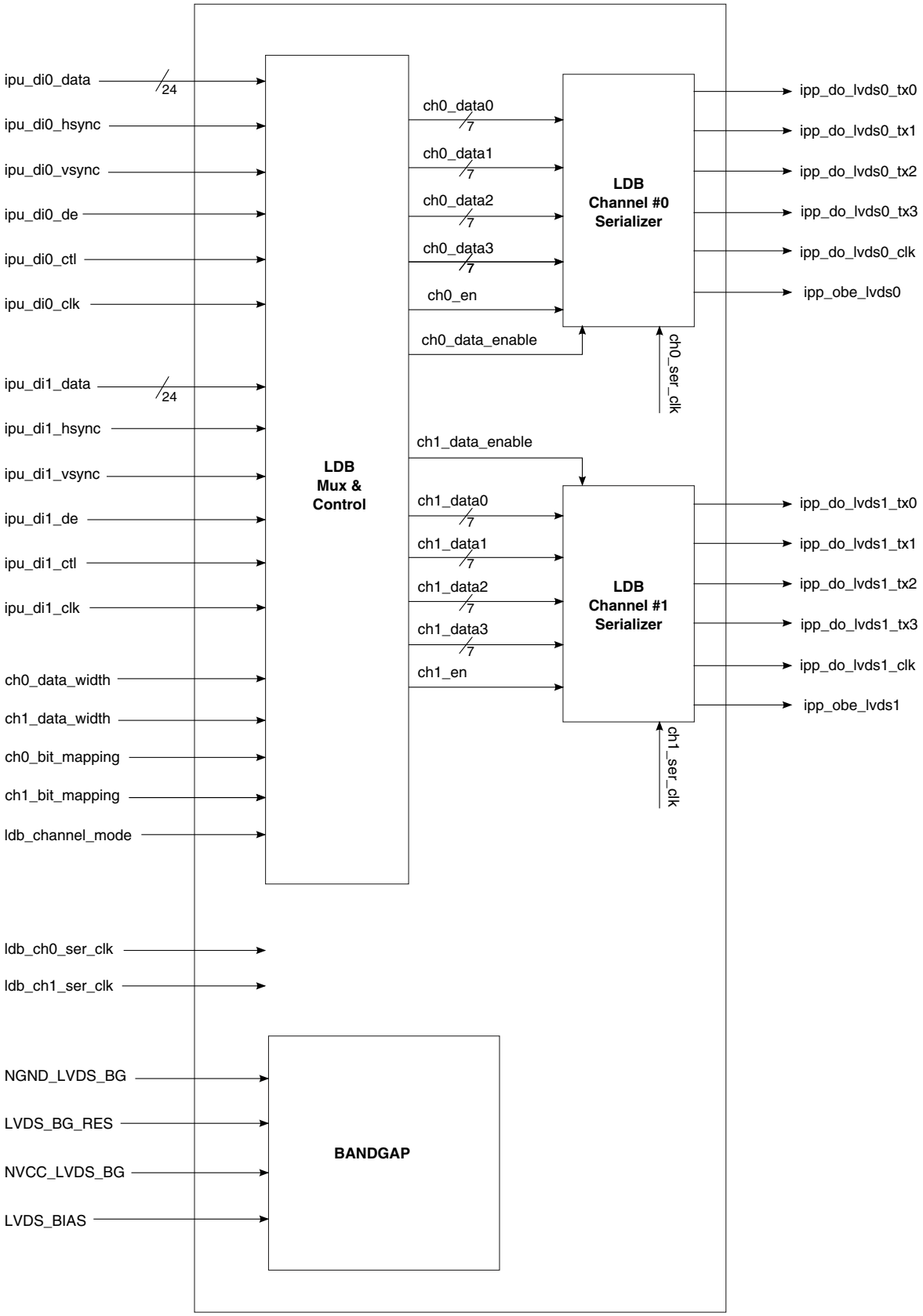


Figure 47-1. LDB - Block Diagram

Table 47-2. LDB - Block Description

Block	Description
LDB mux & control	Gets control signals from SoC and determines parameters of LDB
Channel Serializers	LDB has 2 channel serializers. Each serializer does parallel to serial conversion 7:1 to 3 or 4 data lines
Bandgap	Provides reference current to the LVDS I/O pads.

47.2 External Ports

The LDB has the following ports:

- Two input parallel display ports.
- Two Output LVDS channels - Each channel consisting of 4 data pairs, and 1 clock pair (pair=LVDS pad contains PadP, PadM).
- Control signals - to configure LDB parameters and operations.
- Clocks from SoC DPLLs.

47.2.1 Input Parallel Display Ports

One or Two (DI0, DI1) parallel RGB input ports are supported (configurable). Only synchronous access mode is supported.

Each RGB data interface contains the following:

- RGB Data of 18 or 24 bits
- Pixel clock
- Control signals: HSYNC, VSYNC, DE, and 1 additional optional general purpose control.

Total of up to 28 bits per data interface are transferred per pixel clock cycle.

Rates supported:

- Overall: LDB supports rates needed by WSXGA+ 16:10 aspect ratio (1866 x 1050 @ 60 frames per second, data rate supported up to 150 MHz)
- For single input data intrerface case: Up to 150 MHz pixel clock (WSXGA+ 16:10)
- For dual input data interface case: Up to 85 MHz per interface. (WXGA 1366x768 @ 60 frames per second, 35% blanking).

47.2.2 Output LVDS Ports

There are 2 LVDS channels. These outputs are used to communicate RGB data and controls to external LCD displays.

The LVDS ports may be used as follows:

- Single channel output
- Dual channel output (one input source, two channels outputs for two displays)
- Split channel output (one input source, splitted to 2 channels on output)
- Separate 2 channel output (2 input sources from IPU).

The output LVDS port must comply to 1.

47.3 Clock Sources

Table 47-3. LDB Clock Sources

Name	Symbol	Source	Rate	Comments
IPU DI0 interface pixel clock	DI_0_CLK	Clock control Module	Up to 150 MHz	See note below [1]
IPU DI1 interface pixel clock	DI_1_CLK	Clock control Module	Up to 150 MHz	This input also goes to IPU DI1 as input. See note below
CH0 interface serializer clock	CH0_SERIAL_CLK	Clock control Module	Up to 595 MHz	This is x7 the rate of the DI0 interface pixel clock. See note below.
CH1 interface serializer clock	CH1_SERIAL_CLK	Clock control Module	Up to 595 MHz	This is x7 the rate of the DI1 interface pixel clock. See note below

1. In case of single-channel or separate-channels use-case, the IPU DI_CLK is identical to the LVDS DI_CLK. In case of dual-channel use-case, the IPU DI_CLK has x2 higher frequency than that of the LVDS DI_CLK. Still both need to be synchronized

47.4 Processing

LDB data processing stages are as follows:

- Receive input data from 1 or 2 (configurable) parallel input interfaces. 18/24 RGB data + up to 4 controls and map them to LVDS channels.
  - If needed (dual-channel) split the input bus to two half-rate busses.
- Re-arrange the input data according to channel configuration, and muxing scheme.
- Serialize the 22/28 bit input bus (per channel) on 3-4 output serial data lines (7:1)

47.4.1 Mapping of Input Data Busses

Mapping of Parallel input interfaces (DI0, DI1) to output LVDS channels (Channel 0, Channel 1). See Table 47-2.

Table 47-4. Channel Mapping

Use Case	LVDS Channel 0	LVDS Channel 1
Single Channel DI0	DI0	Disabled
Single Channel DI1 on Channel 1	Disabled	DI1
Separate Channels	DI0	DI1
Dual Channels DI0	DI0	DI0
Dual Channels DI1	DI1	DI1
Split Channel DI0	DI0 (first pixel)	DI0 (second pixel)
Split Channel DI1	DI1 (first pixel)	DI1 (second pixel)

47.4.2 Bit Mapping

LDB supports two mapping standards:

- SPWG mapping
- JEIDA mapping

Table 47-5. SPWG/PSWG/VESA 18/24 bpp Data Mapping

Serializer input	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
CHx_DATA0	G0	R5	R4	R3	R2	R1	R0
CHx_DATA1	B1	B0	G5	G4	G3	G2	G1
CHx_DATA2	DE	VS	HS	B5	B4	B3	B2
CHx_DATA3 (for 24 bpp only)	CTL	B7	B6	G7	G6	R7	R6

Table 47-6. JEIDA 24bpp Data Mapping

Serializer input	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
CHx_DATA0	G2	R7	R6	R5	R4	R3	R2
CHx_DATA1	B3	B2	G7	G6	G5	G4	G3
CHx_DATA2	DE	VS	HS	B7	B6	B5	B4
CHx_DATA3	CTL	B1	B0	G1	G0	R1	R0

NOTE

Several options of control usage can be available. some display devices use only DE, some others use all 3 controls, some use only HS, VS. "CTL" is an optional general purpose control which is usually unused by display.

47.5 Programmable Registers

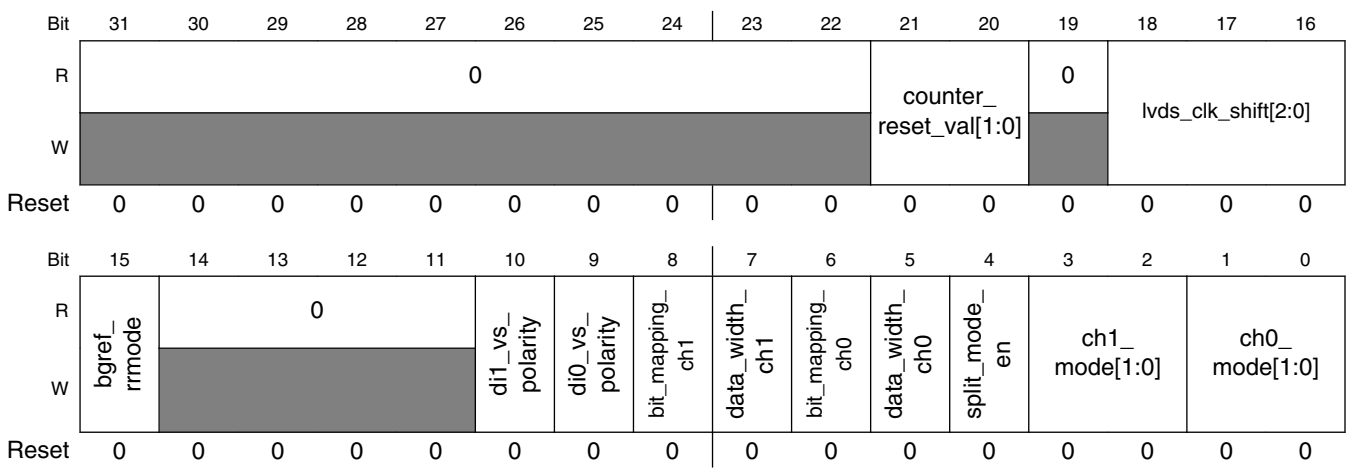
LDB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_8008	LDB Control Register (LDB_CTRL)	32	R/W	0000_0000h	47.5.1/ 3439

47.5.1 LDB Control Register (LDB_CTRL)

The register is implemented in the IOMUX Controller block (IOMUXC), as the register IOMUXC_GPR2.

Address: LDB_CTRL is 53FA_8008h base + 0h offset = 53FA_8008h



LDB_CTRL field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

LDB_CTRL field descriptions (continued)

Field	Description
21–20 counter_reset_val[1:0]	Reset value for the LDB counter which determines when the shift registers are loaded with data. **NOTE:** Used for debug purposes only. In normal functional operation must be '00' 00 Reset value is 5 01 Reset value is 3 10 Reset value is 4 11 Reset value is 6
19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18–16 lvds_clk_shift[2:0]	Shifts the LVDS output clock in relation to the data. **NOTE:** Used for debug purposes only. In normal functional operation must be '000' 000 Output clock is '1100011' (normal operation) 001 Output clock is '1110001' 010 Output clock is '1111000' 011 Output clock is '1000111' 100 Output clock is '0001111' 101 Output clock is '0011111' 110 Output clock is '0111100' 111 Output clock is '1100011'
15 bgref_rrmode	Select reference resistor for bandgap 0 External resistor of 29kOhm is selected 1 Internal resistor is selected
14–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10 di1_vs_polarity	Vsync polarity for IPU's DI1 interface. 0 ipu_di1_vsync is active high. 1 ipu_di1_vsync is active low.
9 di0_vs_polarity	Vsync polarity for IPU's DI0 interface. 0 ipu_di0_vsync is active high. 1 ipu_di0_vsync is active low.
8 bit_mapping_ch1	Data mapping for LVDS channel 1. 0 Use SPWG standard. 1 Use JEIDA standard.
7 data_width_ch1	Data width for LVDS channel 1. **NOTE:** This bit must be set when using JEIDA standard (bit_mapping_ch1 is set) 0 Data width is 18 bits wide (lvds1_tx3 is not used) 1 Data width is 24 bits wide.
6 bit_mapping_ch0	Data mapping for LVDS channel 0.

Table continues on the next page...

LDB_CTRL field descriptions (continued)

Field	Description
	0 Use SPWG standard. 1 Use JEIDA standard.
5 data_width_ch0	Data width for LVDS channel 0. **NOTE:** This bit must be set when using JEIDA standard (bit_mapping_ch0 is set) 0 Data width is 18 bits wide (lvds0_tx3 is not used) 1 Data width is 24 bits wide.
4 split_mode_en	Enable split mode. **NOTE:** In this mode both channels should be enabled and working with the same DI (ch0_mode and ch1_mode should both be either '01' or '11') 0 Split mode is disabled. 1 Split mode is enabled.
3–2 ch1_mode[1:0]	LVDS channel 1 operation mode 00 Channel disabled. 01 Channel enabled, routed to DI0 10 Channel disabled. 11 Channel enabled, routed to DI1.
1–0 ch0_mode[1:0]	LVDS channel 0 operation mode 00 Channel disabled. 01 Channel enabled, routed to DI0 10 Channel disabled. 11 Channel enabled, routed to DI1.

Chapter 48
Low-Dropout Regulator (LDO)

48.1 Introduction

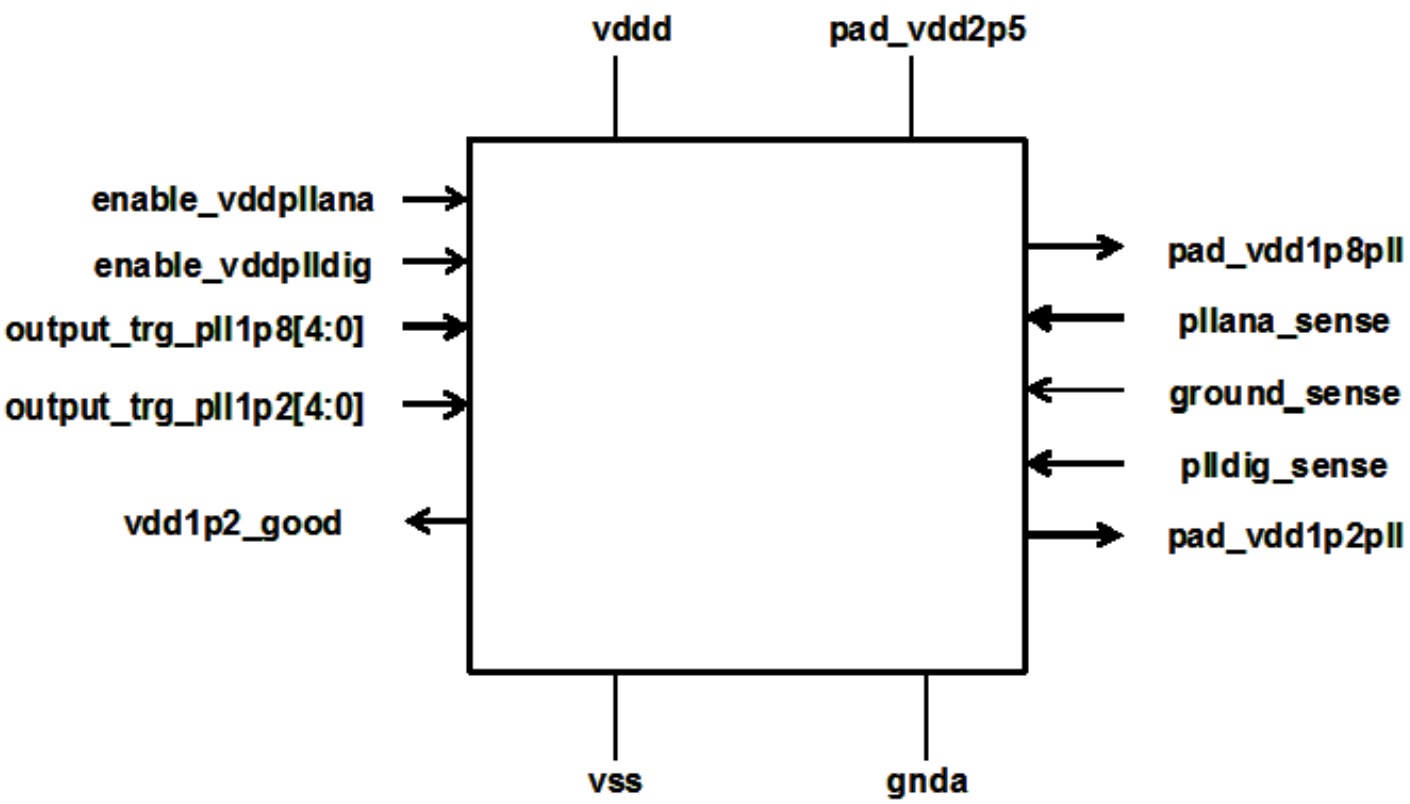


Figure 48-1. LDO Block Diagram

48.1.1 Overview

The LDO is an integrated 1.8 V/1.2 V linear regulator.

48.2 Features

The LDO includes the following list of features:

- Integrated 1.8 V/1.2 V linear regulators with independent controls
- Integrated bandgap reference for stable output voltage over process, temperature and voltage conditions
- Typical quiescent current consumption with everything enabled:
  - Analog 2.5 V Iddq = 0.5 mA
  - Digital 1.2 V Iddq = 0.2 uA
- Power-down current with everything disabled:
  - Analog 2.5 V Iddq = 8 uA
  - Digital 1.2 V Iddq = 0.2 uA

48.3 1.2 V Regulator (plldig) Specifications

- Input supply is 2.5 V±5%.
- 5-bit programmable output voltage in 25 mV steps: nominal output voltage is 1.2 V (0.9 V min 1.4 V max)
- Designed to provide a minimum of 250 mA output current under WCS conditions and nominal output voltage of 1.2 V
- DC output voltage stability is 3% over the ±5% input supply range and -40°C to 125°C junction temperature range.
- Minimum 22 uF external capacitor

48.4 1.8 V Regulator (pllana) Specification

- Input supply is 2.5V±5%
- 5-bit programmable output voltage in 25 mV steps: nominal output voltage is 1.8 V (1.5 V min 2.275 V max)
- Designed to provide a minimum of 75 mA output current under WCS conditions and nominal output voltage of 1.8 V
- DC output voltage stability is 3% over the ±5% input supply range and -40°C to 125°C junction temperature range
- Minimum 22 uF external capacitor

48.5 Register Definition

The 1.2V and 1.8V supply voltage are controllable via software. The 5-bit programming fields are implemented in the IOMUXC in the IOMUXC_GPR1 register.

Table 48-1. Configurable Output Voltage Bits

PLL1P8_VREG[4:0]	Defines the output for voltage regulators. For the 1.8 V regulator: 0x00 = 1.5 V and 0x1F = 2.275 V in 25 mV steps. This puts the reset value for 1.8 V at 0x0C
PLL1P2_VREG[4:0]	Defines the output for voltage regulators. For the 1.2 V regulator: 0x00 = 0.8 V and 0x1F = 1.575 in 25 mV steps. This puts the reset value for 1.2 V at 0x10.

Chapter 49
Multi Master Multi Memory Interface (M4IF)

49.1 Introduction

The Multi-Master Multi Memory Interface (M4IF) controls memory accesses from one or more masters through various port interfaces to external memory controllers (ESDCTL, NFC, and EIM), and to two internal memories in the system. The M4IF block diagram is shown in Figure 49-1.

49.1.1 Overview

M4IF is the external memory controller's (EXTMC) interconnect between system masters and external or internal memory controllers. M4IF is responsible for major functions in EXTMC. It contains the heart of arbitration logic for the memory interfaces. It contains the write and read buffers to support best performance data flow with minimum latency. It contains AXI port gaskets for each master. It manages the AXI access type and transfers it to the arbitration logic. From the arbitration, it moves on to a dedicated memory controller.

M4IF is capable of handling several AXI accesses of different types. In addition, all of EXTMC's handshake protocols with the SoC, such as Low Power Mode entry and DVFS, are handled by M4IF. M4IF interfaces between the SoC's requests and the internal logic, including memory controllers limitations.

M4IF contains the following sub-blocks:

49.1.1.1 AXI Port Gasket

This is the EXTMC interface to AXI masters. M4IF contains 8 AXI port gaskets. This samples the access signals, address and controls into a FIFO and sorts the accesses to their appropriate channels - fast, slow, internal1 memory or internal2 memory channels. The gasket controls the AXI interface and complies with the AXI bus protocol.

49.1.1.2 Dedicated Write Buffers

Each AXI port gasket (master) has a dedicated write buffer. Write data from a master is sampled in the buffer and later transferred to the memory controller upon request. The write buffer size varies from master to master.

49.1.1.3 Read Shared Buffers

The read shared buffer serves all 8 AXI ports. M4IF contains 4 read buffers, one for each channel: slow, fast, internal1 and internal2 memory channel. The data from the memory controller is written to the read buffer. The gasket provides the data back to the master when necessary. Current design configuration supports buffer size of 512Bytes (data only) for fast arbitration, 256Bytes (data only) for slow, and 128Bytes (data only) for internal 1 memory and internal 2 memory arbitration.

For exact buffer sizes please refer to Buffers Size Table.

49.1.1.4 Fast Arbiter

The Fast Arbiter arbitrates master's accesses to the enhanced SDRAM controller (ESDCTL) by using very comprehensive information to optimize accesses to the ESDCTL. It takes into account parameters such as page hit/miss, read/write access etc. If needed, the effect of this additional information can be disabled and the arbitration will function in a fixed priority mode, based on the user defined priority. More details on the dynamic priority arbitration can be found in the functional description section. Fast Arbiter Functional Description - 1st Degree.

49.1.1.5 Slow Arbiter

The Slow Arbiter arbitrates between the masters using a pre-defined bus division. It then sends the accesses to the external interface module (EIM) and NAND Flash Controller (NFC). The division is configurable with high resolution. Further details will be provided in Arbitration Scheme when Masters have Same Priority (Bus Division).

49.1.1.6 Internal 1 Memory Arbiter

The Internal 1 Arbiter arbitrates between the masters using a pre-defined bus division. It then sends the accesses to the internal RAM1 memory controller. The division is configurable with high resolution. Further details will be provided in Arbitration Scheme when Masters have Same Priority (Bus Division).

49.1.1.7 Internal 2 Memory Arbiter

The Internal 2 Arbiter arbitrates between the masters using a pre-defined bus division. It then sends the accesses to the internal RAM2 memory controller. The division is configurable with high resolution. Further details will be provided in Arbitration Scheme when Masters have Same Priority (Bus Division).

49.1.1.8 Debug Unit-Overview

The Debug Unit gives the ability to monitor several internal important EXTMC signals, in order to have better debug capability. It is recommended that these signals be routed out of the chip based on SoC debug definition. In addition, the debug unit would give the option to do a profiling of the arbitration unit of each channel. In that case the user would be able to trace its master bus performance and to fine-tune the arbitration's parameters. The debug unit contains several registers, so that the information can be read from the registers, observed on the pads, or analyzed by the profiling unit data. The debug unit is described in detail in the functional section.

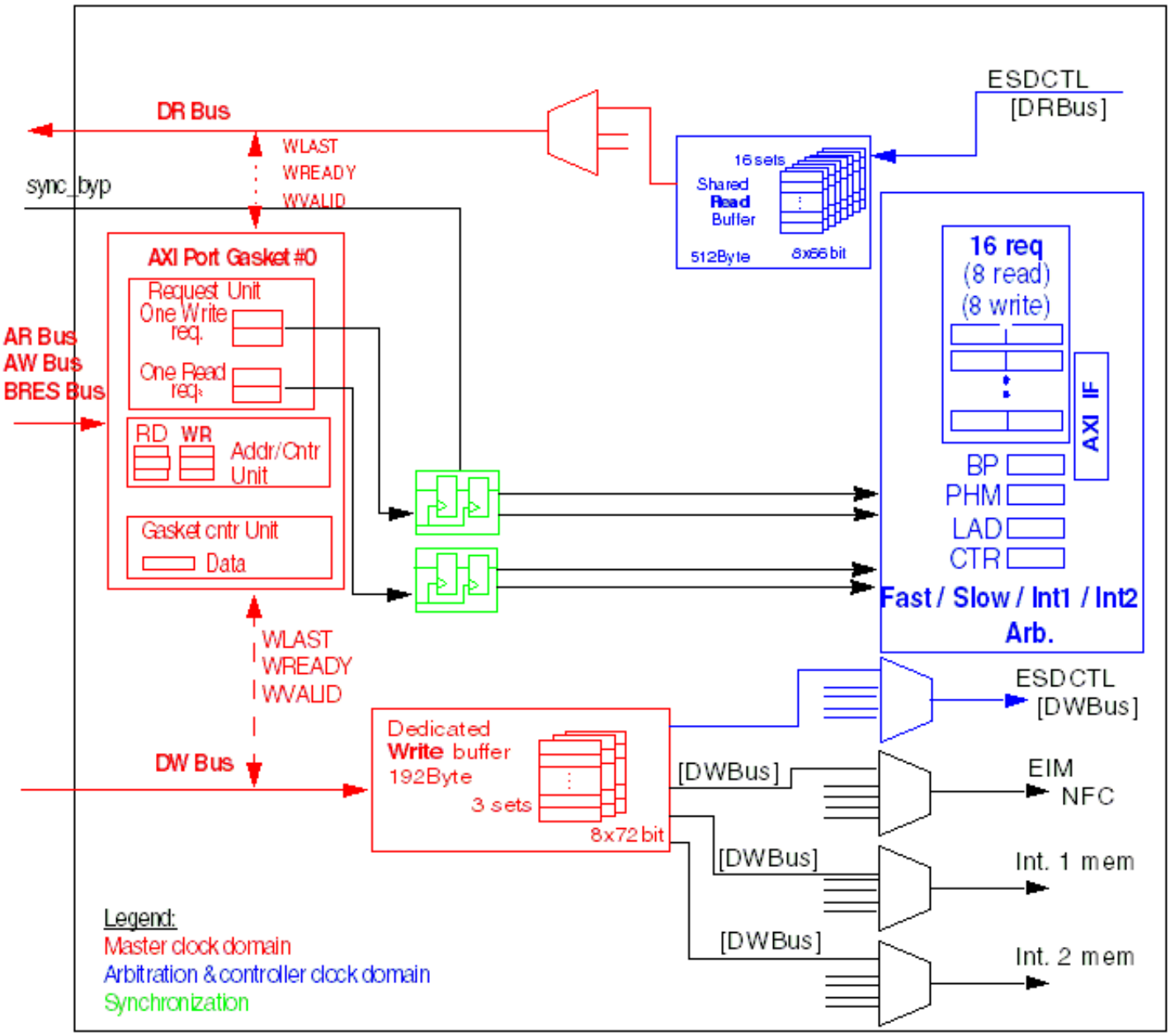


Figure 49-1. M4IF detailed block diagram

49.1.2 Features

The M4IF includes the following features:

- Supports multiple requests from up to 8 masters through different input ports interfaces. Each port can support either of the following two data width options:
  - x32 AXI port.
  - x64 AXI port.

- Arbitrates any master access to 5 main memory controllers, ESDCTL, EIM, NFC, internal 1 salve 0 / slave 1 memory controller and internal 2 memory controller at system level.
- Supports different asynchronous master clock domains for each AXI port, using synchronization system with bypass configuration option in case clocks are aligned and balanced.
- Supports memory watermark protection for up to 2 different chip selects (2 of ESDCTL for preselected masters).
  - Configurable two WM configurations, for two separate masters/domains.
  - Configurable protected memory region (base address with 4 KB resolution) for each one of the watermark memory regions.
  - DDR protected regions must not cross 256MByte boundary.
  - One status bit for each memory region that indicates security violation.
  - Maskable watermark interrupt generation capability in case of security violation for the two WM configurations.
- Separate read and write data buffers for better performance and minimum latency.
- Separate arbitration logic for fast memory interfaces - ESDCTL, slow memory interfaces - EIM & NFC, internal 1 slave 0 / slave 1 memory interfaces and for internal 2 memory interfaces like internal ROM and RAM.
- Separate clocks for each arbiter as mentioned above.
- Enhanced arbitration scheme for fast channel, consider page hit/miss, last access details (read/write), fixed priority configuration, M4IF mechanism and M4IF prediction.
- Support AXI exclusive and lock access per arbitration path, so that locked access of one of the masters to EIM does not lock all EXTMC and access to ESDCTL for example can still be served.
- Bus Division arbitration scheme with programmable resolution for the slow, internal memory and internal 2 memory arbitration logics.
- Supports low power mode techniques. User can put the entire M4IF or partial M4IF (slow arbitration only, etc.) into low power mode.
- Supports auto power saving features for minimum power consumption in idle mode.
- Supports warm reset to allow dynamic reset configuration with minimum system impact.
- Supports several debug capabilities as described in the following section.
- Parametric design capability

49.1.3 Modes of Operation

The following sub-sections describe operation modes supported in M4IF.

49.1.3.1 Normal Operating Modes

In the normal operation mode, M4IF can serve each master access read or write in parallel. All clocks are enabled, and READY signal of the AXI protocol is at high condition. Access from any master can be served. For more details about normal operation mode please refer to Functional Description.

49.1.3.2 Low Power Modes

Low Power mode of operation is supported in M4IF. A low power mode request coming from the SoC will start the low power mode routine. In this mode, no access can be fired through M4IF between any master and memory controller (external or internal). Entering into this mode will cause the whole EXTMC to be in low power mode, meaning DDR external devices will be in self refresh mode and new accesses will not be served. All arbiters will finish serving all pending requests and clean all the buffers. Pending accesses on the memory controller bus will be finished while address READY signals on AXI port gaskets will be LOW. All AXI gaskets will change their address READY signals to be in low state and will not get any new request from any master.

In parallel to this routine, M4IF will pass the LPMD request to memory controllers to initialize its low power mode sequence. At the end of this process each memory controller will send LPMD acknowledge back to M4IF. M4IF would send Low Power Mode Acknowledge signal to the system based on its own process finish flag and on memory controllers acknowledge. This will allow the system clock controller module to disable EXTMC clocks without any risk.

Getting out of low power mode is done by first enabling all EXTMC clocks and then changing low power mode request signal to LOW state. After enabling the internal sub-blocks, gaskets and arbiters, M4IF will change address READY signals in the gaskets to high level to enable new access entry to EXTMC.

In addition to the above, M4IF supports entry to low power mode by SW definition. Please refer to M4IF control register for more details.

M4IF also supports partial low power mode. The system can enter each memory arbitration path into low power mode separately, so that one path can be in idle mode while the other is in low power mode. For example, if DDR is currently working and no accesses to EIM or NFC are needed, the system can set lpmd_slow signal to high so that slow path will be entered into low power mode. Another option is to rely on auto power saving mode to allow M4IF to enter this path into low power mode automatically.

For more details on power saving mode please refer to Power Saving Mode

Please note, in partial low power mode, the READY signals at the AXI gaskets will not be affected. They would remain high as long as one of the arbitration paths are functional. It is the system responsible to stop accesses to an arbitration path, which is in low power mode.

This mode can help the user look in detail, every transaction that crosses each one of the arbiters.

NOTE

In order to get out of SW low-power modes, there has to be an ability to access the IP-registers in M4IF. If access to the M4IF IP-registers is done through M4IF, a dead-lock can occur. Because M4IF is in low-power mode, all AXI accesses through it are blocked and if the IP registers are accessed through the AXI paths, they will not go through. This will leave the system hanging, because it cannot release the SW low-power mode request.

It is the system's responsibility to make sure that this dead-lock will not occur as it is dependant of the M4IF connectivity in the SOC.

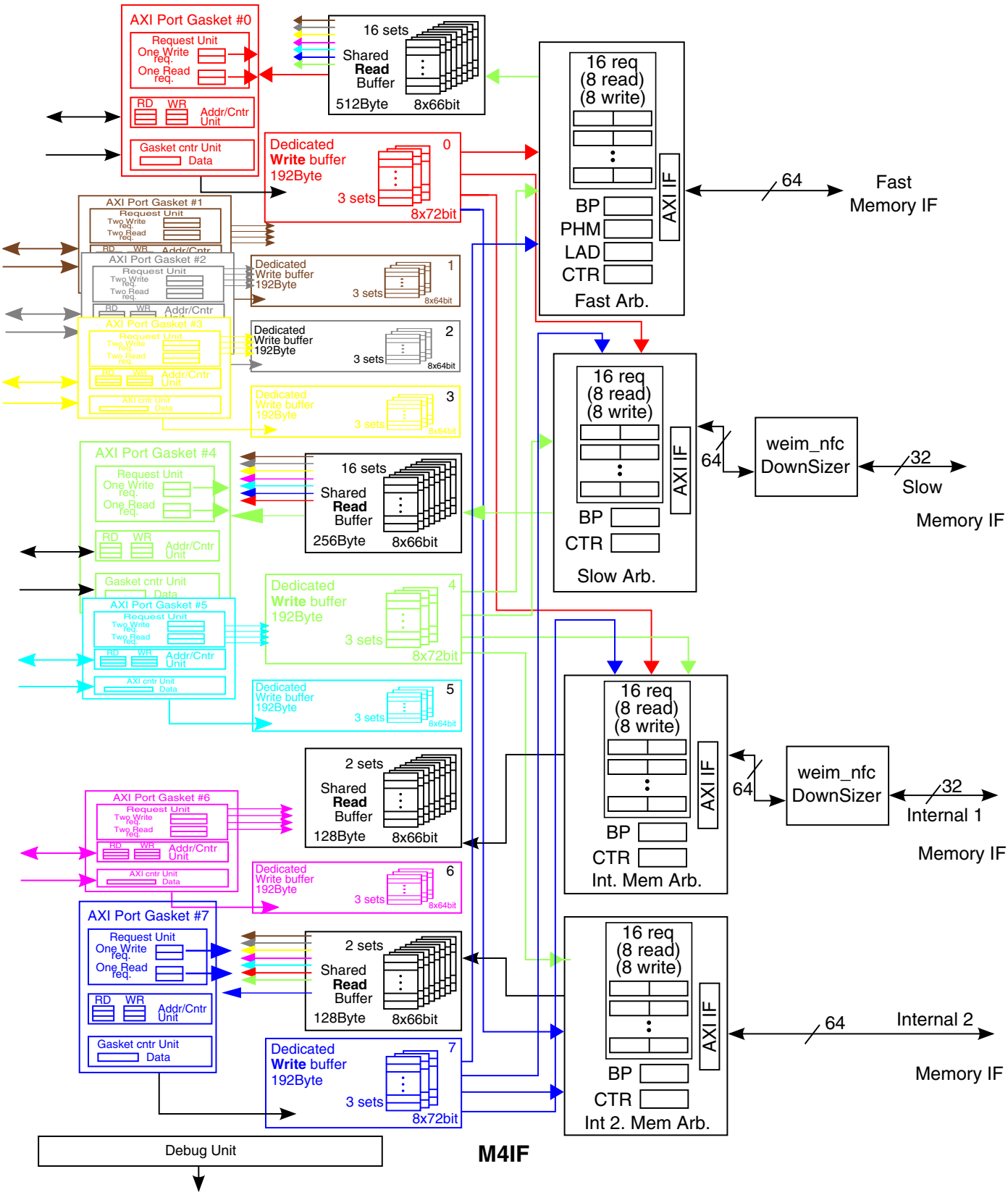


Figure 49-2. M4IF Block Diagram

49.1.3.3 Debug Mode

49.1.3.3.1 Step By Step Mode

This mode allows the user to freeze a specific arbitration after every new transaction that is sent outside the M4IF. The user can then monitor several internal signals of M4IF (while the specific arbitration is on hold) and later write to a register that will release the current transaction and freeze the next one.

Currently, there is no option to enter this mode in the middle of a working system. Entering this mode must be done when there are no transactions in the system, preferably immediately after reset, before any AXI accesses has been issued to M4IF/EXTMC.

Another way to ensure a quiet system is to use the LPMD/LPACK mechanism, either by HW or SW (see Control Register 0 (M4IF_CR0)). By using the LPMD mechanism, M4IF guarantees that all pending accesses have been served.

Once the "step by step" mode is enabled all accesses to the monitored slave (mem. controller) are halted. The properties of the access at the top of the PA fifo can be read via IP registers. (See Step By Step Address (M4IF_SSA).)

In order to release the access at the top of the fifo, the appropriate bit in the M4IF_MDCR (Debug Control Register (M4IF_DCR)) should be set. Either one of the 4 bits: FSBS,SSBS,ISBS or I2SBS depending on which arbitration is being halted.

49.1.3.3.2 Debug Unit - Functional Description

For details about functionality of the Debug Unit in M4IF please refer to Functional Description.

49.1.3.3.3 Debug Signals

The signal list that can be reflected outside is shown in Table 49-1 :

Table 49-1. M4IF Debug Signals

Signal Name	Number of Bits	Description
master_fast	[2:0]	EXTMC Master number of the selected access at fast arbitration bus.
master_slow	[2:0]	EXTMC Master number of the selected access at slow arbitration bus.
master_int1	[2:0]	EXTMC Master number of the selected access at internal 1 memory arbitration bus.

Table continues on the next page...

Table 49-1. M4IF Debug Signals (continued)

Signal Name	Number of Bits	Description
master_int2	[2:0]	EXTMC Master number of the selected access at internal 2 memory arbitration bus.
master_id_fast	[3:0]	Master ID of the selected access at fast arbitration bus.
master_id_slow	[3:0]	Master ID of the selected access at slow arbitration bus.
master_id_int1	[3:0]	Master ID of the selected access at internal 1 memory arbitration bus.
master_id_int2	[3:0]	Master ID of the selected access at internal 2 memory arbitration bus.
id_fast	[3:0]	AXI transaction ID of the selected access at the fast memory arbitration bus.
id_slow	[3:0]	AXI transaction ID of the selected access at the slow memory arbitration bus.
id_int1	[3:0]	AXI transaction ID of the selected access at the internal 1 memory arbitration bus.
id_int2	[3:0]	AXI transaction ID of the selected access at the internal 2 memory arbitration bus.
dyn_pr_fast	[5:0]	Fast Dynamic priority of a certain request. (configured by MDCR[20:16])
dyn_pr_slow	[5:0]	Equal 0, there is no Dynamic Priority mechanism in slow arbitration
dyn_pr_int1	[5:0]	Equal 0, there is no Dynamic Priority mechanism in internal 1 arbitration
dyn_pr_int2	[5:0]	Equal 0, there is no Dynamic Priority mechanism in internal 2 arbitration
acc_type_fast		access type of the selected access at fast arbitration bus.
acc_type_slow		access type of the selected access at slow arbitration bus.
acc_type_int1		access type of the selected access at internal 1 memory arbitration bus.
acc_type_int2		access type of the selected access at internal 2 memory arbitration bus.
addr_fast	[31:0]	AXI ADDRESS of the selected access at fast arbitration bus.
addr_slow	[31:0]	AXI ADDRESS of the selected access at slow arbitration bus.
addr_int1	[31:0]	AXI ADDRESS of the selected access at internal 1 memory arbitration bus.
addr_int2	[31:0]	AXI ADDRESS of the selected access at internal 2 memory arbitration bus.

The selected arbitration's signals are routed outside via a bus called: ipp_do_emi_debug. Only one arbitration can be viewed on the debug bus.

They are ordered on this bus in the following way:

ipp_do_emi_debug[50:0] = {valid_strobe,master,master_id,id,dyn_pr,access_type,addr}

where:

valid_strobe - a signal to indicate that a valid request is selected, this signal will be 3 clocks wide. The clock depends on the arbitration which is selected.

master - EXTMC master port (0:7), 3 bits wide. (Selected between: master_fast, master_slow, master_int1, master_int2)

master_id - system MASTER ID, 4 bits wide.

id - AXI transaction ID, 4 bits wide.

dyn_pr - the updated dynamic priority of a request, 6 bits wide.

access_type - write (0) or read (1), 1 bit.

addr - the address that is accessed (base address), 32 bits wide.

49.1.3.4 Dynamic Voltage and Frequency Scaling (DVFS)

DVFS is supported in M4IF. When DVFS request is high, M4IF sends DVFS request to the memory controllers while stopping all pending requests to the memory controllers. No new access will pass from M4IF to the memory controller but only piped access in the memory controller will be cleaned. At the end, the memory controller pipeline will be cleaned, and a DVFS ack signal will be sent back to M4IF, which will then send a DVFS ACK signal to the system to allow changing of frequency.

In addition to the description above, M4IF supports DVFS request to be enabled by register configuration. Please refer to Control Register 0 (M4IF_CR0) for more details.

49.1.3.5 Power Saving Mode

49.1.3.5.1 Arbitration Power Saving

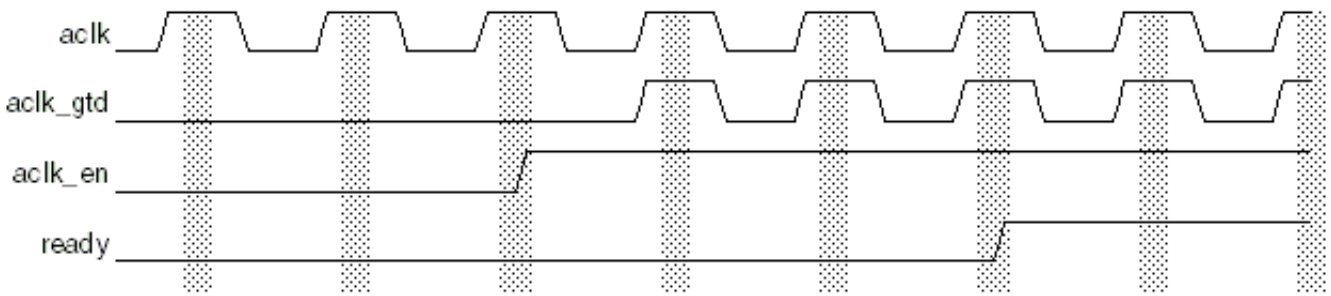
In order to support power saving mode, M4IF should be able to detect whether there is a request pending on any of the arbiters. The power saving register defines the time period that should pass from the last served request in a certain arbitration. If during the specified time period, no request was observed on the arbitration, the arbiter would send low power mode request to its memory controller and would enter itself in low power mode. The power saving mode is defined and functioned separately for each arbitration. In that case fast arbitration can be in normal mode while the slow arbitration can be in power saving mode. Getting out of this mode would be initiated automatically when a new request to a specific arbitration is issued by one of the masters. It is important to mention that it would take few cycles to serve the request since getting out of power saving mode routine takes a few cycles in M4IF as well as in the memory controller.

49.1.3.5.2 Gasket Power Saving

Power saving feature is also supported for each gasket individually. The gasket will detect the number of cycles it was idle. Upon reaching this configurable number of cycles, the gasket will de-assert its clock gating enable, and almost all the FFs inside will stop. There will be a few FFs still running on a free running clock, in order to be able to wake up from this mode. In this mode the 3 ready signals: AWREADY, ARREADY and

WREADY will be de-asserted until an access is issued to this gasket. When detecting a new access (AWVALID, ARVALID or WVALID asserted), the gasket will assert its clock enable signal (in the next cycle). There is a minimum number of cycles that the READY signals have to be de-asserted for, after the clock enable is asserted. Please note these are master clock cycles. For each master the clock period is different. The number of cycles is configurable via a register. The number of idle cycles that will initiate this process is also a configurable number. Refer to Power Saving Masters 0 (M4IF_PSM0) for further details.

The following diagram illustrates the behavior of the clock enable and of the ready signals upon exiting from power saving. In this diagram the "ready off cycles" (ROC) is at its default value - 3



49.1.3.5.3 SW Power Saving

M4IF does not support getting into power saving mode by SW. However, enabling/disabling this mode can be done by SW configuration. Please refer to M4IF control registers for more details. (PSM0-PSM3 for gaskets and MCR0 for arbitration).

Control Register 0 (M4IF_CR0)

Power Saving Masters 0 (M4IF_PSM0)

Power Saving Masters 1 (M4IF_PSM1)

Power Saving Masters 2 (M4IF_PSM2)

Power Saving Masters 3 (M4IF_PSM3)

NOTE

Power saving can only be configured once after reset. Changing the power saving configurations after the system was active (even if now idle) will result in unexpected behavior.

49.1.3.6 Error Handling And Interrupts

49.1.3.6.1 LEN > 8

M4IF does not support AXI accesses longer than 8 beats of data. In the event of an access of this kind arriving to M4IF, it will not be served. The master which sent the response will get an error response (DECERR). For read accesses, the read data will be returned as 0.

In addition to the error response, an interrupt will be issued (LEN interrupt). A status register will indicate which master made the un-supported access. This register will also be used to clear the interrupt. Please refer to Master Len Interrupt (M4IF_MLI).

49.1.3.6.2 Watermark

For detailed description about the watermark feature please refer to WaterMark Functionality Overview.

When a watermark violation occurs, it will be captured at the AXI gasket, and will not be served. The master which sent the request will get an error response (DECERR). For read accesses, the read data will be returned as 0.

In addition to the error response, one of two interrupts will be issued depending on the region which was violated (ARM Platform or Baseband Platform). Status registers will indicate which CS was violated. These registers will also be used to clear the interrupt. Please refer to Watermark Interrupt and Status 0 Register (M4IF_WMIS0).

49.2 Functional Description

This section and the sub-section below provide a functional description of the internal logic of M4IF. This section describes how the M4IF provides access paths between the system and the memory controller. The sub-sections below describe in more detail each part of M4IF and its functionality.

49.2.1 Write Access Description

An AXI access is detected on one or more AXI port gaskets and latched into the addr/ctrl internal buffer. The data buffer is separate from the addr buffer. This enables data to be accepted into the data buffer even before the address arrives on the bus. When WLAST

signal is detected on the bus, an internal logic inside the gasket generates a request to the right arbiter based on the address decoding. Because the request is generated in the master clock domain it is synchronized to the arbitration clock domain. This way it is guaranteed that all the data is stored in the buffer when the request is valid in the arbitration.

The arbiter calculates priority to this new valid request and chooses it according to the arbitration's criterion. See Fast Arbiter Functional Description - 1st Degree for further details. At the moment the access is chosen by the arbitration, it passes to the memory controller. The memory controller gets the data from the write data buffer.

In terms of write response, the gasket sends a bufferable response on the AXI response channel the moment the last data has reached the specific memory controller. In case of a non-bufferable access the response would be sent only after the write access is finished in the memory controller, and the memory controller itself sends a response.

49.2.2 Read Access Description

In a similar way when the access is a read access, the AXI addr and control signals are latched in a read control buffer inside the gasket. They are then transferred to the right arbiter according to the address decoding in the gasket. Based on the arbitration scheme the request is served. The addr and control signals are sent to the memory controller. Data that comes out of the controller is written directly to the read buffer. The read buffer is dedicated to the specific memory controller, and is shared between all 8 AXI masters. The master can pull the data out of the buffer at the moment it gets RVALID signal from the memory controller.

A detailed description of M4IF's functionality is described in the following sub-sections.

49.2.3 AXI Port Gasket Functional Description

The AXI port gasket is an AXI interface to any master in the system. The gasket is a x64 AXI bus interface and can be fitted to a x32 AXI interface. In order to connect a x32 master to the gasket, the wdata and wstrb bus should be duplicated. The rdata bus should connect to the lower 32 bits out of the 64 that come out of the read buffer.

The gasket works with Data Interleaving Depth (DID) of 1. This is a big difference from previous versions of M4IF, where the gasket worked with DID of 2. It can hold as many accesses as its buffer depth (which is parametric) regardless of the AXI ID. This is another noticeable change from the previous M4IF, which was limited to 2 pending IDs.

Each gasket can generate up to two parallel requests to the arbitration logic, one for write and one for read. All requests can be valid in parallel to the same or to different arbiters.

The read addr FIFO and read requests logic is similar to the write logic. The same FIFO exist also for other AXI control signal so it can give the whole information to the arbitration whenever a request is valid at the arbiter.

The write data comes on the bus from the master directly into the write buffer. Because the write buffer is dedicated to each master, that data is latched in the same clock domain of the master and no synchronization is needed.

49.2.4 Read/Write Buffer Functional Description

M4IF includes up to 12 data buffers, eight write data buffers, and four read data buffers. The write buffers are dedicated for each master in the system. If a master is removed from the system, its gasket and its write buffer are removed, saving a lot of area. The read data buffers are dedicated for each arbitration. In the read buffer, data is written by the specific controller and pulled out by any of the eight masters, while in the write buffer the dedicated master writes the data into the buffer and each of the four arbiters can pull out the data.

Therefore, the data in the write buffer is written in the master clock domain, while the data in the read buffer is written in the specific arbitration clock domain.

Both of the read and write buffer types are divided into fixed data slots in which data can be written. The slots are of depth eight (the maximal burst length) and are 64 bits in size. In addition to the data, the buffers store the write strobe (8 bits) for write, and the read response (rresp, 2 bits) for read. Each slot has a pointer_in and a pointer_out to mange the two way data, the way in which it is written into the slot and the way it is pulled out of the slot. It is important to mention that data of a certain access can not be stored in two different slots.

There are two configuration options to pull the data out of the read buffers for the fast arbitration control only, "read through" and "store & forward". In "store & forward" operation mode, the data can be pulled out by the slave only when all the datum of the burst is stored in the buffer, that is, after RVALID, RREADY and RLAST signals are high on the bus. In "read trough" operation mode, the data can be pulled out of the buffer by the slave a few cycles after the first data was written to buffer, due to synchronization reasons. For fast memory controllers it is recommended to use the "read through" (default) and therefore to achieve a minimum latency between the controller and the master.

On the write data buffer it is simpler; there is only one operation mode, the "store & forward" mode. In order to achieve best performance, the arbitration guarantees that when a request is served and moves to the memory controller, all the data is already latched in the write buffer and can be pulled out by the memory controller without any busy or abort cycles in between each data word of the access. This happens because the access request is asserted at the arbitration only after the last data was written to the buffer.

The write buffer gets data from its master. Data can be pulled out of the write buffers by the four memory controllers in the parallel. In the read buffer, only one memory controller can write to the buffer, but several masters can pull the data at the same time, depending on the number of slots in the buffer.

49.2.5 Fast Arbiter Functional Description - 1st Degree

The fast arbiter is responsible for all the accesses that go to ESDCTL memory controller, DDR2/DDR3 memory devices. As was described in the sections before, the addr decoding is done at the AXI port gasket. Any access that is pointed to the ESDCTL memory controller would be sent to the fast arbiter. The fast arbitration arbitrates between all masters request, read and write. The overall number of request is 16, 8 request for read and 8 for write. Each request can be valid on the arbitration logic at any time, it is guaranteed by EXTMC design that whenever a write request is valid on the arbitration logic all the data is already stored in the write buffer. This means that data is ready on the bus between the buffer and the specific memory controller whenever a specific request is chosen by the arbiter. The fast arbitration uses a dynamic priority engine to perform a high performance low latency data path arbitration. The dynamic arbitration uses the following parameters in its engine and calculate the priority of each request accordingly.

49.2.5.1 Page Hit / Miss

DDR memory device is divided into several banks depending on the specific memory configuration detailed by the memory manufacturer. Before any READ or WRITE commands can be issued to a bank in the DDR, a row in that bank must be opened. This is accomplished by the ACTIVE command, by which both the Bank and the Row are selected. More than one bank can be active at any time. Once a row is open, a READ or WRITE command could be issued to that row. A subsequent ACTIVE command to another row in the same bank can only be issued after the previous row has been closed. The minimum time interval between two successive ACTIVE commands on the same bank is defined by tRC timing parameter of the DDR device. A Subsequent ACTIVE

command to another bank can be issued while the first bank is being accessed, which results in a reduction of total row-access overhead. The minimum time interval between two successive ACTIVE commands on the different banks is defined by tRRD timing parameter of the DDR device.

Therefore, selecting access to the same Bank and the same Row one after the other would be issued in a minimum time at the controller. The arbitration would give high score to a page hit access (same Bank and same Row) than any other access in order to get the highest performance from the controller. (The score level can be configured by the user.)

The decoding of the address to bank, row and cs is done in the gasket per each access that arrives from the master. It is later passed on to the "scoring" mechanism in the fast arbitration, in order to calculate the appropriate score for each request. The order of elements on the address is: {cs,bank,row,column}. There is an option to work in "bank interleaving" or "address interleaving" mode. In this mode the order would be: {cs,row,bank,column}. For further details, please refer to the ESDCTL spec.

49.2.5.2 Last Access Details

In order to perform high density on the DDR external bus it is needed to consider last access type when choosing the next request (access) on the arbitration to be served. In addition to Page Hit / Miss parameter in the arbitration, a consecutive READ command (READ after READ) would be issued faster than a WRITE after READ. The same is true for a consecutive WRITE command, (WRITE after WRITE).

The dynamic arbitration logic would prefer a consecutive command access in order to get the highest performance from the controller.

49.2.5.3 Basic Priority Configuration

In continue to the FBP register description, each master in EXTMC will have a basic priority configuration based on the user configuration. This basic priority value will be the base value for each master. The dynamic arbitration, as described below, starts its calculation from the basic priority value per each master.

49.2.5.4 Priority Calculation

The Fast arbiter is designed to arbitrate read and write AXI accesses from a maximum number of 8 masters towards the ESDCTL memory controller.

This arbitration takes into account several parameters in order to decide which access to the controller is the most efficient one of all pending requests.

The arbitration handles a dynamic priority mechanism to allow for improved decision making between the masters. Each master gets a basic priority, which is configurable via a Basic Priority Register, defined below. A dynamic priority as well as page/hit miss score, are added to this basic priority. Thus a total score is formed and according to this score the request is evaluated by the "decision mechanism" of the arbitration.

The dynamic priority is calculated per each request. Each master has 2 possible requests (1 read and 1 write) and therefore a total of 16 dynamic priorities are evaluated per arbitration. If a request is pending on the "decision mechanism" and other requests are chosen at that time, then after a defined amount of selections (configurable by register), the dynamic priority of this request will increase by a defined number of "points" (also configurable via a register). If, however, a request is chosen, then at that cycle the dynamic priority of this request will be decreased by the same number of "points" times the number of data words in the burst (the number of cycles it was taken to served the access).

There is a maximum limit for the dynamic priority. The limit is the same for all masters, and can be configured via a register.

When a master's priority reaches its limit, it cannot surpass that limit. If theoretically, there is a situation in which the limit needs to be exceeded, the priority will remain at its limit value.

Figure 49-3 illustrates the dynamic priority mechanism for a limit offset of 50 "points":

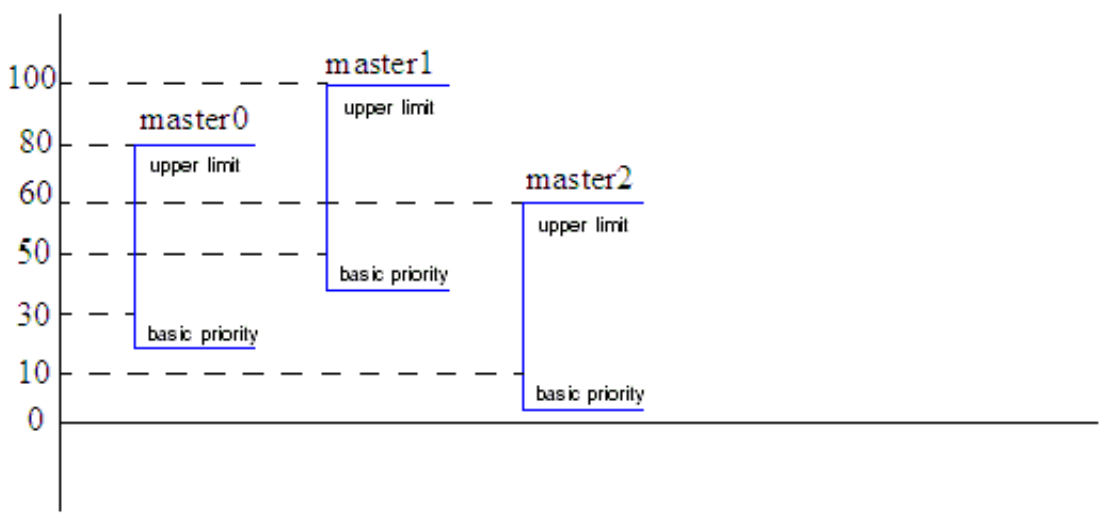


Figure 49-3. Master's Dynamic Priority Scheme

From the above illustration, it is seen that one master can "take control" over the arbitration if its basic priority is too high in comparison to the other masters.

Therefore, the system configuration needs to be such that the odds of such a situation would be as low as possible. The default settings of the fast arbitration and the entire M4IF system must be determined based on intensive modeling simulations. Changes to these values must be made with extreme care.

In case all of the masters have the same score, a simple arbitration logic is used to chose the next request to be served.

49.2.5.5 2nd Degree Arbitration (M4IF)

After choosing the best suited request out of the possible 16, it is latched inside a buffer of size 16 and then it goes through second degree arbitration. The requests which are stored in this buffer are arbitrated in a similar way to the first degree. Here there's a simpler dynamic priority mechanism. Each request gets a single point for each time another request was chosen over it. When a request is chosen its dynamic counter is reset. The page hit and access hit get smaller amount of points. Not 10 and 5 (default values) like in the first degree, but 4 and 2. These parameters are configurable using the M4IF control register. See Control Register (M4IF_CR).

The order of accesses from the same master with the same AXI ID is kept. Even if an access has a high score due to page and access hit, it will not be able to bypass an access from the same master with the same AXI ID that arrived before it.

From the second degree arbitration, the selected request is stored in a FIFO of depth 2. This FIFO handles the AXI accesses between the M4IF and the ESDCTL.

The second degree arbitration also has three more features in its engine:

49.2.5.5.1 M4IF Bypass

It is possible to prioritize a single master over the others in a form of bypass. In the M4IF control register (Control Register (M4IF_CR)) the fields: M4IF_MAS, M4IF_ID,M4IF_IDTOO and M4IF_BYP_EN can be used to do that. Once the M4IF_BYP_EN bit is set to '1', this feature is enabled. There are 2 options for bypass:

1. All accesses from a specific master configured in the M4IF_MAS field will bypass the M4IF arbitration and go directly to the ESDCTL. In this case the M4IF_IDTOO bit should be '0'.

2. Only accesses from M4IF_MAS that have a specific AXI ID, that is configured in the M4IF_ID field, will bypass the M4IF arbitration. For this option to be enabled the M4IF_IDTOO bit must be set to '1'.

Normally, the M4IF bypass feature will be used for IPU's screen refresh channels. Screen refresh channels have a real time nature. It is recommended to have the M4IF bypass feature enabled in any use case where IPU screen refresh is involved.

49.2.5.5.2 Guarding Mechanism

The second degree arbitration is guarded from starvation by using a counter that prevents a specific request from being stuck inside the 2nd degree buffer for more then a predefined number of clock cycles. For more information about this mechanism please refer to the M4IF Scoring register Control Register (M4IF_CR).

49.2.5.5.3 Prediction

The second degree arbitration has the ability to predict the {cs,bank,row} that will be used by the ESDCTL before the transaction is actually sent on the AXI bus. By doing so, we are able to prepare the memory device for future transactions and improve our overall performance. For more information regarding this feature please refer to the ESDCTL spec.

49.2.6 Slow Arbiter Functional Description

The slow arbiter is responsible for all access that goes to EIM and NFC, Nor / PSRAM memory devices or NAND devices. The addr decoding is done at the AXI port gasket. Any access that is pointed to one of these memory controllers would be sent to the slow arbiter. The slow arbiter arbitrates between all masters read and write request. The overall number of requests is 16, 8 requests for read and 8 for write. Each request can be valid on the arbitration logic at any time. EXTMC design guarantees that whenever a request is valid on the arbitration logic all the data is already stored in the write shared buffer. It means that data is ready on the bus whenever a specific request was chosen by the arbiter. The slow arbitration uses the "bus division" mechanism, as described in Arbitration Scheme when Masters have Same Priority (Bus Division).

49.2.7 Internal 1 Memory Arbiter Functional Description

The Internal 1 memory arbiter is responsible for all access that goes to system internal 1 memory controller, like ROM and RAM devices. As described in previous sections, the addr decoding is done at the AXI port gasket. Any access that is pointed to internal 1 memory space is sent to the internal 1 memory arbiter. The internal 1 memory arbiter arbitrates between all master requests, read and write. The overall number of requests is 16, 8 requests for read and 8 for write. Each request can be valid on the arbitration logic at any time. EXTMC design guarantees that whenever a request is valid on the arbitration logic all the data is already stored in the write shared buffer. It means that data is ready on the bus whenever a specific request was chosen by the arbiter. The internal 1 memory arbitration uses the "bus division" mechanism, as described in Arbitration Scheme when Masters have Same Priority (Bus Division).

49.2.8 Internal 2 Memory Arbiter Functional Description

The Internal 2 memory arbiter is responsible for all access that goes to system internal 2 memory controller, like ROM and RAM devices. The addr decoding is done at the AXI port gasket. Any access that is pointed to internal 2 memory space is sent to the internal 2 memory arbiter. The internal 2 memory arbiter arbitrates between all master requests, read and write. The overall number of requests is 16, 8 requests for read and 8 for write. Each request can be valid on the arbitration logic at any time. EXTMC design guarantees that whenever a request is valid on the arbitration logic all the data is already stored in the write shared buffer. It means that data is ready on the bus whenever a specific request was chosen by the arbiter. The internal 2 memory arbitration uses the "bus division" mechanism, as described in Arbitration Scheme when Masters have Same Priority (Bus Division).

49.2.9 Arbitration Scheme when Masters have Same Priority (Bus Division)

The following section describes the arbitration mechanism when the priority of two or more master requests are the same value. It is most likely that this scheme will be dominant in the slow and internal memory arbitration, where the special priority parameters are disabled or even when the dynamic priority is disabled.

The low level arbitration scheme exist in all arbiters. It is built of several levels of the same logic units which are connected in a tree structure. Figure 49-4 describes the general build of the low level arbitration structure.

Each level of arbitration (4 levels total) is built of the same logic unit. Each logic unit can be configured to a different arbitration scheme up on 3 input signals that can be partially configured by the user. It is important to mention that the logic unit follows its configured arbitration scheme only when both inputs (masters or requests) are with the same priority level.

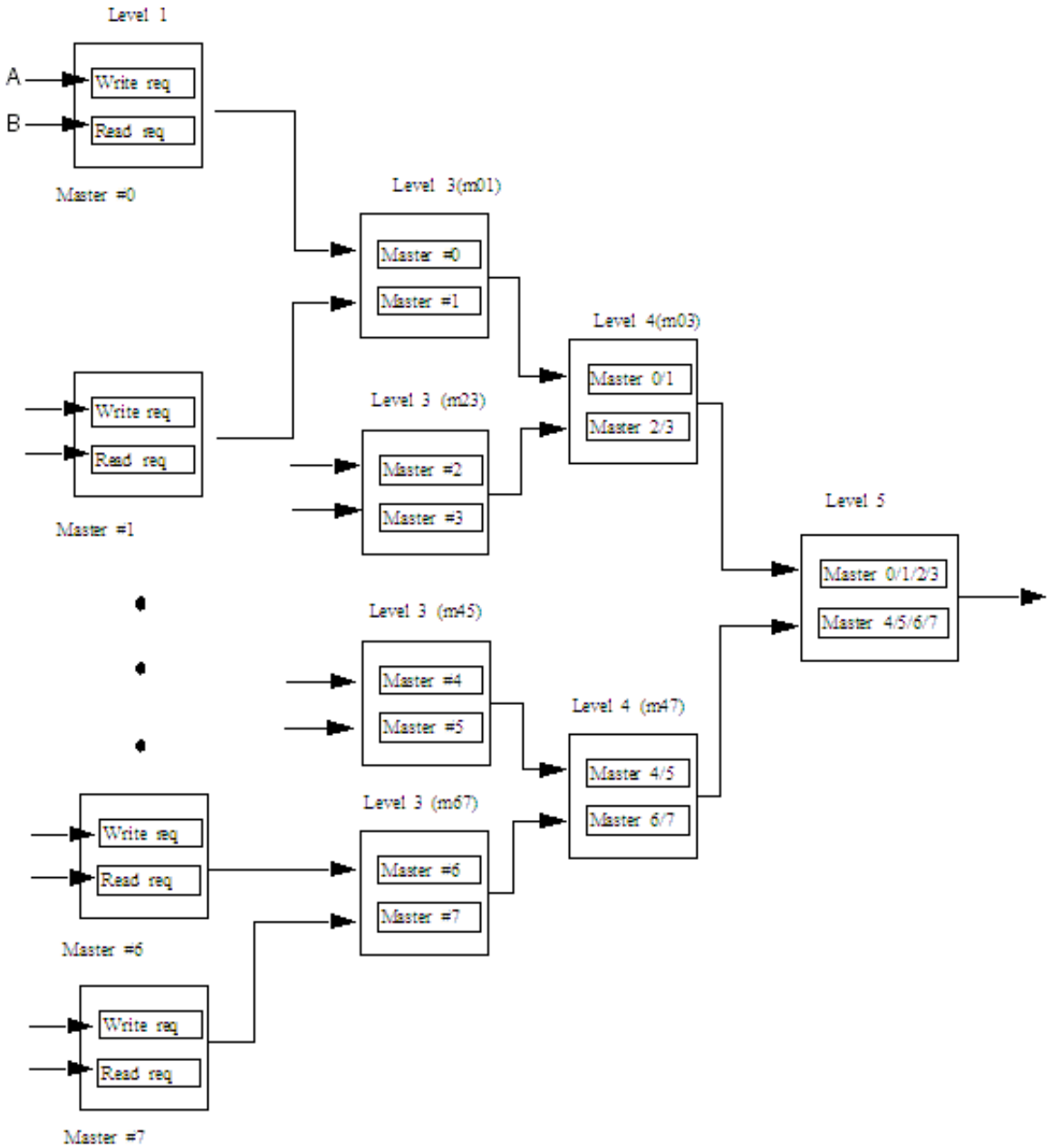


Figure 49-4. Low level arbitration structure

Table 49-2. logic unit fast arbitration configuration

#	bit 2	bit 1	bit0	arbitration scheme
0	0	0	0	100% priority to B.
1	0	0	1	25% priority to A
2	0	1	0	single 50%
3	0	1	1	double 50%
4	1	0	0	75% priority to A
5	1	0	1	100% priority to A
6	1	1	0	Reserved
7	1	1	1	Reserved

The user can configure only part of the logic unit levels. Level 1 has a fixed priority scheme configuration and can not be changed, while levels 3, 4, and 5 can be configured for one of the arbitration configurations as shown in Table 49-2.

Figure 49-5 describes schematically the arbitration scheme of each condition.

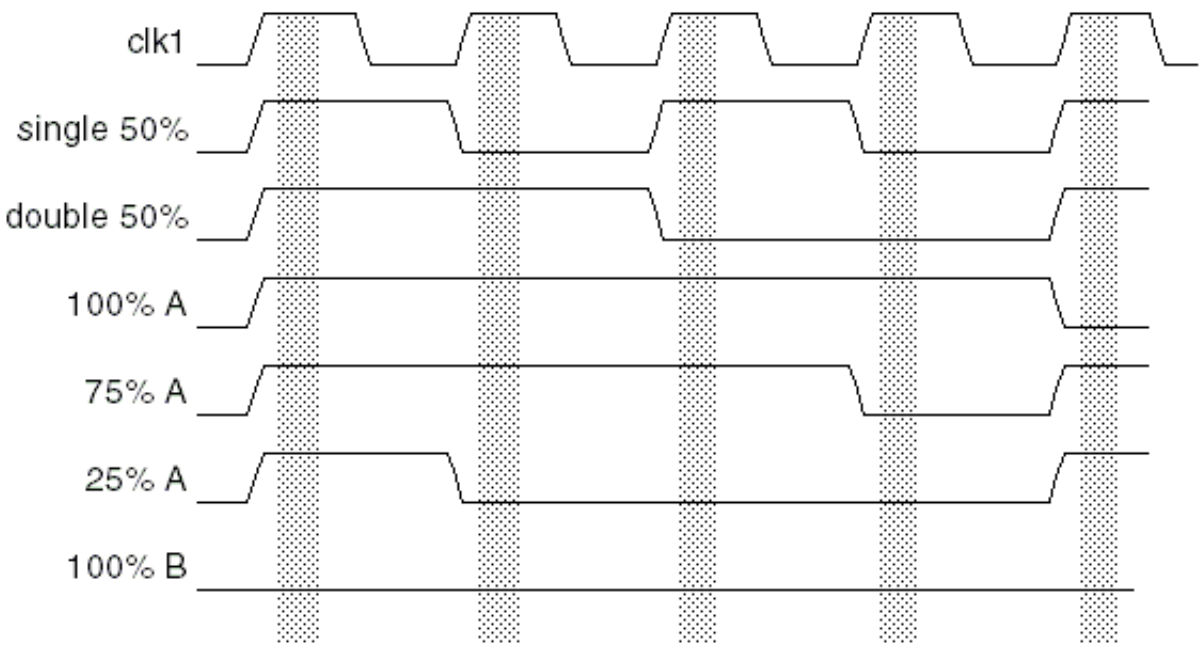


Figure 49-5. Logic units arbitration scheme.

The configuration of the logic units in levels 2, 3 and 4 can be done via the F_Unit_Level_Arbitration_ Register (M4IF_F_ULAR). However, configuration of logic units in level 1 is fixed. Unit level 1 is hard wired to 'double 50%' state.

For 'slow' and 'intr' arbitrations, the scheme has changed from previous versions of M4IF. Now there are more configurations that were added, as described in the table below:

Table 49-3. slow and intr arbitration bus division configuration

#	bit 2	bit 1	bit0	arbitration scheme
0	0	0	0	100% priority to B.
1	0	0	1	12.5% to A.
2	0	1	0	25% to A.
3	0	1	1	37.5% to A.
4	1	0	0	50% - 50%
5	1	0	1	62.5% A.
6	1	1	0	75% to A.
7	1	1	1	100% to A.

49.2.10 EIM-NFC Downsizer

The slow channel of EXTMC ultimately splits into 2 memory controllers, EIM and NFC. These 2 controllers' AXI data bus width is 32 bits. The slow arbitration has an AXI data bus width of 64. Therefore, a 64bit_to_32bit downsizer is required. Such a downsizer is placed inside the M4IF between the slow arbitration and the AXI interfaces to EIM and NFC. The downsizer handles bursts of length up to 8 (awlen/areln <= 7) for 64 (awsize/arsize=3) and 32 (awsize/arsize=2) bits. As for 8 and 16 bit accesses, the downsizer only supports single accesses (awlen/arlen=0). The downsizer is also a splitter and can determine the destination of an accesses, whether EIM or NFC, based on the address of that access.

Single length bursts will always have an output burst type (awburst_weim or awburst_nfc) INCR (2'b01). The awsize field, in case of 64bit accesses, will change from 64bit (2'b11) to 32bit (2'b10).

When sending burst of size 32 bit through the downsizer, wstrb[7:4] must be tied to wstrb[3:0].

49.2.10.1 Endianess In Downsizing (EIM-NFC Downsizer)

The downsizer takes into consideration the endianess of each accesses. If the endianess as LE (0) then the data for addresses with 3 LS bits decoded to: 0, 1, 2 or 3, will be taken from the lower 32 bits out of 64. The data for addresses with 3 LS bits decoded to: 4, 5, 6 or 7 will be taken from the upper 32 bits. If the endianess is BE (1) then for addresses

with 3 LS bits decoded to: 0, 1, 2 or 3 data will be taken from the upper 32 bits, and data corresponding to addresses with 3 LS bits decoded to:4, 5, 6 or 7 will be taken from the lower 32 bits.

NOTE

The information about the destination of an AXI access is not taken into account in the arbiter. The slow arbitration does not select accesses based on the memory controller's condition (busy /idle, etc.).

49.2.11 Internal 1 Downsizer

The internal 1 memory channel of EXTMC ultimately splits to 2 memory controllers, INT1_S0 and INT1_S1. These 2 controllers' AXI data bus width is 32 bits. The internal 1 memory arbitration has an AXI data bus width of 64. Therefore, a 64bit_to_32bit downsizer is required. Such a downsizer is placed inside the M4IF between the internal 1 memory arbitration and the AXI interfaces to INT1_S0 and INT1_S1. The downsizer handles bursts of length up to 8 (awlen/areln <= 7) for 64 (awsize/arsize=3) and 32 (awsize/arsize=2) bits. As for 8 and 16 bit accesses, the downsizer only supports single accesses (awlen/arlen=0). The downsizer is also a splitter and can determine the destination of an accesses, whether INT1_S0 or INT1_S1, based on the address of that access.

Single length bursts will always have an output burst type (awburst_int1_s0 or awburst_int1_s1) INCR (2'b01). The awsize field, in case of 64bit accesses, will change from 64bit (2'b11) to 32bit (2'b10).

Whenever sending burst of size 32 bit through the downsizer, wstrb[7:4] has to be tied to wstrb[3:0].

49.2.11.1 Endianess In Downsizing (Internal 1 Downsizer)

The downsizer takes into consideration the endianess of each accesses. If the endianess as LE (0) then the data for addresses with 3 LS bits decoded to: 0, 1, 2 or 3, will be taken from the lower 32 bits out of 64. The data for addresses with 3 LS bits decoded to: 4, 5, 6 or 7 will be taken from the upper 32 bits. If the endianess is BE (1) then for addresses with 3 LS bits decoded to: 0, 1, 2 or 3 data will be taken from the upper 32 bits, and data corresponding to addresses with 3 LS bits decoded to: 4, 5, 6 or 7 will be taken from the lower 32 bits.

NOTE

The information about the destination of an AXI access is not taken into account in the arbiter. The internal 1 memory arbitration does not select accesses based on the memory controller's condition (busy/idle etc.)

49.2.12 Clocks - i.MX53 Specific

M4IF contains the following clocks:

- ESDCTL main clock (up to 400MHz)
- Slow arbitration clock (up to 133MHz)
- Internal 1 memory arbitration clock (up to 166MHz)
- Internal 2 memory arbitration clock (up to 166MHz)
- 8 x masters clocks, can be asynchronous to EXTMC arbitration clocks (up to 200MHz)
- IP Bus clock (up to 66MHz)

M4IF assumes all of its 4 main clocks: aclk_fast, aclk_slow,aclk_intr and aclk_int2 are on by default (after reset). If the system wishes to turn one of the clocks off, it must do so by going through the specific LPMD sequence. For example, if the system wishes to turn "aclk_fast" off, it should wake up with this clock on, send an "lpmd_fast" request, wait for "lpack_fast" acknowledge, and only then turn "aclk_fast" off. "lpmd_fast" will have to be asserted for the whole duration of "aclk_fast" being off.

Due to synchronization issues, the general "LPMD"/"DVFS" sequences will only be available when "aclk_fast" is on. If "aclk_fast" is turned off, these sequences will not work.

The AXI master clocks, clocks aclk_m0-aclk_m7 are assumed to be "always on". These clocks can only be turned off as a result of proper LPMD sequence. Turning the clocks off without a proper LPMD sequence will result in an unexpected behavior, probably even a dead-lock.

49.2.12.1 Clock Ratios

The IP Bus clock's (ipg_clk_s) frequency should be slower (at least 1/2) than that of aclk_fast, aclk_slow, aclk_int1, aclk_int2 and all the master clocks at all times.

49.2.13 Reset

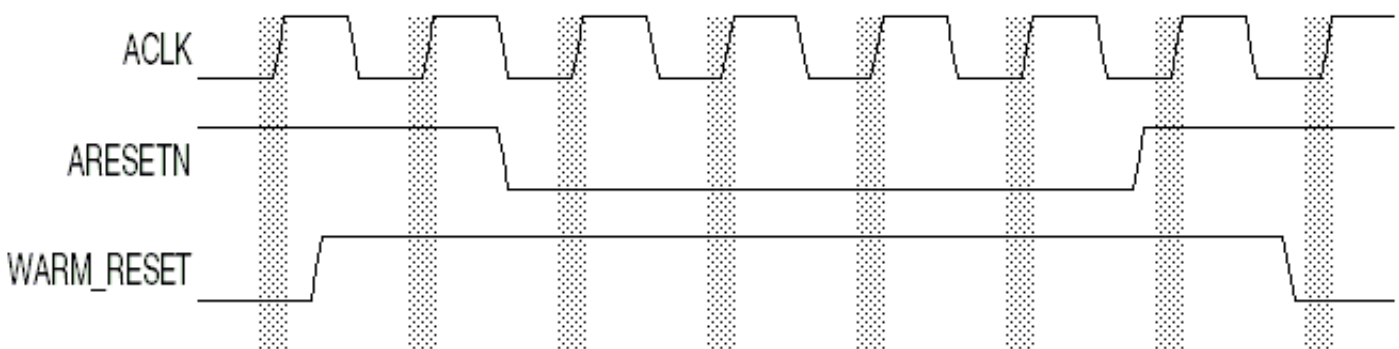
M4IF, as well as the entire EXTMC, supports 3 main reset signals, hard_reset, warm_reset and sw_reset. The reset tree at M4IF as well as in EXTMC is built in a way it provides the maximum flexibility to SoC so it can control the reset scheme according to its needs.

49.2.13.1 Software Reset

The software reset is a reset given to the entire M4IF FFs, except the IP configuration registers. This is done via IP write to bit 31 (SW_RST) in the MCR0 register. Please refer to Control Register 0 (M4IF_CR0).

49.2.13.2 Warm Reset

Warm reset is a reset given to the whole system except the configuration register. The warm reset is issued by asserting the regular reset (aresetn) and by asserting the "warm_reset" signal along with it. The "warm_reset" signal must be asserted before the assertion of the "arestn" signal, and must be deasserted after the deassertion of the "aresetn" signal. As illustrated below:



49.2.13.3 EXTMC Programing Sequence After Warm Reset

This section describes the flow of getting into self-refresh mode before and during warm reset, as well as required configuration to ESDCTL when getting out of warm reset.

Please note that after issuing a precharge all via special command register in the exit warm reset, it is recommended by devices to issue a manual autorefresh command, but not necessary.

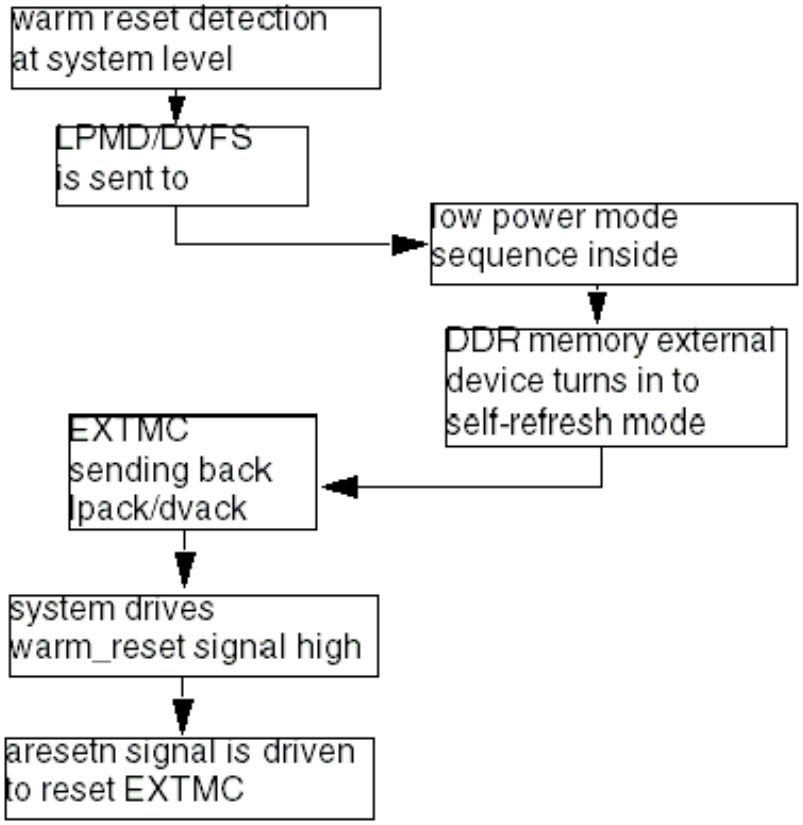


Figure 49-6. Getting into self refresh before warm_reset assertion

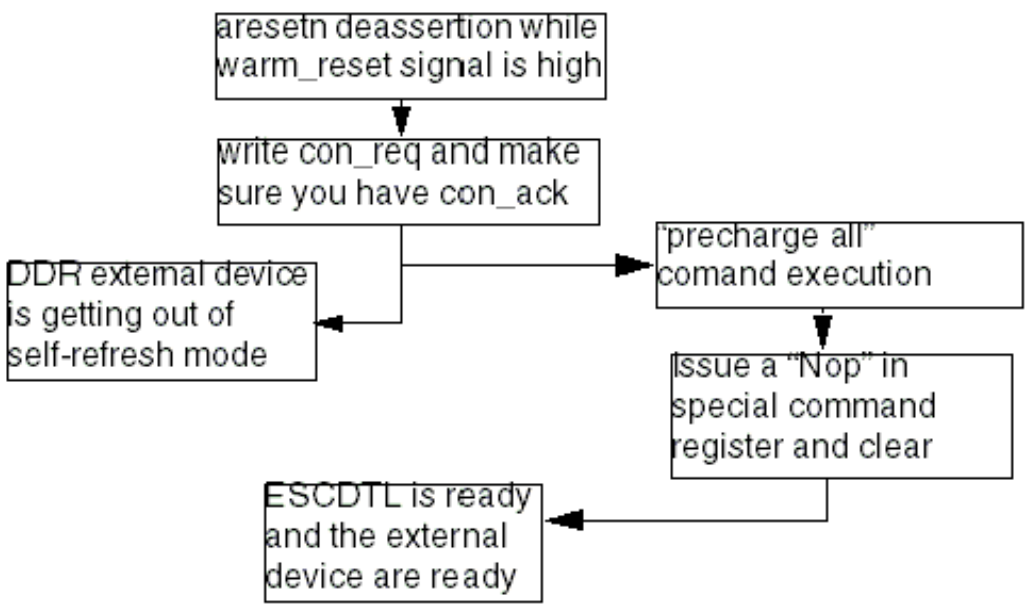


Figure 49-7. Sequence of getting out of self refresh after warm reset

49.2.14 Interrupts

The following interrupts are supported in M4IF.

- Watermark violation interrupt ARM platform.
- Watermark violation interrupt BP domain.
- Master burst length bigger then 8 violation interrupt.

49.2.15 Endianness

Only little endian is supported for i.MX53.

49.2.16 AXI Interface Restrictions

This section describes the built-in limitation in the current M4IF design that lead to some restriction on the EXTMC AXI IF support.

NOTE

Not complying to these restrictions will cause an error at best, and an unknown state (possibly a dead lock) at worst.

49.2.16.1 General Interface Limitations

1. M4IF supports write data interleaved depth of 1 (no interleaving of data is allowed).
2. awcache/arcache - Non-bufferable accesses are supported. In write access there will be a single bit to support bufferable and non-bufferable access types. According to AXI protocol, AWCACHE[0] would reflect the type of the access, AWCACHE[0]=1'b0 means a non-bufferable access and AWCACHE[0]=1'b1 means a bufferable access. The rest of the "cache" signals are not used inside the M4IF. The 'awcache' and 'arcache' output signals (inputs to the memory controller) are set according to the corresponding 'awcache' and 'arcache' inputs from the master.
3. A master must ensure that the written data has reached the destination memory, has to apply non-bufferable access and wait for the response channel.
4. If large block write transfers must be done, it is recommend that the master drive at least one non-bufferable write access for each ID at the end of the write block transfer to ensure transfer complete at the memory device.
5. In order to keep data access coherency between write and read access of the same master, the response signal will be sent as follows:

- bufferable write access - BRESP will be sent when last data of the access has entered the memory controller
- non-bufferable write access - BRESP will be sent when the data is physically written into the external memory device, and valid response was issued by the controller.

6. The AXI ID bus contains four bits total in order to reflect the AXI transaction ID as configured by the master.

NOTE

Inside M4IF an additional three bits will be added for internal use only to reflect master number between 0 and 7.

7. Master ID is supported in EXTMC in a way similar to former versions. Master ID is supported only for address buses and should be considered as ADDR timing. AWMASTER is used for Master ID in write address and ARMASTER for master ID in read address. Each field includes four bits max.

NOTE

The master ID is not an AXI requirement and it is supported for AHB masters option mainly.

8. Burst length up to eight double-words (64 bits), is supported. Access types incr and wrap are supported, fixed access type is not supported. AWSIZE/ARSIZE are two bits wide. AWLEN/ARLEN are three bits wide. When a burst of length greater than eight is issued on the bus, the M4IF will not forward it to the memory controllers, and a "DECERR" will be sent back to the master.
9. The maximum data bus size is 64 bits.
10. Unmapped addresses are not supported. All unmapped addresses will be routed to DDR. It is the SoC's responsibility not to send unmapped addresses to M4IF/EXTMC.

49.2.16.2 Atomic Accesses

M4IF support AXI atomic accesses. The atomic access logic is separate fof each arbiter, so that one CS can be locked or under exclusive monitor while other master can continue accessing another CS on a different arbitration. For example, if one of the masters issued a lock access to DDR's CSD0, the atomic logic will block all accesses to the fast channel (CSD0 & CSD1) from all the other masters. At the same time if a master issues accesses to EIM CS0 these accesses will be served as usual.

49.2.16.2.1 AXI Locked Accesses

The M4IF allows a master to perform lock on a slave region. The lock means that only this master, when using the specific AXI ID of the lock access, can access the locked region.

The master is determined based on the EXTMC master port (0:7) and not based on the MASTER_ID field (AWMASTER/ARMASTER sideband signals).

All other requests from other masters will be masked till the lock access is done. The lock access request will be active only when the request is served by the arbitration. In case there are other pending accesses of the same master that were issues before the lock access, they will be served first by the arbitration. The lock is removed upon selecting request from this master, with the lock ID, and ARLOCK/AWLOCK signals indicates an unlocked access (can be either regular or exclusive access).

49.2.16.2.2 Exclusive Accesses

The M4IF deals with exclusive access for each arbitration path separately. Each arbitration has its own exclusive access monitor and it is not affected by other exclusive accesses to other arbitrations. Thus, once an exclusive read from a certain master with a specific AXI ID has been selected by the arbiter, the specific monitor would be active following this access. If another exclusive access occurs to the same arbitration, the monitor will be updated by the new exclusive access and the old access will be dropped from the monitor. In case a write exclusive access related to the first exclusive read access arrives, it would be treated as a regular access and not as an exclusive one. The data of this write exclusive access would not be written and the response will be sent as OKAY and not EXOKAY.

M4IF supports an exclusive access of a single burst type only. Meaning the exclusive monitor will monitor a single address only. In case an exclusive access of burst length bigger than 1 is done, only the first address of this burst will be monitored.

49.2.16.3 Write Data Interleaving

M4IF has a write data interleaving depth of 1 as opposed to previous version where the depth was 2. This means that write data interleaving is not supported in M4IF.

49.2.17 IPS Interface

All M4IF registers are memory mapped. The registers will be configurable read/write via IPS bus interface only.

The IPS interface unit inside EXTMC will combine all IPS interfaces from EXTMC sub-blocks and M4IF as one of them. The IPS IF unit would route the access to the right sub-block based on ips_module_en signal and address decoding.

M4IF registers uses an accessibility policy as described in the EXTMC spec Chapter.

All M4IF registers are 32bit wide. Only IP accesses of 32bit data width are supported.

M4IF IP accesses should be used in one of the following 2 manners:

1. Check all the status bit are idle. Only if all of them are idle, an IP access could be made.

Status bit list: FPSS, SPPS, IPSS, I2PSS, M#_PSS (#= Master number)

2. Send a LPMD request to a specific arbitration and wait for LPACK before configuring its redigests.

49.2.18 WaterMark Functionality Overview

The WaterMark security feature is used to protect configurable memory regions (WaterMark space) of up to two predefined different CS. Two CS of ESDCT. No watermark capabilities exist for NFC CS and the internal memory interface.

The access (read or write) to the WaterMark space is permitted only if the appropriate control bit of this particular access is high. For read - m#_wtrmrk_ap(bp)_rd, for write - m#_wtrmrk_ap(bp)_wr. The watermark regions are configurable via the watermark registers where watermark space #0 is for AP masters use and watermark space #1 is for BP masters use.

AP masters and BP masters can configure each of its own registers only. An AP master cannot configure BP master registers and vice versa. AP and BP masters are being identified by the predefined via at the EXTMC boundary signals. SoC integration should configure these vias with the right values according to the master ID's in the system. Only a predefined privilege master (AP or BP) can configure the watermark registers.

Access to the WaterMark space when watermark bits are low is considered a WaterMark violation. Even if the accessing master is part of the accessed domain. A WaterMark interrupt will be generated in such a violation. In addition the master who caused the violation will not get access, and an error response (DECERR) will be issued. In case an access was made to watermark space #0 when wtrmrk_ap is in low, an AP watermark interrupt will be generated. The same goes for BP watermarked regions. In both cases the address and the master ID of the first master which made a violated access would be kept in the watermark status registers till the interrupt is cleared. Please refer to Watermark Interrupt and Status 0 Register (M4IF_WMIS0).

If a master violates the same region twice (AP or BP), only the first violation will be captured in the status registers.

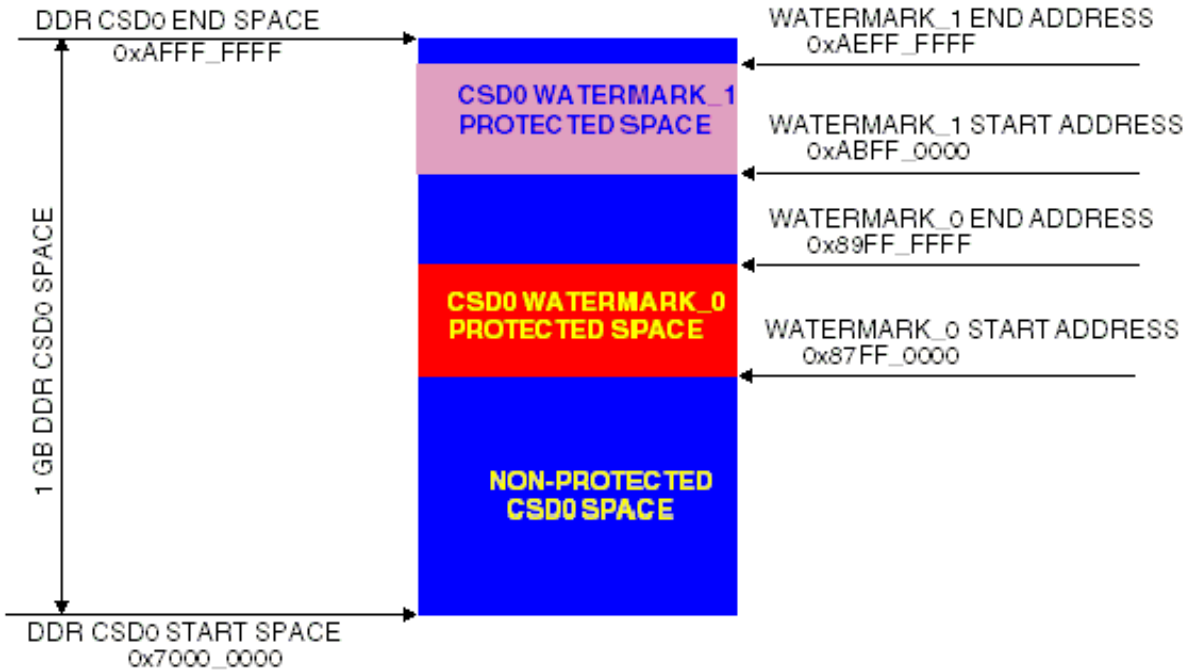
If a "clear" command is issued (by writing to the status register) at the exact same time that a violation occurs, the "clear" will win. Meaning the violation will not be captured in the status register. The access will be treated as a violated access and will not pass through.

Up to two WaterMark spaces can be defined for each predefined CS mapped by the M4IF. The WaterMark space is configurable through a set of configuration registers that contains the WaterMark space start and end address for each chip select. Please refer to Watermark Start ADDR_0 Register n (M4IF_WMSA0_*n*) for specific details.

The watermark address register has a resolution of 4 KByte, meaning:

- The start address can be set at the end of the first 4 KByte.
- The start and end address can be set in steps of 4 KByte.

The figure below illustrates a WaterMark space in DDR CSD0 memory region. The red shaded area in the figure is the WaterMark #0 (AP) space defined in DDR CSD0 memory region while the purple shaded area is the WaterMark #1 (BP) space. These memory regions are accessible only by those masters that have wtrmrk_ap(bp) bits high during the access. Accesses to this region by other masters when wtrmrk_ap(bp) are low generates an interrupt (if enabled through the WaterMark control register) and sets the respective WaterMark space (clear by one) status bit. The blue shaded area is the non protected DDR CSD0 memory region, means that it is accessible by all masters connected to the M4IF.



DDR CSD0 WaterMark Space Diagram

49.2.19 Debug Unit

The M4IF has a debug unit in order to allow for better debug capabilities.

The debug unit is consisted of 2 main parts, visibility unit and profiling units.

49.2.19.1 Visibility Unit

This unit muxes and reflects outside, several crucial signals from within M4IF. These signals are described in detail in Debug Signals. The muxing is done between the 4 channels: fast, slow, internal 1 memory and internal 2 memory. The desired channel is configured via M4IF_MDCR register.

The visibility unit is disabled by default. In order to enable it the user must write to the M4IF_MDCR register's "DBG_EN" bit. Please refer to Debug Control Register (M4IF_DCR) for details.

Note that this unit monitors the signals from within the M4IF, even in a case of fail write exclusive attempt.

49.2.19.2 Profiling Units

There are four profiling units inside the debug unit. Each unit is responsible for profiling a single channel (fast, slow, internal memory and internal 2 memory).

A single profiling unit contains counters which record the number of accesses through the specific channel for each master. The counters are read by the IP bus. The information in the IP register does not contain the lower five bits of the counter, meaning the numbers which are read in the MDSR2-5 registers indicate the number of groups of 32 accesses. For example, if master 5 performed a total of 274 accesses, the number which will be read in the upper 16 bits of MDSR4 will be: trunc(274/32)=8. In order to get the approximate real number of accesses, the user will have to multiply the number in the register by 32. In the above example the outcome will be: 8x32=256. The error will be (274-256)/274=7%. The profiling unit is designed to handle massive use-cases, in which there will be thousands of accesses. In these cases the error will become negligible.

The profiling unit also calculates the maximal dynamic priority of a specific request. The request is selected by M4IF_MDCR configuration register. It also calculates the maximum time the master of that request was pending on the bus without getting a response. These are read by MDSR0 register.

In order to calculate the average time it took a master to get the bus, the profiling unit sums the time (in memory-controller's clock cycles) it took each access to get the bus. This sum can later be read via the M4IF_MDSR8 register. The average can be calculated by reading the counter for the number of accesses that this master performed (via the MDSR2-5 registers), and dividing the 2 numbers.

The two additional registers which contain information about the profiling unit are M4IF_MDSR6 and M4IF_MDSR7. M4IF_MDSR6 contains the total number of accesses a specific request (configured in the MDCR) has accessed the bus. M4IF_MDSR7 contains the sum of "time for bus" for this specific request. An average of "time for bus" for this request can be calculated by dividing the two numbers. There is an option to use these two registers not only for a specific request, but for a type of access (read or write). Each master has two requests for read and two for write. The M4IF_MDSR6-7 registers can be used to accumulate the statistics for a single request or for both. This is configured by the "SPC_REQ" bit of M4IF_MDCR.

There is an option to reset all counters of the profiling unit by setting to "1" the "DBG_RST" bit in the M4IF_MDCR. As long as this bit is asserted, all counters will remain 0.

The debug unit's profiling is disabled by default. In order to enable it the user must write to the M4IF_MDCR register's "DBG_EN" bit. Please refer to Debug Control Register (M4IF_DCR) for details.

49.2.20 Buffers Size Table

Table 49-4. Master Buffers

Master#	Master Port Name	Master Port Write Buffers	Master Port Write Command Buffers	Master Port Read Command Buffers
0	IPU	72bx8x4	65bx4	65bx6
1	VPU	72bx8x6	65bx6	65bx12
2	MAX	36bx8x1 (32bit)	65bx1	65bx1
3	ARM	72bx8x6	65bx6	65bx6
4	SDMA	36bx8x2 (32bit)	65bx2	65bx2
5	GPU	72bx8x8	65bx8	65bx10
6	GPU2D	72bx8x4	65bx4	65bx6
7	USB	36bx8x2 (32bit)	65bx2	65bx2

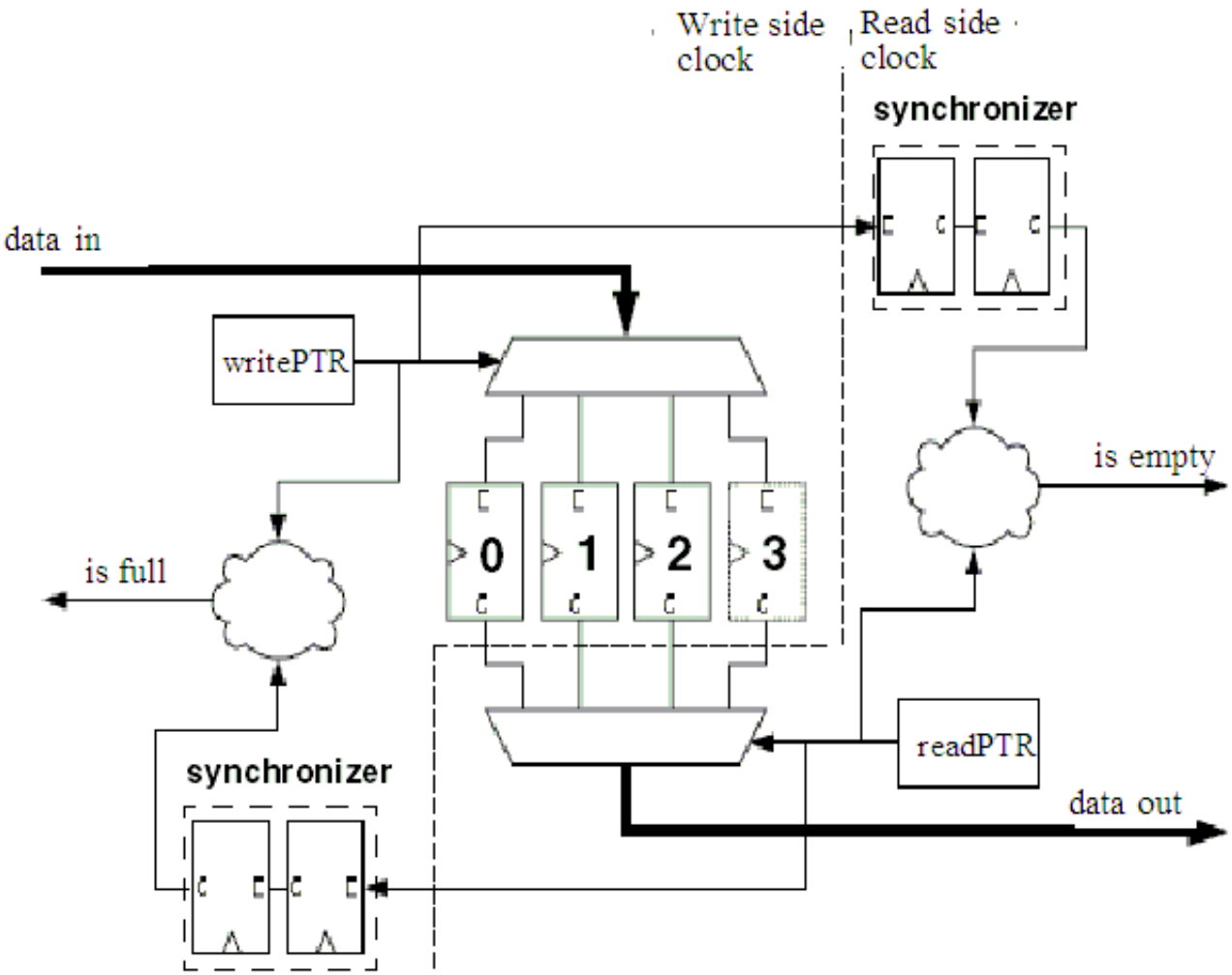
Table 49-5. Slave Buffers

Slave#	Slave Port Name	Slave Port Read Buffers	Slave Pending Address fifo	Slave Last Address Read fifo	Slave Last address Write fifo
0	FAST	66bx16x8	85bx1	13bx16	13bx3
1	SLOW	66bx4x8	67bx2	13bx4	13bx3
2	INT1	66bx2x8	67bx2	13bx2	13bx3
3	INT2	66bx2x8	67bx3	13bx2	13bx4

49.2.21 Synchronization

There are 12 AXI clock domains involved in the M4IF. 8 master clocks and 4 memory-controller clocks. Clock domain crossings are possible only from master clock to arbitration clock and vice versa. There cannot be a clock domain crossing between master clocks.

The controls which cross clock domains are synchronized using the "dual clock FIFO (DCF)".



The DCF allows for synchronization from "fast to slow" clock domains and from "slow to fast", allowing flexibility of the clock ratios between master and arbitration.

49.2.21.1 Synchronization Table

The following table shows the clock-domain relations between all 8 domains of the gaskets/masters, and 4 domains of the arbitrations/slaves.

S - always synchronizer on this path

[] - no synchronizer (domains are synchronized)

NC - No hardware connection at all

Table 49-6. Sync Table

Master#	Master Port Name	FAST	SLOW	INT1	INT2
0	IPU	S	NC	S	S
1	VPU	S	NC	S	S
2	MAX	S	S	[]	[]
3	ARM	S	S	S	S
4	SDMA	S	S	[]	[]
5	GPU	S	NC	S	NC
6	GPU2D	S	NC	S	S
7	USB	S	NC	[]	[]

49.2.22 Supporting 8/16 bit Bursts

M4IF now supports bursts of data size 8/16 bit, and of length up to 8 beat. INCR and WRAP type bursts are supported (according to the AXI protocol spec restrictions).

The M4IF splits the 8/16 bit bursts into a series of single bursts. For instance, an INCR burst of length 6 will be split and issued to the memory-controller/downsizer as a series of 6 single bursts.

4 status bits allow indication whether this kind of burst (8/16 bit) occurred. Please refer to: Master Len Interrupt (M4IF_MLI) for further details.

49.2.23 Dead-Lock Prevention in Read Accesses

M4IF allows flexibility by working in a "parallel routes" scheme. Because it has four separate arbiters, it can allow parallel work of EXTMC against all four slaves. For example, master #0 can send two requests to two different memory controllers (DDR and NFC, for example) and both controllers can handle these requests at the same time without interrupting each other.

This parallel scheme provides enhanced latency hiding when several masters work in parallel against several slaves, or even when a single master works versus multiple slaves.

However, there is a rare theoretical case in which this parallel scheme can cause a dead-lock for read requests. This rare case can only occur if there are two or more masters that work against the two or more slaves in a manner of interleaving the read accesses

between the slaves. That is, a master issues read requests back and forth between two slaves. One request to the first slave and the other to the second slave repeating this over and over. A second master does exactly the same (working with the same slaves).

In order to prevent this case from happening with a 100% certainty, a protection mechanism has been added to the M4IF called "Arbit Stop".

This mechanism limits a master from sending interleaved read requests. If there is an unfinished access in the master's read address buffer, M4IF will accept read accesses that are directed to the same slave as the access that is already inside the buffer. If an access to a different location is issued, it will not be inserted to the buffer. The access will wait on the bus until all accesses in the buffer are served and only then will be accepted.

The limitation is only on the read path. It does affect the write path.

Each master can be configured independently to have this limitation or not. Please refer to Control Register 0 (M4IF_CR0) for specific details.

49.3 Programmable Registers

M4IF memory area is in the IP memory region dedicated for its registers. M4IF registers are located at 0xBASE_0000 - 0xBASE_0FFF. "BASE" address is determined according to the SOC definitions.

The sections below describe the register definition of M4IF. It contains all information on registers definition like watermark and control registers. It contains all user programmable master priority register per arbitration path as well as priority specific definition for fast arbitration.

M4IF memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FD_8000	Power Saving Masters 0 (M4IF_PSM0)	32	R/W	0A07_0A07h	49.3.1/ 3488
63FD_8004	Power Saving Masters 1 (M4IF_PSM1)	32	R/W	0A07_0A07h	49.3.2/ 3490
63FD_800C	General Purpose Register (M4IF_GPR)	32	R/W	0000_0001h	49.3.3/ 3492
63FD_8018	Debug Status Register 6 (M4IF_DSR6)	32	R	0000_0000h	49.3.4/ 3493

Table continues on the next page...

M4IF memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FD_801C	Debug Status Register 7 (M4IF_DSR7)	32	R	0000_0000h	49.3.5/ 3493
63FD_8020	Debug Status Register 8 (M4IF_DSR8)	32	R	0000_0000h	49.3.6/ 3494
63FD_8024	Debug Status Register 0 (M4IF_DSR0)	32	R	0000_0000h	49.3.7/ 3494
63FD_8028	Debug Status Register 1 (M4IF_DSR1)	32	R	0000_0000h	49.3.8/ 3495
63FD_802C	Debug Status Register 2 (M4IF_DSR2)	32	R	0000_0000h	49.3.9/ 3495
63FD_8030	Debug Status Register 3 (M4IF_DSR3)	32	R	0000_0000h	49.3.10/ 3496
63FD_8034	Debug Status Register 4 (M4IF_DSR4)	32	R	0000_0000h	49.3.11/ 3496
63FD_8038	Debug Status Register 5 (M4IF_DSR5)	32	R	0000_0000h	49.3.12/ 3497
63FD_8040	F_Basic Priority Reg 0 (M4IF_F_BPR0)	32	R/W	0000_2233h	49.3.13/ 3498
63FD_8044	F_Basic Priority Reg 1 (M4IF_F_BPR1)	32	R/W	0000_0000h	49.3.14/ 3500
63FD_8048	Control Register (M4IF_CR)	32	R/W	8099_01A3h	49.3.15/ 3502
63FD_8074	I2_Unit_Level_Arbitration_ Register (M4IF_I2_ULAR)	32	R/W	0012_4924h	49.3.16/ 3503
63FD_8078	Int. 2 Memory Arbitration Control Register (M4IF_I2MACR)	32	R/W	0000_0000h	49.3.17/ 3504
63FD_807C	Internal 2 Control Register (M4IF_I2CR)	32	R/W	0000_0051h	49.3.18/ 3505
63FD_8084	Step By Step Address (M4IF_SSA)	32	R	0000_0000h	49.3.19/ 3506
63FD_8088	Step By Step Address Controls (M4IF_SSAC)	32	R	0000_0000h	49.3.20/ 3507
63FD_808C	Control Register 0 (M4IF_CR0)	32	R/W	8000_0007h	49.3.21/ 3508
63FD_8090	Control Register 1 (M4IF_CR1)	32	R/W	0028_140Ah	49.3.22/ 3511
63FD_8094	Debug Control Register (M4IF_DCR)	32	R/W	0000_0000h	49.3.23/ 3513
63FD_8098	Fast Arbitration Control Register (M4IF_FACR)	32	R/W	0000_01A1h	49.3.24/ 3516

Table continues on the next page...

M4IF memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FD_809C	F_Priority Weighting Configuration Register (M4IF_F_PWCR)	32	R/W	0012_0125h	49.3.25/ 3516
63FD_80A0	Slow Arbitration Control Register (M4IF_SACR)	32	R/W	0000_0000h	49.3.26/ 3518
63FD_80A4	Power Saving Masters 2 (M4IF_PSM2)	32	R/W	0A07_0A07h	49.3.27/ 3519
63FD_80A8	Int. Memory Arbitration Control Register (M4IF_IMACR)	32	R/W	0000_0000h	49.3.28/ 3521
63FD_80AC	Power Saving Masters 3 (M4IF_PSM3)	32	R/W	0A07_0A07h	49.3.29/ 3521
63FD_80B0	F_Unit_Level_Arbitration_ Register (M4IF_F_ULAR)	32	R/W	0000_0000h	49.3.30/ 3524
63FD_80B4	S_Unit_Level_Arbitration_ Register (M4IF_S_ULAR)	32	R/W	0012_4924h	49.3.31/ 3525
63FD_80B8	I_Unit_Level_Arbitration_ Register (M4IF_I_ULAR)	32	R/W	0012_4924h	49.3.32/ 3526
63FD_80BC	Fast_Dynamic_Priority_Status Register (M4IF_FDPSR)	32	R	0000_0000h	49.3.33/ 3527
63FD_80C0	Fast_Dynamic_Priority_Control Register (M4IF_FDPCR)	32	R/W	0000_0000h	49.3.34/ 3528
63FD_80C4	Master Len Interrupt (M4IF_MLI)	32	w1c	0000_0000h	49.3.35/ 3530
63FD_80EC	Watermark Start ADDR_0 Register n (M4IF_WMSA0_6)	32	R/W	0000_0000h	49.3.36/ 3532
63FD_80F0	Watermark Start ADDR_0 Register n (M4IF_WMSA0_7)	32	R/W	0000_0000h	49.3.36/ 3532
63FD_810C	Watermark End ADDR_0 Register (M4IF_WMEA0_6)	32	R/W	0000_0000h	49.3.37/ 3532
63FD_8110	Watermark End ADDR_0 Register (M4IF_WMEA0_7)	32	R/W	0000_0000h	49.3.37/ 3532
63FD_8114	Watermark Interrupt and Status 0 Register (M4IF_WMIS0)	32	R/W	0000_0000h	49.3.38/ 3533
63FD_8118	Watermark Violation Address 0 Register (M4IF_WMVA0)	32	R	0000_0000h	49.3.39/ 3534
63FD_8138	Watermark Start ADDR_1 Register n (M4IF_WMSA1_6)	32	R/W	0000_0000h	49.3.40/ 3535
63FD_813C	Watermark Start ADDR_1 Register n (M4IF_WMSA1_7)	32	R/W	0000_0000h	49.3.40/ 3535
63FD_8158	Watermark End ADDR_1 Register (M4IF_WEAR1_6)	32	R/W	0000_0000h	49.3.41/ 3535

Table continues on the next page...

M4IF memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FD_815C	Watermark End ADDR_1 Register (M4IF_WEAR1_7)	32	R/W	0000_0000h	49.3.41/ 3535
63FD_8160	Watermark Interrupt and Status 1 Register (M4IF_WISR1)	32	R/W	0000_0000h	49.3.42/ 3537
63FD_8164	Watermark Violation Address 1 Register (M4IF_WMVA1)	32	R	0000_0000h	49.3.43/ 3538

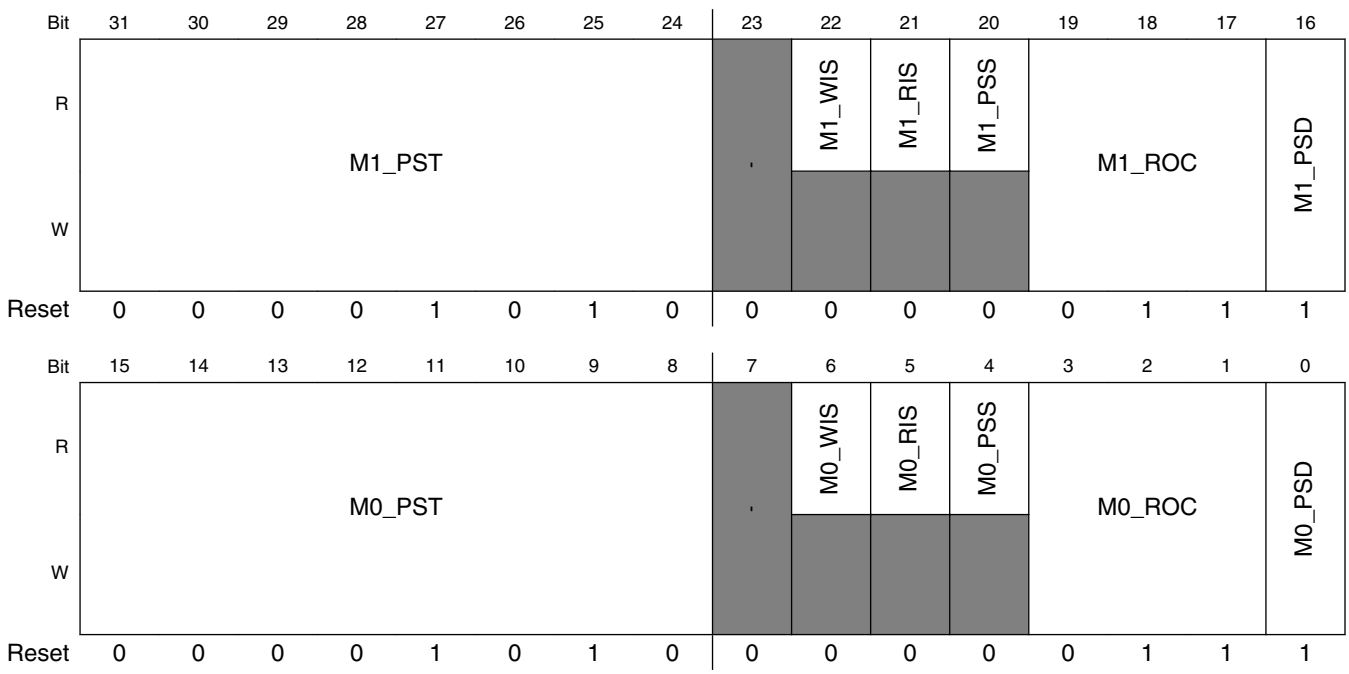
49.3.1 Power Saving Masters 0 (M4IF_PSM0)

The M4IF.PSM0 determines the power saving features for masters #0 and #1.

NOTE

The 'PST' and 'ROC' parameters can only be changed once after reset and before enabling the power saving feature (PSD=0).

Address: M4IF_PSM0 is 63FD_8000h base + 0h offset = 63FD_8000h



M4IF_PSM0 field descriptions

Field	Description
31–24 M1_PST	master #1 Power Saving Timer. The real value which is used is register-value multiplied by 100. Default value is set to 1000 clock cycles. 00000000 Reserved this value is forbidden. 00000001 timer is configured to 100 clock cycles.

Table continues on the next page...

M4IF_PSM0 field descriptions (continued)

Field	Description
	00001010 Default value- 1000 clock cycles. 11111111 timer clock is defined to 25500 clock cycles.
23 -	Reserved.
22 M1_WIS	Master #1 Write Idle Status.This read only bit indicates whether master #1 write request buffer is idle (empty) or not. 0 idle 1 not idle
21 M1_RIS	Master #1 Read Idle Status.This read only bit indicates whether master #1 read request buffer is idle (empty) or not. 0 idle 1 not idle
20 M1_PSS	Master #1 Power Saving Status. This read only bit indicates whether master #1 gasket is in power saving mode. 0 not in power saving 1 power saving
19–17 M1_ROC	master #1 Ready Off Cycles. Number of minimum cycles that the ready signals must be low before they can go back to high. 001-111 (000 is a forbidden value)
16 M1_PSD	master #1 Power Saving Disable. 0 power saving enabled 1 power saving disabled (default)
15–8 M0_PST	master #0 power saving timer. The real value which is used is register-value multiplied by 100. Default value is set to 1000 clock cycles. 00000000 Reserved - this value is forbidden. 00000001 timer is configured to 100 clock cycles. 00001010 Default value- 1000 clock cycles. 11111111 timer clock is defined to 25500 clock cycles.
7 -	Reserved.
6 M0_WIS	Master #0 Write Idle Status.This read only bit indicates whether master #0 write request buffer is idle (empty) or not. 0 idle 1 not idle
5 M0_RIS	Master #0 Read Idle Status.This read only bit indicates whether master #0 read request buffer is idle (empty) or not. 0 idle 1 not idle

Table continues on the next page...

M4IF_PSM0 field descriptions (continued)

Field	Description
4 M0_PSS	Masrer #0 Power Saving Status. This read only bit indicates whether master #0 gasket is in power saving mode. 0 not in power saving 1 power saving
3–1 M0_ROC	master #0 Ready Off Cycles. Number of minimum cycles that the ready signals must be low before they can go back to high. Default value is 3 cycles. 001 111 (000 is a forbidden value)
0 M0_PSD	master #0 Power Saving Disable. 0 power saving enabled 1 power saving disabled (default)

49.3.2 Power Saving Masters 1 (M4IF_PSM1)

The M4IF_PSM1 determines the power saving features for masters #2 and #3.

NOTE

The 'PST' and 'ROC' parameters can only be changed once after reset and before enabling the power saving feature (PSD=0).

Address: M4IF_PSM1 is 63FD_8000h base + 4h offset = 63FD_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	M3_PST								-	M3_WIS	M3_RIS	M3_PSS	M3_ROC			M3_PSD
W																
Reset	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M2_PST								-	M2_WIS	M2_RIS	M2_PSS	M2_ROC			M2_PSD
W																
Reset	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	1

M4IF_PSM1 field descriptions

Field	Description
31–24 M3_PST	master #3 Power Saving Timer. Default value is set to 1000 clock cycles. The real value which is used is register-value multiplied by 100. 00000000 Reserved - this value is forbidden. 00000001 timer is configured to 100 clock cycles. 00001010 Default value- 1000 clock cycles. 11111111 timer clock is defined to 25500 clock cycles.
23 -	Reserved.
22 M3_WIS	Master #3 Write Idle Status.This read only bit indicates whether master #3 write request buffer is idle (empty) or not. 0 idle 1 not idle
21 M3_RIS	Master #3 Read Idle Status.This read only bit indicates whether master #3 read request buffer is idle (empty) or not. 0 idle 1 not idle
20 M3_PSS	Masrer #3 Power Saving Status. This read only bit indicates whether master #3 gasket is in power saving mode. 0 not in power saving 1 power saving
19–17 M3_ROC	master #3 Clock Off Cycles. Number of minimum cycles that the ready signals must be low before they can go back to high. 001-111 (000 is a forbidden value)
16 M3_PSD	master #3 Power Saving Disable. 0 power saving enabled 1 power saving disabled (default)
15–8 M2_PST	master #2 Power Saving Timer. The real value which is used is register-value multiplied by 100. Default value is set to 1000 clock cycles. 00000000 Reserved - this value is forbidden. 00000001 timer is configured to 100 clock cycles. 00001010 Default value- 1000 clock cycles. 11111111 timer clock is defined to 25500 clock cycles.
7 -	Reserved.
6 M2_WIS	Master #2 Write Idle Status.This read only bit indicates whether master #2 write request buffer is idle (empty) or not. 0 idle 1 not idle

Table continues on the next page...

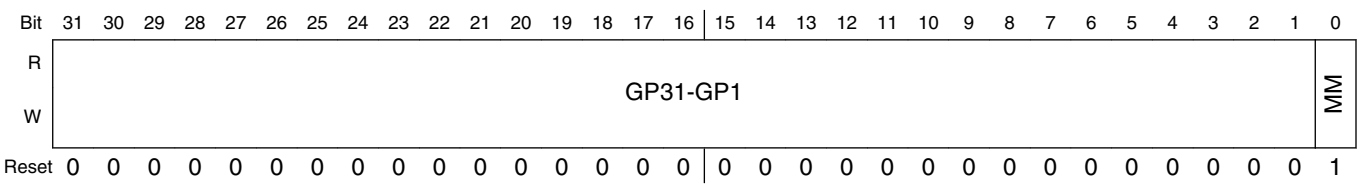
M4IF_PSM1 field descriptions (continued)

Field	Description
5 M2_RIS	Master #2 Read Idle Status.This read only bit indicates whether master #2 read request buffer is idle (empty) or not. 0 idle 1 not idle
4 M2_PSS	Masrer #2 Power Saving Status. This read only bit indicates whether master #2 gasket is in power saving mode. 0 not in power saving 1 power saving
3–1 M2_ROC	master #2 Ready Off Cycles. Number of minimum cycles that the ready signals must be low before they can go back to high. Default value is 3 cycles. 001-111 (000 is a forbidden value)
0 M2_PSD	master #2 Power Saving Disable. 0 power saving enabled 1 power saving disabled (default)

49.3.3 General Purpose Register (M4IF_GPR)

The M4IF_GENP is a general 32-bit read/write register.

Address: M4IF_GPR is 63FD_8000h base + Ch offset = 63FD_800Ch



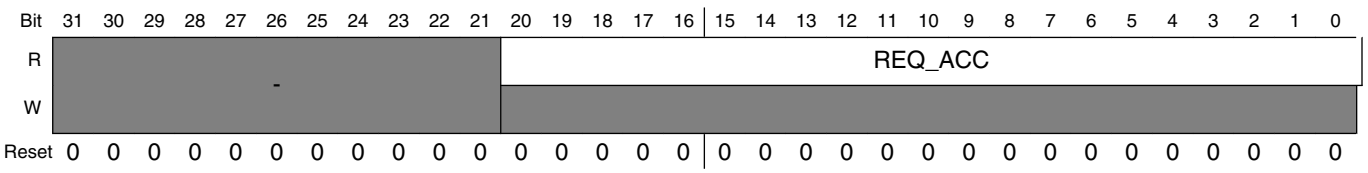
M4IF_GPR field descriptions

Field	Description
31–1 GP31-GP1	General purpose read/write bits. Can be used for anything.
0 MM	MM=1 means that NFC_D is muxed with EIM DATA on NANDF_D (on PATA_DATA[15:0] pins). If MM=0, NFC_D is muxed with EIM A/D on EIM_DA[15:1]. In this case the corresponding CSxGCR2[12] must also be set to notify EIM that its address is muxed with NF data. This MM bit is also selected by boot according to BOOT_CFG1[6] fuse.

49.3.4 Debug Status Register 6 (M4IF_DSR6)

The M4IF_MDSR6 reflects the total number of accesses made by a specific requesting the desired arbitration. The arbitration can be fast, slow, intr1 or intr2. See DCRDebug Control Register.

Address: M4IF_DSR6 is 63FD_8000h base + 18h offset = 63FD_8018h



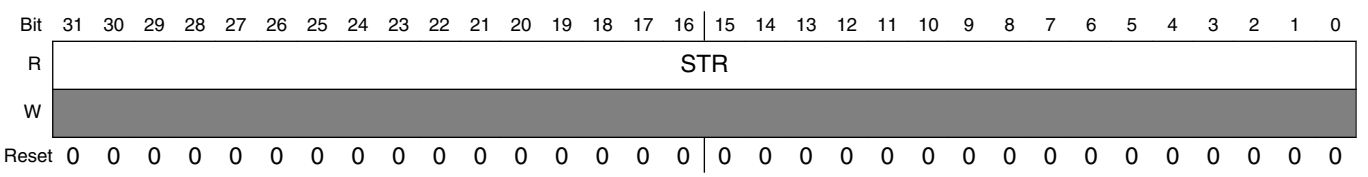
M4IF_DSR6 field descriptions

Field	Description
31–21 -	Reserved.
21–0 REQ_ACC	Number of accesses made by a specific request to a specific arbitration. The request and arbitration are configured in the M4IF_MDCR register.

49.3.5 Debug Status Register 7 (M4IF_DSR7)

The M4IF_MDSR7 reflects the sum of time all the requests of a specific kind were pending in the desired arbitration. The arbitration can be fast, slow, intr1 or intr2 and is configured in the M4IF_MDCR register. See DCRDebug Control Register.

Address: M4IF_DSR7 is 63FD_8000h base + 1Ch offset = 63FD_801Ch



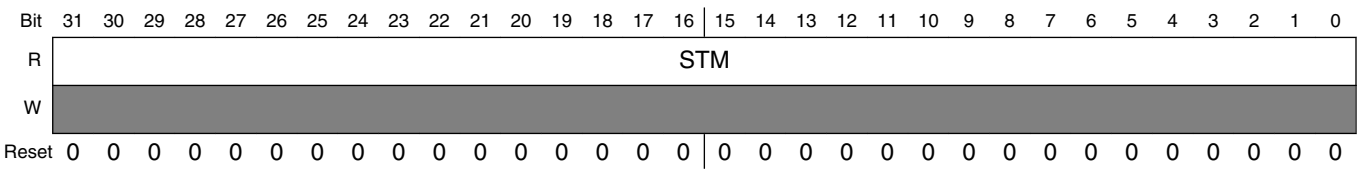
M4IF_DSR7 field descriptions

Field	Description
31–0 STR	Sum Time Request. These bits reflect the sum of time (in arbitration clock cycles) all the requests of a specific kind were pending at a specific arbitration. (The arbitration is configured in the M4IF_MDCR register)

49.3.6 Debug Status Register 8 (M4IF_DSR8)

The M4IF_MDSR8 reflects the sum of time all the requests of a specific master were pending in the desired arbitration. The arbitration can be fast, slow, intr1 or intr2 and is configured in the M4IF_MDCR register. See DCRDebug Control Register.

Address: M4IF_DSR8 is 63FD_8000h base + 20h offset = 63FD_8020h



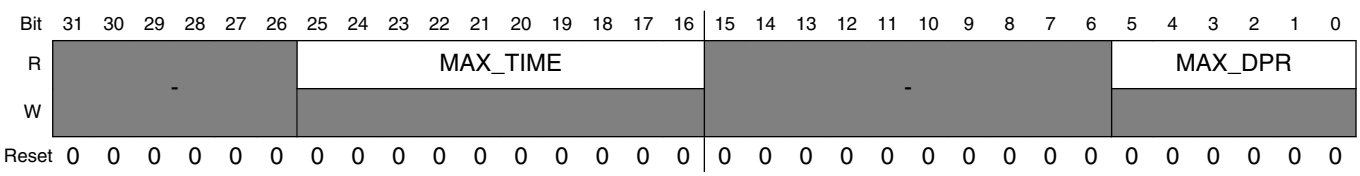
M4IF_DSR8 field descriptions

Field	Description
31–0 STM	Sum Time Master. These bits reflect the sum of time (in arbitration clock cycles) all the requests of a specific master were pending at a specific arbitration. (The arbitration is configured in the M4IF_MDCR register)

49.3.7 Debug Status Register 0 (M4IF_DSR0)

The M4IF_MDSR0 reflects the maximum values of the chosen dynamic priority and the "time for bus" of a master. The desired dynamic priority is configured in the MDCR. Please refer to DCRDebug Control Register.

Address: M4IF_DSR0 is 63FD_8000h base + 24h offset = 63FD_8024h



M4IF_DSR0 field descriptions

Field	Description
31–26 -	Reserved.
25–16 MAX_TIME	Maximum Time for Bus. This field represents the maximum number of cycles the selected master (see DDPM field in MDCR) was pending on the bus.
15–6 -	Reserved.

Table continues on the next page...

M4IF_DSR0 field descriptions (continued)

Field	Description
5–0 MAX_DPR	Maximum Dynamic Priority. These bits reflect the maximum value of the dynamic priority which was selected by the MDCR.

49.3.8 Debug Status Register 1 (M4IF_DSR1)

The M4IF_MDSR1 reflects the total number of accesses that were made through the desired arbitration. The arbitration can be fast, slow or intr, and is configured in the M4IF_MDCR register. Please refer to DCRDebug Control Register.

Address: M4IF_DSR1 is 63FD_8000h base + 28h offset = 63FD_8028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-								TOT_ACC																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

M4IF_DSR1 field descriptions

Field	Description
31–24 -	Reserved.
23–0 TOT_ACC	Total Accesses. These bits reflect the total number of accesses that were made to a specific arbitration. The selected arbitration is configured in the MDCR.

49.3.9 Debug Status Register 2 (M4IF_DSR2)

The M4IF_MDSR2 reflects the number of groups of 32 accesses (resolution is of 32 accesses) that master0 and master1 made through the desired arbitration. The arbitration can be fast, slow, int1 or int2 and configured in the M4IF_MDCR register. Please refer to DCRDebug Control Register

Address: M4IF_DSR2 is 63FD_8000h base + 2Ch offset = 63FD_802Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M1_ACC																M0_ACC															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

M4IF_DSR2 field descriptions

Field	Description
31–16 M1_ACC	Master1 Accesses. These bits reflect the total number of master1 accesses that were made to a specific arbitration. The selected arbitration is configured in the MDCR.
15–0 M0_ACC	Master0 Accesses. These bits reflect the total number of master0 accesses that were made to a specific arbitration. The selected arbitration is configured in the MDCR.

49.3.10 Debug Status Register 3 (M4IF_DSR3)

The M4IF_MDSR3 register reflects the number of groups of 32 accesses (resolution is of 32 accesses) that master2 and master3 made through the desired arbitration. The arbitration can be fast, slow or internal and is configured in the M4IF_MDCR register. Please refer to DCRDebug Control Register.

Address: M4IF_DSR3 is 63FD_8000h base + 30h offset = 63FD_8030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M3_ACC																M2_ACC															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

M4IF_DSR3 field descriptions

Field	Description
31–16 M3_ACC	Master3 Accesses. These bits reflect the total number of master3 accesses that were made to a specific arbitration. The selected arbitration is configured in the MDCR.
15–0 M2_ACC	Master2 Accesses. These bits reflect the total number of master2 accesses that were made to a specific arbitration. The selected arbitration is configured in the MDCR.

49.3.11 Debug Status Register 4 (M4IF_DSR4)

The M4IF_MDSR4 reflects the number of groups of 32 accesses (resolution is of 32 accesses) that master4 and master5 made through the desired arbitration. The arbitration can be fast, slow or intr and is configured in the M4IF_MDCR register.Please refer to DCRDebug Control Register

Address: M4IF_DSR4 is 63FD_8000h base + 34h offset = 63FD_8034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M5_ACC																M4_ACC															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

M4IF_DSR4 field descriptions

Field	Description
31–16 M5_ACC	Master5 Accesses. These bits reflect the total number of master5 accesses that were made to a specific arbitration. The selected arbitration is configured in the MDCR.
15–0 M4_ACC	Master4 Accesses. These bits reflect the total number of master4 accesses that were made to a specific arbitration. The selected arbitration is configured in the MDCR.

49.3.12 Debug Status Register 5 (M4IF_DSR5)

The M4IF_MDSR5 reflects the number of groups of 32 accesses (resolution is of 32 accesses) that master6 and master7 made through the desired arbitration. The arbitration can be fast, slow or intr and is configured in the M4IF_MDCR register.Please refer to DCRDebug Control Register.

Address: M4IF_DSR5 is 63FD_8000h base + 38h offset = 63FD_8038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M7_ACC																M6_ACC															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

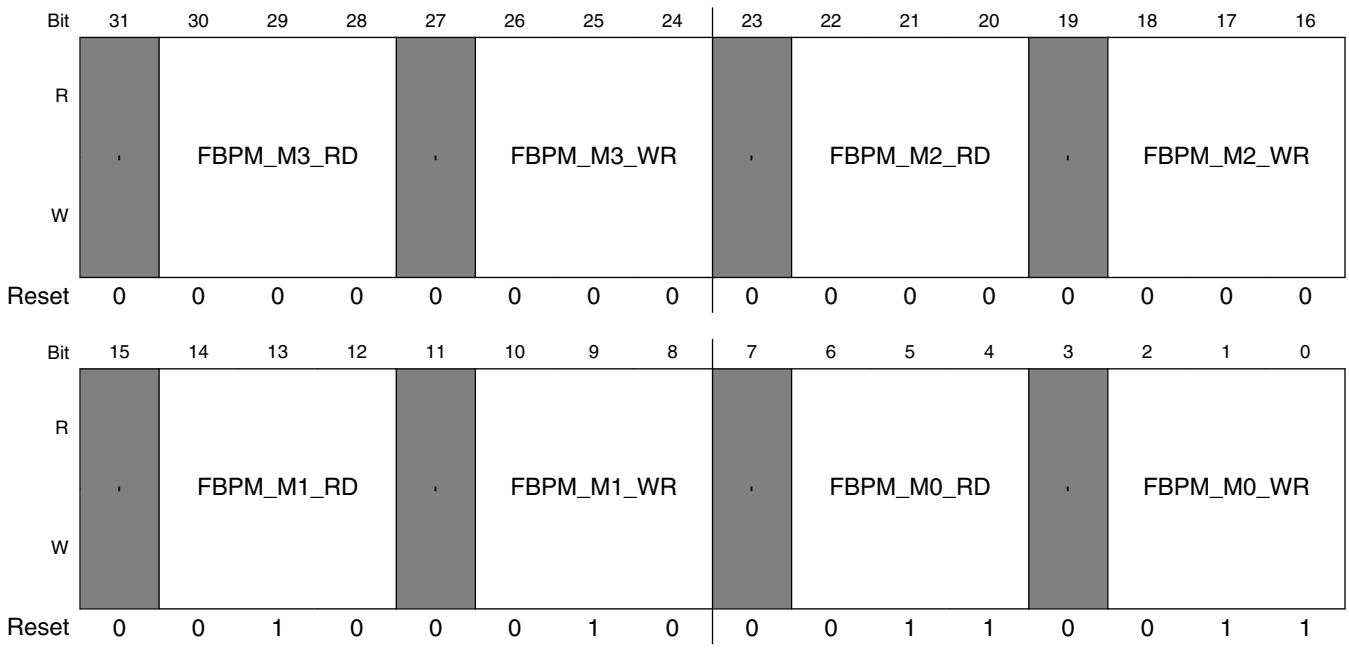
M4IF_DSR5 field descriptions

Field	Description
31–16 M7_ACC	Master7 Accesses. These bits reflect the total number of master7 accesses that were made to a specific arbitration. The selected arbitration is configured in the MDCR.
15–0 M6_ACC	Master6 Accesses. These bits reflect the total number of master6 accesses that were made to a specific arbitration. The selected arbitration is configured in the MDCR.

49.3.13 F_Basic Priority Reg 0 (M4IF_F_BPR0)

M4IF_FBPM0 register contains the basic priority value for masters 0, 1, 2, and 3 for the fast arbitration logic only. Different registers are used for slow arbitration and internal memory arbitration logic.

Address: M4IF_F_BPR0 is 63FD_8000h base + 40h offset = 63FD_8040h



M4IF_F_BPR0 field descriptions

Field	Description
31 -	Reserved
30–28 FBPM_M3_RD	User defined value. These bits represent the basic priority value for read requests as configured by the user to master 3 000 encoding 0 001 encoding 1 111 encoding 7
27 -	Reserved.
26–24 FBPM_M3_WR	User defined value. These bits represent the basic priority value for write requests as configured by the user to master 3 000 encoding 0 001 encoding 1 111 encoding 7
23 -	Reserved

Table continues on the next page...

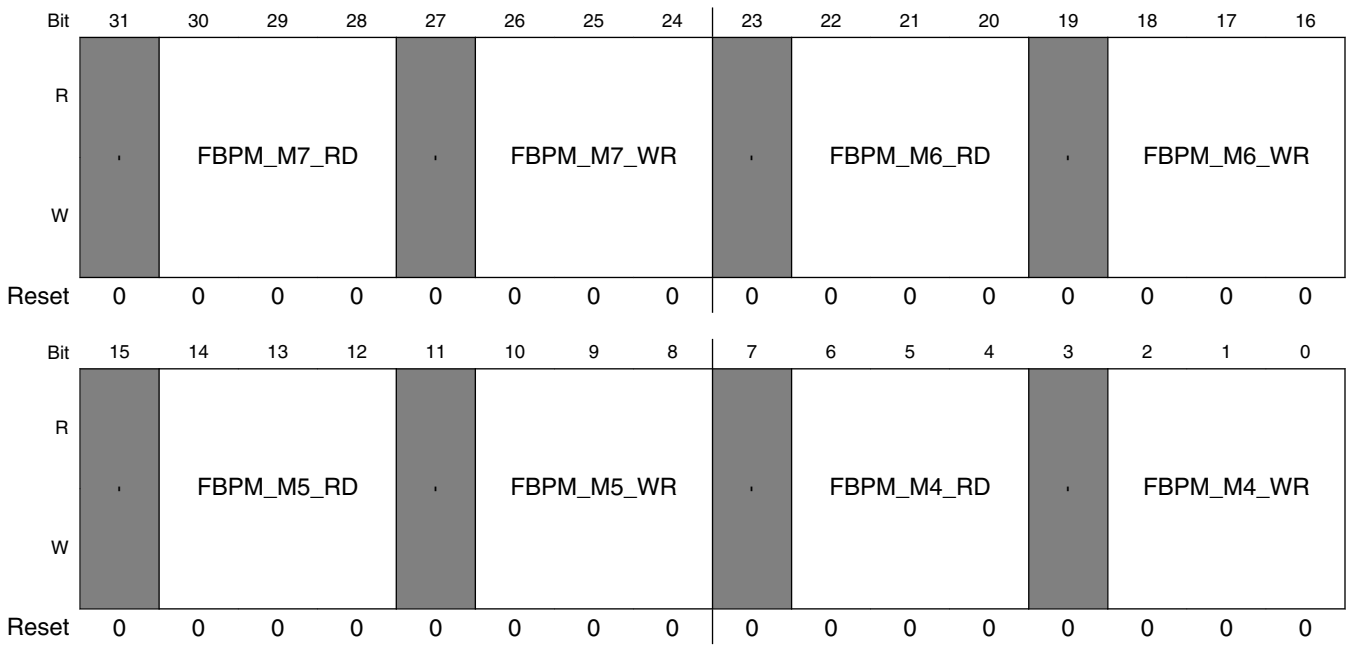
M4IF_F_BPR0 field descriptions (continued)

Field	Description
22–20 FBPM_M2_RD	User defined value. These bits represent the basic priority value for read requests as configured by the user to master 2. 000 encoding 0 001 encoding 1 111 encoding 7
19 -	Reserved
18–16 FBPM_M2_WR	User defined value. These bits represent the basic priority value for write requests as configured by the user to master 2. 000 encoding 0 001 encoding 1 111 encoding 7
15 -	Reserved
14–12 FBPM_M1_RD	User defined value. These bits represent the basic priority value for read requests as configured by the user to master 1. 000 encoding 0 001 encoding 1 111 encoding 7
11 -	Reserved
10–8 FBPM_M1_WR	User defined value. These bits represent the basic priority value for write requests as configured by the user to master 1. 000 encoding 0 001 encoding 1 111 encoding 7
7 -	Reserved
6–4 FBPM_M0_RD	User defined value. These bits represent the basic priority value for read requests as configured by the user to master 0. 000 encoding 0 001 encoding 1 111 encoding 7
3 -	Reserved
2–0 FBPM_M0_WR	User defined value. These bits represent the basic priority value for write requests as configured by the user to master 0. 000 encoding 0 001 encoding 1 111 encoding 7

49.3.14 F_Basic Priority Reg 1 (M4IF_F_BPR1)

M4IF_FBPM1 register contains the basic priority value for masters 4,5,6, and 7 for the fast arbitration logic only. Different registers are used for slow arbitration and internal memory arbitration logic.

Address: M4IF_F_BPR1 is 63FD_8000h base + 44h offset = 63FD_8044h



M4IF_F_BPR1 field descriptions

Field	Description
31 -	Reserved
30–28 FBPM_M7_RD	User defined value. These bits represent the basic priority value for read requests as configured by the user to master 3 000 encoding 0 001 encoding 1 111 encoding 7
27 -	Reserved.
26–24 FBPM_M7_WR	User defined value. These bits represent the basic priority value for write requests as configured by the user to master 7. 000 encoding 0 001 encoding 1 111 encoding 7
23 -	Reserved

Table continues on the next page...

M4IF_F_BPR1 field descriptions (continued)

Field	Description
22–20 FBPM_M6_RD	User defined value. These bits represent the basic priority value for read requests as configured by the user to master 6. 000 encoding 0 001 encoding 1 111 encoding 7
19 -	Reserved
18–16 FBPM_M6_WR	User defined value. These bits represent the basic priority value for write requests as configured by the user to master 6. 000 encoding 0 001 encoding 1 111 encoding 7
15 -	Reserved
14–12 FBPM_M5_RD	User defined value. These bits represent the basic priority value for read requests as configured by the user to master 5. 000 encoding 0 001 encoding 1 111 encoding 7
11 -	Reserved
10–8 FBPM_M5_WR	User defined value. These bits represent the basic priority value for write requests as configured by the user to master 5. 000 encoding 0 001 encoding 1 111 encoding 7
7 -	Reserved
6–4 FBPM_M4_RD	User defined value. These bits represent the basic priority value for read requests as configured by the user to master 4. 000 encoding 0 001 encoding 1 111 encoding 7
3 -	Reserved
2–0 FBPM_M4_WR	User defined value. These bits represent the basic priority value for write requests as configured by the user to master 4. 000 encoding 0 001 encoding 1 111 encoding 7

49.3.15 Control Register (M4IF_CR)

This register determines the values of the weights used for the MIF4 arbitration engine. It is also used for configuring the bypass features of the MIF4 (2nd degree) arbitration.

Address: M4IF_CR is 63FD_8000h base + 48h offset = 63FD_8048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MIF4_BYP_EN	MIF4_MAS			MIF4_ID				MIF4_IDTOO	-	MIF4_PAG_HIT			MIF4_ACC_HIT		
W																
Reset	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-				MIF4_DYN_JMP				MIF4_DYN_MAX				MIF4_GUARD			
W																
Reset	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1

M4IF_CR field descriptions

Field	Description
31 MIF4_BYP_EN	Enable the MIF4 bypass feature. M4IF Bypass 0 - MIF4 bypass is disabled. 1 - MIF4 bypass is enabled. (default)
30–28 MIF4_MAS	This 3 bit field determines which master will be prioritized in the MIF4 arbitration. See M4IF Bypass Default value of MIF4_MAS is 0x000 - master #0.
27–24 MIF4_ID	This 4 bit field determines which AXI ID of the M4IF_MAS will be prioritized provided that the MIF4_IDTOO bit is set to '1'. M4IF Bypass Default value of is MIF4_ID 0x0000 - AXI ID 0.
23 MIF4_IDTOO	This bit defines whether all accesses of the MIF4_MAS master will be prioritized in the MIF4 arbitration, or if only accesses with the specific MIF4_ID will be prioritized. See M4IF Bypass Default value is 1, meaning only a combination of the AXI ID and the master will be prioritized. 0 all access will be prioritized. 1 only accesses with the special AXI ID will be prioritized.
22 -	Reserved

Table continues on the next page...

M4IF_CR field descriptions (continued)

Field	Description
21–19 MIF4_PAG_HIT	MIF4 Page Hit Rate. This value will be added by the MIF4 arbitration logic to any pending request that is considered as page hit. Default value of MIF4_PAG_HIT is 0x00100 - encoding 4.
18–16 MIF4_ACC_HIT	MIF4 Access Hit Rate. Any pending access that is considered as page hit is checked to be a consecutive access type or not. If it is a consecutive access than the weighting value of this request will increase by MIF4_ACC_HIT value. Default value of is MIF4_ACC_HIT 0x0010 - encoding 2.
15–12 -	Reserved
11–8 MIF4_DYN_JMP	MIF4 Dynamic Jump. Each time a request is being popped outside the MIF4 arbitration, the other pending requests will increase there dynamic score value by MIF4_DYN_JMP value. Default MIF4_DYN_JMP value is 0x0001 - encoding 1
7–4 MIF4_DYN_MAX	MIF4 Dynamic Maximum. MIF4_DYN_MAX is the maximum dynamic score value that each request inside the MIF4 mechanism can get. Default MIF4_DYN_MAX value is 0x1000 - encoding 8
3–0 MIF4_GUARD	MIF4 Guard. After a request reached the maximum dynamic score value, it will wait another MIF4_GUARD cycles and then force the request to popped outside the MIF4 arbitration. If MIF4_GUARD equal 0, this guarding mechanism is disabled. See Guarding Mechanism Default MIF4_GUARD value is 0x0000 - encoding 0 (Disabled)

49.3.16 I2_Unit_Level_Arbitration_ Register (M4IF_I2_ULAR)

The M4IF_I2ULA register configures the unit level arbitration configuration bits of each logic unit in levels 3, 4 and 5. Please refer to Arbitration Scheme when Masters have Same Priority (Bus Division) for more details on the arbitration logic units.

Address: M4IF_I2_ULAR is 63FD_8000h base + 74h offset = 63FD_8074h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-											I2L5			I2L4M47		I2L4M03				I2L3M67				I2L3M45				I2L3M23				I2L3M01			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0

M4IF_I2_ULAR field descriptions

Field	Description
31–21 -	Reserved.
20–18 I2L5	Int 2 Level 5 - These bits configure the arbitration mode of level 5 arbitration unit in the int2 arbitration. The arbitration modes configuration options are shown in Table 49-2. Default value of FL5 is 0x100 - encoding 4.

Table continues on the next page...

M4IF_I2_ULAR field descriptions (continued)

Field	Description
17–15 I2L4M47	Int 2 Level 4 (m47) - These bits configure the arbitration mode of level 4(m47) arbitration unit in the int2 arbitration. The arbitration modes configuration options are shown in Table 49-2. Default value of FL4M47 is 0x100 - encoding 4.
14–12 I2L4M03	Int 2 Level 4 (m03) - These bits configure the arbitration mode of level 4(m03) arbitration unit in the int2 arbitration. The arbitration modes configuration options are shown in Table 49-2. Default value of FL4M03 is 0x100 - encoding 4.
11–9 I2L3M67	Int 2 Level 3 (m67) - These bits configure the arbitration mode of level 3(m67) arbitration unit in the int2 arbitration. The arbitration modes configuration options are shown in Table 49-2. Default value of FL3M67 is 0x100 - encoding 4.
8–6 I2L3M45	Int 2 Level 3 (m45) - These bits configure the arbitration mode of level 3(m45) arbitration unit in the int2 arbitration. The arbitration modes configuration options are shown in Table 49-2. Default value of FL3M45 is 0x100 - encoding 4.
5–3 I2L3M23	Int 2 Level 3 (m23) - These bits configure the arbitration mode of level 3(m23) arbitration unit in the int2 arbitration. The arbitration modes configuration options are shown in Table 49-2. Default value of FL3M23 is 0x100 - encoding 4.
2–0 I2L3M01	Int 2 Level 3 (m01) - These bits configure the arbitration mode of level 3(m01) arbitration unit in the int2 arbitration. The arbitration modes configuration options are shown in Table 49-2. Default value of FL3M01 is 0x100 - encoding 4.

49.3.17 Int. 2 Memory Arbitration Control Register (M4IF_I2MACR)

Address: M4IF_I2MACR is 63FD_8000h base + 78h offset = 63FD_8078h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-							-	-							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

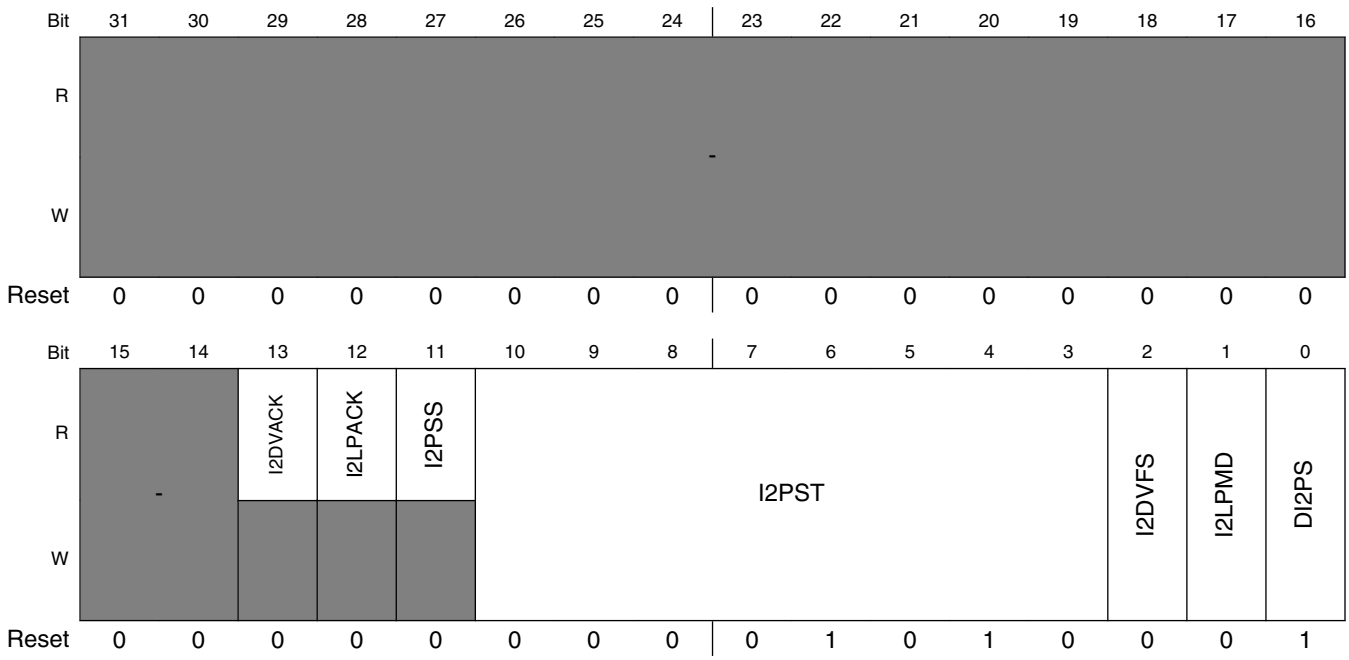
M4IF_I2MACR field descriptions

Field	Description
31–9 -	Reserved
8 -	Reserved must be cleared
7–0 -	Reserved

49.3.18 Internal 2 Control Register (M4IF_I2CR)

The M4IF_RINT2 holds control bits for the int2. memory channel arbitration.

Address: M4IF_I2CR is 63FD_8000h base + 7Ch offset = 63FD_807Ch



M4IF_I2CR field descriptions

Field	Description
31–14 -	Reserved
13 I2DVACK	Intr 2 DVFS acknowledge. This read only bit indicates whether a DVFS acknowledge on the internal 2 memory channel was asserted.
12 I2LPACK	Intr 2 low-power acknowledge. This read only bit indicates whether a low-power acknowledge on the internal 2 memory channel was asserted.

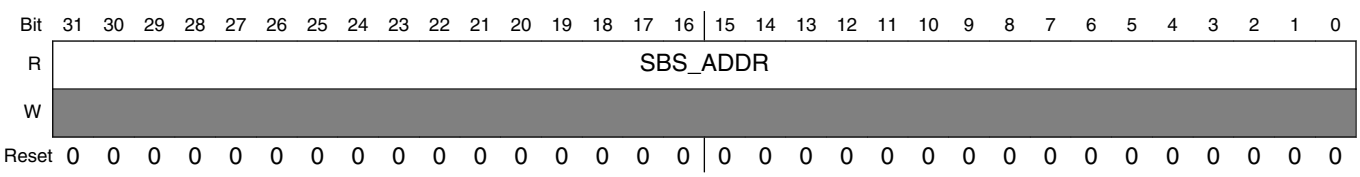
Table continues on the next page...

M4IF_I2CR field descriptions (continued)

Field	Description
11 I2PSS	Intr 2 Power Saving Status. This read only bit indicates whether internal 2 mem. channel is in power saving mode. 0 not in power saving 1 power saving
10–3 I2PST	Internal 2memory arbitration Power Saving Timer - these bits defines the timer value of the power saving mode of the internal 2 memory arbitration. The value in this field represent clock cycles of the internal 2 memory arbitration clock multiplied by 8. Default value is set to 80 clock cycles. 00000000 Reserved - this value is forbidden. 00000001 timer is configured to 8 clock cycles. 00001010 Default value- 80 clock cycles. 11111111 timer clock is defined to 2040 clock cycles.
2 I2DVFS	Internal mem. 2 DVFS request. SW request for DVFS of internal 2 mem. channel. 0 no DVFS request 1 DVFS request
1 I2LPMD	Internal mem. 2 LPMD request. SW request for LPMD of internal 2 mem. channel. 0 no LPMD request 1 LPMD request
0 DI2PS	Disable Internal 2 memory Power Saving mode - this bit when configured can disable auto power saving mode of the internal 2 memory interface. 0 power saving is enabled 1 power saving is disabled (default mode)

49.3.19 Step By Step Address (M4IF_SSA)

Address: M4IF_SSA is 63FD_8000h base + 84h offset = 63FD_8084h



M4IF_SSA field descriptions

Field	Description
31–0 SBS_ADDR	Step By Step Address. These bits reflect the address of a request pending on a specific arbitration/ memory controller, as configured in bits [9:8] (RARB) of the M4IF_MDCR register.

49.3.20 Step By Step Address Controls (M4IF_SSAC)

Address: M4IF_SSAC is 63FD_8000h base + 88h offset = 63FD_8088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-		SBS_VLD	SBS_TYPE	SBS_LEN			SBS_SIZE		SBS_BURST		SBS_PROT			SBS_LOCK	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SBS_CACHE				SBS_AXI_ID				SBS_MASTER_ID				SBS_MASTER_ID			SBS_END
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

M4IF_SSAC field descriptions

Field	Description
31–30 -	Reserved
29 SBS_VLD	Step By Step Valid. This bit reflects whether there is a pending request in the arbitration that is configured in bits [9:8] (RARB) of the M4IF_MDCR register. 0 not valid 1 valid
28 SBS_TYPE	Step By Step Request Type. These bits reflect the pending request type (read/write) as configured in bits [9:8] (RARB) of the M4IF_MDCR register. 0 write 1 read
27–25 SBS_LEN	Step By Step Length. These bits reflect the AXI "LEN" field of the pending request as configured in bits [9:8] (RARB) of the M4IF_MDCR register. 000 burst of length 1 001 burst of length 2 111 burst of length 8
24–23 SBS_SIZE	Step By Step Size. These bits reflect the AXI "SIZE" field of the pending request as configured in bits [9:8] (RARB) of the M4IF_MDCR register. 00 8 bits 01 16 bits 10 32 bits 11 64 bits
22–21 SBS_BURST	Step By Step Burst. These bits reflect the AXI "BURST" field of the pending request as configured in bits [9:8] (RARB) of the M4IF_MDCR register.

Table continues on the next page...

M4IF_SSAC field descriptions (continued)

Field	Description
	00 reserved 01 INCR burst 10 WRAP burst 11 reserved
20–18 SBS_PROT	Step By Step Protection. These bits reflect the AXI "PROT" field of the pending request as configured in bits [9:8] (RARB) of the M4IF_MDCR register.
17–16 SBS_LOCK	Step By Step Lock. These bits reflect the AXI "LOCK" field of the pending request as configured in bits [9:8] (RARB) of the M4IF_MDCR register.
15–12 SBS_CACHE	Step By Step Cache. These bits reflect the AXI "CACHE" field of the pending request as configured in bits [9:8] (RARB) of the M4IF_MDCR register.
11–8 SBS_AXI_ID	Step By Step AXI ID. These bits reflect the AXI "ID" field of the pending request as configured in bits [9:8] (RARB) of the M4IF_MDCR register.
7–4 SBS_MASTER_ID	Step By Step Master ID. These bits reflect the Master ID of the pending request as configured in bits [9:8] (RARB) of the M4IF_MDCR register.
3–1 SBS_MASTER_ID	Step By Step Master ID. These bits reflect the EXTMC Master port of the pending request as configured in bits [9:8] (RARB) of the M4IF_MDCR register.
0 SBS_END	Step By Step Endianess. This bit reflects the endianess of the pending request as configured in bits [9:8] (RARB) of the M4IF_MDCR register.

49.3.21 Control Register 0 (M4IF_CR0)

Address: M4IF_CR0 is 63FD_8000h base + 8Ch offset = 63FD_808Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_RST	-			FDVACK	SDVACK	IDVACK	DVACK	FLPACK	SLPACK	ILPACK	LPACK	-	FPSS	SPSS	IPSS
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EAS3	EAS2	EAS1	EAS0	FDVFS	SDVFS	IDVFS	DVFS	FLPMD	SLPMD	ILPMD	LPMD	-	DFPS	DSPS	DIPS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

M4IF_CR0 field descriptions

Field	Description
31 SW_RST	Software reset. This bit resets all FFs but the configuration registers. 0 reset 1 no reset
30–28 -	Reserved.
27 FDVACK	Fast DVFS acknowledge. This read only bit indicates whether a DVFS acknowledge on the fast channel was asserted.
26 SDVACK	Slow DVFS acknowledge. This read only bit indicates whether a DVFS acknowledge on the slow channel was asserted.
25 IDVACK	Intr DVFS acknowledge. This read only bit indicates whether a DVFS acknowledge on the internal memory channel was asserted.
24 DVACK	General DVFS acknowledge. This read only bit indicates whether a DVFS acknowledge on the whole EXTMC was asserted.
23 FLPACK	Fast low-power acknowledge. This read only bit indicates whether a low-power acknowledge on the fast channel was asserted.
22 SLPACK	Slow low-power acknowledge. This read only bit indicates whether a low-power acknowledge on the slow channel was asserted.
21 ILPACK	Intr low-power acknowledge. This read only bit indicates whether a low-power acknowledge on the internal memory channel was asserted.
20 LPACK	General low-power acknowledge. This read only bit indicates whether a low-power acknowledge on the whole EXTMC was asserted.
19 -	Reserved.
18 FPSS	Fast Power Saving Status. This read only bit indicates whether fast channel is in power saving mode. 0 not in power saving 1 power saving
17 SPSS	Slow Power Saving Status. This read only bit indicates whether slow channel is in power saving mode. 0 not in power saving 1 power saving
16 IPSS	Intr Power Saving Status. This read only bit indicates whether internal mem. channel is in power saving mode. 0 not in power saving 1 power saving
15 EAS3	Enable Arbit Stop M3 - Stop all read accesses to M3 that are not from to the same slave as the ones already inside the gasket. 0 disabled (not stopping) 1 enabled (stopping) - default
14 EAS2	Enable Arbit Stop M2 - Stop all read accesses to M2 that are not from to the same slave as the ones already inside the gasket.

Table continues on the next page...

M4IF_CR0 field descriptions (continued)

Field	Description
	0 disabled (not stopping) - default 1 enabled (stopping)
13 EAS1	Enable Arbit Stop M1 - Stop all read accesses to M1 that are not from to the same slave as the ones already inside the gasket. 0 disabled (not stopping) - default 1 enabled (stopping)
12 EAS0	Enable Arbit Stop M0 - Stop all read accesses to M0 that are not from to the same slave as the ones already inside the gasket. 0 disabled (not stopping) 1 enabled (stopping) - default
11 FDVFS	Fast DVFS request. SW request for DVFS of fast channel. 0 no DVFS request 1 DVFS request
10 SDVFS	Slow DVFS request. SW request for DVFS of slow channel. 0 no DVFS request 1 DVFS request
9 IDVFS	Internal mem. DVFS request. SW request for DVFS of internal mem. channel. 0 no DVFS request 1 DVFS request
8 DVFS	General DVFS request. SW request for DVFS of whole EXTMC. 0 no DVFS request 1 DVFS request
7 FLPMD	Fast LPMD request. SW request for LPMD of fast channel. 0 no LPMD request 1 LPMD request
6 SLPMD	Slow LPMD request. SW request for LPMD of slow channel. 0 no LPMD request 1 LPMD request
5 ILPMD	Internal mem. LPMD request. SW request for LPMD of internal mem. channel. 0 no LPMD request 1 LPMD request
4 LPMD	General LPMD request. SW request for LPMD of whole EXTMC. 0 no LPMD request 1 LPMD request
3 -	Reserved.

Table continues on the next page...

M4IF_CR0 field descriptions (continued)

Field	Description
2 DFPS	Disable Fast arb. Power Saving mode - this bit when configured can disable auto power saving mode of the fast arbitration and its memory controller. 0 power saving is enabled 1 power saving is disabled (default mode)
1 DSPS	Disable Slow arb. Power Saving mode - this bit when configured can disable auto power saving mode of the slow arbitration and its memory controllers. 0 power saving is enabled 1 power saving is disabled (default mode)
0 DIPS	Disable Internal memory Power Saving mode - this bit when configured can disable auto power saving mode of the internal memory interface. 0 power saving is enabled 1 power saving is disabled (default mode)

49.3.22 Control Register 1 (M4IF_CR1)

The M4IF_MCR1 register defines the timer values of each arbitration to be used for auto power saving mode.

Address: M4IF_CR1 is 63FD_8000h base + 90h offset = 63FD_8090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EAS7	EAS6	EAS5	EAS4	-				IPST						-	SPST [-8:7]
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPST[6:0]							-	FPST							
Reset	0	0	0	1	0	1	0	0	0	0	0	0	1	0	1	0

M4IF_CR1 field descriptions

Field	Description
31 EAS7	Enable Arbit Stop M7 - Stop all read accesses to M7 that are not from to the same slave as the ones already inside the gasket. 0 disabled (not stopping) 1 enabled (stopping) - default
30 EAS6	Enable Arbit Stop M6 - Stop all read accesses to M6 that are not from to the same slave as the ones already inside the gasket. 0 disabled (not stopping) - default 1 enabled (stopping)
29 EAS5	Enable Arbit Stop M5 - Stop all read accesses to M5 that are not from to the same slave as the ones already inside the gasket. 0 disabled (not stopping) - default 1 enabled (stopping)
28 EAS4	Enable Arbit Stop M4 - Stop all read accesses to M4 that are not from to the same slave as the ones already inside the gasket. 0 disabled (not stopping) - default 1 enabled (stopping)
27–26 -	Reserved
25–18 IPST	Internal 1 memory arbitration Power Saving Timer - these bits defines the timer value of the power saving mode of the internal 1 memory arbitration. The value in this field represent clock cycles of the internal 1 memory arbitration clock multiplied by 8. Default value is set to 80 clock cycles. 00000000 Reserved - this value is forbidden. 00000001 timer is configured to 8 clock cycles. 00001010 Default value- 80 clock cycles. 11111111 timer clock is defined to 2040 clock cycles.
17 -	Reserved
16–9 SPST	Slow arbitration Power Saving Timer - these bits defines the timer value of the power saving mode of the slow arbitration. The value in this field represent clock cycles of the slow memory arbitration clock multiplied by 8. Default value is set to 80 clock cycles. 00000000 Reserved - this value is forbidden. 00000001 timer is configured to 8 clock cycles. 00001010 Default value- 80 clock cycles. 11111111 timer clock is defined to 2040 clock cycles.
8 -	Reserved

Table continues on the next page...

M4IF_CR1 field descriptions (continued)

Field	Description
7–0 FPST	Fast arbitration Power Saving Timer - these bits defines the timer value of the power saving mode of the fast arbitration. The value in this field represent clock cycles of the fast memory arbitration clock multiplied by 8. Default value is set to 80 clock cycles. 00000000 Reserved - this value is forbidden. 00000001 timer is configured to 8 clock cycles. 00001010 Default value- 80 clock cycles. 11111111 timer clock is defined to 2040 clock cycles.

49.3.23 Debug Control Register (M4IF_DCR)

The M4IF_MDCR controls several functions in the debug unit.

Address: M4IF_DCR is 63FD_8000h base + 94h offset = 63FD_8094h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DBG_EN	-			DBG_RST	-			I2SBS_EN	I1SBS_EN	SSBS_EN	DDPT	DDPM			-
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FSBS_EN	FSBS	SSBS	I1SBS	I2SBS	-	RARB		-						VARB	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

M4IF_DCR field descriptions

Field	Description
31 DBG_EN	Debug Enable. Enable debug units counters and summing. default is "disable" 0 disable 1 enable

Table continues on the next page...

M4IF_DCR field descriptions (continued)

Field	Description
30–28 -	Reserved.
27 DBG_RST	Debug Reset. Reset all debug unit's counters and summing. 0 no reset 1 reset
26–24 -	Reserved.
23 I2SBS_EN	Step By Step Enable int2. Enable step by step mode for int2 channel. default is "disable" 0 disable 1 enable
22 I1SBS_EN	Step By Step Enable int1. Enable step by step mode for int1 channel. default is "disable" 0 disable 1 enable
21 SSBS_EN	Step By Step Enable slow. Enable step by step mode for slow channel. default is "disable" 0 disable 1 enable
20 DDPT	Debug Dynamic Priority Type. Read or Write request reflect on the debug unit registers. 0 write 1 read
19–17 DDPM	Debug Dynamic Priority Master. Which master to reflect on the debug unit registers.
16 -	Reserved.
15 FSBS_EN	Step By Step Enable fast. Enable step by step mode for fast channel. default is "disable" 0 disable 1 enable
14 FSBS	Fast Step By Step. This bit trigger the fast arbitration in step by step mode. Write '1' to this register in order to initiate a new transaction inside the fast arbitration. This bit goes down to '0' when the new transaction is pending on AXI bus and then waiting for another write '1' in order to initiate the next transaction. 1 Fast arbitration is waiting to a new transaction to go out on the AXI bus. 0 There is a pending transaction on the AXI bus.
13 SSBS	Slow Step By Step. This bit trigger the slow arbitration in step by step mode. Write '1' to this register in order to initiate a new transaction inside the slow arbitration. This bit goes down to '0' when the new

Table continues on the next page...

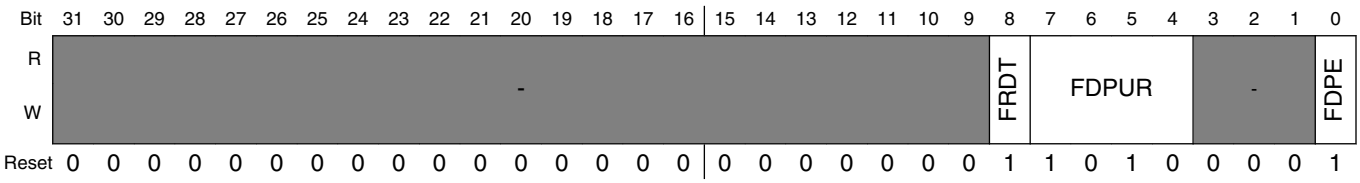
M4IF_DCR field descriptions (continued)

Field	Description
	transaction is pending on AXI bus and then waiting for another write '1' in order to initiate the next transaction. 1 Slow arbitration is waiting to a new transaction to go out on the AXI bus. 0 There is a pending transaction on the AXI bus.
12 I1SBS	Int1 Step By Step. This bit trigger the int1 arbitration in step by step mode. Write '1' to this register in order to initiate a new transaction inside the int1 arbitration. This bit goes down to '0' when the new transaction is pending on AXI bus and then waiting for another write '1' in order to initiate the next transaction. 1 Int1 arbitration is waiting to a new transaction to go out on the AXI bus. 0 There is a pending transaction on the AXI bus.
11 I2SBS	Int2 Step By Step. This bit trigger the int2 arbitration in step by step mode. Write '1' to this register in order to initiate a new transaction inside the int2 arbitration. This bit goes down to '0' when the new transaction is pending on AXI bus and then waiting for another write '1' in order to initiate the next transaction. 1 Int2 arbitration is waiting to a new transaction to go out on the AXI bus. 0 There is a pending transaction on the AXI bus.
10 -	Reserved.
9–8 RARB	Read Arbitration. These 2 bits decide which arbitration will be reflected on the debug status registers when reading them. 00 fast arbitration 01 slow arbitration 10 internal 1 mem. arbitration 11 internal 2 mem. arbitration
7–2 -	Reserved
1–0 VARB	Visibility Arbitration. These 2 bits decide which arbitration to reflect on the EXTMC debug output pins and on the step by step registers. 00 fast arbitration 01 slow arbitration 10 internal 1 mem. arbitration 11 internal 2 mem. arbitration

49.3.24 Fast Arbitration Control Register (M4IF_FACR)

The M4IF_FACR register is the control register of the fast arbitration logic. It contains several configuration field in order to let the user optimize the arbitration.

Address: M4IF_FACR is 63FD_8000h base + 98h offset = 63FD_8098h



M4IF_FACR field descriptions

Field	Description
31–9 -	Reserved
8 FRDT	Read Through bit - this bit indicates the read mode from fast channel 0 store and forward 1 read through (default)
7–4 FDPUR	Dynamic Priority Update Resolution - This field defines the minimum time resolution (in number of arbitration selections) that each priority master would be update by the dynamic priority logic.The minimum value is 2 selections. The value can be changed by the user in order to achieve lower resolution to reduce dynamic priority influence on the arbitration logic. It is recommended not to use higher value than average access time of the memory controller. 0000 reserved 0001 encoding 2 1010 Default value- encoding 11. 1111 Dynamic priority calculation updates will be done once per 15 selections minimum.
3–1 -	Reserved
0 FDPE	Dynamic Priority Enable. This bit defines whether arbitration logic would use dynamic priority logic or not. 0 Dynamic Priority is not used, disabled. 1 Dynamic Priority is used, enabled.

49.3.25 F_Priority Weighting Configuration Register (M4IF_F_PWCR)

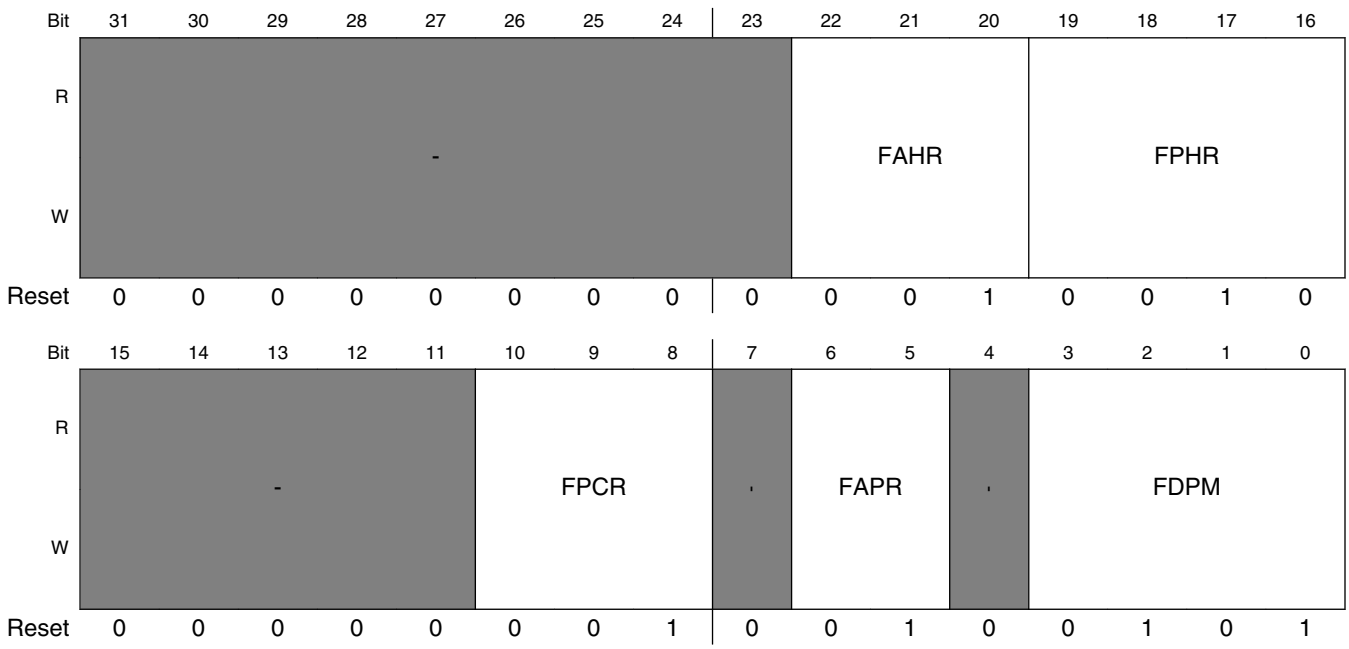
The M4IF_FPWC register configures the weighting values of each field in the arbitration. Each weighting field has a default value based on model simulation. It is highly recommended to use these default values and and to refrain from changing them. Any change to these values should be done with extreme care.

NOTE

At any configuration of these values, the user must make sure that the following restriction is kept:

FPHR+FAHR+FDPM+[highest basic priority of fast channel] < 16

Address: M4IF_F_PWCR is 63FD_8000h base + 9Ch offset = 63FD_809Ch



M4IF_F_PWCR field descriptions

Field	Description
31–23 -	Reserved.
22–20 FAHR	Access Hit Rate. Any pending access that is considered as page hit is checked to be a consecutive access type or not. If it is a consecutive access than the weighting value of this request will increase by FAHR value. Default value of FAHR is 0x001 - encoding 1.
19–16 FPHR	Page Hit Rate. This value will be added by the dynamic arbitration logic to any pending request that is considered as page hit. Default value of FPHR is 0x0010 - encoding 2.
15–11 -	Reserved.
10–8 FPCR	Pending Cycles Rate. Each timing resolution the dynamic arbitration updates a pending request, it will increase its weighting value by FPCR value. Default FPCR value is 0x001 - encoding 1.
7 -	Reserved.
6–5 FAPR	Access Penalty Rate. This value is decreased from any data in a served request. For example, for served access of burst2 type, the weighting value will be decreased by 2xFAPR.

Table continues on the next page...

M4IF_F_PWCR field descriptions (continued)

Field	Description
	Default FAPR value is 0x01 - encoding 1.
4 -	Reserved.
3–0 FDPM	Dynamic Priority Maximum. This field defines the maximum value of the dynamic priority. This value can be changed in order to get larger window for the dynamic arbitration. Default value of FDPM is 0x0101 - encoding 5.

49.3.26 Slow Arbitration Control Register (M4IF_SACR)

Address: M4IF_SACR is 63FD_8000h base + A0h offset = 63FD_80A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-							'	-							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

M4IF_SACR field descriptions

Field	Description
31–9 -	Reserved
8 -	Reserved. Must be cleared
7–0 -	Reserved

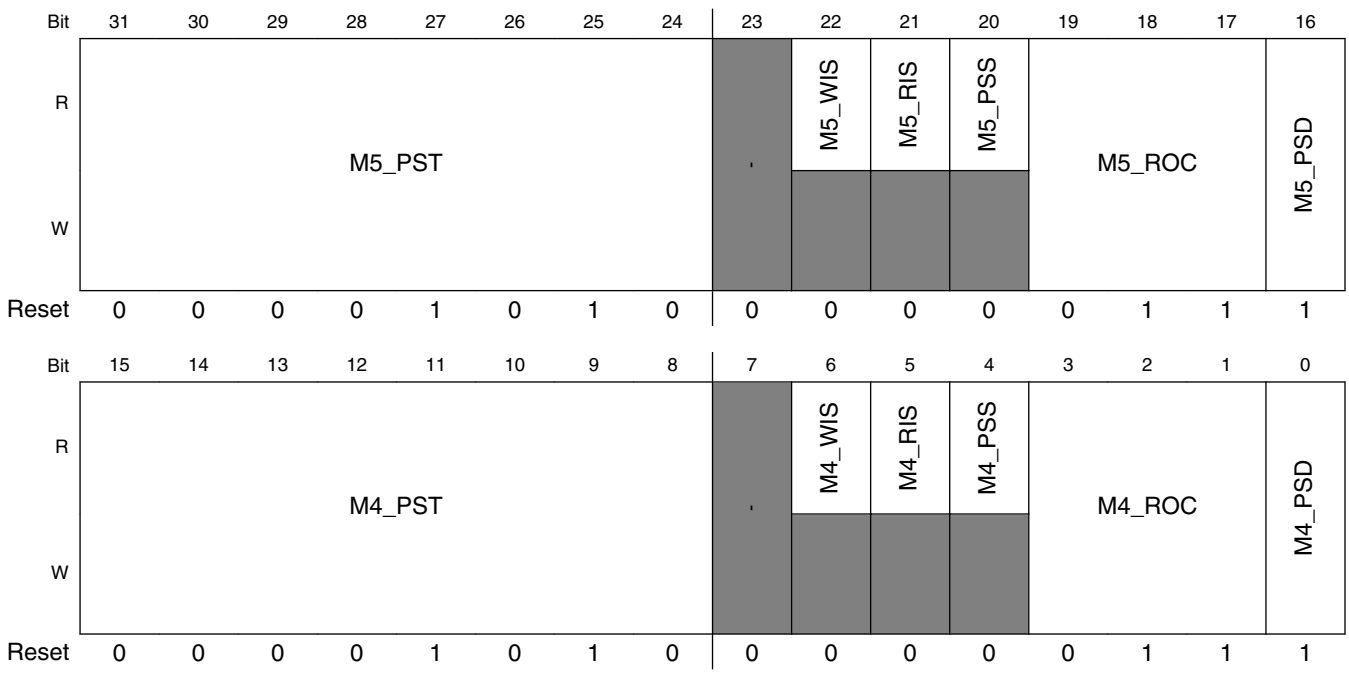
49.3.27 Power Saving Masters 2 (M4IF_PSM2)

The M4IF_PSM2 determines the power saving features for masters #4 and #5.

NOTE

The 'PST' and 'ROC' parameters can only be changed once after reset and before enabling the power saving feature (PSD=0)

Address: M4IF_PSM2 is 63FD_8000h base + A4h offset = 63FD_80A4h



M4IF_PSM2 field descriptions

Field	Description
31–24 M5_PST	master #5 power saving timer. The real value which is used is register-value multiplied by 100. Default value is set to 1000 clock cycles. 00000000 Reserved - this value is forbidden. 00000001 timer is configured to 100 clock cycles. 00001010 Default value- 1000 clock cycles. 11111111 timer clock is defined to 25500 clock cycles.
23 -	Reserved.
22 M5_WIS	Master #5 Write Idle Status.This read only bit indicates whether master #5 write request buffer is idle (empty) or not. 0 idle 1 not idle

Table continues on the next page...

M4IF_PSM2 field descriptions (continued)

Field	Description
21 M5_RIS	Master #5 Read Idle Status.This read only bit indicates whether master #5 read request buffer is idle (empty) or not. 0 idle 1 not idle
20 M5_PSS	Masrer #5 Power Saving Status. This read only bit indicates whether master #5 gasket is in power saving mode. 0 not in power saving 1- power saving
19–17 M5_ROC	master #5 ready off cycles. Number of minimum cycles that the ready signals must be low before they can go back to high. 001-111 (000 is a forbidden value)
16 M5_PSD	master #5 power saving disable. 0 power saving enabled 1 power saving disabled (default)
15–8 M4_PST	master #4 power saving timer. The real value which is used is register-value multiplied by 100. Default value is set to 1000 clock cycles. 00000000 Reserved - this value is forbidden. 00000001 timer is configured to 100 clock cycles. 00001010 Default value- 1000 clock cycles. 11111111 timer clock is defined to 25500 clock cycles.
7 -	Reserved.
6 M4_WIS	Master #4 Write Idle Status.This read only bit indicates whether master #4 write request buffer is idle (empty) or not. 0 idle 1 not idle
5 M4_RIS	Master #4 Read Idle Status.This read only bit indicates whether master #4 read request buffer is idle (empty) or not. 0 idle 1 not idle
4 M4_PSS	Masrer #4 Power Saving Status. This read only bit indicates whether master #4 gasket is in power saving mode. 0 not in power saving 1 power saving
3–1 M4_ROC	master #4 ready off cycles. Number of minimum cycles that the ready signals must be low before they can go back to high. Default value is 3 cycles.

Table continues on the next page...

M4IF_PSM2 field descriptions (continued)

Field	Description
	001-111 (000 is a forbidden value)
0 M4_PSD	master #4 power saving disable. 0 power saving enabled 1 power saving disabled (default)

49.3.28 Int. Memory Arbitration Control Register (M4IF_IMACR)

Address: M4IF_IMACR is 63FD_8000h base + A8h offset = 63FD_80A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-							-	-							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

M4IF_IMACR field descriptions

Field	Description
31–9 -	Reserved
8 -	Reserved. Must be cleared
7–0 -	Reserved

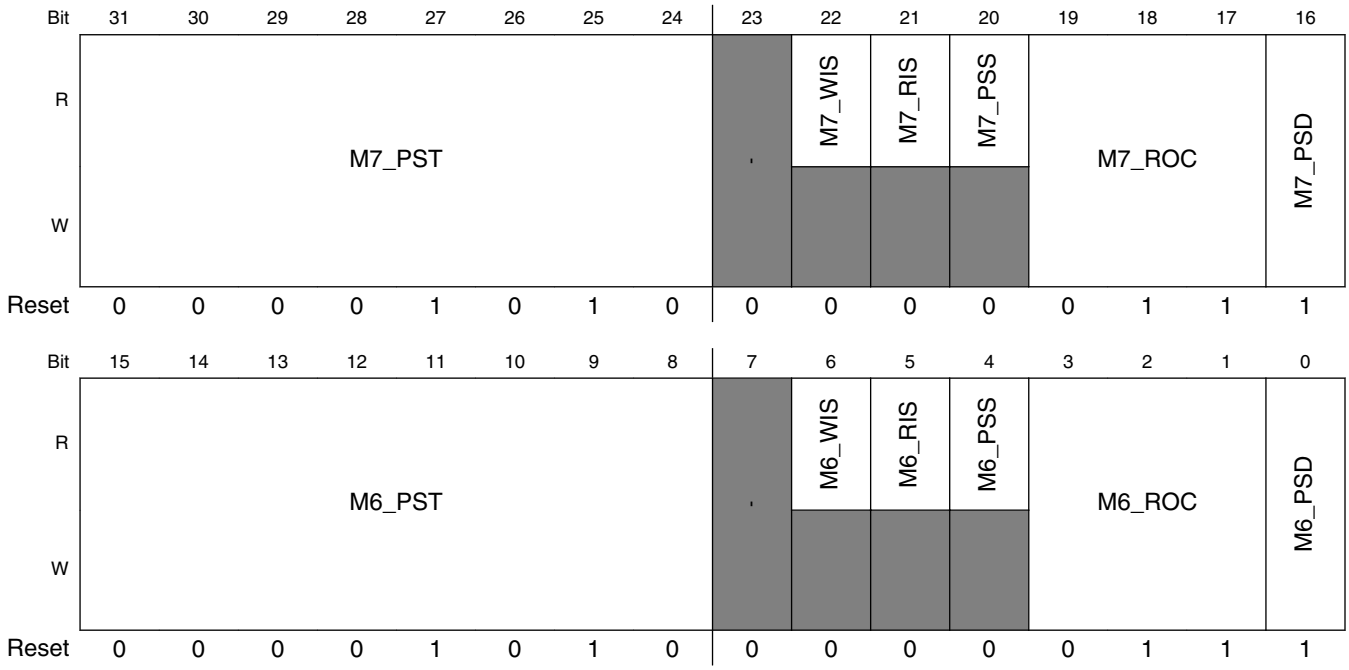
49.3.29 Power Saving Masters 3 (M4IF_PSM3)

The M4IF_PSM3 determines the power saving features for masters #6 and #7.

NOTE

The 'PST' and 'ROC' parameters can only be changed once after reset and before enabling the power saving feature (PSD=0)

Address: M4IF_PSM3 is 63FD_8000h base + ACh offset = 63FD_80ACh



M4IF_PSM3 field descriptions

Field	Description
31–24 M7_PST	master #7 power saving timer. The real value which is used is register-value multiplied by 100. Default value is set to 1000 clock cycles. 00000000 Reserved - this value is forbidden. 00000001 timer is configured to 100 clock cycles. 00001010 Default value- 1000 clock cycles. 11111111 timer clock is defined to 25500 clock cycles.
23 -	Reserved.
22 M7_WIS	Master #7 Write Idle Status.This read only bit indicates whether master #7 write request buffer is idle (empty) or not. 0 idle 1 not idle
21 M7_RIS	Master #7 Read Idle Status.This read only bit indicates whether master #7 read request buffer is idle (empty) or not. 0 idle 1 not idle
20 M7_PSS	Masrer #7 Power Saving Status. This read only bit indicates whether master #7 gasket is in power saving mode.

Table continues on the next page...

M4IF_PSM3 field descriptions (continued)

Field	Description
	0 not in power saving 1 power saving
19–17 M7_ROC	master #7 ready off cycles. Number of minimum cycles that the ready signals must be low before they can go back to high. 001-111 (000 is a forbidden value)
16 M7_PSD	master #7 power saving disable. 0 power saving enabled 1 power saving disabled (default)
15–8 M6_PST	master #6 power saving timer. The real value which is used is register-value multiplied by 100. Default value is set to 1000 clock cycles. 00000000 Reserved - this value is forbidden. 00000001 timer is configured to 100 clock cycles. 00001010 Default value 1000 clock cycles. 11111111 timer clock is defined to 25500 clock cycles.
7 -	Reserved.
6 M6_WIS	Master #6 Write Idle Status.This read only bit indicates whether master #6 write request buffer is idle (empty) or not. 0 idle 1 not idle
5 M6_RIS	Master #6 Read Idle Status.This read only bit indicates whether master #6 read request buffer is idle (empty) or not. 0 idle 1 not idle
4 M6_PSS	Masrer #6 Power Saving Status. This read only bit indicates whether master #6 gasket is in power saving mode. 0 not in power saving 1 power saving
3–1 M6_ROC	master #6 ready off cycles. Number of minimum cycles that the ready signals must be low before they can go back to high. Default value is 3 cycles. 001-111 (000 is a forbidden value)
0 M6_PSD	master #6 power saving disable. 0 power saving enabled 1 power saving disabled (default)

49.3.30 F_Unit_Level_Arbitration_ Register (M4IF_F_ULAR)

The M4IF_FULA register configures the unit level arbitration configuration bits of each logic unit in levels 3, 4 and 5. Please refer to Arbitration Scheme when Masters have Same Priority (Bus Division) for more details on the arbitration logic units.

Address: M4IF_F_ULAR is 63FD_8000h base + B0h offset = 63FD_80B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-											FL5			FL4M47		FL4M03				FL3M67			FL3M45			FL3M23			FL3M01		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

M4IF_F_ULAR field descriptions

Field	Description
31–21 -	Reserved.
20–18 FL5	Fast Level 5 - These bits configure the arbitration mode of level 5 arbitration unit in the fast arbitration. The arbitration modes configuration options are shown in Table 49-2. Default value of FL5 is 0x000 - encoding 0.
17–15 FL4M47	Fast Level 4 (m47) - These bits configure the arbitration mode of level 4(m47) arbitration unit in the fast arbitration. The arbitration modes configuration options are shown in Table 49-2. Default value of FL4M47 is 0x000 - encoding 0.
14–12 FL4M03	Fast Level 4 (m03) - These bits configure the arbitration mode of level 4(m03) arbitration unit in the fast arbitration. The arbitration modes configuration options are shown in Table 49-2. Default value of FL4M03 is 0x000 - encoding 0.
11–9 FL3M67	Fast Level 3 (m67) - These bits configure the arbitration mode of level 3(m67) arbitration unit in the fast arbitration. The arbitration modes configuration options are shown in Table 49-2. Default value of FL3M67 is 0x000 - encoding 0.
8–6 FL3M45	Fast Level 3 (m45) - These bits configure the arbitration mode of level 3(m45) arbitration unit in the fast arbitration. The arbitration modes configuration options are shown in Table 49-2. Default value of FL3M45 is 0x000 - encoding 0.
5–3 FL3M23	Fast Level 3 (m23) - These bits configure the arbitration mode of level 3(m23) arbitration unit in the fast arbitration. The arbitration modes configuration options are shown in Table 49-2. Default value of FL3M23 is 0x000 - encoding 0.
2–0 FL3M01	Fast Level 3 (m01) - These bits configure the arbitration mode of level 3(m01) arbitration unit in the fast arbitration. The arbitration modes configuration options are shown in Table 49-2. Default value of FL3M01 is 0x000 - encoding 0.

49.3.31 S_Unit_Level_Arbitration_ Register (M4IF_S_ULAR)

The M4IF_SULA register configures the unit level arbitration configuration bits of each logic unit in levels 3, 4 and 5. Please refer to Arbitration Scheme when Masters have Same Priority (Bus Division) for more details on the arbitration logic units.

Address: M4IF_S_ULAR is 63FD_8000h base + B4h offset = 63FD_80B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-											SL5			SL4M47		SL4M03			SL3M67			SL3M45			SL3M23			SL3M01		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0

M4IF_S_ULAR field descriptions

Field	Description
31–21 -	Reserved.
20–18 SL5	Slow Level 5 - These bits configure the arbitration mode of level 5 arbitration unit in the slow arbitration. The arbitration modes configuration options are shown in Table 49-3. Default value of SL5 is 0x100 - encoding 4.
17–15 SL4M47	Slow Level 4 (m47) - These bits configure the arbitration mode of level 4(m47) arbitration unit in the slow arbitration. The arbitration modes configuration options are shown in Table 49-3. Default value of SL4M47 is 0x100 - encoding 4.
14–12 SL4M03	Slow Level 4 (m03) - These bits configure the arbitration mode of level 4(m03) arbitration unit in the slow arbitration. The arbitration modes configuration options are shown in Table 49-3. Default value of SL4M03 is 0x100 - encoding 4.
11–9 SL3M67	Slow Level 3 (m67) - These bits configure the arbitration mode of level 3(m67) arbitration unit in the slow arbitration. The arbitration modes configuration options are shown in Table 49-3. Default value of SL3M67 is 0x100 - encoding 4.
8–6 SL3M45	Fast Level 3 (m45) - These bits configure the arbitration mode of level 3(m45) arbitration unit in the slow arbitration. The arbitration modes configuration options are shown in Table 49-3. Default value of SL3M45 is 0x100 - encoding 4.
5–3 SL3M23	Slow Level 3 (m23) - These bits configure the arbitration mode of level 3(m23) arbitration unit in the slow arbitration. The arbitration modes configuration options are shown in Table 49-3. Default value of SL3M23 is 0x100 - encoding 4.
2–0 SL3M01	Slow Level 3 (m01) - These bits configure the arbitration mode of level 3(m01) arbitration unit in the slow arbitration. The arbitration modes configuration options are shown in Table 49-3. Default value of SL3M01 is 0x100 - encoding 4.

49.3.32 I_Unit_Level_Arbitration_ Register (M4IF_I_ULAR)

The M4IF_IULA register configures the unit level arbitration configuration bits of each logic unit in levels 3, 4 and 5. Please refer to Arbitration Scheme when Masters have Same Priority (Bus Division) for more details on the arbitration logic units.

Address: M4IF_I_ULAR is 63FD_8000h base + B8h offset = 63FD_80B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-											IL5			IL4M47		IL4M03				IL3M67			IL3M45			IL3M23			IL3M01		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0

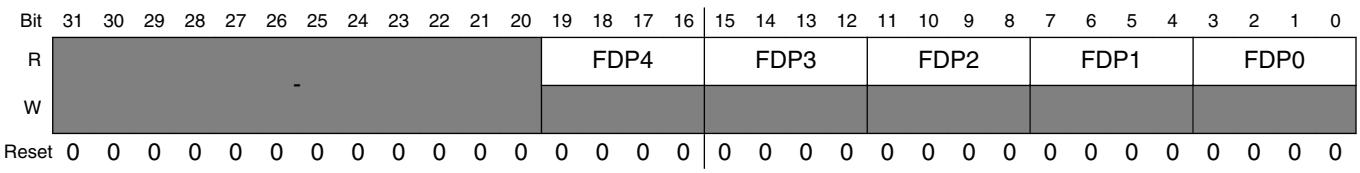
M4IF_I_ULAR field descriptions

Field	Description
31–21 -	Reserved.
20–18 IL5	int1 arbitration Level 5 - These bits configure the arbitration mode of level 5 arbitration unit in the int1 arbitration. The arbitration modes configuration options are shown in Table 49-3. Default value of IL5 is 0x0100 - encoding 4.
17–15 IL4M47	int1 arbitration Level 4 (m47) - These bits configure the arbitration mode of level 4(m47) arbitration unit in the int1 arbitration. The arbitration modes configuration options are shown in Table 49-3. Default value of IL4M47 is 0x100 - encoding 4.
14–12 IL4M03	int1 arbitration Level 4 (m03) - These bits configure the arbitration mode of level 4(m03) arbitration unit in the int1 arbitration. The arbitration modes configuration options are shown in Table 49-3. Default value of IL4M03 is 0x100 - encoding 4.
11–9 IL3M67	int1 arbitration Level 3 (m67) - These bits configure the arbitration mode of level 3(m67) arbitration unit in the int1 arbitration. The arbitration modes configuration options are shown in Table 49-3. Default value of IL3M67 is 0x100 - encoding 4.
8–6 IL3M45	int1 arbitration Level 3 (m45) - These bits configure the arbitration mode of level 3(m45) arbitration unit in the int1 arbitration. The arbitration modes configuration options are shown in Table 49-3. Default value of IL3M45 is 0x100 - encoding 4.
5–3 IL3M23	int1 arbitration Level 3 (m23) - These bits configure the arbitration mode of level 3(m23) arbitration unit in the int1 arbitration. The arbitration modes configuration options are shown in Table 49-3. Default value of IL3M23 is 0x100 - encoding 4.
2–0 IL3M01	int1 arbitration Level 3 (m01) - These bits configure the arbitration mode of level 3(m01) arbitration unit in the int1 arbitration. The arbitration modes configuration options are shown in Table 49-3. Default value of IL3M01 is 0x100 - encoding 4.

49.3.33 Fast_Dynamic_Priority_Status Register (M4IF_FDPSR)

The M4IF_FDPS register is a read-only register. This register reflects the dynamic priority value of 5 requests in the fast arbitration, as configured in the M4IF_FDPC register.

Address: M4IF_FDPSR is 63FD_8000h base + BCh offset = 63FD_80BCh



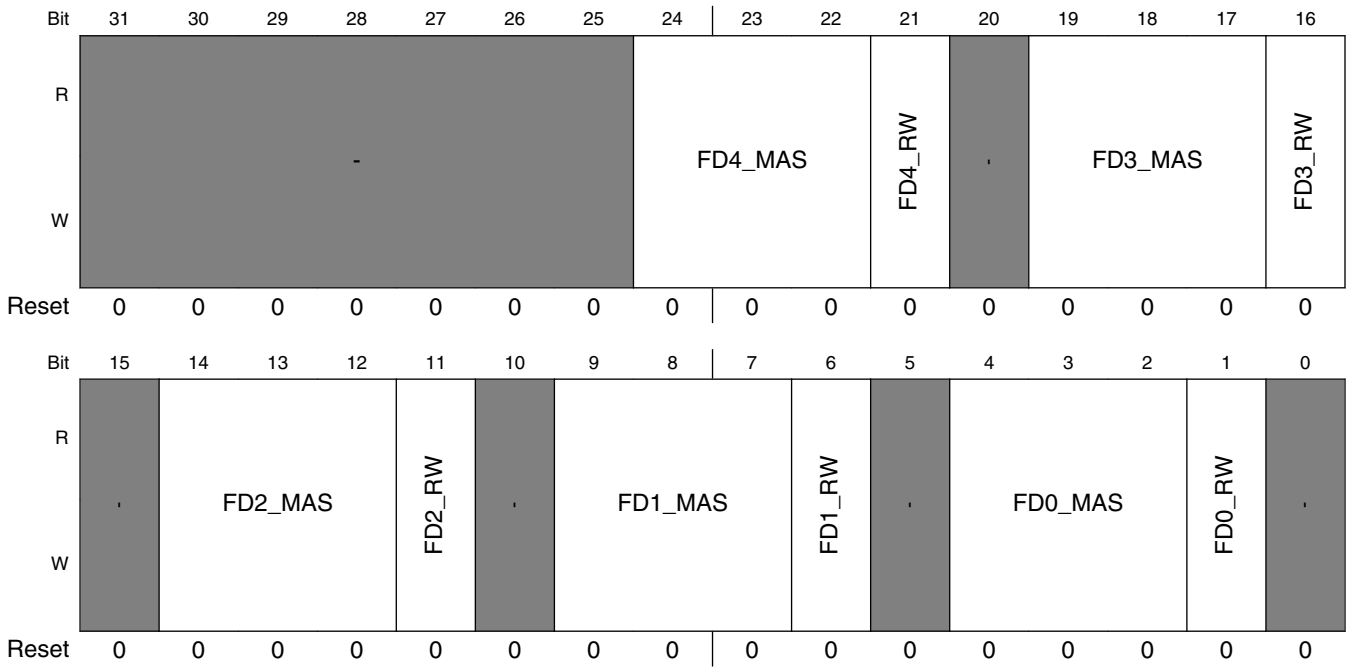
M4IF_FDPSR field descriptions

Field	Description
31–20 -	Reserved.
19–16 FDP4	Fast Dynamic Priority4 - these bits reflect the dynamic priority value of a request as configured in bits [24:20] of the FDPC register.
15–12 FDP3	Fast Dynamic Priority3 - these bits reflect the dynamic priority value of a request as configured in bits [19:15] of the FDPC register.
11–8 FDP2	Fast Dynamic Priority2 - these bits reflect the dynamic priority value of a request as configured in bits [14:10] of the FDPC register.
7–4 FDP1	Fast Dynamic Priority1 - these bits reflect the dynamic priority value of a request as configured in bits [9:5] of the FDPC register.
3–0 FDP0	Fast Dynamic Priority0 - these bits reflect the dynamic priority value of a request as configured in bits [4:0] of the FDPC register.

49.3.34 Fast_Dynamic_Priority_Control Register (M4IF_FDPCR)

The M4IF_FDPC register is a read-write register. This register is used to configure which 5 requests' dynamic priorities will be reflected in the M4IF_FDPS register.

Address: M4IF_FDPCR is 63FD_8000h base + C0h offset = 63FD_80C0h



M4IF_FDPCR field descriptions

Field	Description
31–25 -	Reserved.
24–22 FD4_MAS	Fast Dynamic 4 master - These bits indicate which master is to be reflected in the FDPS register, for dynamic priority 4.
21 FD4_RW	Fast Dynamic 4 read/write - This bit indicates whether a read or a write request is to be reflected in the FDPS register, for dynamic priority 4. 0 write 1 read
20 -	Reserved.
19–17 FD3_MAS	Fast Dynamic 3 master - These bits indicate which master is to be reflected in the FDPS register, for dynamic priority 3.
16 FD3_RW	Fast Dynamic 3 read/write - This bit indicates whether a read or a write request is to be reflected in the M4IF_FDPS register, for dynamic priority 3. 0 write 1 read

Table continues on the next page...

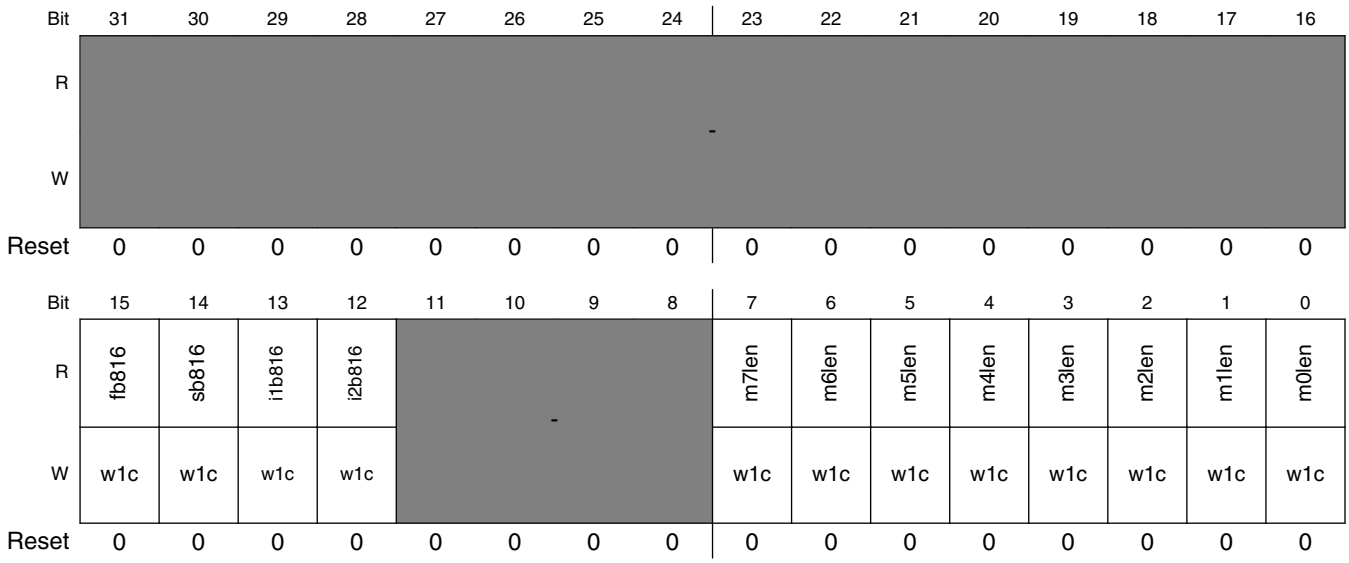
M4IF_FDPCR field descriptions (continued)

Field	Description
15 -	Reserved.
14–12 FD2_MAS	Fast Dynamic 2 master - These bits indicate which master is to be reflected in the M4IF_FDPS register, for dynamic priority 2.
11 FD2_RW	Fast Dynamic 2 read/write - This bit indicates whether a read or a write request is to be reflected in the M4IF_FDPS register, for dynamic priority 2. 0 write 1 read
10 -	Reserved.
9–7 FD1_MAS	Fast Dynamic 1 master - These bits indicate which master is to be reflected in the M4IF_FDPS register, for dynamic priority 1.
6 FD1_RW	Fast Dynamic 1 read/write - This bit indicates whether a read or a write request is to be reflected in the M4IF_FDPS register, for dynamic priority 1. 0 write 1 read
5 -	Reserved.
4–2 FD0_MAS	Fast Dynamic 0 master - These bits indicate which master is to be reflected in the M4IF_FDPS register, for dynamic priority 0.
1 FD0_RW	Fast Dynamic 0 read/write - This bit indicates whether a read or a write request is to be reflected in the M4IF_FDPS register, for dynamic priority 0. 0 write 1 read
0 -	Reserved.

49.3.35 Master Len Interrupt (M4IF_MLI)

Master Len Interrupt is a status register that indicates whether a "LEN>8" interrupt had occurred.

Address: M4IF_MLI is 63FD_8000h base + C4h offset = 63FD_80C4h



M4IF_MLI field descriptions

Field	Description
31–16 -	Reserved
15 fb816	A burst of 16 bit or 8 bit was handled in the fast arbitration. This bit can be cleared by writing 1 to it. 0 no burst occurred 1 burst occurred
14 sb816	A burst of 16 bit or 8 bit was handled in the slow arbitration. This bit can be cleared by writing 1 to it. 0 no burst occurred 1 burst occurred
13 i1b816	A burst of 16 bit or 8 bit was handled in the int1 arbitration. This bit can be cleared by writing 1 to it. 0 no burst occurred 1 burst occurred
12 i2b816	A burst of 16 bit or 8 bit was handled in the int2 arbitration. This bit can be cleared by writing 1 to it. 0 no burst occurred 1 burst occurred
11–8 -	Reserved

Table continues on the next page...

M4IF_MLI field descriptions (continued)

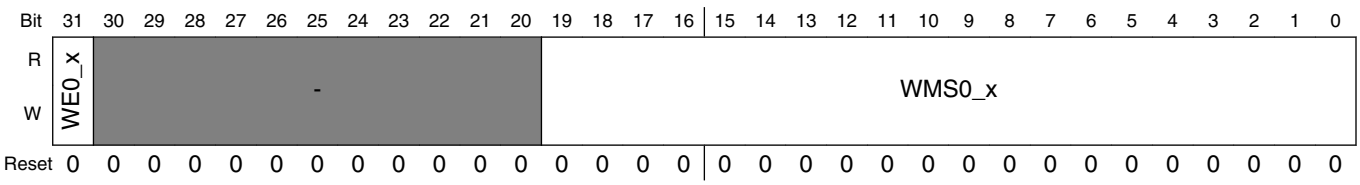
Field	Description
7 m7len	Master 7 burst length value bigger then 8 status. This bit indicates if a transaction with a burst length bigger then 8 has been sent. In such case, a violation would occurred. m0len is cleared by writing one to the bit. 0 violation did not occur. 1 violation had occurred.
6 m6len	Master 6 burst length value bigger then 8 status. This bit indicates if a transaction with a burst length bigger then 8 has been sent. In such case, a violation would occurred. m0len is cleared by writing one to the bit. 0 violation did not occur. 1 violation had occurred.
5 m5len	Master 5 burst length value bigger then 8 status. This bit indicates if a transaction with a burst length bigger then 8 has been sent. In such case, a violation would occurred. m0len is cleared by writing one to the bit. 0 violation did not occur. 1 violation had occurred.
4 m4len	Master 4 burst length value bigger then 8 status. This bit indicates if a transaction with a burst length bigger then 8 has been sent. In such case, a violation would occurred. m0len is cleared by writing one to the bit. 0 violation did not occur. 1 violation had occurred.
3 m3len	Master 3 burst length value bigger then 8 status. This bit indicates if a transaction with a burst length bigger then 8 has been sent. In such case, a violation would occurred. m0len is cleared by writing one to the bit. 0 violation did not occur. 1 violation had occurred.
2 m2len	Master 2 burst length value bigger then 8 status. This bit indicates if a transaction with a burst length bigger then 8 has been sent. In such case, a violation would occurred. m0len is cleared by writing one to the bit. 0 violation did not occur. 1 violation had occurred.
1 m1len	Master 1 burst length value bigger then 8 status. This bit indicates if a transaction with a burst length bigger then 8 has been sent. In such case, a violation would occurred. m0len is cleared by writing one to the bit. 0 violation did not occur. 1 violation had occurred.
0 m0len	Master 0 burst length value bigger then 8 status. This bit indicates if a transaction with a burst length bigger then 8 has been sent. In such case, a violation would occurred. m0len is cleared by writing one to the bit. 0 violation did not occur. 1 violation had occurred.

49.3.36 Watermark Start ADDR_0 Register n (M4IF_WMSA0_*n*)

The M4IF_WMSA0 register defines the watermark start address of space area #0. Watermark space #0 corresponds in general to AP privilege master which is defined by SoC AP via at the EXTMC boundary. AP privilege master is the only master that can configure/read this register.

Addresses: M4IF_WMSA0_6 is 63FD_8000h base + ECh offset = 63FD_80ECh

M4IF_WMSA0_7 is 63FD_8000h base + F0h offset = 63FD_80F0h



M4IF_WMSA0_*n* field descriptions

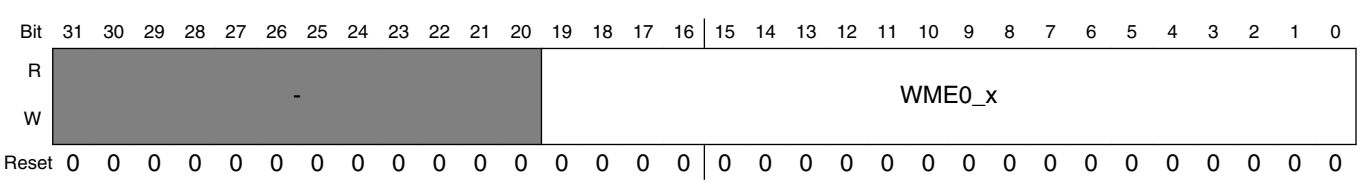
Field	Description
31 WE0_x	WaterMark Region Enable. This bit enables the WaterMark mechanism of space #0 Region X. 0 WaterMark Region X is disabled. 1 WaterMark Region X is enabled.
30–20 -	Reserved.
19–0 WMS0_x	Watermark Start Addr space #0 - As described in the watermark section in this Chapter, the watermark start address can be configured in resolution of 4KByte only. Therefore, WMSA0 register defines the upper 20bits of the start address, bits [31:12]. The lower 12bits that are not configurable via the register are zero always, 12'b0000_0000_0000 (0x000). ESDCTL protected regions must not cross 256MByte boundary.

49.3.37 Watermark End ADDR_0 Register (M4IF_WMEA0_*n*)

The M4IF_WMEA0 register defines the watermark end address of space area #0. Watermark space #0 corresponds in general to AP privilege master which is defined by SoC AP via at the EXTMC boundary. AP privilege master is the only master that can configure/read this register.

Addresses: M4IF_WMEA0_6 is 63FD_8000h base + 10Ch offset = 63FD_810Ch

M4IF_WMEA0_7 is 63FD_8000h base + 110h offset = 63FD_8110h



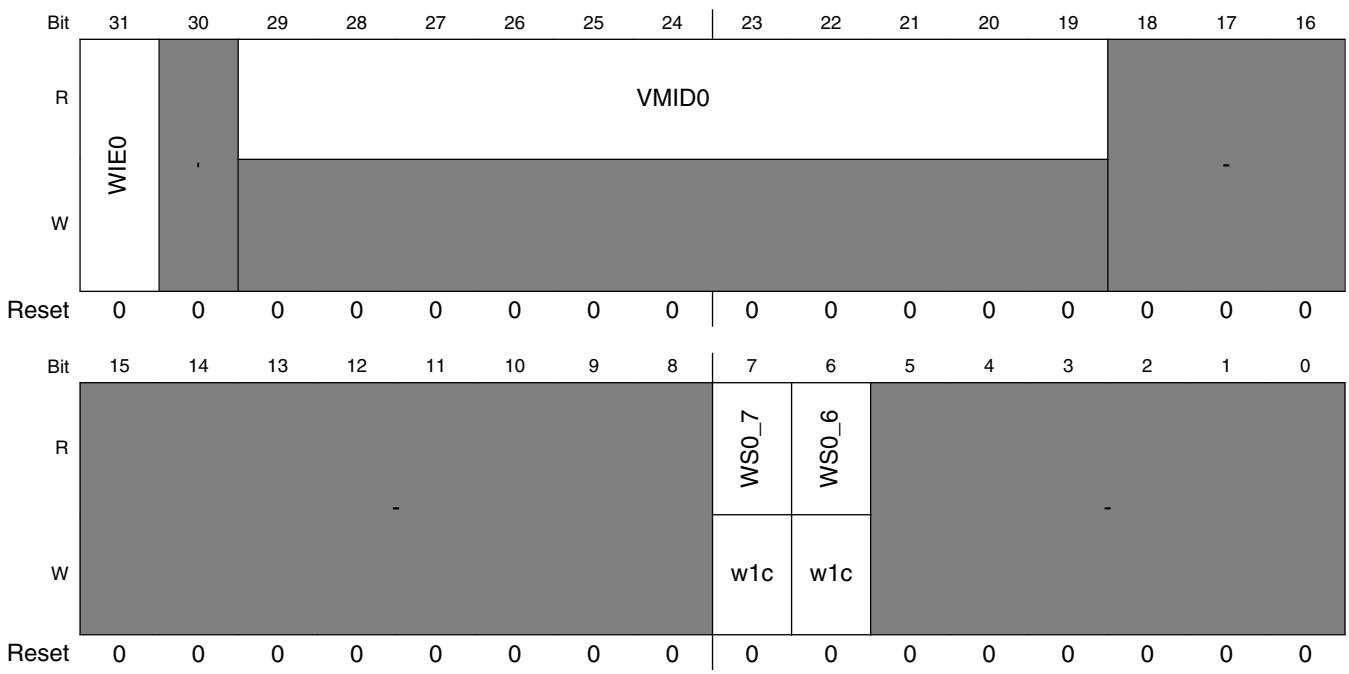
M4IF_WMEA0_*n* field descriptions

Field	Description
31–20 -	Reserved.
19–0 WME0_x	Watermark end Addr space #0 - As described in the watermark section in this Chapter, the watermark end address can be configured in resolution of 4KByte only. Therefore, WMEA0 register defines the upper 20bits of the end address, bits [31:12]. The lower 12bits that are not configurable via the register are one always, 12'b1111_1111_1111 (0xFFF). ESDCTL protected regions must not cross 256MByte boundary.

49.3.38 Watermark Interrupt and Status 0 Register (M4IF_WMIS0)

The M4IF_WMIS0 register defines the watermark interrupt enable of watermark space #0. The register enables the AP interrupt and the clear bits for each watermark space #0 region. AP privilege master is the only master that can configure/read this register.

Address: M4IF_WMIS0 is 63FD_8000h base + 114h offset = 63FD_8114h



M4IF_WMIS0 field descriptions

Field	Description
31 WIE0	WaterMark Enable. This bit enables the WaterMark mechanism of space #0 including the interrupt generation. For example, if WME is set and a WaterMark violation is detected a WaterMark interrupt is generated. 0 WaterMark is disabled. 1 WaterMark is enabled.

Table continues on the next page...

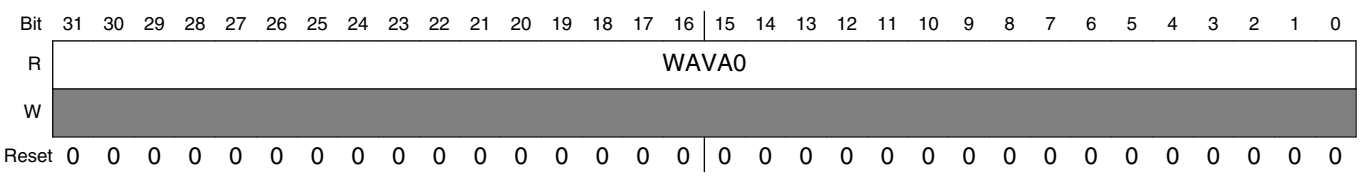
M4IF_WMIS0 field descriptions (continued)

Field	Description
30 -	Reserved.
29–19 VMID0	Violation Master ID - these bits contain the Master [29:27] Master ID [26:23] and AXI ID [22:19] of the master which tried to access a watermark area in the fast channel with WM[0] in low state.
18–8 -	Reserved.
7 WS0_7	WaterMark Status0_7. This bit indicates if WaterMark violation had occurred in the WaterMark space #0 memory region of DDR CS1 as defined in WMSA0_7 and WMEA0_7. WS0_7 is cleared by writing a one to the bit. 0 DDR/MDDR CSD1 WaterMark violation did not occur. 1 DDR CSD1 WaterMark violation had occurred.
6 WS0_6	WaterMark Status0_6. This bit indicates if WaterMark violation had occurred in the WaterMark space #0 memory region of DDR CS0 as defined in WMSA0_6 and WMEA0_6. WS0_6 is cleared by writing a one to the bit. 0 DDR CSD0 WaterMark violation did not occur. 1 DDR CSD0 WaterMark violation had occurred.
5–0 -	Reserved.

49.3.39 Watermark Violation Address 0 Register (M4IF_WMVA0)

The M4IF_WMVA0 register contains the address of the access that violated the AP watermark space. AP privileged master is the only master that can read this register.

Address: M4IF_WMVA0 is 63FD_8000h base + 118h offset = 63FD_8118h



M4IF_WMVA0 field descriptions

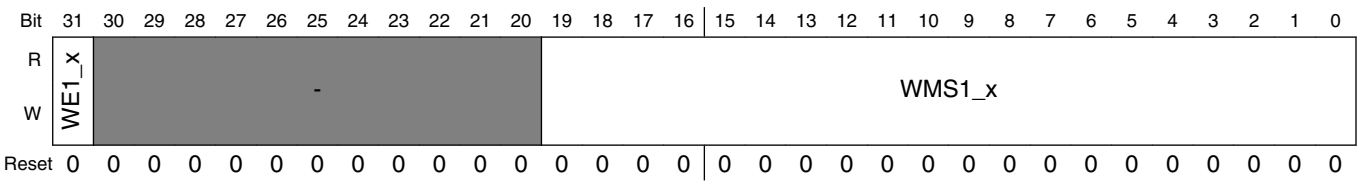
Field	Description
31–0 WAVA0	WaterMark Access violation address - These bits reflect the address of the access that caused the watermark space #0 violation.

49.3.40 Watermark Start ADDR_1 Register n (M4IF_WMSA1_*n*)

The M4IF_WMSA1 register defines the watermark start address of space area #1. Watermark space #1 corresponds in general to BP privilege master which is defined by SoC BP via at the EXTMC boundary. BP privilege master is the only master that can configure/read this register.

Addresses: M4IF_WMSA1_6 is 63FD_8000h base + 138h offset = 63FD_8138h

M4IF_WMSA1_7 is 63FD_8000h base + 13Ch offset = 63FD_813Ch



M4IF_WMSA1_*n* field descriptions

Field	Description
31 WE1_x	WaterMark Region Enable. This bit enables the WaterMark mechanism of space #1 Region X. 0 WaterMark Region X is disabled. 1 WaterMark Region X is enabled.
30–20 -	Reserved.
19–0 WMS1_x	Watermark Start Addr space #1 - As described in the watermark section in this Chapter, the watermark start address can be configured in resolution of 4KByte only. Therefore, M4IF_WMSAO register defines the upper 20bits of the start address, bits [31:12]. The lower 12bits that are not configurable via this register are zero always, 12'b0000_0000_0000 (0x000). ESDCTL protected regions must not cross 256MByte boundary.

49.3.41 Watermark End ADDR_1 Register (M4IF_WEAR1_*n*)

The M4IF_WMEA1 register defines the watermark end address of space area #1. Watermark space #1 corresponds in general to BP privilege master which is defined by SoC BP via at the EXTMC boundary. BP privilege master is the only master that can configure/read this register.

The following table maps watermark registers to their corresponding CS in EXTMC.

Table 49-55. Watermark register mapping

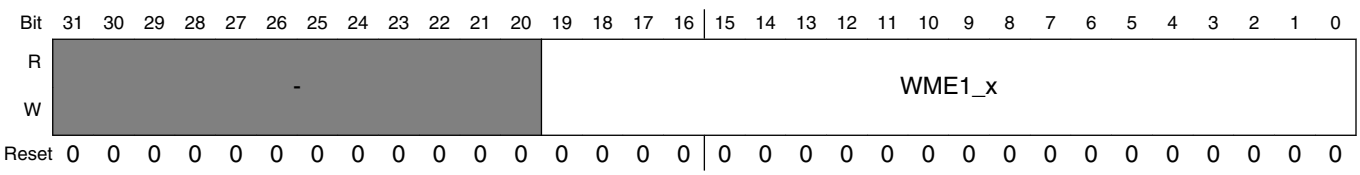
Watermark registers	CS #	Block name
M4IF_WMSA0_6 M4IF_WMEA0_6 M4IF_WMSA1_6 M4IF_WMEA1_6	CSD0	ESDCTL **NOTE:** ESDCTL protected regions must not cross 256MByte boundary.
M4IF_WMSA0_7 M4IF_WMEA0_7 M4IF_WMSA1_7 M4IF_WMEA1_7	CSD1	ESDCTL **NOTE:** ESDCTL protected regions must not cross 256MByte boundary.
Watermark is not supported		NFC
Watermark is not supported		Internal Memory

NOTE

If a watermark register is configured to addresses which are not included in the CS it belongs to, unknown behavior will occur.

Addresses: M4IF_WEAR1_6 is 63FD_8000h base + 158h offset = 63FD_8158h

M4IF_WEAR1_7 is 63FD_8000h base + 15Ch offset = 63FD_815Ch



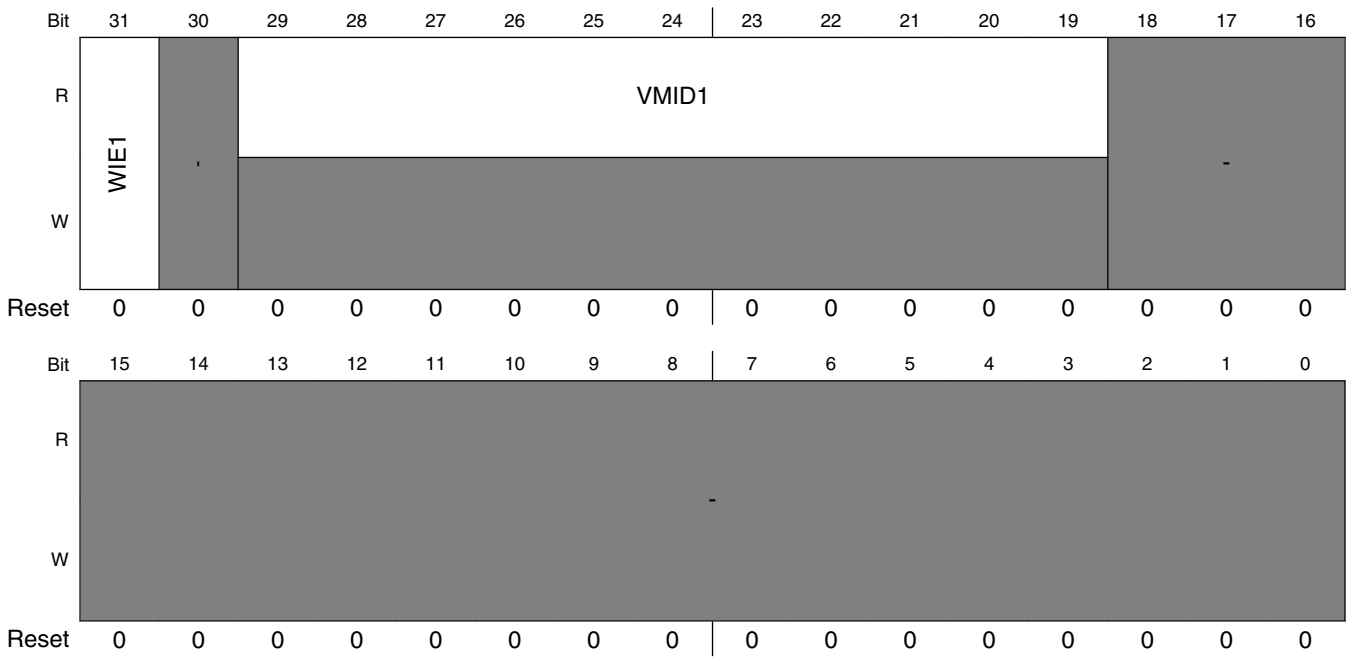
M4IF_WEAR1_*n* field descriptions

Field	Description
31–20 -	Reserved.
19–0 WME1_x	Watermark end Addr space #1 - As described in the watermark section in this Chapter, the watermark end address can be configured in resolution of 4KByte only. Therefore, M4IF_WMEA0 register defines the upper 20bits of the end address, bits [31:12]. The lower 12bits that are not configurable via the register are one always, 12'b1111_1111_1111 (0xFFF). ESDCTL protected regions must not cross 256MByte boundary.

49.3.42 Watermark Interrupt and Status 1 Register (M4IF_WISR1)

The M4IF_WMIS1 register defines the watermark interrupt enable of watermark space #1. The register enables the BP interrupt and the clear bits for each watermark space #1 region. BP privilege master is the only master that can configure/read this register.

Address: M4IF_WISR1 is 63FD_8000h base + 160h offset = 63FD_8160h



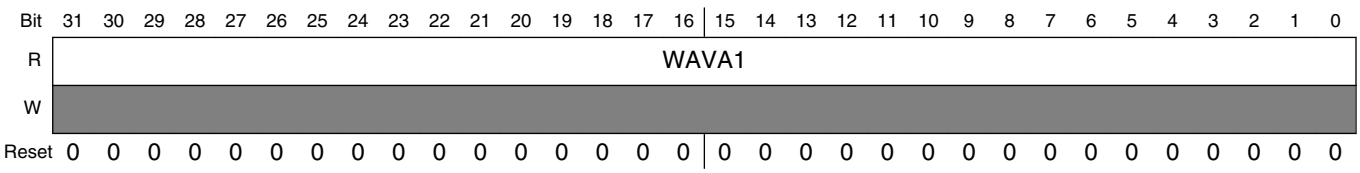
M4IF_WISR1 field descriptions

Field	Description
31 WIE1	WaterMark Enable. This bit enables the WaterMark mechanism of space #1 including the interrupt generation. For example, if WIE is set and a WaterMark violation is detected a WaterMark interrupt is generated. 0 WaterMark is disabled. 1 WaterMark is enabled.
30 -	Reserved.
29–19 VMID1	Violation Master ID - these bits contain the Master [29:27] Master ID [26:23] and AXI ID [22:19] of the master which tried to access a watermark area in the fast channel with WM[1] in low state.
18–0 -	Reserved.

49.3.43 Watermark Violation Address 1 Register (M4IF_WMVA1)

The M4IF_WMVA0 register contains the address of the access that violated the watermark space BP. Corresponding to watermark interrupt and status #1 register. BP privilege master is the only master that can read this register.

Address: M4IF_WMVA1 is 63FD_8000h base + 164h offset = 63FD_8164h



M4IF_WMVA1 field descriptions

Field	Description
31–0 WAVA1	WaterMark Access violation address - These bits reflect the address of the access that caused the watermark space #1 violation.

Chapter 50
Media Local Bus (MediaLB) Block (MLB)

50.1 Introduction

A block diagram of the MLB can be found in Figure 50-1.

50.1.1 Overview

The Media Local Bus (MLB) block implements the Physical Layer and Link Layer of the MediaLB specification, interfacing to an MLB controller. The MLB implements the 3-pin MLB mode and can run at speeds up to 1024Fs. It does not implement MLB controller functionality.

All MLB devices support a set of physical channels for sending data over the MLB. Each physical channel is 4 bytes in length (quadlet) and grouped into logical channels with one or more physical channels allocated to each logical channel. These logical channels can be in any combination of channel type (synchronous, asynchronous, control, or isochronous) and direction (transmit or receive).

The MLB provides support for up to 16 logical channels and up to 31 physical channels with a maximum of 124 bytes of data per frame. Each logical channel is referenced using an unique channel address and represents a unidirectional data path between the MLB device transmitting the data and the MLB device(s) receiving the data.

The MLB controller generates a unique frame sync pattern once per MOST network frame. This pattern defines the frame and channel boundaries of the signal information and data lines.

The MLB Controller initiates all communication over the MLB by sending out the logical channel address on the signal information line for each physical channel. This logical address indicates to the appropriate MLB device that it can transmit data for that logical channel during the next physical channel slot. One quadlet later, the transmitting MLB device sends out a MLB command byte on the signal information line and the

corresponding data on the data line. All other MLB devices (including the controller) have already compared the logical channel address with their internal table of addresses to determine if they are the intended recipient of the data on this logical channel.

The receiving MLB device responds with a receive status response on the signal information line one byte after the transmitting device sends the MLB command byte. Note that synchronous data transmissions (which are the only data formats to support multiple receivers) are not acknowledged, but asynchronous, control and isochronous data transmissions are acknowledged.

For more information about the MediaLB protocol, please refer to the MediaLB Specification.

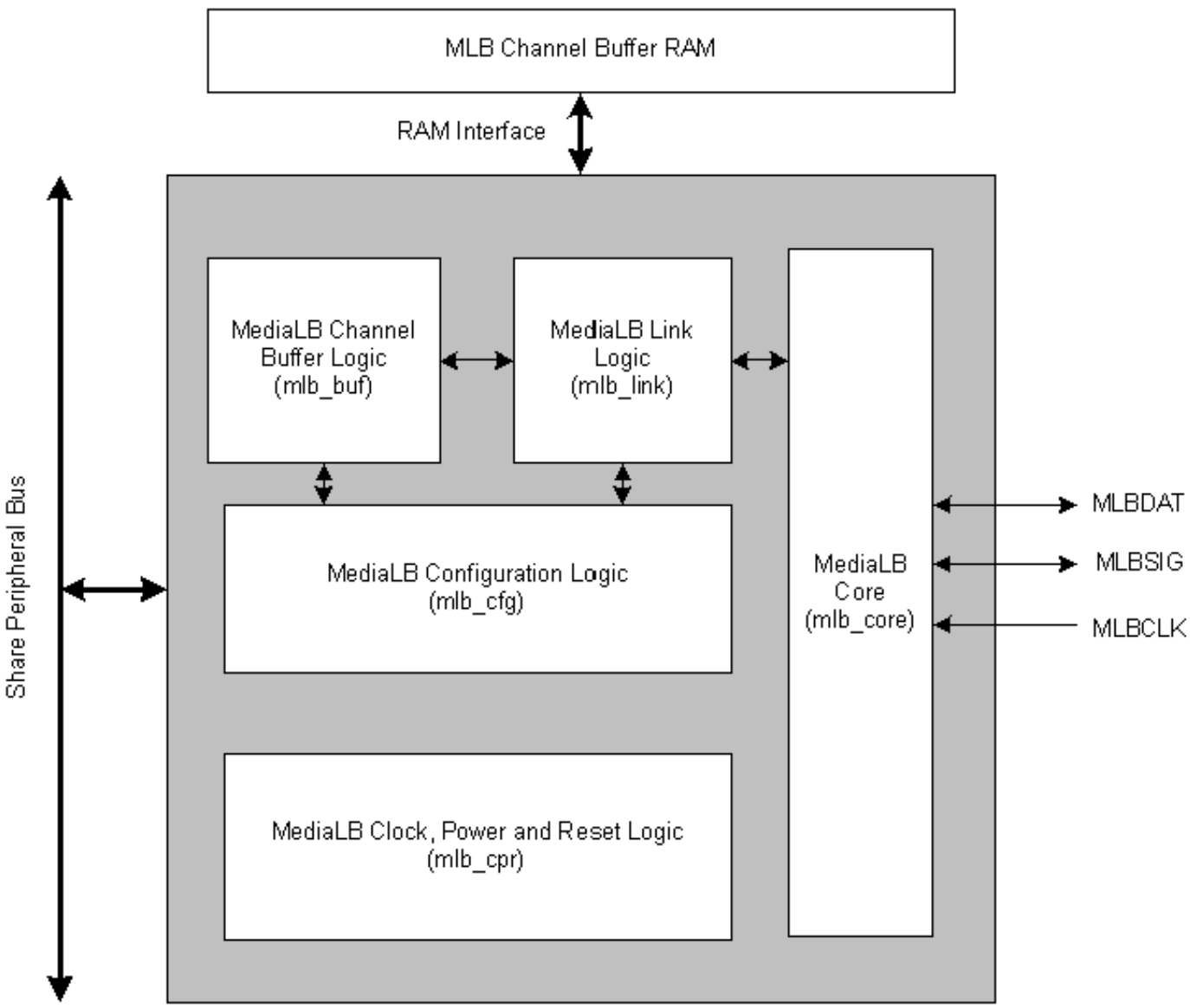


Figure 50-1. Block Diagram

50.1.2 Features

- Support for up to 16 logical channels and 31 physical channels running at a maximum speed of 1024Fs
- Transmission of commands and data, and reception of receive status when functioning as the transmitting device associated with a logical channel address
- Reception of commands and data, and transmission as receive status responses when functioning as the receiving device associated with a logical channel address
- MLB lock detection
- System channel command handling
- Local buffer memory of 2k quadlets shared between all logical channels

50.1.3 Logic Blocks

The MediaLB design is split into five major components:

MediaLB Core- The MediaLB Core implements the physical layer of the MediaLB interface. This physical layer performs serial-to-parallel and parallel-to-serial data transformations and MLB frame synchronization.

MediaLB CPR Logic- The MLB Clocks, Power, and Reset (CPR) Logic implements clock and reset control.

MediaLB Link Logic- The MediaLB Link Logic implements the link layer functionality of the MediaLB interface, including:

- checking of synchronous, asynchronous, control, and isochronous channel protocol,
- handling of both RX and TX initiated breaks,
- generating RX responses to the MediaLB Core,
- generating TX commands for the MediaLB Core,
- processing of and responding to the system channel commands,
- detection of MLB bus lock/unlock, and
- recognition and pipe-lining of logical Channel Addresses.

MediaLB Configuration Logic- The MediaLB Configuration Logic implements the memory space for the Configuration Control Registers and Channel Configuration Registers. These configuration and control registers are used to define various parameters and control the operation of the MediaLB Device.

MediaLB Channel Buffer Logic- The function of the MediaLB Channel Buffer logic block includes:

- buffering of logical channel data for bus latency issues,
- multiplexing of logical channel data in Big- and Little-Endian mode, and
- implementation of hardware loop-back mode between logical channels.

50.1.4 Modes of Operation

The MediaLB uses DMA mode to transfer data between hardware channels and system memory. The byte order in which data is transferred is determined by enabling either Big-Endian or Little-Endian mode through the DCCR.MLE bit.

The MediaLB also provides customers with a single test mode, called Loop-Back Test Mode. This mode provides basic testing capabilities for the MLB pads, physical layer, link layer, channel protocol, and local channel buffer, by enabling a single RX channel and a single TX channel. The RX channel is used to transfer data directly to the enabled TX channel.

50.2 External Signal Description

The MediaLB has three external signals which are summarized in Table 50-1.

Table 50-1. Signal Properties

Name	Function	I/O	Reset	Pull
MLBCLK	MLB Clock	I	In	Down
MLBDAT	MLB Data	I/O	In	Down
$\overline{\text{MLBSIG}}$	MLB Signal	I/O	In	Down

50.2.1 Detailed Signal Descriptions

Table 50-2. MediaLB Interface

Signal	I/O	Description	
MLBCLK	I	MLB Clock	
		State Meaning	Asserted/Negated-Supports a 256Fs, 512Fs or 1024Fs clock input from the MediaLB controller.
		Timing	Assertion/Negation-Supports maximum frequency of 49.2 MHz for a 48 kHz sample rate.

Table continues on the next page...

Table 50-2. MediaLB Interface (continued)

Signal	I/O	Description	
MLBDAT	I/O	MLB Data	
		State Meaning	Asserted/Negated-MediaLB data for serial receive/transmit channel data.
		Timing	Assertion/Negation-Registered on the falling edge of MLBCLK.
MLBSIG	I/O	MLB Signal	
		State Meaning	Asserted/Negated-MediaLB signal information for serial transmit channel commands, serial receive channel responses, and logical channel address information.
		Timing	Assertion/Negation-Registered on the falling edge of MLBCLK.

50.3 Programmable Registers

MLB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FE_4000	Device Control Configuration Register (MLB_DCCR)	32	R/W	0000_0000h	50.3.1/ 3548
63FE_4004	System Status Configuration Register (MLB_SSCR)	32	R/W	0000_0000h	50.3.2/ 3550
63FE_4008	System Data Configuration Register (MLB_SDCR)	32	R	0000_0000h	50.3.3/ 3552
63FE_400C	System Mask Configuration Register (MLB_SMCR)	32	R/W	0000_0060h	50.3.4/ 3552
63FE_401C	Version Control Configuration Register (MLB_VCCR)	32	R	0200_0202h	50.3.5/ 3553
63FE_4020	Synchronous Base Address Configuration Register (MLB_SBCR)	32	R	0000_0000h	50.3.6/ 3554
63FE_4024	Asynchronous Base Address Configuration Register (MLB_ABCR)	32	R	0000_0000h	50.3.7/ 3554
63FE_4028	Control Base Address Configuration Register (MLB_CBCR)	32	R	0000_0000h	50.3.8/ 3555
63FE_402C	Isochronous Base Address Configuration Register (MLB_IBCR)	32	R	0000_0000h	50.3.9/ 3555
63FE_4030	Channel Interrupt Configuration Register (MLB_CICR)	32	R	0000_0000h	50.3.10/ 3556

Table continues on the next page...

MLB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FE_4040	Channel n Entry Configuration Register (MLB_CECR0)	32	R/W	0000_0000h	50.3.11/ 3557
63FE_4044	Channel n Status Configuration Register (MLB_CSCR0)	32	R/W	8000_0000h	50.3.12/ 3560
63FE_4048	Channel n Current Buffer Configuration Register (MLB_CCBCR0)	32	R/W	0000_0000h	50.3.13/ 3563
63FE_404C	Channel n Next Buffer Configuration Register (MLB_CNBCR0)	32	R/W	0000_0000h	50.3.14/ 3564
63FE_4050	Channel n Entry Configuration Register (MLB_CECR1)	32	R/W	0000_0000h	50.3.11/ 3557
63FE_4054	Channel n Status Configuration Register (MLB_CSCR1)	32	R/W	8000_0000h	50.3.12/ 3560
63FE_4058	Channel n Current Buffer Configuration Register (MLB_CCBCR1)	32	R/W	0000_0000h	50.3.13/ 3563
63FE_405C	Channel n Next Buffer Configuration Register (MLB_CNBCR1)	32	R/W	0000_0000h	50.3.14/ 3564
63FE_4060	Channel n Entry Configuration Register (MLB_CECR2)	32	R/W	0000_0000h	50.3.11/ 3557
63FE_4064	Channel n Status Configuration Register (MLB_CSCR2)	32	R/W	8000_0000h	50.3.12/ 3560
63FE_4068	Channel n Current Buffer Configuration Register (MLB_CCBCR2)	32	R/W	0000_0000h	50.3.13/ 3563
63FE_406C	Channel n Next Buffer Configuration Register (MLB_CNBCR2)	32	R/W	0000_0000h	50.3.14/ 3564
63FE_4070	Channel n Entry Configuration Register (MLB_CECR3)	32	R/W	0000_0000h	50.3.11/ 3557
63FE_4074	Channel n Status Configuration Register (MLB_CSCR3)	32	R/W	8000_0000h	50.3.12/ 3560
63FE_4078	Channel n Current Buffer Configuration Register (MLB_CCBCR3)	32	R/W	0000_0000h	50.3.13/ 3563
63FE_407C	Channel n Next Buffer Configuration Register (MLB_CNBCR3)	32	R/W	0000_0000h	50.3.14/ 3564
63FE_4080	Channel n Entry Configuration Register (MLB_CECR4)	32	R/W	0000_0000h	50.3.11/ 3557
63FE_4084	Channel n Status Configuration Register (MLB_CSCR4)	32	R/W	8000_0000h	50.3.12/ 3560
63FE_4088	Channel n Current Buffer Configuration Register (MLB_CCBCR4)	32	R/W	0000_0000h	50.3.13/ 3563
63FE_408C	Channel n Next Buffer Configuration Register (MLB_CNBCR4)	32	R/W	0000_0000h	50.3.14/ 3564

Table continues on the next page...

MLB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FE_4090	Channel n Entry Configuration Register (MLB_CECR5)	32	R/W	0000_0000h	50.3.11/ 3557
63FE_4094	Channel n Status Configuration Register (MLB_CSCR5)	32	R/W	8000_0000h	50.3.12/ 3560
63FE_4098	Channel n Current Buffer Configuration Register (MLB_CCBCR5)	32	R/W	0000_0000h	50.3.13/ 3563
63FE_409C	Channel n Next Buffer Configuration Register (MLB_CNBCR5)	32	R/W	0000_0000h	50.3.14/ 3564
63FE_40A0	Channel n Entry Configuration Register (MLB_CECR6)	32	R/W	0000_0000h	50.3.11/ 3557
63FE_40A4	Channel n Status Configuration Register (MLB_CSCR6)	32	R/W	8000_0000h	50.3.12/ 3560
63FE_40A8	Channel n Current Buffer Configuration Register (MLB_CCBCR6)	32	R/W	0000_0000h	50.3.13/ 3563
63FE_40AC	Channel n Next Buffer Configuration Register (MLB_CNBCR6)	32	R/W	0000_0000h	50.3.14/ 3564
63FE_40B0	Channel n Entry Configuration Register (MLB_CECR7)	32	R/W	0000_0000h	50.3.11/ 3557
63FE_40B4	Channel n Status Configuration Register (MLB_CSCR7)	32	R/W	8000_0000h	50.3.12/ 3560
63FE_40B8	Channel n Current Buffer Configuration Register (MLB_CCBCR7)	32	R/W	0000_0000h	50.3.13/ 3563
63FE_40BC	Channel n Next Buffer Configuration Register (MLB_CNBCR7)	32	R/W	0000_0000h	50.3.14/ 3564
63FE_40C0	Channel n Entry Configuration Register (MLB_CECR8)	32	R/W	0000_0000h	50.3.11/ 3557
63FE_40C4	Channel n Status Configuration Register (MLB_CSCR8)	32	R/W	8000_0000h	50.3.12/ 3560
63FE_40C8	Channel n Current Buffer Configuration Register (MLB_CCBCR8)	32	R/W	0000_0000h	50.3.13/ 3563
63FE_40CC	Channel n Next Buffer Configuration Register (MLB_CNBCR8)	32	R/W	0000_0000h	50.3.14/ 3564
63FE_40D0	Channel n Entry Configuration Register (MLB_CECR9)	32	R/W	0000_0000h	50.3.11/ 3557
63FE_40D4	Channel n Status Configuration Register (MLB_CSCR9)	32	R/W	8000_0000h	50.3.12/ 3560
63FE_40D8	Channel n Current Buffer Configuration Register (MLB_CCBCR9)	32	R/W	0000_0000h	50.3.13/ 3563
63FE_40DC	Channel n Next Buffer Configuration Register (MLB_CNBCR9)	32	R/W	0000_0000h	50.3.14/ 3564

Table continues on the next page...

MLB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FE_40E0	Channel n Entry Configuration Register (MLB_CECR10)	32	R/W	0000_0000h	50.3.11/ 3557
63FE_40E4	Channel n Status Configuration Register (MLB_CSCR10)	32	R/W	8000_0000h	50.3.12/ 3560
63FE_40E8	Channel n Current Buffer Configuration Register (MLB_CCBCR10)	32	R/W	0000_0000h	50.3.13/ 3563
63FE_40EC	Channel n Next Buffer Configuration Register (MLB_CNBCR10)	32	R/W	0000_0000h	50.3.14/ 3564
63FE_40F0	Channel n Entry Configuration Register (MLB_CECR11)	32	R/W	0000_0000h	50.3.11/ 3557
63FE_40F4	Channel n Status Configuration Register (MLB_CSCR11)	32	R/W	8000_0000h	50.3.12/ 3560
63FE_40F8	Channel n Current Buffer Configuration Register (MLB_CCBCR11)	32	R/W	0000_0000h	50.3.13/ 3563
63FE_40FC	Channel n Next Buffer Configuration Register (MLB_CNBCR11)	32	R/W	0000_0000h	50.3.14/ 3564
63FE_4100	Channel n Entry Configuration Register (MLB_CECR12)	32	R/W	0000_0000h	50.3.11/ 3557
63FE_4104	Channel n Status Configuration Register (MLB_CSCR12)	32	R/W	8000_0000h	50.3.12/ 3560
63FE_4108	Channel n Current Buffer Configuration Register (MLB_CCBCR12)	32	R/W	0000_0000h	50.3.13/ 3563
63FE_410C	Channel n Next Buffer Configuration Register (MLB_CNBCR12)	32	R/W	0000_0000h	50.3.14/ 3564
63FE_4110	Channel n Entry Configuration Register (MLB_CECR13)	32	R/W	0000_0000h	50.3.11/ 3557
63FE_4114	Channel n Status Configuration Register (MLB_CSCR13)	32	R/W	8000_0000h	50.3.12/ 3560
63FE_4118	Channel n Current Buffer Configuration Register (MLB_CCBCR13)	32	R/W	0000_0000h	50.3.13/ 3563
63FE_411C	Channel n Next Buffer Configuration Register (MLB_CNBCR13)	32	R/W	0000_0000h	50.3.14/ 3564
63FE_4120	Channel n Entry Configuration Register (MLB_CECR14)	32	R/W	0000_0000h	50.3.11/ 3557
63FE_4124	Channel n Status Configuration Register (MLB_CSCR14)	32	R/W	8000_0000h	50.3.12/ 3560
63FE_4128	Channel n Current Buffer Configuration Register (MLB_CCBCR14)	32	R/W	0000_0000h	50.3.13/ 3563
63FE_412C	Channel n Next Buffer Configuration Register (MLB_CNBCR14)	32	R/W	0000_0000h	50.3.14/ 3564

Table continues on the next page...

MLB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FE_4130	Channel n Entry Configuration Register (MLB_CECR15)	32	R/W	0000_0000h	50.3.11/ 3557
63FE_4134	Channel n Status Configuration Register (MLB_CSCR15)	32	R/W	8000_0000h	50.3.12/ 3560
63FE_4138	Channel n Current Buffer Configuration Register (MLB_CCBCR15)	32	R/W	0000_0000h	50.3.13/ 3563
63FE_413C	Channel n Next Buffer Configuration Register (MLB_CNBCR15)	32	R/W	0000_0000h	50.3.14/ 3564
63FE_4280	Local Channel n Buffer Configuration Register (MLB_LCBCR0)	32	R/W	See section	50.3.15/ 3565
63FE_4290	Local Channel n Buffer Configuration Register (MLB_LCBCR1)	32	R/W	See section	50.3.15/ 3565
63FE_42A0	Local Channel n Buffer Configuration Register (MLB_LCBCR2)	32	R/W	See section	50.3.15/ 3565
63FE_42B0	Local Channel n Buffer Configuration Register (MLB_LCBCR3)	32	R/W	See section	50.3.15/ 3565
63FE_42C0	Local Channel n Buffer Configuration Register (MLB_LCBCR4)	32	R/W	See section	50.3.15/ 3565
63FE_42D0	Local Channel n Buffer Configuration Register (MLB_LCBCR5)	32	R/W	See section	50.3.15/ 3565
63FE_42E0	Local Channel n Buffer Configuration Register (MLB_LCBCR6)	32	R/W	See section	50.3.15/ 3565
63FE_42F0	Local Channel n Buffer Configuration Register (MLB_LCBCR7)	32	R/W	See section	50.3.15/ 3565
63FE_4300	Local Channel n Buffer Configuration Register (MLB_LCBCR8)	32	R/W	See section	50.3.15/ 3565
63FE_4310	Local Channel n Buffer Configuration Register (MLB_LCBCR9)	32	R/W	See section	50.3.15/ 3565
63FE_4320	Local Channel n Buffer Configuration Register (MLB_LCBCR10)	32	R/W	See section	50.3.15/ 3565
63FE_4330	Local Channel n Buffer Configuration Register (MLB_LCBCR11)	32	R/W	See section	50.3.15/ 3565
63FE_4340	Local Channel n Buffer Configuration Register (MLB_LCBCR12)	32	R/W	See section	50.3.15/ 3565
63FE_4350	Local Channel n Buffer Configuration Register (MLB_LCBCR13)	32	R/W	See section	50.3.15/ 3565
63FE_4360	Local Channel n Buffer Configuration Register (MLB_LCBCR14)	32	R/W	See section	50.3.15/ 3565
63FE_4370	Local Channel n Buffer Configuration Register (MLB_LCBCR15)	32	R/W	See section	50.3.15/ 3565

50.3.1 Device Control Configuration Register (MLB_DCCR)

The Device Control Configuration Register (DCCR) is used to control basic features of the MediaLB, such as clock rate, lock status, enable, and reset behavior.

Address: MLB_DCCR is 63FE_4000h base + 0h offset = 63FE_4000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MDE	LBM	MCS		--	MLK	MLE	MHRE	MRS	--[-1:8]						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	--[7:0]								MDA							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_DCCR field descriptions

Field	Description
31 MDE	MediaLB Device Enable. When set, enables the MediaLB Interface based on the other bits in the register. 0 MediaLB Device Disabled 1 MediaLB Device Enabled
30 LBM	Loop-Back Mode Enable. When set, enables the loop-back testing of the MediaLB bus between logical channel N (RX) and logical channel N+1 (TX). {N=0, 2, 4, 6,...,14} 0 Loop-Back Mode Disabled Loop-Back Mode Enabled
29–28 MCS	MediaLB Clock Select. These field must be programmed by system software to reflect the MLBCLK_IN speed. 00 256Fs: supports 8 quadlets per frame 01 512Fs: supports 16 quadlets per frame 10 1024Fs: supports 32 quadlets per frame 11 Reserved
27 --	Reserved. Should be written as zero for compatibility.
26 MLK	MLB Lock. When set, indicates that the MLB Port is synchronized to the incoming MLB frame. If MLK is clear (unlocked), MLK is set after FRAMESYNC is detected at the same position for three consecutive frames. If MLK is set (locked), MLK is cleared after not receiving FRAMESYNC at the expected time for two consecutive frames. While MLK is set, FRAMESYNC patterns occurring at locations other than the expected one are ignored.
25 MLE	MLB Little Endian.This field determines how MLB quadlet based data is stored in system memory.

Table continues on the next page...

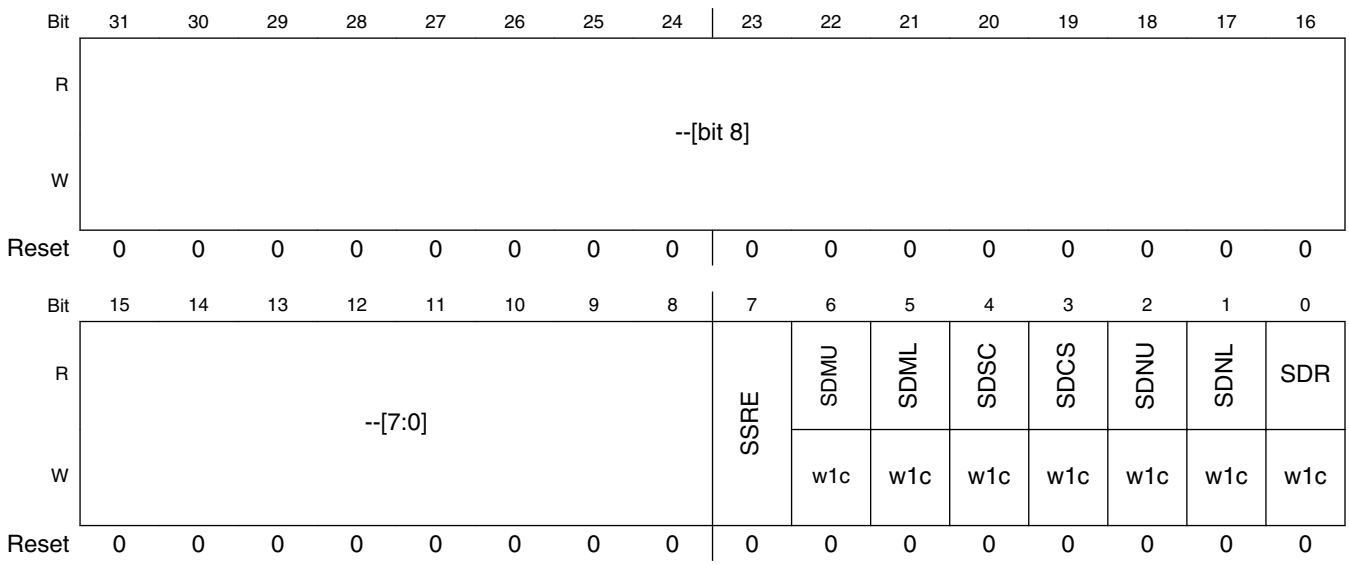
MLB_DCCR field descriptions (continued)

Field	Description
	0 Big Endian mode 1 Little Endian mode
24 MHRE	MLB Hardware Reset Enable. When set, enables hardware to automatically reset the MediaLB physical and link layer logic upon the reception of either a global (SDCR.MDS = 8'h0000) or device specific (SDCR.MDS = DA) MlbReset (FEh) System Command. 0 MediaLB device does not reset on reception of system reset command. 1 MediaLB device is reset on reception of system reset command.
23 MRS	MLB Software Reset. When set, resets the MLB physical and link layer logic. Hardware clears this bit automatically. 0 MediaLB device is not reset by software. 1 MediaLB device is reset by software.
22–8 --	Reserved. Should be written as zero for compatibility.
7–0 MDA	MLB Device Address. Determines the unique DeviceAddress (DA) for the MLB Device. DeviceAddresses are used by the system channel MlbScan command. DA[15:0] = {7'h00, MDA[8:1], 1'b0}

50.3.2 System Status Configuration Register (MLB_SSCR)

The System Status Configuration register (SSCR) allows system software to monitor and control the status of the MLB network. The register is updated once per frame by hardware during the MLB System Channel. Except for the bits associated with MLB lock and unlock (SSCR.SDMU and SSCR.SDML), the bits of the SSCR register are not valid until the MLB is locked to the MLB interface. System software must service status events before the start of the next MLB frame to prevent the current frame status from being lost.

Address: MLB_SSCR is 63FE_4000h base + 4h offset = 63FE_4004h



MLB_SSCR field descriptions

Field	Description
31–8 --	Reserved. Should be written as zero for compatibility.
7 SSRE	System Service Request Enable. System software can set this bit to indicate that this MLB Device is present and needs service. An RxStatus DeviceServiceRequest (82h) will be sent in response to a MlbScan System Command from the MLB Controller. Hardware clears this bit after the RxStatus is sent. 0 MLB device responds to System Scan Command with Device Present (80h). 1 MLB device responds to System Scan Command with Device Service Request (82h).
6 SDMU	System Detects MLB Unlock. This bit is set to indicate that the MLB Device has unlocked from the MLB frame. Detecting a MLB unlock generates a maskable system interrupt to system software. Once set, this bit is sticky until cleared by software. 0 MLB device has not unlocked from MediaLB frame. 1 MLB device has unlocked from MediaLB frame.

Table continues on the next page...

MLB_SSCR field descriptions (continued)

Field	Description
5 SDML	System Detects MLB Lock. This bit is set to indicate that the MLB Device has locked to the MLB frame. Detecting a MLB lock generates a maskable system interrupt to system software. Once set, this bit is sticky until cleared by software. 0 MLB device has not locked to MediaLB frame. 1 MLB device has locked to MediaLB frame.
4 SDSC	System Detects SubCommand. This bit is set to indicate that the MLB Device has received the MlbSubCmd (E6h) System Command. The user-defined software command is stored in the SDCR register. The decoding of this command is left up to software. Detecting MlbSubCmd generates a maskable system interrupt to system software. Once set, this bit is sticky until cleared by software. 0 MLB device has not detected a Sub-command System Command. 1 MLB device has detected a Sub-command System Command.
3 SDCS	System Detects Channel Scan. This bit is set to indicate that the MLB Device has received the MlbScan (E4h) System Command. The target DeviceAddress is stored in the SDCR register. Detecting MlbScan generates a maskable system interrupt to system software. Once set, this bit is sticky until cleared by software. 0 MLB device has not detected a System Scan Command. 1 MLB device has detected a System Scan Command.
2 SDNU	System Detects Network Unlock. This bit is set to indicate that the MLB Device has received the MOST_Unlock (E2h) System Command. Detecting MOST_Unlock generates a maskable system interrupt to system software. Once set, this bit is sticky until cleared by software. 0 MLB device has not detected an Unlock Command. 1 MLB device has detected an Unlock Command.
1 SDNL	System Detects Network Lock. This bit is set to indicate that the MLB Device has received the MOST_Lock (E0h) System Command. Detecting MOST_Lock generates a maskable system interrupt to system software. Once set, this bit is sticky until cleared by software 0 MLB device has not detected a Lock Command. 1 MLB device has detected a Lock Command.
0 SDR	System Detects Reset. This bit is set to indicate that the MLB Device has received the MlbReset (FEh) System Command. The target DeviceAddress is stored in the SDCR register. Detecting MLBReset generates a maskable system interrupt to system software. Once set, this bit is sticky until cleared by software. 0 MLB device has not detected a Reset Command. 1 MLB device has detected a Reset Command.

50.3.3 System Data Configuration Register (MLB_SDCR)

The System Data Configuration Register (SDCR) allows system software to receive control information from the MLB Controller. The register is updated once per frame by hardware during the MLB System Channel. System software must read SDCR before the start of the next MLB frame to prevent the current frame data from being lost.

Address: MLB_SDCR is 63FE_4000h base + 8h offset = 63FE_4008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSD																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_SDCR field descriptions

Field	Description
31–0 MSD	MLB System Data. This register is loaded with the data from MLBDAT during the System Channel quadlet.

50.3.4 System Mask Configuration Register (MLB_SMCR)

The System Mask Configuration register (SMCR) allows system software to mask system status interrupts.

Address: MLB_SMCR is 63FE_4000h base + Ch offset = 63FE_400Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	--																									SMMU	SMML	SMSC	SMCS	SMNU	SMNL	SMR
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

MLB_SMCR field descriptions

Field	Description
31–7 --	Reserved. Should be written as zero for compatibility.
6 SMMU	System Masks MLB Unlock. When set, this bit masks system interrupts generated when a MLB unlock is detected. At reset, MediaLB unlock events are masked (SMMU = 1) 0 MLB unlock system interrupt is enabled. 1 MLB unlock system interrupt is disabled.
5 SMML	System Masks MLB Lock. When set, this bit masks system interrupts generated when MLB lock is detected. At reset, MediaLB lock events are masked (SMML = 1).

Table continues on the next page...

MLB_SMCR field descriptions (continued)

Field	Description
	0 MLB lock system interrupt is enabled. 1 MLB lock system interrupt is disabled.
4 SMSC	System Masks SubCommand. When set, this bit masks system interrupts for the MlbSubCmd (E6h) System Command. 0 MLB SubCommand system interrupt is enabled. 1 MLB SubCommand system interrupt is disabled.
3 SMCS	System Masks Channel Scan. When set, this bit masks system interrupts for the MlbScan (E4h) System Command. 0 MLB Channel Scan system interrupt is enabled. 1 MLB Channel Scan system interrupt is disabled.
2 SMNU	System Masks Network Unlock. When set, this bit masks system interrupts for the MOST_Unlock (E2h) System Command 0 MLB Network Unlock system interrupt is enabled. 1 MLB Network Unlock system interrupt is disabled.
1 SMNL	System Masks Network Lock. When set, this bit masks system interrupts for the MOST_Lock (E0h) System Command. 0 MLB Network Lock system interrupt is enabled. 1 MLB Network Lock system interrupt is disabled.
0 SMR	System Masks Reset. When set, this bit masks system interrupts for the MlbReset (FEh) System Command. 0 MLB Reset system interrupt is enabled. 1 MLB Reset system interrupt is disabled.

50.3.5 Version Control Configuration Register (MLB_VCCR)

The Version Control Configuration Register (VCCR) allows system software to verify the version of the MediaLB.

Address: MLB_VCCR is 63FE_4000h base + 1Ch offset = 63FE_401Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UMA								UMI								MMA								MMI							
W																																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

MLB_VCCR field descriptions

Field	Description
31–24 UMA	User Major Revision Code. This field identifies the major revision of MLB Device in i.MX53. The current major revision code is 02h.

Table continues on the next page...

MLB_VCCR field descriptions (continued)

Field	Description
23–16 UMI	User Minor Revision Code. This field identifies the minor revision of MLB Device in i.MX53.The current minor revision code is 00h.
15–8 MMA	MLB Major Revision Code. This field identifies the major revision of the MLB Device. The current major revision code is 02h.
7–0 MMI	MLB Minor Revision Code. This field identifies the minor revision of the MLB Device. The current minor revision code is 02h.

50.3.6 Synchronous Base Address Configuration Register (MLB_SBCR)

The Synchronous Base Address Configuration Register (SBCR) allows system software to define the base address for synchronous RX/TX system memory buffers.

Address: MLB_SBCR is 63FE_4000h base + 20h offset = 63FE_4020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SRBA																STBA															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_SBCR field descriptions

Field	Description
31–16 SRBA	Synchronous Receive Base Address. This base address is shared by all synchronous RX channels and defines the upper 16 bits of the 32-bit AHB address for these channels.
15–0 STBA	Synchronous Transmit Base Address. This base address is shared by all synchronous TX channels and defines the upper 16 bits of the 32-bit AHB address for these channels.

50.3.7 Asynchronous Base Address Configuration Register (MLB_ABCR)

The Asynchronous Base Address Configuration Register (ABCR) allows system software to define the base address for synchronous RX/TX system memory buffers.

Address: MLB_ABCR is 63FE_4000h base + 24h offset = 63FE_4024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ARBA																ATBA															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_ABCR field descriptions

Field	Description
31–16 ARBA	Asynchronous Receive Base Address. This base address is shared by all asynchronous RX channels and defines the upper 16 bits of the 32-bit AHB address for these channels.
15–0 ATBA	Asynchronous Transmit Base Address. This base address is shared by all asynchronous TX channels and defines the upper 16 bits of the 32-bit AHB address for these channels.

50.3.8 Control Base Address Configuration Register (MLB_CBCR)

The Control Base Address Configuration Register (CBCR) allows system software to define the base address for synchronous RX/TX system memory buffers.

Address: MLB_CBCR is 63FE_4000h base + 28h offset = 63FE_4028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CRBA																CTBA															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_CBCR field descriptions

Field	Description
31–16 CRBA	Control Receive Base Address. This base address is shared by all control RX channels and defines the upper 16 bits of the 32-bit AHB address for these channels.
15–0 CTBA	Control Transmit Base Address. This base address is shared by all control TX channels and defines the upper 16 bits of the 32-bit AHB address for these channels.

50.3.9 Isochronous Base Address Configuration Register (MLB_IBCR)

The Isochronous Base Address Configuration Register (IBCR) allows system software to define the base address for synchronous RX/TX system memory buffers.

Address: MLB_IBCR is 63FE_4000h base + 2Ch offset = 63FE_402Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRBA																ITBA															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_IBCR field descriptions

Field	Description
31–16 IRBA	Isochronous Receive Base Address. This base address is shared by all Isochronous RX channels and defines the upper 16 bits of the 32-bit AHB address for these channels.
15–0 ITBA	Isochronous Transmit Base Address. This base address is shared by all Isochronous TX channels and defines the upper 16 bits of the 32-bit AHB address for these channels.

50.3.10 Channel Interrupt Configuration Register (MLB_CICR)

The Channel Interrupt Configuration Register (CICR) reflects the channel interrupt status of the individual MLB logical channels. These bits are set by hardware when a channel interrupt is generated. The channel interrupt bits are sticky and can only be reset by software.

Address: MLB_CICR is 63FE_4000h base + 30h offset = 63FE_4030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	--																CSU															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_CICR field descriptions

Field	Description
31–16 --	Reserved. Should be written as zero for compatibility.
15–0 CSU	Channel Status Update for Logical Channels 15 through 0. When set, these bits indicate that hardware has generated an interrupt for the appropriate channel. These bits are sticky and can only be cleared by a software write. Writing to the CICR register has no affect. To clear a particular bit in the CICR, software must clear all of the unmasked status bits in the corresponding CSCRn register. 0 Channel n has not generated an interrupt. 1 Channel n has generated an interrupt.

50.3.11 Channel n Entry Configuration Register (MLB_CECR*n*)

The Channel n Entry Configuration Register (CECRn) defines basic attributes about a given logical channel, such as the channel enable, channel type, channel direction, and channel address. The definition of some of the bit fields in the CECRn register vary dependant on the selected channel type.

Addresses: 63FE_4000h base + 40h offset + (16d × *n*), where *n* = 0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CE	TR	CT		Rsvd/FSE/FCE	MDS		--		MLFS		MBE	MBS	MBD	MDB	MPE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Rsvd/FSCD/IPL[7]	Rsvd/IPL[6:5]		Rsvd/FSPC/IPL[4:0]					CA							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_CECR*n* field descriptions

Field	Description
31 CE	Channel n Enable. 0 Channel n disabled (default) 1 Channel n enabled
30 TR	Channel n Transmit Select. 0 Receive 1 Transmit
29–28 CT	Channel n Type Select. 00 Synchronous 01 Isochronous 10 Asynchronous 11 Control
27 Rsvd/FSE/FCE	The meaning of this bit depends on the type of channel: Rsvd - For Asynchronous and Control Channels

Table continues on the next page...

MLB_CECR*n* field descriptions (continued)

Field	Description
	FSE - For Synchronous Channels - Frame Synchronization Enable. When set, enables Streaming Channel Frame Synchronization for this logical synchronous channel FCE - For Isochronous Channels - Flow Control Enable. When set, allows an isochronous RX channel to generate the ReceiverBusy (10h) response.
26–25 MDS	Channel n Mode Select. 00 ping-pong buffering used 01 circular buffering used 10-11 Reserved
24–23 --	Reserved. Should be written as zero for compatibility.
22 MLFS	Mask Lost Frame Synchronization. When set, masks *Lost Frame Synchronization* channel interrupts for this logical channel. 0 Enable *Lost Frame Synchronization* channel interrupts for this logical channel 1 Disable *Lost Frame Synchronization* channel interrupts for this logical channel
21 --	Reserved. Should be written as zero for compatibility.
20 MBE	Mask Buffer Error. When set,masks *Buffer Error* channel interrupts for this logical channel. 0 Enable *Buffer Error* channel interrupts for this logical channel 1 Disable *Buffer Error* channel interrupts for this logical channel
19 MBS	Mask Buffer Start. When set, masks *Buffer Start* channel interrupts for this logical channel. 0 Enable *Buffer Start* channel interrupts for this logical channel 1 Disable *Buffer Start* channel interrupts for this logical channel
18 MBD	Mask Buffer End. When set, masks *Buffer Done* channel interrupts for this logical channel. 0 Enable *Buffer Done* channel interrupts for this logical channel 1 Disable *Buffer Done* channel interrupts for this logical channel
17 MDB	Mask Detect Break. When set, masks detect break channel interrupts for this logical channel. This bit is valid for asynchronous and control channels only. 0 Enable detect break channel interrupts for this logical channel 1 Disable detect break channel interrupts for this logical channel
16 MPE	Mask Protocol Error. When set, masks Protocol error channel interrupts for this logical channel. This bit is valid for all RX channel types and valid for only asynchronous and control TX channels. 0 Enable protocol error channel interrupts for this logical channel 1 Disable protocol error channel interrupts for this logical channel
15 Rsvd/FSCD/IPL[7]	The meaning of this bit depends on the type of channel: Rsvd: For Asynchronous and Control channels only IPL[7]: For Isochronous channels only - Isochronous Packet Length bit 7 field Isochronous Packet Length. For Isochronous TX channels, defines the number of packet bytes. The smallest isochronous packet size per frame is 5 bytes (IPL[7:0] >= 5). A packet length of 256 bytes can be represented as IPL[7:0]=00h. For Isochronous RX channels, software must program IPL[7:2] to indicate the expected number of bytes per packet, where IPL[1:0] always equals'00'.

Table continues on the next page...

MLB_CECR*n* field descriptions (continued)

Field	Description
	FSCD: For Synchronous channels only - Frame Synchronization Channel Disable. When set, disables this logical channel (set CECRn.CE = 0) when Lost Frame Synchronization occurs. 0 Do not disable this logical channel when frame synchronization is lost 1 Disable this logical channel when frame synchronization is lost
14–13 Rsvd/IPL[6:5]	The meaning of this bit depends on the type of channel: Rsvd: For Asynchronous, control and synchronous channels only IPL[6:5]: For Isochronous channels only - Isochronous Packet Length bit6-bit5 fields
12–8 Rsvd/FSPC/IPL[4:0]	The meaning of this bit depends on the type of channel: Rsvd: For Asynchronous, Control channels only FSPC: For Synchronous channels only - Frame Synchronization Physical Channels Count bit4-bit0 fields. IPL: For Isochronous channels only - Isochronous Packets Length bit4-bit0 fields
7–0 CA	Channel Address. These bits determine the *ChannelAddress* associated with this logical channel. This value is matched against the *ChannelAddress* received each physical channel from the MLB Controller. There is a *ChannelAddress* match if and only if the *ChannelAddress* recovered from the MediaLB input, *MLBSIG*, equals the *ChannelAddress* defined by: CA[15:0] = {7'h00, CA[8:1], 1'b0}.

50.3.12 Channel n Status Configuration Register (MLB_CSCR*n*)

The Channel n Status Configuration Register (CSCRn) reflects the status of the given logical channel. The definition of some of the bit fields in the CSCRn register vary dependant on the selected channel type.

Addresses: 63FE_4000h base + 44h offset + (16d × *n*), where *n* = 0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BM	BF	--										IVB		GIRB	RDY
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	--				PBS	PBD	PBDB	PBPE		LFS	ABE	BE	CBS	CBD	CBDB	CBPE
W					w1c	w1c	w1c	w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_CSCR*n* field descriptions

Field	Description
31 BM	Buffer Empty. When set, the local channel buffer (for channel n) is empty. This bit is set and cleared by hardware. At reset, the local channel buffer is empty (BM = 1).
30 BF	Buffer Full. When set, the local channel buffer (for channel n) is full.This bit is set and cleared by hardware.
29–20 --	Reserved. Should be written as zero for compatibility.
19–18 IVB	Isochronous Valid Bytes. These bits are loaded by hardware with the number of valid bytes in the last packet of a broken Isochronous RX channel. Used in conjunction with CCBCRn.BCA, IVB[1:0] can be used by software to determine the final valid byte of the local channel buffer.(Valid for local channel buffers configured for isochronous RX data). 00 Final valid byte = (CCBCRn.BCA - 5) 01 Final valid byte = (CCBCRn.BCA - 4) 10 Final valid byte = (CCBCRn.BCA - 3) 11 Final valid byte = (CCBCRn.BCA - 2)
17 GIRB	GIRB: For isochronous data - Generate Isochronous Receive Break. When set, this bit causes hardware to terminate the current packet, flush the local channel buffer, clear the RDY bit, and load IVB[1:0]. This bit is set by system software and cleared by hardware.

Table continues on the next page...

MLB_CSCR*n* field descriptions (continued)

Field	Description
	GB: For asynchronous and control data - Generate Break. When the local channel buffer is configured for TX data, the setting of this bit causes hardware to send the AsyncBreak (26h) or ControlBreak (36h) command and stop the transfer. When the local channel buffer is configured for RX data, the setting of this bit causes hardware to send the MLB RxStatus ReceiverBreak (70h) and stop the transfer. This bit is set by system software and cleared by hardware.
16 RDY	Next Buffer Ready. System software should set this bit when all the registers, data, and program memory variables are setup and ready to transmit or receive data. For TX data, the system memory buffer should also be filled. For ping-pong buffering mode, hardware clears this bit after the buffer begins to be processed. For circular buffering mode, software should clear this bit only when buffer processing needs to halted. 0 Next buffer is not ready in system memory 1 Next buffer is ready in system memory
15–12 --	Reserved. Should be written as zero for compatibility.
11 PBS	Previous Buffer Start. When set, this bit indicates the first quadlet of the *Previous Buffer* has been successfully transmitted or received. The setting of this bit generates a maskable channel interrupt to system software. This bit is valid for all channel types. 0 First quadlet of the *Previous Buffer* has not been successfully transmitted or received 1 First quadlet of the *Previous Buffer* has been successfully transmitted or received
10 PBD	Previous Buffer Done. When set, this bit indicates that the last quadlet of the *Previous Buffer* has been successfully transmitted or received. The setting of this bit generates a maskable channel interrupt to system software. This bit is valid for all channel types. 0 Last quadlet of the *Previous Buffer* has not been successfully transmitted or received 1 Last quadlet of the *Previous Buffer* has been successfully transmitted or received
9 PBDB	Previous Buffer Detect Break. When set, this bit indicates that either a TX channel has detected a receiver break response, *ReceiverBreak* (70h), or an RX channel has detected a transmitter break command, *ControlBreak* (36h) or *AsyncBreak* (26h), while processing the *Previous Buffer*. The setting of this bit generates a maskable channel interrupt to system software. This bit is valid for all channel types 0 Break response was not detected while processing the Previous Buffer 1 Break response was detected while processing the Previous Buffer
8 PBPE	Previous Buffer Protocol Error. When set, this bit indicates that either a TX channel has detected an RxStatus of *ReceiverProtocolError* (72h), a RX channel has detected an invalid command for this channel type, or an additional *AsyncStart* (20h) or *ControlStart* (30h) command has been received while in the middle of a packet. The setting of this bit generates a maskable channel interrupt to system software. This bit is valid for all RX channels and valid for only asynchronous and control TX channels. 0 Protocal error was not detected while processing the Previous Buffer 1 Protocal error was detected while processing the Previous Buffer
7 --	Reserved. Should be written as zero for compatibility.
6 LFS	Lost Frame Synchronization. When set, this bit indicates that the logical channel has lost synchronization with the MLB frame. The setting of this bit generates a maskable channel interrupt to system software. This bit is valid for synchronous channels only. 0 Frame synchronization not lost 1 Frame synchronization lost

Table continues on the next page...

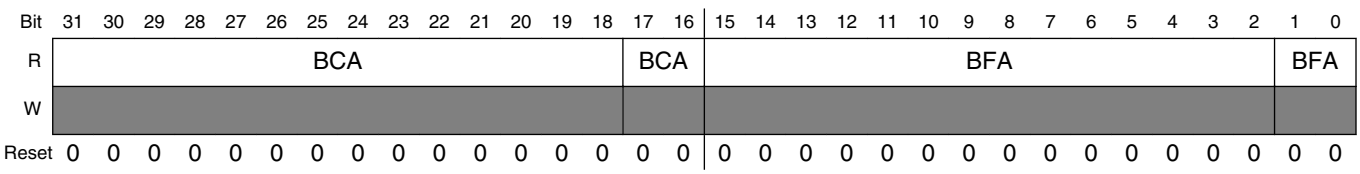
MLB_CSCR*n* field descriptions (continued)

Field	Description
5 ABE	AHB Bus Error. When set, this bit indicates that an AHB bus error has been detected. The setting of this bit generates a non-maskable channel interrupt to system software.
4 BE	Buffer Error. When set, this bit indicates that either a TX channel has detected a buffer underflow (for example, attempt to pop data from an empty buffer), or an RX channel has detected a buffer overflow (for example, attempt to push data onto a full buffer). The setting of this bit generates a maskable channel interrupt to system software. This bit is valid for synchronous RX/TX and isochronous RX (CECRn.FCE = 0) channels only. 0 TX underflow or RX underflow not detected 1 TX underflow or RX underflow detected
3 CBS	Current Buffer Start. When set, this bit indicates that the DMA controller has started processing the *Current Buffer*. This bit is set after the contents of CNBCRn have been loaded into CCBCRn, the CSCRn.RDY bit has been cleared (for ping-pong buffering), and hardware is available to accept the next buffer. The setting of this bit generates a maskable channel interrupt to system software. This bit is valid for all channel types. 0 First quadlet of the Current Buffer has not been successfully transmitted or received 1 First quadlet of the Current Buffer has been successfully transmitted or received
2 CBD	Current Buffer Done. When set, this bit indicates that the last quadlet from the last packet (in the *Current Buffer*) has been successfully transmitted or received. The setting of this bit generates a maskable channel interrupt to system software. This bit is valid for all channel types. 0 Last quadlet of the *Current Buffer* has not been successfully transmitted or received 1 Last quadlet of the *Current Buffer* has been successfully transmitted or received
1 CBDB	Current Buffer Detect Break. When set, this bit indicates that either a TX channel has detected a receiver break response, *ReceiverBreak* (70h), or an RX channel has detected a transmitter break command, *ControlBreak* (36h) or *AsyncBreak* (26h), while processing the *Current Buffer*. The setting of this bit generates a maskable channel interrupt to system software. This bit is valid for asynchronous and control channels only. 0 Break response was not detected while processing the Current Buffer 1 Break response was detected while processing the Current Buffer
0 CBPE	Current Buffer Protocol Error. This bit indicates that either a TX channel has detected an RxStatus of *ReceiverProtocolError* (72h), an RX channel has detected an invalid command for a given channel type, or an additional *ControlStart* (30h) or *AsyncStart* (20h) command has been received while in the middle of a packet. The setting of this bit generates a maskable channel interrupt to system software. This bit is valid for all RX channel types and valid for only asynchronous and control TX channels. 0 Protocal error was not detected while processing the Current Buffer 1 Protocal error was detected while processing the Current Buffer

50.3.13 Channel n Current Buffer Configuration Register (MLB_CCBCR*n*)

The Channel n Current Buffer Configuration Register (CCBCRn) allows system software to monitor the address pointer and buffer length of the Current Buffer in system memory for the logical channel.

Addresses: 63FE_4000h base + 48h offset + (16d × *n*), where *n* = 0d to 15d



MLB_CCBCR*n* field descriptions

Field	Description
31–18 BCA	Buffer Current Address. The BCA field defines a 16-bit address pointer, which identifies the lower half of the beginning address of the *Current Buffer* in system memory. The BCA[15:2] bits are loaded from CNBCRn.BSA[15:2] when the Next Buffer is ready for processing. This Current Buffer address pointer should always be quadlet aligned (for example, BCA[1:0] equals 2'b00). During the processing of the *Current Buffer*, the BCA field marks which quadlet of the buffer is currently being processed.
17–16 BCA	For Synchronous, Asynchronous, and Control Channels - Hard-wired to'00' For Isochronous Channels - Buffer Current Address bits 1:0
15–2 BFA	Buffer Final Address. The BFA field defines a 16-bit address pointer, which identifies the lower half of the ending address of the *Current Buffer* in system memory. The BFA[15:2] bits are loaded from CNBCRn.BEA[15:2] when the Next Buffer is read for processing. This Current Buffer address pointer, except when associated with isochronous channels, should always be quadlet aligned (for example, BFA[1:0] equals 2'b00). During the processing of the Current Buffer, the point at which the BCA field becomes equal to (or greater than) the BFA field indicates that the processing of the Current Buffer will end upon successful completion of the current quadlet (for isochronous and synchronous channels) or upon successful completion of the current packet (for asynchronous and control channels). It is the responsibility of system software to ensure the system memory buffers (for RX asynchronous and control channels) can accommodate overflow in the size of the largest packet supported. Additionally, single-packet buffering can be used by simply programming CNBCRn.BSA[15:2] = CNBCRn.BEA[15:2]
1–0 BFA	For Synchronous, Asynchronous, and Control Channels - Hard-wired to'00' For Isochronous Channels - Buffer Final Address bits 1:0

50.3.14 Channel n Next Buffer Configuration Register (MLB_CNBCR*n*)

The Channel n Next Buffer Configuration Register (CNBCRn) allows system software to set the start and end addresses of the Next Buffer in system memory for the logical channel.

Addresses: 63FE_4000h base + 4Ch offset + (16d × *n*), where *n* = 0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BSA																BEA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MLB_CNBCR*n* field descriptions

Field	Description
31–16 BSA	Buffer Start Address. The BSA field defines a 16-bit address pointer, which identifies the lower half of the beginning address of the *Next Buffer* in system memory. Once system software detects CSCRn.RDY has been cleared by hardware (for ping-pong buffering), the beginning address of the *Next Buffer* may be loaded into BSA[15:2]. System software should then set CSCRn.RDY. Once processing of the *Current Buffer* for the logical channel is complete, the BSA[15:2] field is loaded into the CCBCRn.BCA[15:2] field and processing of the next buffer can begin. This *Next Buffer* address pointer must always be quadlet aligned (for example, BSA[1:0] must be written as 2'b00).
15–0 BEA	Buffer End Address. The BEA field defines a 16-bit address pointer, which identifies the lower half of the ending address of the *Next Buffer* in system memory. Once system software detects CSCRn.RDY has been cleared by hardware (for ping-pong buffering), the ending address of the *Next Buffer* may be loaded into BEA[15:2]. System software should then set CSCRn.RDY. Once processing of the *Current Buffer* for the logical channel is complete, the BEA[15:2] field is loaded into the CCBCRn.BFA[15:2] field and processing of the next buffer can begin. The BEA[15:2] bits are loaded into CCBCRn.BFA[15:2] when the *Current Buffer* is finished being processed. This *Next Buffer* address pointer, except when associated with isochronous channels, should always be quadlet aligned (for example, BEA[1:0] defaults to 2'b00).

50.3.15 Local Channel n Buffer Configuration Register (MLB_LCBCR*n*)

The Local Channel n Buffer Configuration Register (LCBCRn) allows software to optimize use of the Local Channel Buffer RAM. This register should only be written by software while the logical channel is disabled (for example, CECR3.CE clear disables Channel 3; therefore software may write LCBCR3). Writing to this register while the corresponding logical channel is enabled may result in unexpected behavior.

Addresses: 63FE_4000h base + 280h offset + (16d × *n*), where *n* = 0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	--												BD				--							SA								
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	1*	0*	0*	0*	0*	0*	0*	0*	1*	1*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- See descriptions for correct reset values.

MLB_LCBCR*n* field descriptions

Field	Description
31–20 --	Reserved
19–13 BD	Buffer Depth. This field defines the depth of the local channel buffer space in the local buffer RAM. 00h Depth = 4 quadlets 01h Depth = 8 quadlets 02h Depth = 12 quadlets 7Fh Depth = 512 quadlets
12–9 --	Reserved
8–0 SA	Buffer Start Address. This field defines the starting address of the channel buffer space in the local buffer RAM. At reset, the LCBCR.SA[9:0] field is loaded with the channel number multiplied by 32 (for channel number multiplied by 128 quadlets). 000h Start Address = 0 quadlets 001h Start Address = 4 quadlets 002h Start Address = 8 quadlets 3FFh Start Address = 2044 quadlets

50.4 Functional Description

50.4.1 Local Channel Buffer RAM

A single-port RAM is used to implement the memory space for local channel buffering. The size of the RAM is 2k x 36-bits (1 quadlet of data; 4-bit tag). The initial start address, depth and threshold values for logical channel buffer RAM are determined by parameters RAM_SADDR and BUF_DEPTH. See Table 50-99 for parameters that indicate the start address, depth and threshold reset values. After reset, the start address, depth for the logical channels buffered in the RAM are controlled through the LCBCRn registers, the initial values can be overwritten by software.

Table 50-99. Local Channel Buffer RAM Parameters

Logical Channel	RAM_SADDR	BUF_DEPTH
0	0	16
1	16	16
2	32	144
3	176	144
4	320	144
5	464	144
6	608	144
7	752	144
8	896	144
9	1040	144
10	1184	144
11	1328	144
12	1472	144
13	1616	144
14	1760	144
15	1904	144

See Figure 50-97 for more information on using the LCBCRn register to configure the local RAM buffer.

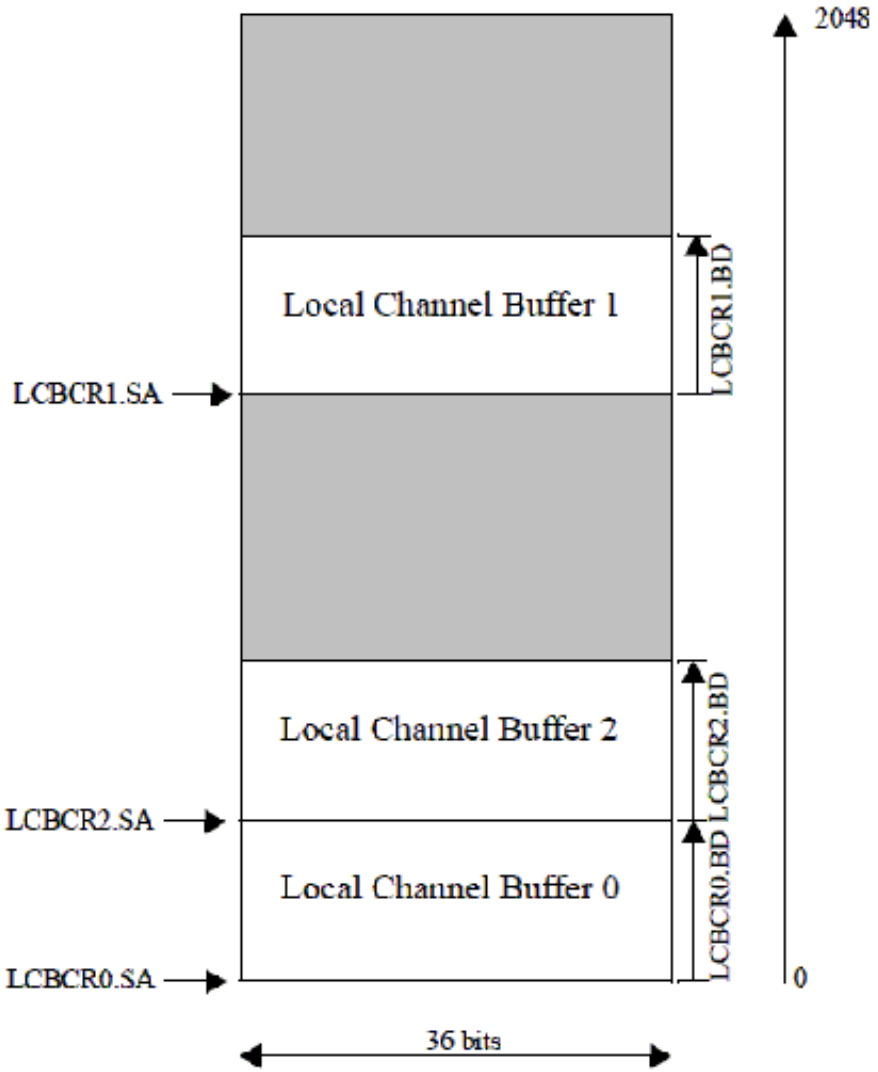


Figure 50-97. Programming Example for LCBCRn

50.4.1.1 Local Buffer Start Address

The initial buffer start address of each local channel buffer is defined by the default (after reset) start address of each local channel buffer (LCBCRn.SA [8:0]). The start address is the location of the beginning of the buffer in the local buffer RAM. Software may change the start address of each local channel buffer after reset by writing to LCBCRn.SA[8:0] directly.

50.4.1.2 Local Channel Buffer Depth

The initial buffer depth of each local channel buffer is configured by the default (after reset) depth of each local channel buffer (LCBCRn.BD[6:3] and LCBCRn.BD[2:0]) in quadlets. The buffer depth should be set based on the worst-case peripheral interface read/write latency. Software may change the depth of each local channel buffer after reset by writing to LCBCRn.BD[6:3] and LCBCRn.BD[2:0] directly.

50.4.2 Streaming Channel Frame Synchronization

Certain types of streaming applications require data to be synchronous with the MediaLB frame, including: stereo, 5.1 audio, and Generic Synchronous Packet Format (GSPF) DTCP. The MediaLB *Streaming Channel Frame Synchronization* feature provides this option.

For example, 24-bit stereo channels require two MLB physical channels (PC) to transmit left (0xLLLLLn) and right (0xRRRRRn) speaker data. Assuming the MLB Controller allocates Physical Channel 1 (PC1) and Physical Channel 2 (PC2) to this stereo channel, the data is synchronized to the MLB frame as shown in Table 50-100.

Table 50-100. Example of 24-bit stereo data synchronous to 256Fs MediaLB frame

Frame	PC=0	PC=1	PC=2	PC=3	PC=4	PC=5	PC=6	PC=7
n=0		0xLLLLLLRR	0xRRRRxxxx					
n=1		0xLLLLLLRR	0xRRRRxxxx					
n=2		0xLLLLLLRR	0xRRRRxxxx					
n=3		0xLLLLLLRR	0xRRRRxxxx					

Without frame synchronization, the MediaLB may begin transmitting or receiving data that is not aligned with the MLB frame. Misalignment, as depicted in Table 50-101, may result in data corruption.

Table 50-101. Example of 24-bit stereo data asynchronous to 256Fs MediaLB frame

Frame	PC=0	PC=1	PC=2	PC=3	PC=4	PC=5	PC=6	PC=7
n=0			0xLLLLLLRR					
n=1		0xRRRRxxxx	0xLLLLLLRR					
n=2		0xRRRRxxxx	0xLLLLLLRR					
n=3		0xRRRRxxxx	0xLLLLLLRR					

The MediaLB supports *Streaming Channel Frame Synchronization* as a programmable option for each logical channel configured for synchronous dataflow. System software can enable the frame synchronization feature for a synchronous logical channel by setting CECRn.FSE. When enabled, the synchronous logical channel begins transmitting and receiving data only at a MediaLB frame boundary.

When the loss of MLB frame synchronization occurs, the MLB detects it and optionally notifies system software through a maskable channel interrupt. In order to use this option, system software must:

- program CECRn.FSPC[4:0] with the expected number of physical channels per frame for the logical channel, and
- unmask the CSCRn.STS[6] bit by setting CECRn.MLFSto 0.

A channel interrupt is generated when the actual number of physical channels detected during a MLB frame does not match the expected value. An additional channel interrupt is generated if the local channel buffer overflows (for RX channels) or underflows (for TX channels).

Additionally, software may instruct the MLB to automatically disable a logical channel when MediaLB frame synchronization is lost. To enable this feature, software must set CSCRn.FSCD, which causes the hardware to automatically clear the Channel Enable bit (CECRn.CE) when synchronization is lost.

Frame synchronization is not supported for asynchronous, control, or isochronous channels.

50.4.3 Loop-Back Test Mode

In order to facilitate silicon debug of the MLB Device, hardware supports the *Loop-Back Test Mode*. This mode allows testing of the MLB pads, physical layer, link layer, channel protocol, and local channel buffer. When the DCCR.LBM bit is set, a data path is enabled which allows RX data from even Channel N to be sent out as TX data on Channel N+1, where N = {0, 2, 4, 6,..., 12, 14}.

Figure 50-98 illustrates the *Loop-Back Test Mode* data path.

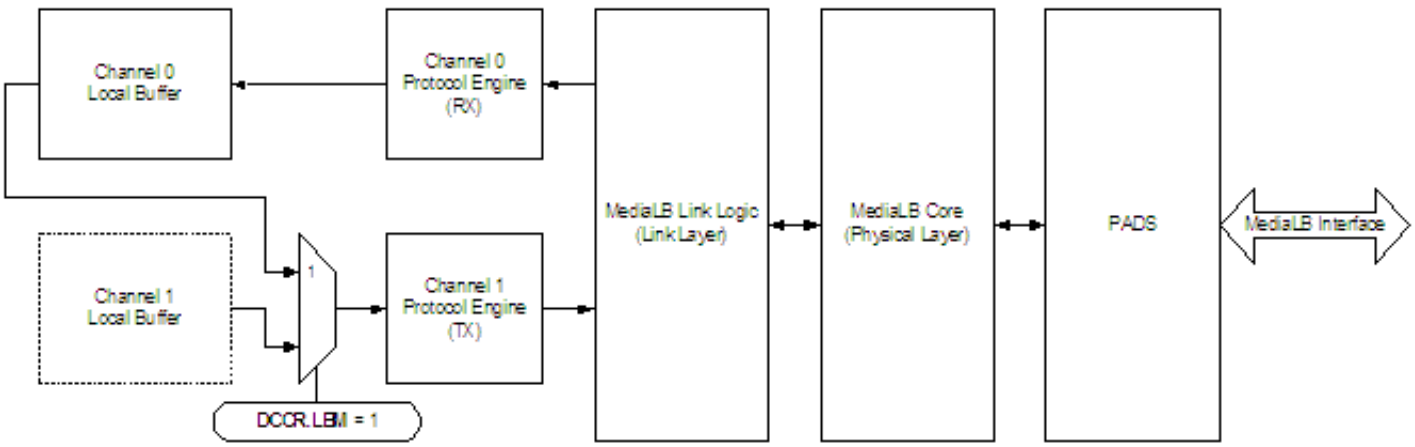


Figure 50-98. Loop-Back Test Mode Data Path

For *Loop-Back Test Mode* operation, software must perform the following steps:

- Set the logical *ChannelAddresses* for Channel N and N+1. (They cannot be the same address.)
- Enable Channel N for receiving synchronous, asynchronous, control, or isochronous data.
- Enable Channel N+1 for transmitting the same channel data type as Channel N.
- Set the Loop-Back Mode bit (DCCR.LBM).

Restrictions on the *Loop-Back Test Mode* are as follows:

- No protocol errors or breaks are allowed on either the RX or TX channel.
- Little-Endian mode must be disabled (DCCR.MLE clear).
- Isochronous packet lengths must be quadlet multiples.

Next Buffer Ready bits for Channels N and N+1 must remain clear (CSCR0.RDY = CSCL1.RDY = 0)

Chapter 51
NAND Flash Controller (NFC)

51.1 Introduction

The Nand Flash Controller (NFC) is composed of various control logic units, a 4.5-Kbyte internal RAM and an internal ECC engine. The NFC can interface standard NAND Flash memory devices. Figure 51-1 shows the block diagram of the NAND Flash controller.

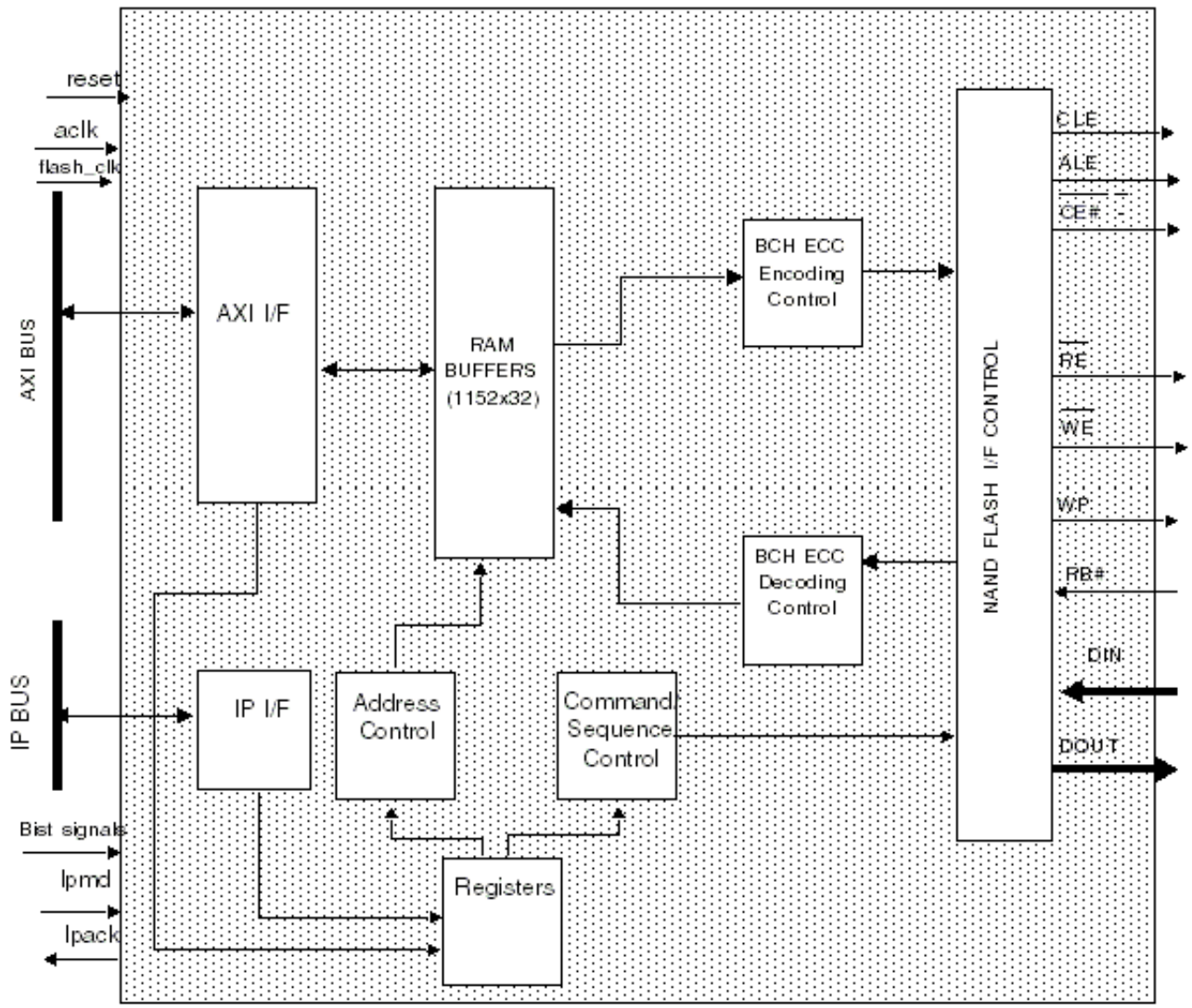


Figure 51-1. NAND Flash Controller Block Diagram

51.2 Overview

The NAND Flash Controller (NFC) can communicate with standard asynchronous NAND Flash devices. It provides a direct interface to both 8-bit and 16-bit NAND Flash devices supporting page sizes of 512B, 2KB, or 4KB.

The NFC is configured by the ARM platform through the AXI and IP Buses. The ARM processor initiates read and write operations (automatic or atomic) towards the NAND flash device. When the operation is finished the NFC issues an interrupt to the host.

The NFC works in two clock domains, aclk (fast clock) for communication with the AXI and ECC calculations and flash_clk (slow clock) for communication with the NAND flash device.

51.3 Features

The NFC includes the following features:

- x8 / x16 NAND Flash bus.
- Page size of 512B, 2KB and 4KB.
- Internal memory mapped RAM of 4KB + spare of up to 512B.
  - Supports page size of 512B + 16B, 2KB + 64B, 4KB + 128B, 4k + 218B, 4KB + 224B (when the spare area is greater than 218B and ECC is enabled then spare area size should be configured to 218B (i.e SPAS = 8'h6D))
- AXI Host interface that can access the internal RAM buffers and several registers. It supports:
  - Read / Write Burst
  - 16-bits / 32-bits bus transfers
- Support atomic (manual) and automatic modes of operation with up to 6 address phases.
- Supports interleaved accesses to up to 4 NAND devices while hiding the busy period of each device
- Supports ECC-BCH(8192) error correction code of 4/8/14/16 error bits in 528/538/548 bytes (512B main data +16B/26B/36B spare).
- Communicates with the system DMA
  - DMA read request after a full page read.
  - DMA write request at the beginning of a full-page programming.
- Multiple Reset
  - AXI reset/Cold Reset/ Warm Reset
- IO pins sharing support
  - Allows sharing of the IO pins with other memory controllers through special arbitration logic.
- Power down handshake mechanism for power saving.

51.4 Restrictions

In case of working with NAND device of x16bits bus then the ratio between aclk and flash clock must be at least 1:3 (i.e the aclk is at least 3 times faster than the flash_clk).

In AUTO_COPY_BACK operation with ADD_OP=01 the NFC can copy data only to the following page.

The address that the NFC drives to the NAND flash device must be aligned to address 0. i.e the columns address must be "0".

51.5 Signals Overview

The following signals shown in Table 51-1 are used to configure and control the NFC and its attached Flash device. Signals ending with _b are active-low signals.

Table 51-1. NFC Signal Properties

Name	Function	I/O	Reset
ipp_nfc_ale_out	Flash Address Latch Enable	O	0
ipp_nfc_ce0_out	Flash #0 Chip Enable	O	1
ipp_nfc_ce1_out	Flash #1 Chip Enable	O	1
ipp_nfc_ce2_out	Flash #2 Chip Enable	O	1
ipp_nfc_ce3_out	Flash #3 Chip Enable	O	1
ipp_nfc_cle_out	Flash Command Latch Enable	O	0
ipp_nfc_rb0_in	Flash #0 Ready/Busy	I	1
ipp_nfc_rb1_in	Flash #1 Ready/Busy	I	1
ipp_nfc_rb2_in	Flash #2 Ready/Busy	I	1
ipp_nfc_rb3_in	Flash #3 Ready/Busy	I	1
ipp_nfc_re_out	Flash Read Enable	O	1
ipp_nfc_read_data_in [15:0]	NFC data input from the NAND Flash	I	
ipp_nfc_we_out	Flash Write Enable	O	1
ipp_nfc_wp_out	Flash Write Protect	O	1
ipp_nfc_write_data_out [15:0]	NFC data output towards the NAND Flash	O	0000

51.6 Detailed Signal Descriptions

The following table gives a detailed description of the NFC signals.

Table 51-2. NFC Detailed Signal Descriptions

Signal	I/O	Description
ipp_nfc_ce0_out	O	Flash #0 Chip Enable. This signal indicates the NAND Flash selection. When the NAND Flash device is accessed, this signal is low. This pin is also called $\overline{\text{CE0\#}}$.
ipp_nfc_ce1_out	O	Flash #1 Chip Enable. This signal indicates the NAND Flash selection. When the NAND Flash device is accessed, this signal is low. This pin is also called $\overline{\text{CE1\#}}$.
ipp_nfc_ce2_out	O	Flash #2 Chip Enable. This signal indicates the NAND Flash selection. When the NAND Flash device is accessed, this signal is low. This pin is also called $\overline{\text{CE2\#}}$.
ipp_nfc_ce3_out	O	Flash #3 Chip Enable. This signal indicates the NAND Flash selection. When the NAND Flash device is accessed, this signal is low. This pin is also called $\overline{\text{CE3\#}}$.
ipp_nfc_re_out	O	Flash Read Enable. This signal controls the read data from the NAND device. The read data is latched on the rising edge of the signal. This pin is also called $\overline{\text{RE\#}}$.
ipp_nfc_we_out	O	Flash Write Enable.This signal controls the write of commands, address and data to the NAND device. The write is latched on the rising edge of the signal. This pin is also called $\overline{\text{WE\#}}$.
ipp_nfc_cle_out	O	Flash Command Latch Enable. The signal controls the command that is being sent to the command register of NAND device. When active high, commands are latched into the command register of NAND device on the rising edge of the $\overline{\text{WE\#}}$ signal. This pin is also called $\overline{\text{CLE\#}}$.
ipp_nfc_ale_out	O	Flash Address Latch Enable. The signal controls address that is being sent to the address register of NAND Flash. When active high, addresses are latched into the NAND device address register of NAND device on the rising edge of the $\overline{\text{WE\#}}$ signal. This pin is also called ALE#.
ipp_nfc_wp_out	O	Flash Write Protect. This signal provides inadvertent program/erase protection. This signal is active (held low) during power-up. This pin is also called WP#.
ipp_nfc_rb0_in	I	Flash #0 Ready/Busy. This signal indicates the status of the NAND Flash operation. When low, it indicates that a program, erase, or read operation of NAND Flash is in process. Upon completion of the process, this signal returns to high state. This pin is also called RB. **NOTE:** This signal is connected to an open drain output, via a 100 KOhm pull-up resister (outside the external NAND Flash memory device).
ipp_nfc_rb1_in	I	Flash #1 Ready/Busy. This signal indicates the status of the NAND Flash operation. When low, it indicates that a program, erase, or read operation of NAND Flash is in process. Upon completion of the process, this signal returns to high state. This pin is also called RB. **NOTE:** This signal is connected to an open drain output, via a 100 KOhm pull-up resister (outside the external NAND Flash memory device).
ipp_nfc_rb2_in	I	Flash #2 Ready/Busy. This signal indicates the status of the NAND Flash operation. When low, it indicates that a program, erase, or read operation of NAND Flash is in process. Upon completion of the process, this signal returns to high state. This pin is also called RB. **NOTE:** This signal is connected to an open drain output, via a 100 KOhm pull-up resister (outside the external NAND Flash memory device).

Table continues on the next page...

Table 51-2. NFC Detailed Signal Descriptions (continued)

Signal	I/O	Description
ipp_nfc_rb3_in	I	Flash #3 Ready/Busy. This signal indicates the status of the NAND Flash operation. When low, it indicates that a program, erase, or read operation of NAND Flash is in process. Upon completion of the process, this signal returns to high state. This pin is also called RB. **NOTE:** This signal is connected to an open drain output, via a 100 KOhm pull-up resister (outside the external NAND Flash memory device).
ipp_nfc_write_data_out[15:0]	O	Flash data output. The NFC uses this bus to drive command/address/data to the NAND device.
ipp_nfc_read_data_in[15:0]	I	Flash data input. The NFC uses this bus to latch data from the NAND device during read operation.

51.7 Memory Map and Register Definition

Table 51-3. Memory Map

Address	Use	Access
0xAXI_BASE_0000–0xAXI_BASE_11FF	Internal RAM	R/W
0xAXI_BASE_1200–0xAXI_BASE_1BFF	AXI Reserved	-
0xAXI_BASE_1E00–0xAXI_BASE_1E43	AXI Reserved	R/W
0xAXI_BASE_1E44–0xAXI_BASE_1FFF	AXI Reserved	-
0xBASE_3000–0xBASE_3003	IP Registers	R/W
0xBASE_3004–0xBASE_3023	IP Registers	-
0xBASE_3024–0xBASE_3037	IP Registers	R/W
0xBASE_3038–0xBASE_3EFC	IP Registers	-

51.7.1 Internal RAM Address Space and Organization

The internal RAM is divided into 4KB of main area and 1/2KB of spare area. The main area is used to hold the data of the NAND device page in read/program operations and is divided into 8 data buffers (also called data sections) of 512B each. The spare area is used to hold user-reserved data as well as the ECC encoding and is divided into 8 buffers of 64B each. Each main area buffer is associated with spare area buffer. For example spare area buffer 0 is associated with main area buffer 0.

Table 51-4 shows the organization of the main and spare area buffers inside the internal RAM. The host can use all of the spare area except for the ECC code areas. For further information of the internal RAM refer to Internal RAM

The NFC automatically generates ECC code for both main and spare area during data programming to NAND Flash. When programming/reading a page, the main and spare area buffer number must be selected using the RAM buffer address (RBA) field at NFC_CONFIGURATION1 Spare Area Buffers

Table 51-4. Internal RAM

Address	Use	Access
0xAXI_BASE_0000-0xAXI_BASE_01FF	Main area Buffer 0	R/W
0xAXI_BASE_0200-0xAXI_BASE_03FF	Main area Buffer 1	R/W
0xAXI_BASE_0400-0xAXI_BASE_05FF	Main area Buffer 2	R/W
0xAXI_BASE_0600-0xAXI_BASE_07FF	Main area Buffer 3	R/W
0xAXI_BASE_0800-0xAXI_BASE_09FF	Main area Buffer 4	R/W
0xAXI_BASE_0A00-0xAXI_BASE_0BFF	Main area Buffer 5	R/W
0xAXI_BASE_0C00-0xAXI_BASE_0DFF	Main area Buffer 6	R/W
0xAXI_BASE_0E00-0xAXI_BASE_0FFF	Main area Buffer 7	R/W
0xAXI_BASE_1000-0xAXI_BASE_103F	Spare area Buffer 0 (SB0)	R/W
0xAXI_BASE_1040-0xAXI_BASE_107F	Spare area Buffer 1 (SB1)	R/W
0xAXI_BASE_1080-0xAXI_BASE_10BF	Spare area Buffer 2 (SB2)	R/W
0xAXI_BASE_10C0-0xAXI_BASE_10FF	Spare area Buffer 3 (SB3)	R/W
0xAXI_BASE_1100-0xAXI_BASE_113F	Spare area Buffer 4 (SB4)	R/W
0xAXI_BASE_1140-0xAXI_BASE_117F	Spare area Buffer 5 (SB5)	R/W
0xAXI_BASE_1180-0xAXI_BASE_11BF	Spare area Buffer 6 (SB6)	R/W
0xAXI_BASE_11C0-0xAXI_BASE_11FF	Spare area Buffer 7 (SB7)	R/W

In case working with a NAND device with spare area of 16B per section of 512B (i.e 512B+spare of 16B, 2KB+ spare of 64B or 4KB + spare of 128B) then only the first 16B of each spare buffer will be used for both user-reserved and ECC code. The number of spare buffers that will be used corresponds the number of main area buffers which are derived from the page size. For example in page size of 4K+ spare of 128B there are 8 main buffers of 512B each and 8 spare buffers of 16B each.

In case working with a NAND device of 4KB + spare of 218B or 4KB + spare of 224B then the size of the first 7 spare buffers (i.e SB0-SB6) will be 26B and the size of the last spare buffer (i.e SB7) will be 36B.

NOTE

In case of working with a NAND device with page size of 4K+ spare of 218B with ECC_MODE of 8/14bits then the last 10B of the last spare buffer (i.e AXI_BASE+0x11DA-AXI_BASE +0x11E3) are not ECC protected. Moreover, in case of 4bits ECC_MODE then the SPAS should be configured to 0x40 (i.e

spare of 128B) and the NFC will use only the first 16B (8B for user-reserved and 8B for ECC) for every section of 512B. That means that the NFC will use only 128B of spare out of the 218B available. In 16 bits ECC_MODE all the bytes of the main and spare buffers are ECC protected.

Table 51-5 shows the sub-organization of the spare area buffers in case of working with NAND device of 4K+218B or 4K+224B (last 6B of the spare area buffer won't be used). Note that the sub-organization of the first 7 buffers (SB0-SB6) are different from the last buffer (SB7).

Table 51-6 shows the sub-organization of the spare area buffers in case of working with NAND devices of 512B+16B, 2K+64B or 4K+128B (i.e 16B per spare buffer)

Table 51-5. Spare Area Buffer for 4K+218B NAND

Address	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
0xAXI_BASE_1000 (SB0)	ECC Byte / Reserved for user(MSB)[1]								ECC Byte / Reserved for user(LSB)[1]							
0xAXI_BASE_1002 (SB0)	ECC Byte / Reserved for user[2]								ECC Byte / Reserved for user[2]							
0xAXI_BASE_1004 (SB0)	ECC Byte / Reserved for user[2]								ECC Byte / Reserved for user[2]							
0xAXI_BASE_1006 (SB0)	ECC Byte / Reserved for user[2]								ECC Byte / Reserved for user[2]							
0xAXI_BASE_1008 (SB0)	ECC Byte / Reserved for user [3]								ECC Byte / Reserved for user [3]							
0xAXI_BASE_100A (SB0)	ECC Byte / Reserved for user [3]								ECC Byte / Reserved for user [3]							
0xAXI_BASE_100C (SB0)	ECC Byte								ECC Byte							
0xAXI_BASE_100E (SB0)	ECC Byte								ECC Byte							
0xAXI_BASE_1010 (SB0)	ECC Byte / Not in Use[4]								ECC Byte / Not in Use [4]							
0xAXI_BASE_1012 (SB0)	ECC Byte / Not in Use [4]								ECC Byte / Not in Use [4]							
0xAXI_BASE_1014 (SB0)	ECC Byte / Not in Use [4]								ECC Byte / Not in Use [4]							
0xAXI_BASE_1016 (SB0)	ECC Byte / Not in Use [4]								ECC Byte / Not in Use [4]							
0xAXI_BASE_1018 (SB0)	ECC Byte / Not in Use [4]								ECC Byte / Not in Use [4]							
0xAXI_BASE_101A-0xAXI_BASE_103F (SB0)	Not in Use															

Table continues on the next page...

Table 51-5. Spare Area Buffer for 4K+218B NAND (continued)

Address	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
0xAXI_BASE_1040-0xAXI_BASE_107F(SB1)	SB1 - SB6 have same assignment like SB0.															
0xAXI_BASE_1080-0xAXI_BASE_10BF (SB2)																
0xAXI_BASE_10C0-0xAXI_BASE_10FF (SB3)																
0xAXI_BASE_1100-0xAXI_BASE_113F(SB4)																
0xAXI_BASE_1140-0xAXI_BASE_117F (SB5)																
0xAXI_BASE_1180-0xAXI_BASE_11BF (SB6)																
0xAXI_BASE_11C0-0xAXI_BASE_11E3 (SB7)	Reserved for user(MSB)								Reserved for user(LSB)							
	ECC Byte / Reserved for user[5]								ECC Byte / Reserved for user [5]							
	ECC Byte / Reserved for user [5]								ECC Byte / Reserved for user [5]							
	ECC Byte / Reserved for user [5]								ECC Byte / Reserved for user [5]							
	.ECC Byte / Reserved for user [6]								ECC Byte / Reserved for user [6]							
	ECC Byte / Reserved for user [3]								ECC Byte / Reserved for user [3]							
	ECC Byte								ECC Byte							
	ECC Byte								ECC Byte							
	ECC Byte / Not in Use [4]								ECC Byte / Not in Use [4]							
	ECC Byte / Not in Use [4]								ECC Byte / Not in Use [4]							
	ECC Byte / Not in Use [4]								ECC Byte / Not in Use [4]							
	ECC Byte / Not in Use [4]								ECC Byte / Not in Use [4]							
	ECC Byte / Not in Use [4]								ECC Byte / Not in User [4]							
	ECC Byte / Reserved for user / Not in Use[7]								ECC Byte / Reserved for user / Not in Use [7]							
	ECC Byte / Reserved for user / Not in Use [7]								ECC Byte / Reserved for user / Not in Use [7]							
	ECC Byte / Reserved for user / Not in Use [7]								ECC Byte / Reserved for user / Not in Use [7]							
	ECC Byte / Reserved for user / Not in Use [7]								ECC Byte / Reserved for user / Not in Use [7]							
	ECC Byte / Reserved for user / Not in Use [7]								ECC Byte / Reserved for user / Not in Use [7]							
0xAXI_BASE_11E4-0xAXI_BASE_11FF	Not in Use															

1. When working in 16 bits ECC_MODE, this Byte is an ECC Byte. When working in 4/8/14 bits ECC_MODE, this Byte is user-reserved.
2. When working in 14/16 bits ECC_MODE, this Byte is an ECC Byte. When working in 4/8 bits ECC_MODE, this Byte is user-reserved.
3. When working in 4/14/16 bits ECC_MODE, this Byte is an ECC Byte. When working in 8bits ECC_MODE, this Byte is user-reserved.

4. When working in 8/14/16 bits ECC_MODE, this Byte is an ECC Byte. When working in 4bits ECC_MODE, this Byte is not used.
5. When working in 14 bits ECC_MODE, this Byte is an ECC Byte. When working in 4/8/16 bits ECC_MODE, this Byte is user-reserved
6. When working in 4/14 bits ECC_MODE, this Byte is an ECC Byte. When working in 8/16 bits ECC_MODE, this Byte is user-reserved
7. When working in 16 bits ECC_MODE, this Byte is an ECC Byte. When working in 8/14 bits ECC_MODE, this Byte is unprotected ECC user-reserved. When working in 4 bits ECC_MODE this Byte is not used.

In case of working with NAND devices of 512B+16B, 2K+64B or 4K+128B (i.e 16B of spare buffer) then only ECC_MODE of 4bits is allowed and then the first 8B of the relevant spare buffers are reserved for user and the rest 8B are reserved for ECC. For example in NAND device of 2K+64B only the first 4 spare buffers are in use.

Table 51-6. Spare Area Buffer for NAND devices of 512B+16B, 2K+64, 4K +128B

Address	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
0xAXI_BASE_1000 (SB0)	Reserved for user(MSB)								Reserved for user (LSB)							
0xAXI_BASE_1002 (SB0)	Reserved for user								Reserved for user							
0xAXI_BASE_1004 (SB0)	Reserved for user								Reserved for user							
0xAXI_BASE_1006 (SB0)	Reserved for user								Reserved for user							
0xAXI_BASE_1008 (SB0)	ECC Byte								ECC Byte							
0xAXI_BASE_100A (SB0)	ECC Byte								ECC Byte							
0xAXI_BASE_100C (SB0)	ECC Byte								ECC Byte							
0xAXI_BASE_100E (SB0)	ECC Byte								ECC Byte							
0xAXI_BASE_1010-0xAXI_BASE_103F(SB0)	Reserved for user															
0xAXI_BASE_1040-0xAXI_BASE_107F (SB1)	SB1 - SB7 have same assignment like SB0.															
0xAXI_BASE_1080-0xAXI_BASE_10BF (SB2)																
0xAXI_BASE_10C0-0xAXI_BASE_10FF (SB3)																
0xAXI_BASE_1100-0xAXI_BASE_113F (SB4)																
0xAXI_BASE_1140-0xAXI_BASE_117F (SB5)																
0xAXI_BASE_1180-0xAXI_BASE_11BF (SB6)																
0xAXI_BASE_11C0-0xAXI_BASE_11FF (SB7)																

51.8 AXI Memory Map

This section contains the NFC and NFC-AXI memory maps.

NFC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F7FF_1E00	NAND Flash command (NFC_NAND_CMD)	32	R/W	0000_0000h	51.8.1/ 3582
F7FF_1E04	NAND Flash address0 (NFC_NAND_ADD0)	32	R/W	0000_0000h	51.8.2/ 3582
F7FF_1E08	NAND address1 (NFC_NAND_ADD1)	32	R/W	0000_0000h	51.8.3/ 3583
F7FF_1E0C	NAND address2 (NFC_NAND_ADD2)	32	R/W	0000_0000h	51.8.4/ 3583
F7FF_1E10	NAND address3 (NFC_NAND_ADD3)	32	R/W	0000_0000h	51.8.5/ 3584
F7FF_1E14	NAND address4 (NFC_NAND_ADD4)	32	R/W	0000_0000h	51.8.6/ 3585
F7FF_1E18	NAND address5 (NFC_NAND_ADD5)	32	R/W	0000_0000h	51.8.7/ 3585
F7FF_1E1C	NAND address6 (NFC_NAND_ADD6)	32	R/W	0000_0000h	51.8.8/ 3586
F7FF_1E20	NAND address7 (NFC_NAND_ADD7)	32	R/W	0000_0000h	51.8.9/ 3586
F7FF_1E24	NAND address8 (NFC_NAND_ADD8)	32	R/W	0000_0000h	51.8.10/ 3587
F7FF_1E28	NAND address9 (NFC_NAND_ADD9)	32	R/W	0000_0000h	51.8.11/ 3587
F7FF_1E2C	NAND address10 (NFC_NAND_ADD10)	32	R/W	0000_0000h	51.8.12/ 3588
F7FF_1E30	NAND address11 (NFC_NAND_ADD11)	32	R/W	0000_0000h	51.8.13/ 3588
F7FF_1E34	NFC configuration (NFC_CONFIGURATION1)	32	R/W	0000_0000h	51.8.14/ 3589
F7FF_1E38	ECC status result (NFC_ECC_STATUS_RESULT)	32	R	0000_0000h	51.8.15/ 3592
F7FF_1E3C	status sum (NFC_STATUS_SUM)	32	R/W	0000_0000h	51.8.16/ 3595
F7FF_1E40	Initiate an NFC operation (NFC_LAUNCH_NFC)	32	R/W	0000_0000h	51.8.17/ 3595

51.8.1 NAND Flash command (NFC_NAND_CMD)

This register contains the commands to be written during a command phase.

Address: NFC_NAND_CMD is F7FF_0000h base + 1E00h offset = F7FF_1E00h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NAND_CMD3								NAND_CMD2								NAND_CMD1								NAND_CMD0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_NAND_CMD field descriptions

Field	Description
31–24 NAND_CMD3	NAND Flash Command3. second NAND Flash command which will be written to the NAND Flash device during an automatic copy-back operations at the write phase.
23–16 NAND_CMD2	NAND Flash Command2. First NAND Flash command which will be written to the NAND Flash device during an automatic copy-back operations at the write phase.
15–8 NAND_CMD1	NAND Flash Command1. Second NAND Flash command (command confirmation) which will be written to the NAND Flash device during an automatic erase/read/program/copy-back operations. In copy-back operation this command is the second command at the read phase.
7–0 NAND_CMD0	NAND Flash Command0. First NAND Flash command which will be written to the NAND Flash device during an automatic and atomic erase/read/program/copy-back operations. In copy-back operation this command is the second command at the read phase. In atomic (manual) operation only this command will hold the command that will be written to the NAND device.

51.8.2 NAND Flash address0 (NFC_NAND_ADD0)

This register contains the 32 lowest address bits to be written during an address phase to address group0.

- The NFC is always aligned to column 0 therefore the address bits of the column suppose to be zero.
- In case of Erase operation the lsb of the page address should be written to NAND_ADD0[0], In other operations the lsb of the column address should be written to NAND_ADD0[0].

NOTE

For example: Assume 4 devices are connected to the NFC, erasing block 0 of device #0 requires writing 48'h0 to NFC_ADDx register. Erasing block 0 requires writing 48'h1 to NFC_ADDx register. Assuming 2KB devices: Programming page 0 requires writing 48'h10000 to NFC_ADDx register (16 lower bits are column address). For further examples refer to

ADD_OP at NFC Operation Configuration3 (NFC_CONFIGURATION3)

Address: NFC_NAND_ADD0 is F7FF_0000h base + 1E04h offset = F7FF_1E04h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NAND_ADD0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_NAND_ADD0 field descriptions

Field	Description
31–0 NAND_ADD0	NAND Flash address group 0. This defines the lower 32 bits of address group 0. This will be used as the address of the page/block in an automatic and atomic operations.

51.8.3 NAND address1 (NFC_NAND_ADD1)

This register contains the 32 lowest address bits to be written during an address phase to address group1.

- The NFC is always aligned to column 0 therefore the address bits of the column suppose to be zero.
- In case of Erase operation the lsb of the page address should be written to NAND_ADD1[0], In other operations the lsb of the column address should be written to NAND_ADD1[0].

Address: NFC_NAND_ADD1 is F7FF_0000h base + 1E08h offset = F7FF_1E08h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NAND_ADD1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_NAND_ADD1 field descriptions

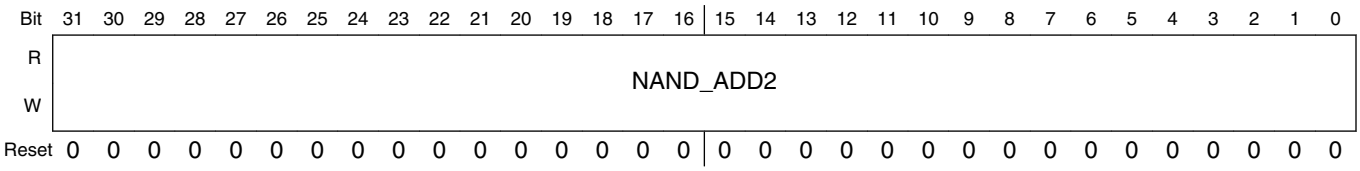
Field	Description
31–0 NAND_ADD1	NAND Flash address group 1. This defines the lower 32 bits of address group 1. This will be used as the address of the page/block in automatic operations depending on ADD_OP.

51.8.4 NAND address2 (NFC_NAND_ADD2)

This register contains the 32 lowest address bits to be written during an address phase to address group2.

- The NFC is always aligned to column 0 therefore the address bits of the column suppose to be zero.
- In case of Erase operation the lsb of the page address should be written to NAND_ADD2[0]. In other operations the lsb of the column address should be written to NAND_ADD2[0].

Address: NFC_NAND_ADD2 is F7FF_0000h base + 1E0Ch offset = F7FF_1E0Ch



NFC_NAND_ADD2 field descriptions

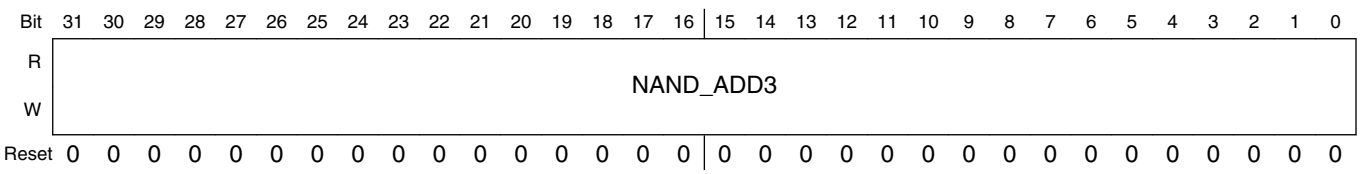
Field	Description
31–0 NAND_ADD2	NAND Flash address group 2. This defines the lower 32 bits of address group 2. This will be used as the address of the page/block in automatic operations depending on ADD_OP.

51.8.5 NAND address3 (NFC_NAND_ADD3)

This register contains the 32 lowest address bits to be written during an address phase to address group3.

- The NFC is always aligned to column 0 therefore the address bits of the column suppose to be zero.
- In case of Erase operation the lsb of the page address should be written to NAND_ADD3[0], In other operations the lsb of the column address should be written to NAND_ADD3[0].

Address: NFC_NAND_ADD3 is F7FF_0000h base + 1E10h offset = F7FF_1E10h



NFC_NAND_ADD3 field descriptions

Field	Description
31–0 NAND_ADD3	NAND Flash address group 3. This defines the lower 32 bits of address group 3. This will be used as the address of the page/block in automatic operations depending on ADD_OP.

51.8.6 NAND address4 (NFC_NAND_ADD4)

This register contains the 32 lowest address bits to be written during an address phase to address group4.

The NFC is always aligned to column 0 therefore the address bits of the column suppose to be zero.

- In case of Erase operation the lsb of the page address should be written to NAND_ADD4[0], In other operations the lsb of the column address should be written to NAND_ADD4[0].

Address: NFC_NAND_ADD4 is F7FF_0000h base + 1E14h offset = F7FF_1E14h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NAND_ADD4																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_NAND_ADD4 field descriptions

Field	Description
31–0 NAND_ADD4	NAND Flash address group 4. This defines the lower 32 bits of address group 4. This will be used as the address of the page/block in automatic operations depending on ADD_OP.

51.8.7 NAND address5 (NFC_NAND_ADD5)

This register contains the 32 lowest address bits to be written during an address phase to address group5.

- The NFC is always aligned to column 0 therefore the address bits of the column suppose to be zero.
- In case of Erase operation the lsb of the page address should be written to NAND_ADD5[0], In other operations the lsb of the column address should be written to NAND_ADD5[0].

Address: NFC_NAND_ADD5 is F7FF_0000h base + 1E18h offset = F7FF_1E18h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NAND_ADD5																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_NAND_ADD5 field descriptions

Field	Description
31–0 NAND_ADD5	NAND Flash address group 5. This defines the lower 32 bits of address group 5. This will be used as the address of the page/block in automatic operations depending on ADD_OP.

51.8.8 NAND address6 (NFC_NAND_ADD6)

This register contains the 32 lowest address bits to be written during an address phase to address group6.

- The NFC is always aligned to column 0 therefore the address bits of the column suppose to be zero.
- In case of Erase operation the lsb of the page address should be written to NAND_ADD6[0], In other operations the lsb of the column address should be written to NAND_ADD6[0].

Address: NFC_NAND_ADD6 is F7FF_0000h base + 1E1Ch offset = F7FF_1E1Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NAND_ADD6																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_NAND_ADD6 field descriptions

Field	Description
31–0 NAND_ADD6	NAND Flash address group 6. This defines the lower 32 bits of address group 6. This will be used as the address of the page/block in automatic operations depending on ADD_OP.

51.8.9 NAND address7 (NFC_NAND_ADD7)

This register contains the 32 lowest address bits to be written during an address phase to address group7.

- The NFC is always aligned to column 0 therefore the address bits of the column suppose to be zero.
- In case of Erase operation the lsb of the page address should be written to NAND_ADD7[0], In other operations the lsb of the column address should be written to NAND_ADD7[0].

Address: NFC_NAND_ADD7 is F7FF_0000h base + 1E20h offset = F7FF_1E20h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NAND_ADD7																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_NAND_ADD7 field descriptions

Field	Description
31–0 NAND_ADD7	NAND Flash address group 7. This defines the lower 32 bits of address group 7. This will be used as the address of the page/block in automatic operations depending on ADD_OP.

51.8.10 NAND address8 (NFC_NAND_ADD8)

This register contains the 16 upper address bits to be written during an address phase to address groups 0 and 1.

Address: NFC_NAND_ADD8 is F7FF_0000h base + 1E24h offset = F7FF_1E24h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NAND_ADD1																NAND_ADD0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_NAND_ADD8 field descriptions

Field	Description
31–16 NAND_ADD1	NAND Flash address group 1. This defines the upper 16 bits of address group 1. This will be used as the address of the page/block in automatic operations depending on ADD_OP.
15–0 NAND_ADD0	NAND Flash address group 0. This defines the upper 16 bits of address group 0. This will be used as the address of the page/block in automatic operations depending on ADD_OP.

51.8.11 NAND address9 (NFC_NAND_ADD9)

This register contains the 16 upper address bits to be written during an address phase to address groups 2 and 3.

Address: NFC_NAND_ADD9 is F7FF_0000h base + 1E28h offset = F7FF_1E28h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NAND_ADD3																NAND_ADD2															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_NAND_ADD9 field descriptions

Field	Description
31–16 NAND_ADD3	NAND Flash address group 3. This defines the upper 16 bits of address group 3. This will be used as the address of the page/block in automatic operations depending on ADD_OP.
15–0 NAND_ADD2	NAND Flash address group 2. This defines the upper 16 bits of address group 2. This will be used as the address of the page/block in automatic operations depending on ADD_OP.

51.8.12 NAND address10 (NFC_NAND_ADD10)

This register contains the 16 upper address bits to be written during an address phase to address groups 4 and 5.

Address: NFC_NAND_ADD10 is F7FF_0000h base + 1E2Ch offset = F7FF_1E2Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NAND_ADD5																NAND_ADD4															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_NAND_ADD10 field descriptions

Field	Description
31–16 NAND_ADD5	NAND Flash address group 5. This defines the upper 16 bits of address group 5. This will be used as the address of the page/block in automatic operations depending on ADD_OP.
15–0 NAND_ADD4	NAND Flash address group 4. This defines the upper 16 bits of address group 4. This will be used as the address of the page/block in automatic operations depending on ADD_OP.

51.8.13 NAND address11 (NFC_NAND_ADD11)

This register contains the 16 upper address bits to be written during an address phase to address groups 6 and 7.

Address: NFC_NAND_ADD11 is F7FF_0000h base + 1E30h offset = F7FF_1E30h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NAND_ADD7[47:32]																NAND_ADD6[47:32]															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

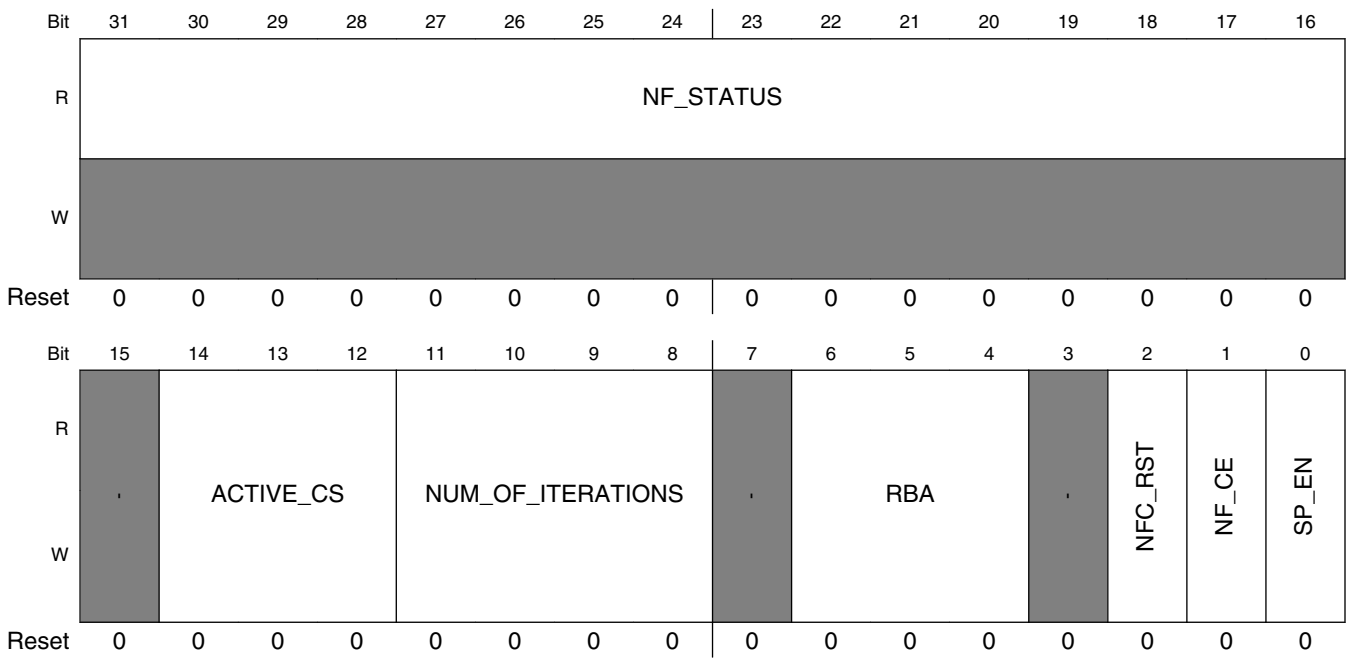
NFC_NAND_ADD11 field descriptions

Field	Description
31–16 NAND_ ADD7[47:32]	NAND Flash address group 7. This defines the upper 16 bits of address group 7. This will be used as the address of the page/block in automatic operations depending on ADD_OP.
15–0 NAND_ ADD6[47:32]	NAND Flash address group 6. This defines the upper 16 bits of address group 6. This will be used as the address of the page/block in automatic operations depending on ADD_OP.

51.8.14 NFC configuration (NFC_CONFIGURATION1)

This register specifies interactive configuration used by the NFC.

Address: NFC_CONFIGURATION1 is F7FF_0000h base + 1E34h offset = F7FF_1E34h



NFC_CONFIGURATION1 field descriptions

Field	Description
31–16 NF_STATUS	NAND Flash Status. This field holds the last NAND device status that was read by the NFC. This register is updated every time that a read status command is executed (AUTO_STAT, atomic read status or when RBB_MODE is set to zero).
15 -	Reserved
14–12 ACTIVE_CS	Active CS. This field indicates the NAND device ID to which the NFC will target its operations. This field is used only during atomic operations (i.e if one of FCMD, FADD, FDI or FDO is set) and during automatic status read operation (AUTO_STAT is set). During other automatic operations the NAND ID is defined based on the ADD_OP register.

Table continues on the next page...

NFC_CONFIGURATION1 field descriptions (continued)

Field	Description
	000 access NAND device connected to CS0 001 access NAND device connected to CS1 010 access NAND device connected to CS2 011 access NAND device connected to CS3 100 access NAND device connected to CS4 101 access NAND device connected to CS5 110 access NAND device connected to CS6 111 access NAND device connected to CS7
11–8 NUM_OF_ ITERATIONS	Number Of Iterations. Defines how many times NFC will execute AUTO_READ/AUTO_PROG/ AUTO_ERASE/AUTO_COPY_BACK without setting an interrupt. When automatic operation bit is set, NFC executes the automatic operation NUM_OF_ITERATIONS+1 times. For example: If NUM_OF_ITERATION is set to 4'h4,and then an AUTO_READ operation is set, then NFC will read 5 pages from the NAND devices. It is the system's responsibility to read the pages content from the NFC's RAM. When NUM_OF_ITERATIONS is greater than 0 and NUM_OF_DEVICES is greater than 0 then an interleaved accesses will be executed and in that case SDMA will be used. if NUM_OF_ITERATION is greater than NUM_OF_DEVICES then after accessing the last device then the NFC will issue another round of accesses to the 1st device and then the 2nd device and etc. until NUM_OF_ITERATIONS is reached. In every round the NFC will increment the address of the page/block. For example if NUM_OF_ITERATIONS=2 (3 iterations) and NUM_OF_DEVICES=1(i.e 2 devices) then the NFC will access 1st device then 2nd device then will increment the address of the page/block and access 1st device and only then issue an interrupt, because 3 iterations were reached. For further examples refer to ADD_OP field at NFC Operation Configuration3 (NFC_CONFIGURATION3) Note: if NUM_OF_ITERATIONS is greater than NUM_OF_DEVICES then this will work only if RBB_MODE is set to "0". 0x0 execute automatic operation 1 time. 0x1 execute automatic operation 2 times. 0xf execute automatic operation 16 times.
7 -	Reserved
6–4 RBA	Ram Buffer Address. Specifies which 1/2K of the internal ram to use when accessing the NAND device. In either Atomic or Automatic operations when NO_SDMA bit is set the value of RBA indicates from which part of the internal RAM to fetch the data in a program operation or to which part of the internal RAM to store the data in a read operation. For example if RBA is set to "100" and the page size is 1/2K (PS is set to "00") and a program operation is configured then the NFC will program the NAND device with the data that is located at the 5th 1/2KB of the internal RAM. In that case it is the user responsibility to store the requested data at the 5th 1/2KB before starting the program operation. Note that in case the page size is 2KB or 4KB then the NFC will fetch/store the data from/to continuos RBA. For example if RBA is set to "000" and page size is 2KB and a program operation is configured then the NFC will program the NAND device with the data that is stored at 1st, 2nd, 3rd, 4th 1/2KB. Note: • In case the page size is 2KB then only RBA of "000" and "100" are allowed and in case the page size is 4KB then only RBA of "000" is allowed. • In case of working with SDMA (NO_SDMA is set to "0") then the NFC will set RBA to "000" in either read or program operations. 000 Use 1st 1/2K of the internal ram.

Table continues on the next page...

NFC_CONFIGURATION1 field descriptions (continued)

Field	Description
	001 Use 2nd 1/2K of the internal ram 010 Use 3rd 1/2K of the internal ram 011 Use 4th 1/2K of the internal ram 100 Use 5th1/2K of the internal ram 101 Use 6th 1/2K of the internal ram 110 Use 7th 1/2K of the internal ram 111 Use 8th 1/2K of the internal ram
3 -	Reserved
2 NFC_RST	NFC Reset. This bit resets the NFC state machines. This bit should be used when issuing a reset command to the NAND Flash device during an operation. When a NAND device received a reset command (0xff) during an operation it aborts its operation and returns to idle. In order to do the same to the NFC, this bit must be set before the reset command is sent. This bit is self cleared. 0 Do not reset the NFC state machine 1 Reset the NFC state machine
1 NF_CE	NAND Flash Force CE. Setting this bit forces the $\overline{\text{CE\#}}$ signal that is driven to the NAND Flash device to 0. There are some NAND devices that require that the CE# will be kept low during the busy period. Therefore when working with such a NAND device with RBB_MODE=0 (in that mode the NFC toggles the CE# while performing a read status during the busy period) NF_CE should set to "1" and then the CE will be forced to "0". This bit is not self cleared. 0 $\overline{\text{CE\#}}$ signal operates normally 1 $\overline{\text{CE}}$ # signal is asserted as long as this bit is set to 1.
0 SP_EN	NAND Flash Spare Enable. This bit determines whether the host will program/read the spare area only or main and spare area together to/from the NAND device. This features is supported only in NAND devices of 1/2K page size. When this bit is enabled then the NFC will program/read 16B of spare area. **NOTE:** There is no ECC when SP_EN=1 0 Program/read main and spare area. 1 Program/read spare area only.

51.8.15 ECC status result (NFC_ECC_STATUS_RESULT)

This register shows the number of bits that were detected/corrected for every page section of 528/538/548 bytes in the main and spare area as a result of the BCH ECC check. For further information of the ECC algorithm refer to 51.12.4/51-77

Address: NFC_ECC_STATUS_RESULT is F7FF_0000h base + 1E38h offset = F7FF_1E38h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NOBER7				NOBER6				NOBER5				NOBER4				NOBER3				NOBER2				NOBER1				NOBER0			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_ECC_STATUS_RESULT field descriptions

Field	Description
31–28 NOBER7	Number Of bit Errors for eighth 528/538/548 bytes section. This field shows the number of bit errors in 512 bytes main area plus 16/26/36 bytes spare (totally 528/538/548 bytes) that were detected/corrected as a result of the BCH ECC check upon read. others > Reserved 0000 No error 0001 >1-3 bit Error (Correctable Error) 0010 > 4-bit Error (Correctable Error) 0011 > 5-7 bit Error (Correctable Error for 8/14/16bit ecc-mode) 0100 > 8-bit Error (Correctable Error for 8/14/16bit ecc-mode) 0101 > 9-13 bit Error (Correctable Error for 14/16bit ecc-mode) 0110 > 14-bit Error (Correctable Error for 14/16bit ecc-mode) 0111 > 15-bit Error (Correctable Error for 16bit ecc-mode) 1000 > 16-bit Error (Correctable Error for 16bit ecc-mode) 1111 > Uncorrectable error.
27–24 NOBER6	Number Of bit Errors for seventh 528/538/548 bytes section. This field shows the number of bit errors in 512 bytes main area plus 16/26/36 bytes spare (totally 528/538/548 bytes) that were detected/corrected as a result of the BCH ECC check upon read. others > Reserved 0000 No error 0001 >1-3 bit Error (Correctable Error) 0010 > 4-bit Error (Correctable Error) 0011 > 5-7 bit Error (Correctable Error for 8/14/16bit ecc-mode) 0100 > 8-bit Error (Correctable Error for 8/14/16bit ecc-mode) 0101 > 9-13 bit Error (Correctable Error for 14/16bit ecc-mode) 0110 > 14-bit Error (Correctable Error for 14/16bit ecc-mode) 0111 > 15-bit Error (Correctable Error for 16bit ecc-mode) 1000 > 16-bit Error (Correctable Error for 16bit ecc-mode) 1111 > Uncorrectable error.

Table continues on the next page...

NFC_ECC_STATUS_RESULT field descriptions (continued)

Field	Description
23–20 NOBER5	Number Of bit Errors for sixth 528/538/548 bytes section. This field shows the number of bit errors in 512 bytes main area plus 16/26/36 bytes spare (totally 528/538/548 bytes) that were detected/corrected as a result of the BCH ECC check upon read. others > Reserved 0000 No error 0001 >1-3 bit Error (Correctable Error) 0010 > 4-bit Error (Correctable Error) 0011 > 5-7 bit Error (Correctable Error for 8/14/16bit ecc-mode) 0100 > 8-bit Error (Correctable Error for 8/14/16bit ecc-mode) 0101 > 9-13 bit Error (Correctable Error for 14/16bit ecc-mode) 0110 > 14-bit Error (Correctable Error for 14/16bit ecc-mode) 0111 > 15-bit Error (Correctable Error for 16bit ecc-mode) 1000 > 16-bit Error (Correctable Error for 16bit ecc-mode) 1111 > Uncorrectable error.
19–16 NOBER4	Number Of bit Errors for fifth 528/538/548 bytes section. This field shows the number of bit errors in 512 bytes main plus area 16/26/36 bytes spare (totally 528/538/548 bytes) that were detected/corrected as a result of the BCH ECC check upon read. others > Reserved 0000 No error 0001 >1-3 bit Error (Correctable Error) 0010 > 4-bit Error (Correctable Error) 0011 > 5-7 bit Error (Correctable Error for 8/14/16bit ecc-mode) 0100 > 8-bit Error (Correctable Error for 8/14/16bit ecc-mode) 0101 > 9-13 bit Error (Correctable Error for 14/16bit ecc-mode) 0110 > 14-bit Error (Correctable Error for 14/16bit ecc-mode) 0111 > 15-bit Error (Correctable Error for 16bit ecc-mode) 1000 > 16-bit Error (Correctable Error for 16bit ecc-mode) 1111 > Uncorrectable error.
15–12 NOBER3	Number Of bit Errors for fourth 528/538/548 bytes section. This field shows the number of bit errors in 512 bytes main area plus 16/26/36 bytes spare (totally 528/538/548 bytes) that were detected/corrected as a result of the BCH ECC check upon read. others > Reserved 0000 No error 0001 >1-3 bit Error (Correctable Error) 0010 > 4-bit Error (Correctable Error) 0011 > 5-7 bit Error (Correctable Error for 8/14/16bit ecc-mode) 0100 > 8-bit Error (Correctable Error for 8/14/16bit ecc-mode) 0101 > 9-13 bit Error (Correctable Error for 14/16bit ecc-mode) 0110 > 14-bit Error (Correctable Error for 14/16bit ecc-mode) 0111 > 15-bit Error (Correctable Error for 16bit ecc-mode) 1000 > 16-bit Error (Correctable Error for 16bit ecc-mode) 1111 > Uncorrectable error.

Table continues on the next page...

NFC_ECC_STATUS_RESULT field descriptions (continued)

Field	Description
11–8 NOBER2	Number Of bit Errors for third528/538/548 bytes section. This field shows the number of bit errors in 512 bytes main area plus 16/26/36 bytes spare (totally 528/538/548 bytes) that were detected/corrected as a result of the BCH ECC check upon read. others > Reserved 0000 No error 0001 >1-3 bit Error (Correctable Error) 0010 > 4-bit Error (Correctable Error) 0011 > 5-7 bit Error (Correctable Error for 8/14/16bit ecc-mode) 0100 > 8-bit Error (Correctable Error for 8/14/16bit ecc-mode) 0101 > 9-13 bit Error (Correctable Error for 14/16bit ecc-mode) 0110 > 14-bit Error (Correctable Error for 14/16bit ecc-mode) 0111 > 15-bit Error (Correctable Error for 16bit ecc-mode) 1000 > 16-bit Error (Correctable Error for 16bit ecc-mode) 1111 > Uncorrectable error.
7–4 NOBER1	Number Of bit Errors for second 528/538/548 bytes section. This field shows the number of bit errors in 512 bytes main area plus 16/26/36 bytes spare (totally 528/538/548 bytes) that were detected/corrected as a result of the BCH ECC check upon read. others > Reserved 0000 No error 0001 >1-3 bit Error (Correctable Error) 0010 > 4-bit Error (Correctable Error) 0011 > 5-7 bit Error (Correctable Error for 8/14/16bit ecc-mode) 0100 > 8-bit Error (Correctable Error for 8/14/16bit ecc-mode) 0101 > 9-13 bit Error (Correctable Error for 14/16bit ecc-mode) 0110 > 14-bit Error (Correctable Error for 14/16bit ecc-mode) 0111 > 15-bit Error (Correctable Error for 16bit ecc-mode) 1000 > 16-bit Error (Correctable Error for 16bit ecc-mode) 1111 > Uncorrectable error.
3–0 NOBER0	Number Of bit Errors for first 528/538/548 bytes section. This field shows the number of bit errors in 512 bytes main area plus 16/26/36 bytes spare (totally 528/538/548 bytes) that were detected/corrected as a result of the BCH ECC check upon read. others > Reserved 0000 No error 0001 >1-3-bit Error (Correctable Error) 0010 > 4-bit Error (Correctable Error) 0011 > 5-7 bit Error (Correctable Error for 8/14/16bit ecc-mode) 0100 > 8-bit Error (Correctable Error for 8/14/16bit ecc-mode) 0101 > 9-13 bit Error (Correctable Error for 14/16bit ecc-mode) 0110 > 14-bit Error (Correctable Error for 14/16bit ecc-mode) 0111 > 15-bit Error (Correctable Error for 16bit ecc-mode) 1000 > 16-bit Error (Correctable Error for 16bit ecc-mode) 1111 > Uncorrectable error.

51.8.16 status sum (NFC_STATUS_SUM)

This register shows a summary status of the NAND device as well as a summary result of the ECC check after a page read.

Address: NFC_STATUS_SUM is F7FF_0000h base + 1E3Ch offset = F7FF_1E3Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-																ECC_SUM								NAND_STATUS_SUM							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_STATUS_SUM field descriptions

Field	Description
31–16 -	Reserved
15–8 ECC_SUM	Ecc Summary report. reports correctable/non-correctable ECC result after a page read for each NAND device. Every NAND device has a bit in this field that indicates whether the last page read from this device had any uncorrectable ECC sections. (bit 0 to Nand0, bit 1 to Nand1, and so on). During a page read, NFC checks for error-bits in each section of the page (for further information of the ECC algorithm refer to ECC Normal Operation . If any of the sections had an uncorrectable ECC result, then the corresponding bit will be set. If all sectors of the page had correctable ECC errors or no error at all, then the corresponding bit will not be set. If NFC reports an uncorrectable ECC status, the user must clear this field manually.
7–0 NAND_ STATUS_SUM	NAND Status Summary. Reports the status of all NAND devices. After NFC executes a status-read operation, as part of the automatic operations (Not status-read of atomic operation (FDO)) to any NAND device, it stores the pass/fail indication of the status register in this field. Each NAND device has 1 bit that stores its status. (bit 0 to Nand0, bit 1 to Nand1, etc.). This field is being cleared at any write to LAUNCH_NFC register, thus if an automatic operation was carried more than once to a specific CS, then this field will indicate if any of the operations resulted in an error by the NAND device.

51.8.17 Initiate an NFC operation (NFC_LAUNCH_NFC)

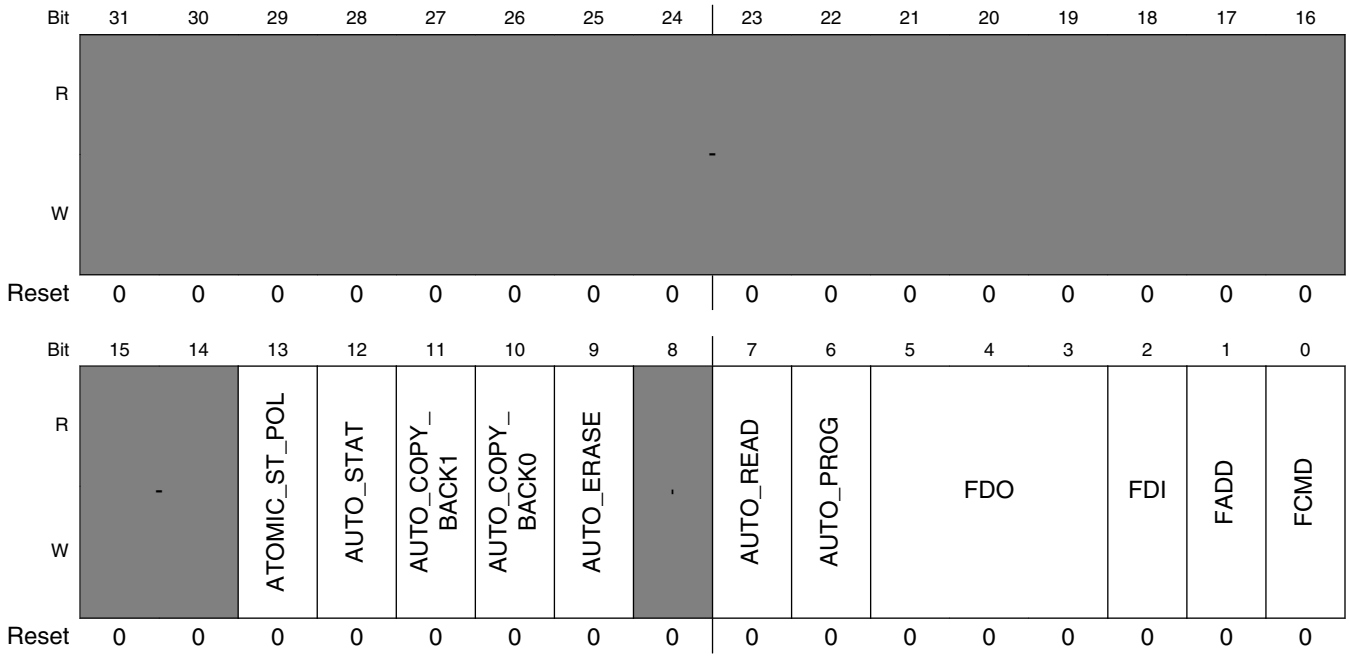
Address of block memory in the NAND Flash that is unlocked in Write Protection mode.

NOTE

When the basic operation is completed, this register is self cleared.

.

Address: NFC_LAUNCH_NFC is F7FF_0000h base + 1E40h offset = F7FF_1E40h



NFC_LAUNCH_NFC field descriptions

Field	Description
31–14 -	Reserved
13 ATOMIC_ST_POL	Atomic Read Status Polling. This bit should be set in Atomic operation when RBB_MODE is set to "0" and when the NAND device is expected to get into busy period. When this bit is set then the NFC will issue a read status command and then issue 1 pulse of RE_B as long as the NAND device is busy (read status polling), When the NAND device will be ready then the NFC will issue an interrupt and clear that bit. For example in read operation from a device that doesn't require a read confirmation command then after the last address cycle the NAND device is expected to get into busy period. Therefore this bit should be set together with the last address cycle and afterwards a read command should be reissued before reading the data in order to read from the data buffer and not from the status buffer of the NAND. So in that example the sequence will be as following: <read command 0x0> <address0>...<addressN + set ATOMIC_ST_POL> <read command 0x0> <read data>. For that example if the device requires a read confirmation then the ATOMIC_ST_POL will have to be set set together with read confirmation command, So in that example the sequence will be as following: <read command 0x0> <address0>...<addressN> <read confirmation command 0x30 + set ATOMIC_ST_POL> <read command 0x0> <read data>. In any automatic operation this bit should be "0" as the whole sequence is executed automatically.ALso when RBB_MODE is set to "1" this bit should be set to "0". For further information refer to Atomic Operations Sequence 0 Do not execute a status read operation. 1 Execute a status read operation.
12 AUTO_STAT	Automatic Status Read operation. When this bit is set, NFC will automatically perform a status read operation: send status read command that is defined in ST_CMD in NFC_CONFIGURATION2, and then issues 1 pulse of RE_B. The status result will be stored in NF_STATUS field. The summary of the status will be recorded in NAND_STATUS_SUM field.

Table continues on the next page...

NFC_LAUNCH_NFC field descriptions (continued)

Field	Description
	The status-read operation will be executed to the NAND device defined by ACTIVE_CS. The status-read operation will be executed only once, and it will be not affected from NUM_OF_ITERATION field. 0 Don't execute a status read operation 1 Execute a status read operation.
11 AUTO_COPY_BACK1	Auto copy back sequence to multiple NAND devices. When this bit is set and interleaved mode is configured (i.e NUM_OF_ITERATION and NUM_OF_DEVICES are greater than 0), NFC will execute a page copy back sequence in interleaved mode as following: <command phase> < address phases> <command phase> <command phase> < address phases> <command phase>. A read status at the end of the sequence is executed only when RBB_MODE is set to "0". For further information refer to AutoMatic Copy-back Operation The commands will be issued from NAND_CMD0, NAND_CMD1, NAND_CMD2 and NAND_CMD3. The source address will be taken from the address group0 and the target address will be taken from address group1 0 Don't execute a page copy back. 1 Execute a page copy back
10 AUTO_COPY_BACK0	Auto copy back sequence. When this bit is set, NFC will execute a page copy back sequence: <command phase> < address phases> <command phase> <command phase> < address phases> <command phase>. A read status at the end of the sequence is executed only when RBB_MODE is set to "0". The first command phase will be taken from NAND_CMD0, the second command phase will be taken from NAND_CMD1 the third command phase will be taken from NAND_CMD2 and the last command phase will be taken from NAND_CMD3. In case of ADD_OP other than 01 then the source address will be taken from the address group0 and the target address will be taken from address group1. In case ADD_OP is set to "01" then the NFC will use only address group0 and copy the page to address group0+1 no matter what is address group1. If the user wish to use a device other than device #0 for this operation, then ADD_OP 2'b01 or 2'b11 must be configured. For further information refer to AutoMatic Copy-back Operation 0 Don't execute a page copy back. 1 Execute a page copy back
9 AUTO_ERASE	Auto erase sequence. When this bit is set, NFC will execute a full block erase sequence: <command phase> < address phases> <command phase>. A read status at the end of the sequence is executed only when RBB_MODE is set to "0". The commands will be issued from NAND_CMD0 and NAND_CMD1. The address will be taken from the active address group. For further information refer to Automatic erase operation 0 Don't execute a full block erase. 1 Execute a full block erase
8 -	Reserved
7 AUTO_READ	Auto read sequence. When this bit is set, NFC will execute a full page-read sequence: <command phase> < address phases> <command phase> <busy polling> <data transfer><read status>. A read status at the end of the sequence is executed when both RBB_MODE is set to "0" or "1".

Table continues on the next page...

NFC_LAUNCH_NFC field descriptions (continued)

Field	Description
	The commands will be issued from NAND_CMD0 and NAND_CMD1. The address will be taken from the active address group. If NUM_OF_ITERATIONS is greater than 0, then NFC will continue reading the next page until NUM_OF_ITERATIONS is reached. NFC will monitor the AXI address in order to decide when next page read should start. For further information refer to FMP and to Automatic Read Operation 0 Don't execute a full page read. 1 Execute a full page read
6 AUTO_PROG	Auto program sequence. When this bit is set, NFC will execute a full page-program sequence: <command phase> < address phases> <data transfer> <command phase>. A read status at the end of the sequence is executed only when RBB_MODE is set to "0". The commands will be issued from NAND_CMD0 and NAND_CMD1. The address will be taken from the active address group. If NUM_OF_ITERATIONS is greater than 0, then NFC will start programming the next page until NUM_OF_ITERATIONS is reached. NFC will monitor the AXI address in order to decide when next page program should start. For further information refer to FPM and to Automatic program operation 0 Do not execute a full page program. 1 Execute a full page program
5–3 FDO	NAND atomic Flash Data Output. This bit enables NAND Flash Data Output. Other options are reserved 001 One page data out. For further information see NAND Flash Atomic Data Output Operation 010 NAND Flash ID data out. Toggles 6 times the read enable signal (re) and store the returned data at the main buffer according to RBA. For further information refer to 51.12.2.6/51-72 100 NAND Flash Status Register data out. This operation toggle once the read enable signal (re) and store the status in NF_STATUS. For further information refer to 51.12.2.7/51-74
2 FDI	NAND Flash atomic Data Input. This field enables NAND Flash Data Input. For further information see NAND Flash Atomic Data Input Operation 0 No NAND Flash data input operation 1 Enable NAND Flash data input operation
1 FADD	NAND Flash atomic Address Input. This field enables NAND Flash Address Input. For further information see NAND Flash Atomic Address Input Operation 0 No NAND Flash Address input operation 1 Enable NAND Flash Address input operation
0 FCMD	NAND Flash atomic Command Input. This field enables the NAND Flash Command Input. For further information see NAND Flash Atomic Command Input Operation 0 No NAND Flash Command input operation 1 Allow NAND Flash Command input operation

51.9 Programmable Registers

NFC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FD_B000	NAND Flash Write Protection (NFC_WR_PROTECT)	32	R/W	0000_0000h	51.9.1/ 3599
63FD_B024	NFC Operation Configuration2 (NFC_CONFIGURATION2)	32	R/W	0040_223Dh	51.9.2/ 3600
63FD_B028	NFC Operation Configuration3 (NFC_CONFIGURATION3)	32	R/W	0002_0600h	51.9.3/ 3603
63FD_B02C	NFC IP Control (NFC_IPC)	32	R/W	0000_0000h	51.9.4/ 3608
63FD_B030	AXI error address (NFC_AXI_ERR_ADD)	32	R	0000_0000h	51.9.5/ 3610
63FD_B034	Delay line parameters (NFC_DELAY_LINE)	32	R/W	0000_0080h	51.9.6/ 3610

51.9.1 NAND Flash Write Protection (NFC_WR_PROTECT)

This register specifies the RAM protection level.

Address: NFC_WR_PROTECT is 63FD_B000h base + 0h offset = 63FD_B000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								BLS		0					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_WR_PROTECT field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–6 BLS	Buffer Lock Set. This field specifies the buffer lock status of first 2KB in the internal buffer (first 2KByte of main data together with 256Bytes of spare data). The other 2KB are always Unlocked. The lock refers only to NAND Accesses. AXI can access all the internal RAM regardless of this field. Any NAND read command, to the first 2KB of the internal RAM, would take place, but, would not be written to the internal RAM (Ecc result on such a read is not valid).(for more details see section 11.10.4/11-46) 00 Locked 01 Locked

Table continues on the next page...

NFC_WR_PROTECT field descriptions (continued)

Field	Description
	10 Unlocked (default) 11 Locked
5–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

51.9.2 NFC Operation Configuration2 (NFC_CONFIGURATION2)

This register is a configuration register for the NAND Flash device to control the ECC-Enable or Disable, Mask Interrupt, endianness, and so on.

.

Address: NFC_CONFIGURATION2 is 63FD_B000h base + 24h offset = 63FD_B024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ST_CMD								SPAS							
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INT_MSK	AUTO_PROG_DONE_MSK	NUM_ADR_PHASES1		EDC		PPB		ECC_MODE		NUM_ADR_PHASES0	NUM_CMD_PHASES	ECC_EN	SYM	PS	
Reset	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0	1

NFC_CONFIGURATION2 field descriptions

Field	Description
31–24 ST_CMD	Status Command. This is the command NFC will use when executing automatic status-read operations. For most NAND devices this value should be 8'h70
23–16 SPAS	Spare Area Size. This field specifies the size of the spare area of the NAND device.This size is in half words and refers to a full page. For example when working with a NAND device with page of 2KB + 64B then spare area is 64B so the SPAS should be set to 8'h20. Note: In case ECC is enabled then the spare area size must be less or equal 218B.
15 INT_MSK	Mask interrupt Bit. This bit enables the interrupt by masking or not masking the interrupt bit. 0 Mask interrupt is disabled (interrupt enabled) 1 Mask interrupt is enabled (interrupt disabled)

Table continues on the next page...

NFC_CONFIGURATION2 field descriptions (continued)

Field	Description
14 AUTO_PROG_ DONE_MSK	Auto Program Done Mask. This bit enables the auto_prog_done by masking or not masking the AUTO_PROG_DONE bit. 0 Mask AUTO_PROG_DONE is disabled (auto_prog_done enabled) 1 Mask AUTO_PROG_DONE is enabled (auto_prog_done disabled)
13–12 NUM_ADR_ PHASES1	Number Of Address Phases1. he number of address phases need to be executed to the NAND Device during read/program operations. 00 3 phases 01 4 phases 10 5 phases 11 6 phases
11–10 EDC	Extra dead cycles. This field specifies number of dead cycles after a read operation, before nfc will allow any other slave to use the shared IO. NFC will delay its ack_en signal according to this field. This will occur any time the NFC stops reading, whether its during a read (because of EIM request), at the end of a read, or at the end of any read operation. The purpose of this field is to prevent contention on the shared IO when using memories with slow data-to-high-z response (large tRHZ field). 00 EDC disabled 01 2 clocks 10 4 clocks 11 6 clocks.
9–8 PPB	Pages Per Block. Indicates how many pages are in 1 Block of the Nand Flash. PPB is used to: - Calculate the next block to erase in AUTO_ERASE. - Calculate in ADD_OP=1/3 where to start the address of the block after extracting the lsb for CS. 00 32 pages per block 01 64 pages per block 10 128 pages per block (default value) 11 256 pages per block
7–6 ECC_MODE	ECC_MODE. This field selects the error correction capabilities, either 4bits correction, 8bits correction, 14bits correction or 16bits correction. In 4bits ECC_MODE the NFC uses 16B of spare area for every 512B data section of the Nand device. (8B for user specific applications and 8B for ECC). In 8bits ECC_MODE the NFC uses 26B of spare area for every 512B data section of the Nand device (12B for user specific application, 14B for ECC). In 14bits ECC_MODE the NFC uses 26B of spare area for every 512B data section of the Nand device (2B for user specific application, 24B for ECC). In 16bit ECC_MODE the NFC uses either 26B or 36B of spare area for every 512B data section of the Nand device. If the spare area has 36B of spare area available for the current main-data section then NFC will use 10B for user specific application and 26B for ECC. If the spare has less than 36B, then NFC will use 26B for ECC, and no Bytes for user-reserved. Note: - In order to use 8/14/16bits ECC_MODE it is needed to work with a NAND device that has at least 26B of spare area for every 512B data section.

Table continues on the next page...

NFC_CONFIGURATION2 field descriptions (continued)

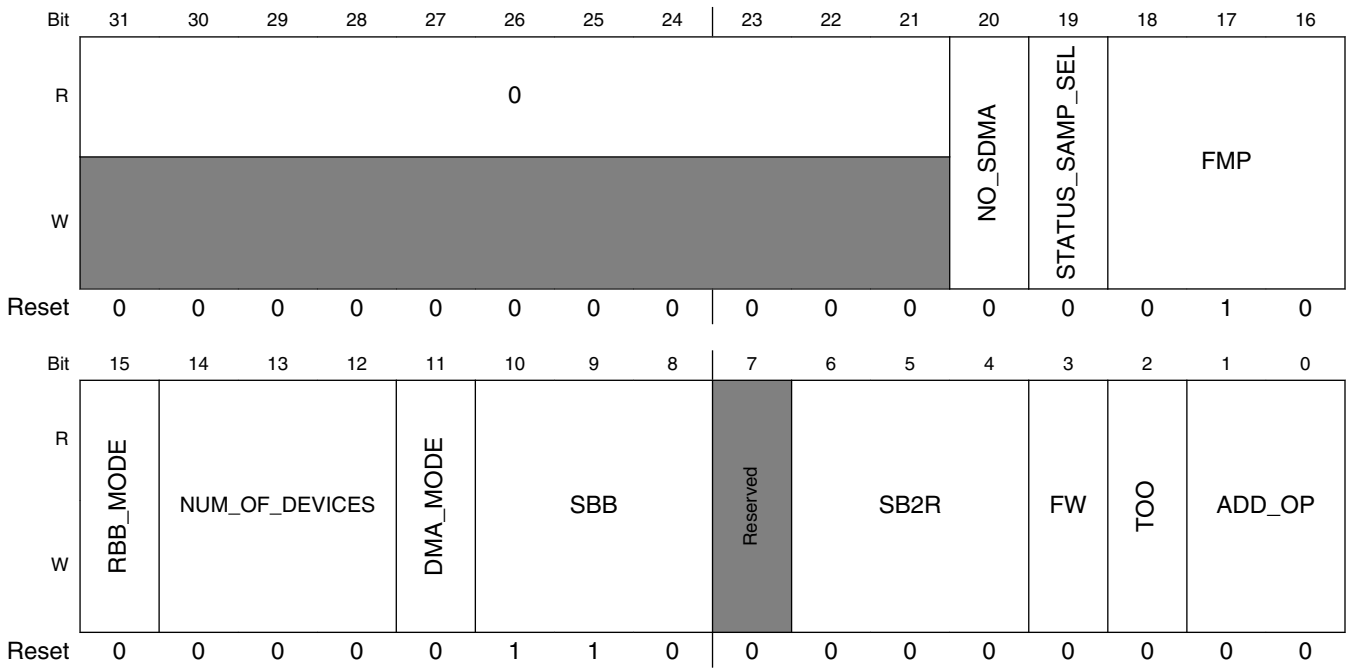
Field	Description
	- In case of working with a NAND device with page size of 4K+ spare of 218B with ECC_MODE of 8/14bits then the last 10B of the last spare buffer (i.e AXI_BASE+0x11DA-AXI_BASE+0x11E3) are not ECC protected. Moreover, in case of 4bits ECC_MODE then the SPAS should be configured to 0x40 (i.e spare of 128B) and the NFC will use only the first 16B (8B for user-reserved and 8B for ECC) for every section of 512B. That means that the NFC will use only 128B of spare out of the 218B available. In 16 bits ECC_MODE all the bytes of the main and spare buffers are ECC protected. For further information of the ECC algorithm refer to ECC Operation. 00 4bit error correction (default value). 01 8bit error correction. 10 14bit error correction 11 16bit error correction.
5 NUM_ADR_PHASES0	Number Of Address Phases0. The number of address phases to be executed to the NAND Device during an AUTO_ERASE operation 0 1 phase less then NUM_ADR_PHASES1 1 2 phases less then NUM_ADR_PHASES1
4 NUM_CMD_PHASES	Number Of Command Phases. The number of command phases to execute to the NAND Device during AUTO_READ operation(i.e, is read-confirm command is needed or not) 0 1 command phase for reading a page 1 2 command phases for reading a page (read command & read-confirm command)
3 ECC_EN	ECC operation Enable. This bit determines whether ECC operation is executed or bypassed. When ECC is bypassed then, in a program operation, the NFC will drive the relevant data sections and the whole data of the associated spare buffers. When ECC is enabled then, in a program operation, the NFC will override the ECC locations in the spare buffers and drive the calculated ECC bytes instead. The default is ECC enable. 0 ECC operation is bypassed. 1 ECC operation is executed.
2 SYM	Symmetric mode. This bit controls the speed of access as well as RE# waveform during read. 0 Two flash clock cycles per access of RE# or WE#, (asymmetric RE waveform). For further information refer to Symmetric/Asymmetric Mode Operation. 1 One flash clock cycle per access of RE# or WE#, (symmetric RE waveform). (default)
1–0 PS	Page Size. Determines the NAND device page size. 00 1/2KB page 01 2KB page 10 4KB page 11 4KB page

51.9.3 NFC Operation Configuration3 (NFC_CONFIGURATION3)

This register is a configuration register for the NAND Flash device to control the way address is extracted, number of nand devices in the system, endianness, and so on.

.

Address: NFC_CONFIGURATION3 is 63FD_B000h base + 28h offset = 63FD_B028h



NFC_CONFIGURATION3 field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero. Reserved
20 NO_SDMA	NO_SDMA. When this bit is set, then the system is transferring data to/from the RAM by a different master then the SDMA. Furthermore, it means that dma_wr_req and dma_rd_req signals, are not being used. When SDMA is being used to transfer data to/from the RAM, then it starts each transfer upon assertion of dma_wr_req or dma_rd_req. This activates FMP protection mechanism, and prevents RBA from automatically incrementing. When not using SDMA, then the system must first copy data to the RAM (for program), and only then set the AUTO_PROG bit of the NFC. NFC will automatically increment RBA to point to the next section in the RAM, so if the system wants to program more than 1 page (using NUM_OF_ITERATIONS field) it can do it up to a full use of the RAM. During read, NFC will write the page to the RAM, and again will automatically increment RBA, so if NUM_OF_ITERATIONS is not 4'h0, NFC will automatically start another read sequence and will write it to the next slot in the RAM.

Table continues on the next page...

NFC_CONFIGURATION3 field descriptions (continued)

Field	Description
	NFC will not stop after the RAM was fully used. It is the system responsibility to configure NUM_OF_ITERATIONS field to prevent RAM overflow/underflow. NOTE: Setting this bit disables FMP protection mechanism, so the system must fill the RAM with data before setting AUTO_PROG bit. 0 SDMA is responsible for data transfer to/from NFC RAM. 1 SDMA is not in being used for NFC operations.
19 STATUS_ SAMP_SEL	Status Sample Select. This bit determines when to sample the read status. For further explanation refer to Delay Line Operation 0 Read status is sampled at the next posedge after the read data is sampled. 1 Read status is sampled at the same sampling edge as read data.
18–16 FMP	Fifo Mode Protect. In order to prevent underrun/overrun of the RAM data, the NFC uses a protection mechanism in which the NFC monitors the AXI-RAM addresses and the NAND-RAM addresses and maintains a safety buffer in which the AXI must be ahead of the NAND operation. That mode can be operated only when NUM_OF_ITERATIONS is greater than 0 and when SDMA is used (i.e NO_SDMA = 0). When the FMP is smaller than PS then the system part and the NAND part can access simultaneously to the internal RAM in a way that when the difference between the AXI-RAM address and the NAND-RAM address is smaller than FMP then the NFC will pause transferring data to/from the NAND device till the difference will be larger than the FMP. This is the system responsibility to make sure that only 1 master is accessing the NFC during such operations otherwise other masters might "confuse" this protection mechanism. This protection mechanism is protecting only the main area of the RAM, so any system operations must access the spare area prior to the main area. (During program, the system should first copy the spare area data into the RAM and only then copy the main data. During read, the system should first read the spare area, and only then read the main data) Note: In read operation the FMP can be set to one of the options below, but in write operation it must be set to PS (page size). 000 Disable protection buffer. 001 64B 010 128B (default) 011 256B 100 512B 101 1KB (Not to be used with 1/2KB devices) 110 2KB (Not to be used with 1/2KB devices) 111 4KB (To be used only with 4KB devices)
15 RBB_MODE	Ready-Busy mode. This bit determines how NFC monitors the ready-busy signals of the NAND devices during both automatic and atomic operations. If this bit is set, NFC monitors the ipp_nfc_rbbX signals (one signal for each device), that are connected directly from the NAND devices. If this bit is cleared, NFC will perform status-read operations to determine the NAND device status.

Table continues on the next page...

NFC_CONFIGURATION3 field descriptions (continued)

Field	Description
	When RBB_MODE is set to "0" and either atomic or automatic operation is executed then the user must connect all ipp_nfc_rbbx signals to "1" (ready). When RBB_MODE is set to "1" then non active ipp_nfc_rbbx signals must also be connected to "1". Note: In Atomic operation with RBB_MODE=0 it is the user responsibility to set ATOMIC_ST_POL together with the operation (FADD,FCMD,FDO or FDI) that will enter the device into a busy state. This in order to guarantee that an interrupt will be issued only when the device returned to ready. For further information refer to Atomic Operations Sequence. 0 NFC monitors ready-busy status by doing a status-read command (default) 1 NFC monitors ready-busy status by checking ipp_nfc_rbbX signals
14–12 NUM_OF_ DEVICES	Number Of Devices. Defines how many NAND devices are connected to the NFC. 000 1 NAND device 001 2 NAND devices 010 3 NAND devices 011 4 NAND devices 100 5 NAND devices 101 6 NAND devices 110 7 NAND devices 111 8 NAND devices
11 DMA_MODE	Dma Mode. This bit determines the functionality of the dma_req signals (dma_rd_req and dma_wr_req). If this bit is set, then NFC will output only 1 dma request signal on dma_rd_req which is a logical OR of dma_rd_req & dma_wr_req. The system can determine the request type by reading DMA_STATUS register. If this bit is used, then the system must clear DMA_STATUS field after reading it. 0 NFC outputs 2 dma request signals 1 NFC outputs 1 dma signal which is the logical OR of the 2 dma requests.
10–8 SBB	Status Busy Bit. Indicates which bit in the NAND status register is the Ready/Busy indication. For example: In Samsung's K9K8GU0M bit 6 of the status register is the Ready/Busy indication.
7 Reserved	This field is reserved. Reserved
6–4 SB2R	Status bit to Record. Indicates which bit in the NAND status register is the Pass/Fail indication. For example: In Samsung's K9K8GU0M bit 0 of the status register is the Pass/Fail indication.
3 FW	Flash Width. Determines the NAND Flash IO width (x8/x16). 0 x16 1 x8
2 TOO	Two On One. When this bit is set, then NFC assumes there are 2 x8 NAND devices connected to every CS line (One device on the lower byte of ipp_nfc_read_data_in/ipp_nfc_write_data_out, and one device on the upper byte). When using 2 devices on one CS line, then NFC should be configured as if an x16 device is connected to it in order to use all data lines and the page size (PS) equals the page size of a single deice.

Table continues on the next page...

NFC_CONFIGURATION3 field descriptions (continued)

Field	Description
	If 2 devices of 1/2KB page are connected in this manner, then each automatic page-program will transfer 1KB of data from the NFC's RAM to the NAND devices, so RBA field must be 1KB aligned (i.e set to 3'h0, 3'h2, 3'h4 or 3'h6) and PS should be configured to 00. In atomic operations, NFC transfers only 1/2KB data each time REG_FDI is set, so its the user's responsibility to update RBA field and to set FDI again. If 2 devices of 2KB page are connected in this manner, then each automatic page-program will transfer 4KB of data from the NFC's RAM to the NAND devices, so RBA field must be 4KB aligned (i.e set to 3'h0) and PS should be configured to 01. In atomic operations, NFC transfers only 2KB data each time FDI is set, so its the user's responsibility to update RBA field and to set FDI again. **NOTE:** • TOO mode can be activated only when NUM_OF_ITERATIONS is set to "0" (i.e non interleaved mode). • If 2 devices of 4KB page are connected in this manner, then the user must not use automatic operations since the RAM can not hold 8KB data. Only atomic operations are permitted in this mode. 0 Only 1 device is connected to all CS lines. 1 2 NAND devices are connected to each active CS line.
1–0 ADD_OP	Addressing Options. This field defines how NFC handles the addressing of the pages/blocks during an automatic operations. 00 NFC will use only address_group0 (i.e NAND_ADD0,NAND_ADD8) for all the relevant devices. So in interleaved mode (i.e NUM_OF_ITERATIONS and NUM_OF_DEVICES are greater than 0) then the NFC will act as following: First operation will be sent to CS0 and NFC will drive the address from address group 0 Second operation will be sent to CS1 and NFC will drive the address from address group 0 Third operation will be sent to CS2 and NFC will drive the address from address group 0 and so on until NUM_OF_ITERATIONS is reached. If NUM_OF_ITERATIONS is greater than the number of NAND devices (NUM_OF_DEVICES) connected to the NFC then after accessing the last device the NFC will increment address group 0 (For page operations NFC will use the same block address, and will increment the page address. for block operations(erase), NFC will increment the block address) and return in a loop to access CS0, then CS1 and so on until NUM_OF_ITERATIONS is reached. For example in Erase: if 4 devices are connected to NFC and PPB is 10 (128 pages per block) and we wish to erase the 5th block (block number 4) in each device (total of 4 iterations (i.e NUM_OF_ITERATIONS=3)), then address group 0 supposes to be 48'h200 (7 lsb are used for the page and the rest are used for the block). Another example in Erase: if 4 devices are connected to NFC and PPB is 10 (128 pages per block) and we wish to erase the 5th,6th blocks (block number 4,5) in each device (total of 8 iterations (i.e NUM_OF_ITERATIONS=7)), then address group 0 supposes to be 48'h200 (7 lsb are used for the page and the rest are used for the block) for the first loop and in the second loop (5th, 6th, 7th, 8th iterations) the NFC will increment the address to 48'h280. For example in Program/Read: if 4 devices are connected to NFC and PPB is 10 (128 pages per block) and we wish to program/read to/from the 2nd page (page number 1) of the 5th block (block number 4) in each device (total of 4 iterations (i.e NUM_OF_ITERATIONS=3)), then address group 0 supposes to be 48'h2010000 (16 lsb are used for the column, 7 bits are used for the page and the rest are used for the block). 01 NFC will use only address group 0 (i.e NAND_ADD0,NAND_ADD8) and will only use one NAND device as following:

Table continues on the next page...

NFC_CONFIGURATION3 field descriptions

Field	Description
	The NFC will use the lowest bits of the NAND_ADD0 as the CS for the operations according to NUM_OF_DEVICES. In case of 1 or 2 devices then the lsb (of the page) is used for the CS. In case of 4 devices the 2 lsb (of the page) are used for the CS and in case of 8 devices then the 3 lsb (of the page) are used for CS. Those lowest bits will be removed from the address that will be driven to the NAND device. Moreover for CS calculations, NFC ignores the column address and leaves it as it is. For example: Assume 4 devices are connected to the NFC, erasing block 0 of device #0 requires writing 48'h0 to address group 0 register. Erasing block 0 of device #1 requires writing 48'h1 to NFC_ADD0 register. Assume 2KB page devices: Programming page 0 of device #1 requires writing 48'h10000 to NFC_ADD register (16 lower bits are column address and don't take place in CS-Address extraction) If NUM_OF_ITERATIONS is greater than "0" then the NFC will increment the address group 0 every iteration (For page operations NFC will use the same block address, and will increment the page address. for block operations(erase), NFC will increment the block address) and access the same device (according to the extracted CS) until NUM_OF_ITERATIONS is reached. For example in Erase: if 4 devices are connected to NFC and PPB is 10 (128 pages per block) and we wish to make two erase iterations (i.e NUM_OF_ITERATIONS=1) of 4th.5th blocks (block number 3,4) of device3 (CS3) then address group 0 supposes to be 48'h603 (2 lsb are used for the CS then 7 bits are used for the page and the rest are used for the block). In the 2nd iteration the address will be incremented to 48'h803 (increment block by 1). For example in Program/Read: if 4 devices are connected to NFC and PPB is 10 (128 pages per block) and we wish to make 3 program/read iterations (i.e NUM_OF_ITERATIONS=2) to/from 2nd, 3rd,4th page (page number 1,2,3) of the 2nd block (block number 1) of device3 (CS3) then address group 0 supposes to be 48'h2070000 (16 lsb are used for the column, 2 lsb are used for the CS, 7 bits are used for the page and the rest are used for the block). In the 2nd iteration the address will be incremented to 48'h20B0000 (increment page by 1) and In the 3rd iteration the address will be incremented to 48'h20F0000 (increment page by 1). Note: - This option can be used only when using 1,2,4 or 8 NAND device. - Only single device will be accessed (No interleaving) even if both NUM_OF_ITERATIONS and NUM_OF_DEVICES are greater than "0"
	10 NFC will use all the address groups. Each device with the associated address group (device0 with address group 0, device1 with address group 1 and so on). So in interleaved mode (i.e NUM_OF_ITERATIONS and NUM_OF_DEVICES are greater than 0) then the NFC will act as following: First operation will be sent to CS0 and NFC will drive the address from address group 0. Second operation will be sent to CS1 and NFC will drive the address from address group 1. Third operation will be sent to CS2 and NFC will drive the address from NAND_ADD3, and so on. If NUM_OF_ITERATIONS is greater than the number of nand devices (NUM_OF_DEVICES) connected to the NFC then after accessing the last device the NFC will increment the associated NAND_ADDx (For page operations NFC will use the same block address, and will increment the page address. for block operations(erase), NFC will increment the block address) and return in a loop to access CS0, then CS1 and so on until NUM_OF_ITERATIONS is reached.
	11 NFC will use all the address groups and also will use the lowest bits of the NAND_ADDx as the CS for the operations. So in interleaved mode (i.e NUM_OF_ITERATIONS and NUM_OF_DEVICES are greater than 0) then the NFC will act as following: First operation will be sent to CSX based on address group 0 (i.e NAND_ADD0, NAND_ADD8). Second operation will be sent to CSY based on address group 1 (i.e NAND_ADD1, NAND_ADD8).

NFC_CONFIGURATION3 field descriptions (continued)

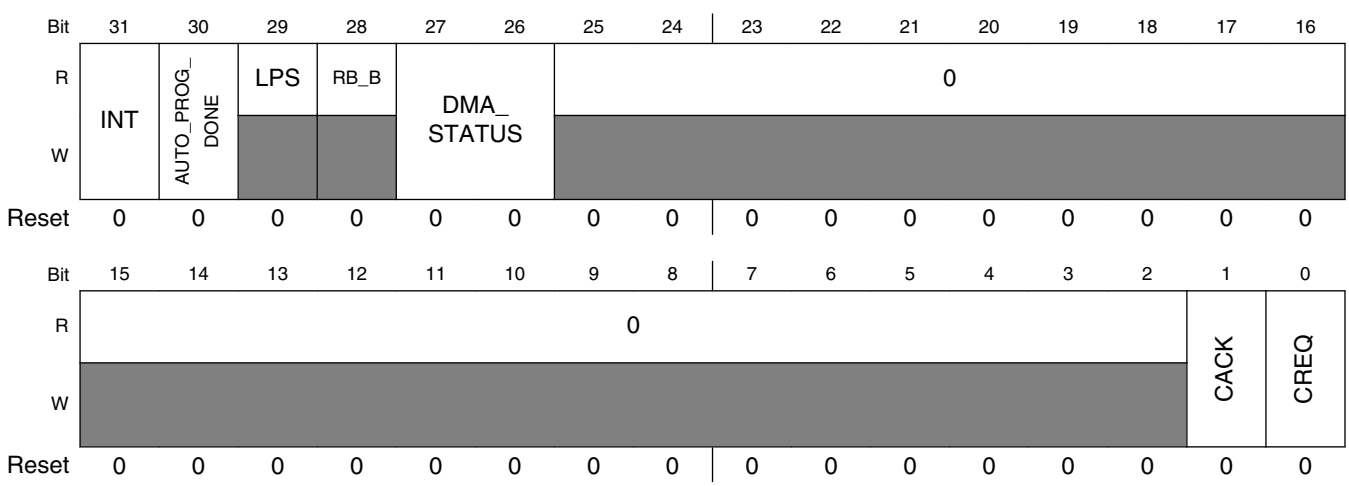
Field	Description
	Third operation will be sent to CSZ based on address group 2 (i.e NAND_ADD2, NAND_ADD9), and so on. In case of 1 or 2 devices then the lsb (of the page) is used for the CS. In case of 4 devices the 2 lsb (of the page) are used for the CS and in case of 8 devices then the 3 lsb (of the page) are used for CS. Those lowest bits will be removed from the address that will be driven to the NAND device. Moreover for CS calculations, NFC ignores the column address and leaves it as it is. For example in Erase: if 4 devices are connected to NFC and PPB is 10 (128 pages per block) and we wish to make 3 erase iterations (i.e NUM_OF_ITERATIONS=2) the first iteration to the 4th block (block number 3) of device3 (CS3), the second iteration to 2nd block (block number 1) of device1(CS1) and the 3rd to the 1st block (block number 0) of device0 (CS0). Then the address for the 1st iteration will be driven from address group 0 that supposes to be 48'h603 (2 lsb will be used for the CS then 7 bits will be used for the page and the rest are used for the block (which is 3)). The address for the second iteration will be driven from address group 1 that supposes to be 48'h201 and the address for the 3rd iteration will be driven from address group 2 that supposes to be 48'h0 Note:This option can be used only when using 1,2,4 or 8 NAND device. In this mode, every address group involved must specify a different CS

51.9.4 NFC IP Control (NFC_IPC)

This register holds the IP write handshake registers as well as the interrupt register.

.

Address: NFC_IPC is 63FD_B000h base + 2Ch offset = 63FD_B02Ch



NFC_IPC field descriptions

Field	Description
31 INT	Interrupt.The output port ipi_int_nfc is a logic AND between this bit and INT_MSK bit. This bit will rise after

Table continues on the next page...

NFC_IPC field descriptions (continued)

Field	Description
	NFC finishes an atomic operation (setting any bit in LAUNCH_NFC[5:0]) or in automatic operation (AUTO_READ/AUTO_PROG/AUTO_ERASE/AUTO_COPY_BACK0/1) it will issue the interrupt only after NUM_OF_ITERATIONS is reached. When interrupt is being set after an automatic operation it means that all status registers from all relevant NAND devices have been read and the status is stored in STATUS_SUM register. This bit is not self cleared, so it must be cleared by the user (by writing "0"). 0 NFC didn't finish its operation. 1 NFC finished its operation and is ready to accept next operation.
30 AUTO_PROG_DONE	Automatic Program Done. Indicates that SDMA finished transferring all NUM_OF_ITERATIONS pages to the NFC. This allows the software to prepare SDMA with the data of next group of pages to be transferred to the NAND devices. This bit is not self cleared, so it must be cleared by the user (by writing "0"). 0 SDMA didn't finish transferring NUM_OF_ITERATIONS pages to NFC. 1 SDMA finished transferring NUM_OF_ITERATIONS pages to NFC.
29 LPS	Low Power Status. This read-only bit reflects the low power mode of the nfc. 0 nfc is not in a low power mode 1 nfc is in a low power mode
28 RB_B	RB_B Status. This read-only bit reflects the RB_B signal of the NAND device. If more than 1 RB_B line are connected to the same port of the RB_Bx of the NFC, then this bit reflects a logical AND of all RB_B signals. 0 NAND device is busy. 1 NAND device is idle.
27–26 DMA_STATUS	Dma Status. This field indicates which dma request signal is asserted. This field is not self cleared, so it must be cleared by the user. 00 All dma request signals are deasserted. 01 dma_wr_req was asserted 10 dma_rd_req was asserted 11 Both dma_rd_req & dma_wr_req were asserted.
25–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1 CACK	IP configuration acknowledge. Whenever this bit is set, IP is permitted to configure NFC's IP registers. 0 Configuration of NFC's registers is forbidden. 1 Configuration of NFC's registers is permitted.
0 CREQ	IP request to configure NFC. Whenever IP wants to configure NFC it must set this bit to one, and poll on CACK until it is high. Only then configuration is permitted. After configuration is done, the user must deassert this bit. This feature prevents configuring the NFC during an operation and causing an unexpected behavior. 0 No request to configure NFC. 1 A request to configure NFC is valid.

51.9.5 AXI error address (NFC_AXI_ERR_ADD)

Address of the first erroneous AXI access.

Address: NFC_AXI_ERR_ADD is 63FD_B000h base + 30h offset = 63FD_B030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																AXI_ERR_ADD															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_AXI_ERR_ADD field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12–0 AXI_ERR_ADD	AXI Error Address. Whenever NFC returns an error response on the AXI interface, it will sample the address in which the error occurred, and store it in this register for debug purposes. NOTE: Only the first error-address will be stored. If second AXI error response will occur, the second error-address will be lost. Reading from this register, will unlock this register, so, second error-address will be sampled.

51.9.6 Delay line parameters (NFC_DELAY_LINE)

The NFC includes 4 delay line units where each unit can adjust the timepoint in which the NFC latches the read data by 1/2 fast clock cycle (aclk) and therefore the 4 delay line units can adjust that timepoint by up to 2 fast clock cycles (aclk)

NFC_DELAY_LINE register represents a single delay line unit. By default the NFC latches the data after a fixed delay of 1 flash_clk cycle + 1 fast clock cycle (aclk).

Writing to NFC_DELAY_LINE register will adjust the fast clock cycle delay.

For further information refer to Delay Line Operation

Address: NFC_DELAY_LINE is 63FD_B000h base + 34h offset = 63FD_B034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																NFC_OFF_DEL								NFC_ABS_DEL							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

NFC_DELAY_LINE field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–8 NFC_OFF_DEL	NFC Delay-line Offset delay. The field represents positive and negative numbers using twos compliment representation. This allows positive and negative offsets from the quarter cycle measured value. For example, to add 3 delay units to the measured delay, this field should be 00000011, to subtract 3 delay units, it will be set to 11111101.
7–0 NFC_ABS_DEL	NFC Delay-line Absolute delay. This field determines the specific delay line absolute delay in fractions of fast cycle terms. The fraction is process and frequency independent. The delay of single delay line would be (NFC_ABS_DEL / 512) * fast_clock. So for the default value of 128 we get a quarter cycle delay.

51.10 Functional Description

This section provides the functional description for the NAND Flash Controller.

51.10.1 Reset

There are several options to reset the NFC:

axi_reset - asserting axi_reset_b will reset all AXI related functionality, such as AXI state machines and axi registers. axi_reset_b resets block registers as well.

cold reset - asserting of ipp_reset_b will reset all the functionality that is responsible to communicate with the NAND include the ECC engine.

warm_reset - this signal is checked during assertion of axi_reset_b, and if it is high, then all configuration registers will not be reset. This will enable a quick return-to-action reset operation.

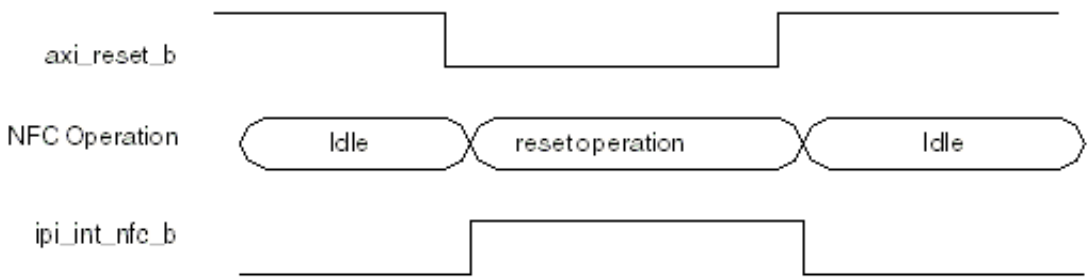


Figure 51-25. Interrupt Functionality During Reset

51.10.2 NAND Flash I/F Control

The NAND Flash I/F Control generates all the following control signals:

CE# (Flash Chip Enable), $\overline{\text{RE\#}}$ (Read Enable for read operations), $\overline{\text{WE\#}}$ (Flash Write Enable), CLE# (Flash Command Latch Enable), ALE (Flash Address Latch Enable). It monitors the RB_B (Flash Ready/Busy indication) or generates read status commands to check if the NAND Flash is in the middle of operation.

Ratio of ipp_flash_clk and aclk should be an integer value and a minimum of 2 (aclk is at least double the frequency of ipp_flash_clk) as long as the clocks are synchronous.

The following will specify the operation of the NFC during program or read:

1/2KB page:

There is a need to configure the address and command sequence before every page of 512B.

Every page is transferred to 1/2KB in the nfc internal RAM buffer so it is important to change RBA register to choose which 1/2KB section of the RAM will be written or read from.

2KB page:

The commands and address cycles occur at the beginning of the program or read sequence. After the command and address sequence, there are four sections of data program/read of 512B each. Every section of 1/2KB is transferred to/from a different 1/2KB in the nfc internal RAM buffer. After command & address cycles are done, upon kicking off a 2K page (writing 16'h4 or 16'h8 to LAUNCH_NFC register), the NFC will transfer the 2K page from/to its internal RAM to/from the NAND device.

Every page is written to 2KB in the nfc internal RAM buffer so it is important to change RBA register to choose which 2KB section of the RAM will be written or read from. Legal values for RBA field are RBA=0 & RBA=4.

4KB page:

The commands and address cycles occur at the beginning of the program or read sequence. After the command and address sequence, there are eight sections of data program/read of 512B each. Every one of the 1/2KB is transferred to/from different 1/2KB in the nfc internal RAM buffer. After command & address cycles are done, upon kicking off a 4KB (writing 16'h4 or 16'h8 to LAUNCH_NFC register), the NFC will transfer the 4KB page from/to its internal RAM from/to the NAND device.

In 4KB page, it is only legal to write 3'b000 to RBA register.

ECC is always calculated for data section of 1/2KB even if the page size is 2KB/4KB.

Figure 51-26, Figure 51-27, and Figure 51-28 show the NAND Flash read, program, and erase timing diagrams.

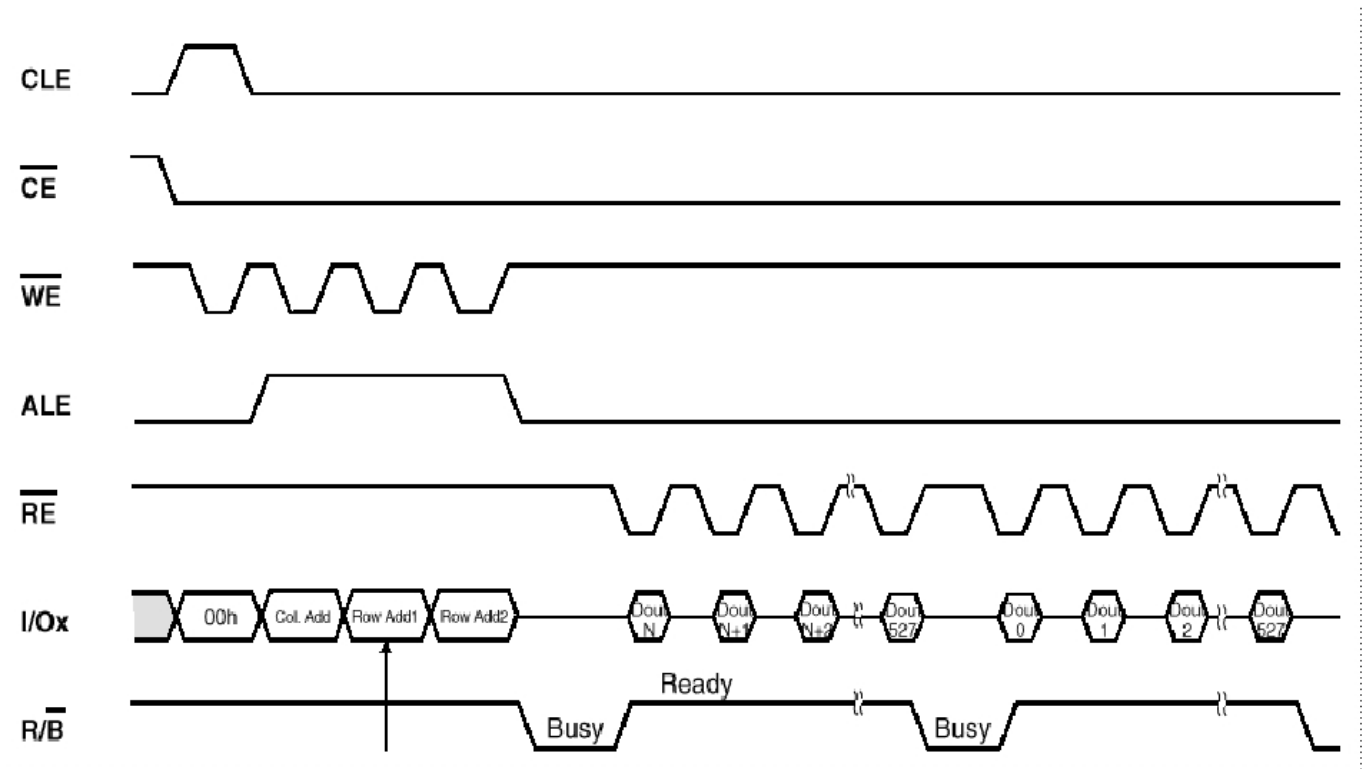


Figure 51-26. Read Operation

ALE can be asserted & deasserted separately for each address phase.

Number of ALE phases is determined by the user.

PAGE PROGRAM OPERATION

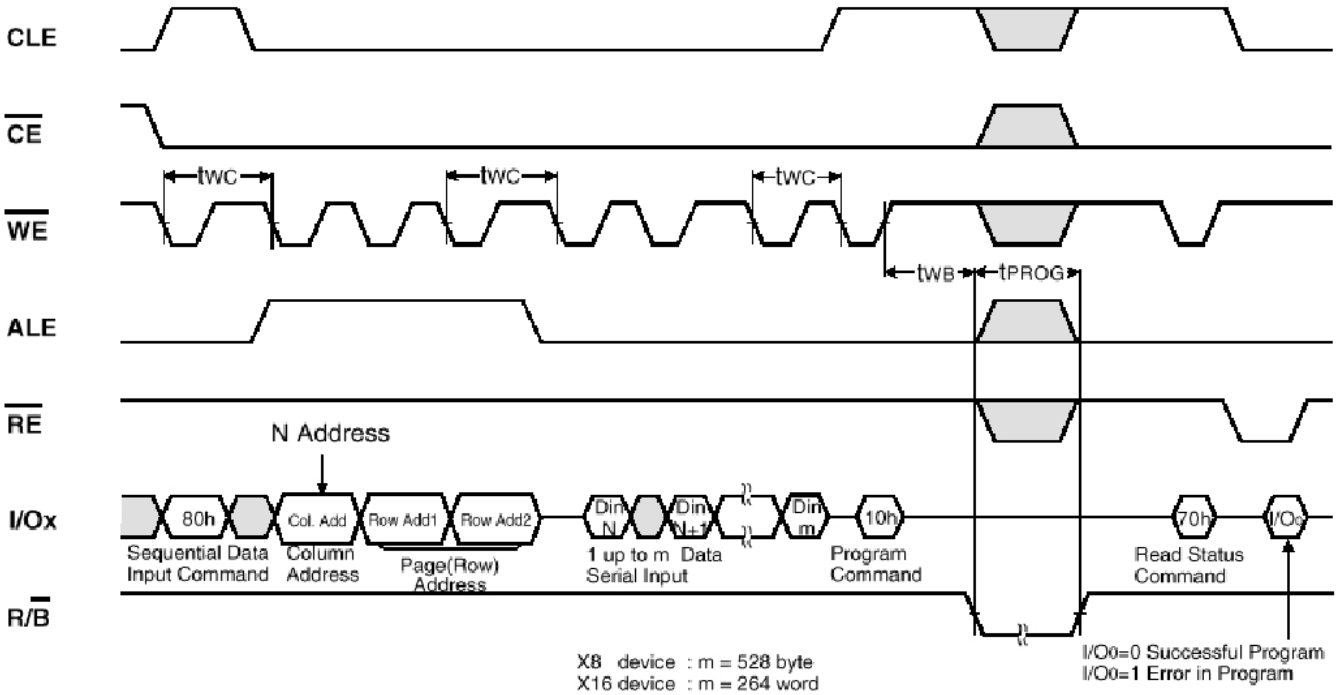


Figure 51-27. Program Operation

BLOCK ERASE OPERATION (ERASE ONE BLOCK)

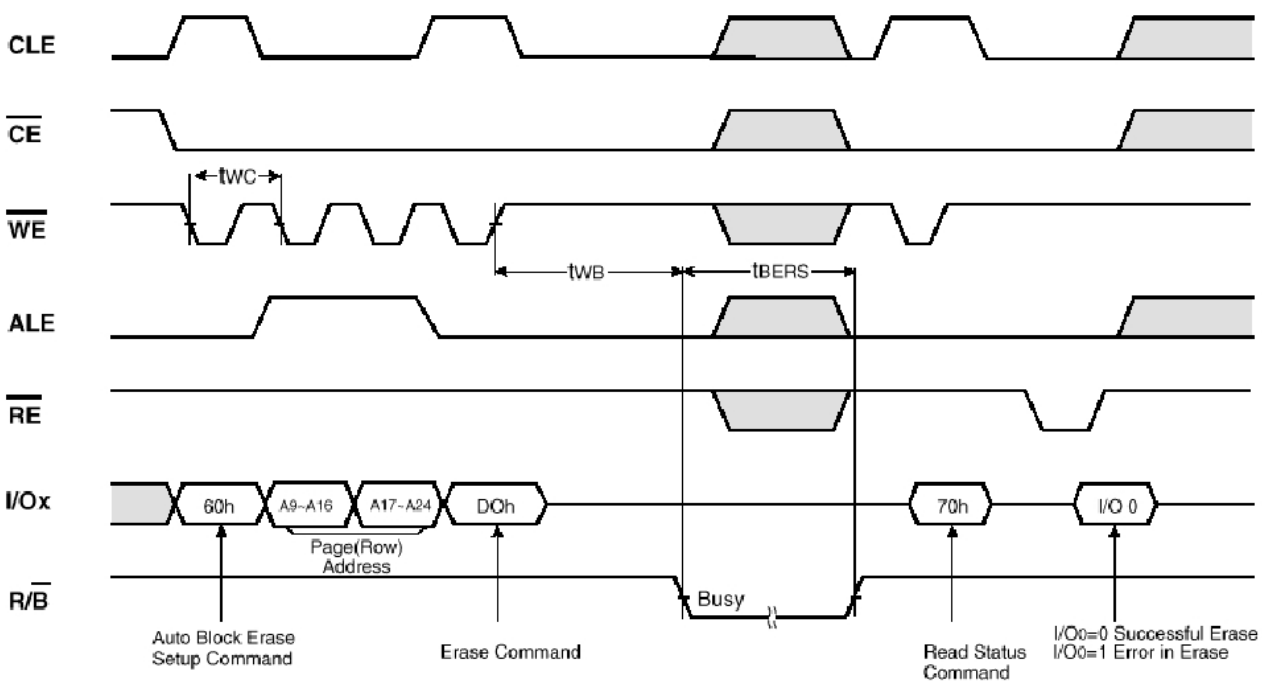


Figure 51-28. Erase Operation

51.10.3 DMA Request Operation

In case NUM_OF_ITERATIONS is greater than "0" then NO_SDMA should be set to "0". In that case the NFC will communicate with the system DMA using two signals as following:

- dma_rd_request-triggers after NFC finished reading a page.
- dma_wr_request-triggers either on write to AUTO_PROG bit, or after NFC finished transferring data from NFC's RAM to the NAND device during an automatic program operation, but only if there are additional pages to program based on NUM_OF_ITERATIONS field.

The configuration bit DMA_MODE selects between a mode of 2 independent dma-request signals, and a combined signal which asserts both on dma_rd_request triggering and on dma_wr_request triggering. This combined dma-request signal is functional on dma_rd_req pin, whereas dma_wr_req pin will be unused in such a mode.

51.10.4 Internal RAM

The internal RAM Buffer is a 4608 Byte single Port RAM buffer. This memory has 1152 words of 32-bits each, where1024 words are used for the main are buffers (data sections) and the remaining 128 words are allotted to spare area buffers, which are used for ECC (Error Correction) and other user-reserved.

Although the RAM is a single port RAM, the NFC holds a mechanism that allows accessing the RAM from two sources at the same time. For example: When NFC reads data from the RAM in order to program it into the NAND device, the host can write to another location in the RAM and prepare the data for the second page program. This mechanism can be achieved since the Flash side is relatively slow, so, there are idle cycles in which the NFC can service the host. Example of a good use for this feature is that the host instead of writing a full page to the NFC, writes only part of the page, and then launches the program operation. Now the host can continue writing the page to the RAM while NFC is reading it out and sends it to the NAND device. Such a use can almost hide the page transfer time from the host to the NFC.

NOTE: There is a limitation for using the dual-port ability of the RAM. The User must not perform AXI read transactions during NFC-program operation, and the user must not perform AXI write transactions during NFC-read operation (If NFC is reading from the RAM - during program, then the user must not read from the RAM also, and while NFC is writing to the RAM, the user must not write to the RAM either.)

When FMP (Fifo Mode Protect field in NFC_CONFIGURATION3 register) is set, the NFC assumes that the host is acting as the example above. NFC will monitor the RAM and makes sure that there is always enough data in the RAM for the NFC to write. i.e if the host is becoming slow when writing the page into the RAM, then the NFC will pause the program operation and waits until the host continues to write the page. Same monitoring is applied in read operations, when the host reads only a part of a read page, and NFC starts reading another page. The NFC may overwrite the previous page before the host can read it - so the NFC monitors the progress of the host and pauses the read operation before an overwrite can occur.

To ensure that this protection mechanism would work properly, the system must make sure that only 1 master is accessing the NFC during the entire operation (for example: SDMA, ARM, etc.).

The NFC logically divides the RAM into 8 sections. Each section contains 512B of main data and 64B of spare area. When reading (or programming) from the NAND device, the NFC writes the main data from the NAND device into the main area, and the spare data from the device is written into the spare section. If the NAND device spare data is less than 64B per 512B main data, then the NFC's spare section in the internal RAM will not be fully used. For example in NAND device that has 2KB main data and 64B spare data, then this device has 16B spare data for every 512B main data, then the spare data in the RAM will be located as following: first 16B of spare data in spare buffer 0,next 16B in spare buffer 1, and so on.

When using a NAND device in which the spare area size is not fully divided by 512B sections, then the remainder will be located at the last spare section. For example in a NAND device with 4KB main data + 218B spare data, then the division will be 27.25B of spare for each 512B of main data. Then the NFC rounds this number down to the nearest even number, which 26B, and that would be the number of Bytes located in each spare buffer. The remaining Bytes will be added to the last section, meaning that the last spare section will contain 26+10=36B.

AXI bus interface

The AXI bus interface is an adapter between ABMA AXI bus and the internal bus.

On the AXI bus side it supports a 16-bit and 32-bit transactions, burst and non burst operations. On the internal bus side it supports 32-bit bus width.

AXI interface supports only INCR accesses (no WRAP nor FIXED), with length of 1-16 data transfers (defined by arlen/awlen of 4 bit).

AXI domain is divided to 2 regions: internal RAM & AXI registers. Any access to AXI registers, must be 32bit wide, with length of 1 data transfer (single access).

In case od reading the reserved area of the AXI address space then the response in unpredictable.

51.10.4.1 LPMD/ DVFS

DVFS is supported using a handshake with the system. The system asserts lpmd signal to request a power down. NFC will respond with assertion of lpack, as soon as it becomes idle from its current operation. lpack assertion signals to the HOST that the NFC is ready for power down.

lpack will be asserted if there is no active axi transaction and there is no active flash accesses.

The NFC holds a register in block domain (LPS) that indicates the status of the lpack signal.

In case the lpmd is asserted during a read operation while NUM_OF_ITERATION is greater than 0 then the NFC will assert the lpack before the assertion of dma_rd_req (DMA read request), Therefore the dma_rd_req will be asserted after returning from DVFS mode.

51.10.4.2 Burst Access Support

Not all AXI burst transactions are supported. Table 51-32 lists NFC supported access burst types.

Table 51-32. NAND Flash Burst Access Support

HBURST	BURST TYPE	SUPPORTED	Description
00	**FIXED**	**No**	**Fixed address burst**
01	**INCR**	**Yes**	**Incrementing-address burst**
10	WRAP	No	Incrementing-address burst that wraps at wrap boundary
11	Reserved	Yes	-

NOTE

NFC supports bursts of 16/32-bit words only. Bursts of byte words (8-bits) are not supported.

51.10.5 Block interface

The block interface is an adapter between block protocol and internal control registers.

NFC's registers are divided between AXI domain & block domain to enable better security and data protection.

Write sequence to block must follow a write handshake protocol:

Whenever the host wants to write to one of the block registers, he must first assert CREQ field. Whenever the NFC will be ready to accept the write, he will assert CACK. Now the write sequence can go on. Upon completion of all write sequences, the host must clear the CREQ field.

The purpose of the handshake mechanism, is to prevent a situation where control registers are being changed during an operation and by that causing an unexpected behavior.

51.10.6 I/O Pins Sharing

The NAND Flash Controller has logic that allows it to share I/O pins with pins of another memory controller.

The arbitration between the NFC and the other memory has hard priority favouring the other memory. Since NFC's accesses are long, when the other memory requests the bus, the NFC will always halt its operation as soon as possible based on its internal state and the other memory will be granted. The only operations that will not halt in the middle are relatively short ones, such as command phase, address phase or spare area read. The host must take into account, that spare area read is the longest operation of the above and can take up to 32 Flash-clock cycles (for 8bit memory in non-symmetric mode).

This hard decision arbitration contains a dead-lock counter in favour of the NFC. The counter counts fast clocks cycles (aclk) in which the NFC requests the bus, but not being granted. If the counter reaches a certain value, it will cause the arbitration to favour the NFC over the other memory. Meaning, that if the other memory will deassert its request it won't get it again until the NFC will be granted at least once. After that grant, arbitration returns to normal mode, meaning, that if the other memory requests the bus, the NFC will pause as soon as possible based on the previous description.

This priority based arbitration mechanism must be taken into account when sharing the I/O bus with another memory.

51.11 NFC Operation

This section describes the operation of the NFC. It is divided to the following sub-sections:

- Automatic operations (Automatic Operations).
- Atomic operations (Atomic Operations).
- Atomic operations sequence (Atomic Operations Sequence).
- ECC operation (ECC Operation).
- Symmetric/Asymmetric operation (Symmetric/Asymmetric Mode Operation).
- Delay line operation(Delay Line Operation).

51.11.1 Automatic Operations

The NFC offers the user a simplification in accessing the NAND device, by automatically performing common NAND sequences. All automatic operations require some kind of preparations from the host, before the automatic operation is performed. NFC informs the host about the completion of the operation.

51.11.1.1 Automatic program operation

Setting AUTO_PROG bit in LAUNCH_NFC register, results in NFC executing a full page program including a status read at the end (The status-read operation will be executed only in case that RBB_MODE=0). NFC will repeat this operation several times as defined in NUM_OF_ITERATIONS field in NFC_CONFIGURATION1 register.

Upon setting AUTO_PROG bit, NFC will immediately set dma_wr_req (in case SDMA is used and NUM_OF_ITERATION is greater than "0"). After setting dma_wr_req the host should start filling the internal RAM while the NFC will start executing the program operation by sending a command defined in NAND_CMD0 register, followed by sending the address from the relevant address group (The address will be divided into several address phases as defined in NUM_ADR_PHASES1 register). After the address phases finished then NFC will start copying the data from the NFC's RAM to the NAND device (If the RAM contains enough data as defined by FMP), and then NFC will issue a program-confirm command defined in NAND_CMD1 register. NFC will monitor the Ready/Busy status of the NAND device, and will read the status of the device as soon as the device returns to ready state.

In case SDMA is not used and NUM_OF_ITERATIONS is set to "0" then it is the user responsibility to fill the RAM with the page data before setting the AUTO_PROG bit.

When NUM_OF_ITERATIONS is greater than 0 and NUM_OF_DEVICES is greater than 0 (meaning interleaved mode. More than 1 page should be programmed to several devices), then NFC will operate a little different:

- NFC will set the next dma_wr_req after the data-transfer phase finished (meaning the RAM can be overwritten)
- NFC will start issuing the next page before the previous device returns to idle (Thus hiding the busy time of the device).

The address that will be sent depends on ADD_OP.

NFC will not set dma_wr_req after the data transfer of the last page (As defined in NUM_OF_ITERATIONS).

Automatic program operation generates 2 types of interrupts:

- INT will be set when all NAND devices that were involved in the operation returned to idle (And their status is recorded inside the NFC).
- AUTO_PROG_DONE will be set when NFC detects that all pages have been copied to its internal RAM (Meaning that the SDMA can be configured with parameters for next set of pages to be programmed).

51.11.1.2 Automatic Read Operation

Setting AUTO_READ bit in LAUNCH_NFC register, results in NFC executing a full page read including a status read at the end (In both RBB_MODE=0 or RBB_MODE=1). NFC will repeat this operation several times as defined in NUM_OF_ITERATIONS field in NFC_CONFIGURATION1 register.

Upon setting AUTO_READ bit, NFC will start executing a page read operation from the NAND device: read-command followed by address phases and a read-confirm-command if needed. Then NFC will monitor the read/busy status of the device, and will start reading the data as soon as the device returns to ready state. NFC will read the status of the operation afterwords.

NFC will set dma_rd_req when it finishes reading a full page from the NAND device and store the data in the internal RAM (After fixing ECC errors if there are any).

If NUM_OF_ITERATIONS is greater than 0, then NFC pauses after reading a page, and waits for the system to read out the data from NFC's RAM. NFC will continue to read the next page as soon as the system reads enough data as defined in FMP register.

INT will be set after NFC has read the status register of all NAND devices that were involved in the operation.

51.11.1.3 Automatic erase operation

Setting AUTO_ERASE bit in LAUNCH_NFC register, results in NFC executing a full block erase including a status read at the end.(The status-read operation will be executed only in case that RBB_MODE=0) NFC will repeat this operation several times as defined in NUM_OF_ITERATIONS field in NFC_CONFIGURATION1 register.

Upon setting AUTO_ERASE bit, NFC will start executing a block erase operation from the NAND device:

Erase-command, followed by erase-address, followed by erase-confirm-command. Then NFC will monitor the read/busy status of the device, and will read the status of the device as soon as the device returns to ready.

INT will be set when all NAND devices that were involved in the operation returned to idle (And their status is recorded inside the NFC).

51.11.1.4 AutoMatic Copy-back Operation

Setting AUTO_COPY_BACK0 bit in LAUNCH_NFC register, results in NFC executing a full copy-back operation including a status read at the end (The status-read operation will be executed only in case that RBB_MODE=0).

NFC will send a command (defined in NAND_CMD0 register), then address phases (defined in the relevant address group), then another command phase (defined in NAND_CMD1 register), then NFC will monitor the ready/busy status of the NAND device, then another command phase (defined in NAND_CMD2 register), then address phases (defined in relevant address group), then last command phase (defined in NAND_CMD3 register), then again NFC will monitor the ready/busy status of the NAND device. Finally NFC will check the status of the operation.

INT will be set when the status of the NAND device is ready.

Setting AUTO_COPY_BACK1 will result in NFC executing AUTO_COPY_BACK0 operation to all NAND devices connected to it (with the same addresses to all the devices).

NFC will perform AUTO_COPY_BACK1 in an interleaved mode (meaning, that if a NAND device is getting busy, NFC will continue to the next device, and will get back to the first device as soon as it becomes ready). So, if the system wants to perform a copy-back operation with the same addresses to all the NAND devices, it is much more efficient to use AUTO_COPY_BACK1 then using AUTO_COPY_BACK0 several times.

51.11.1.5 Automatic Status-read Operation

Setting AUTO_STAT bit in LAUNCH_NFC register, results in NFC executing a full status read operation to the NAND device.

NFC will send status read command (defined in ST_CMD in NFC_CONFIGURATION2 register), then it will read the status of the device and store it in NF_STATUS field in NFC_CONFIGURATION1 register.

51.11.2 Atomic Operations

51.11.2.1 Preset Operation

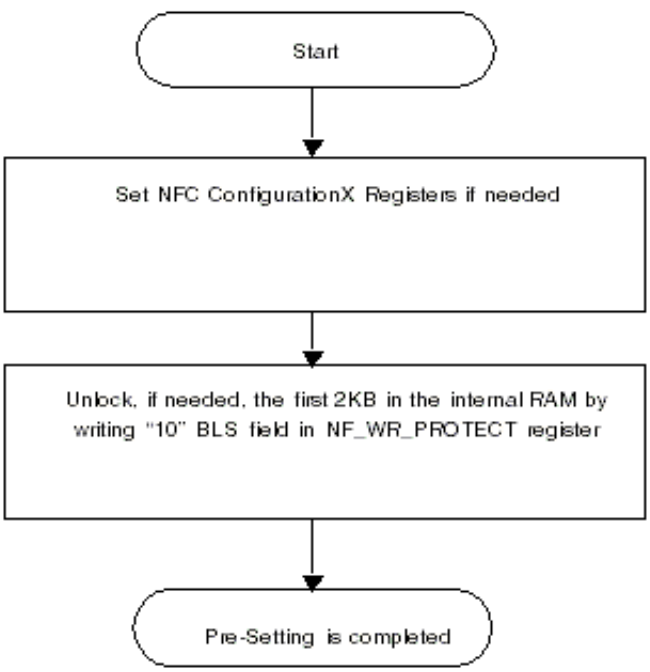


Figure 51-29. Flow Chart of Reset Operation

51.11.2.2 NAND Flash Atomic Command Input Operation

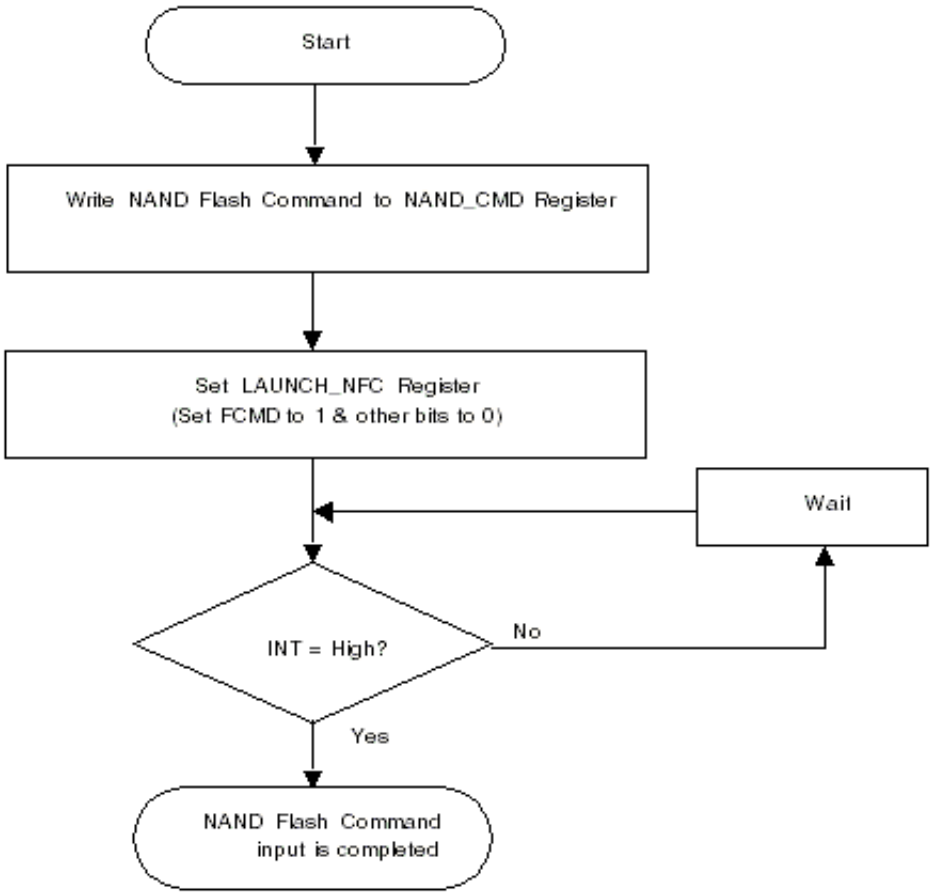


Figure 51-30. Flow Chart of NAND Flash Command Input Operation

51.11.2.3 NAND Flash Atomic Address Input Operation

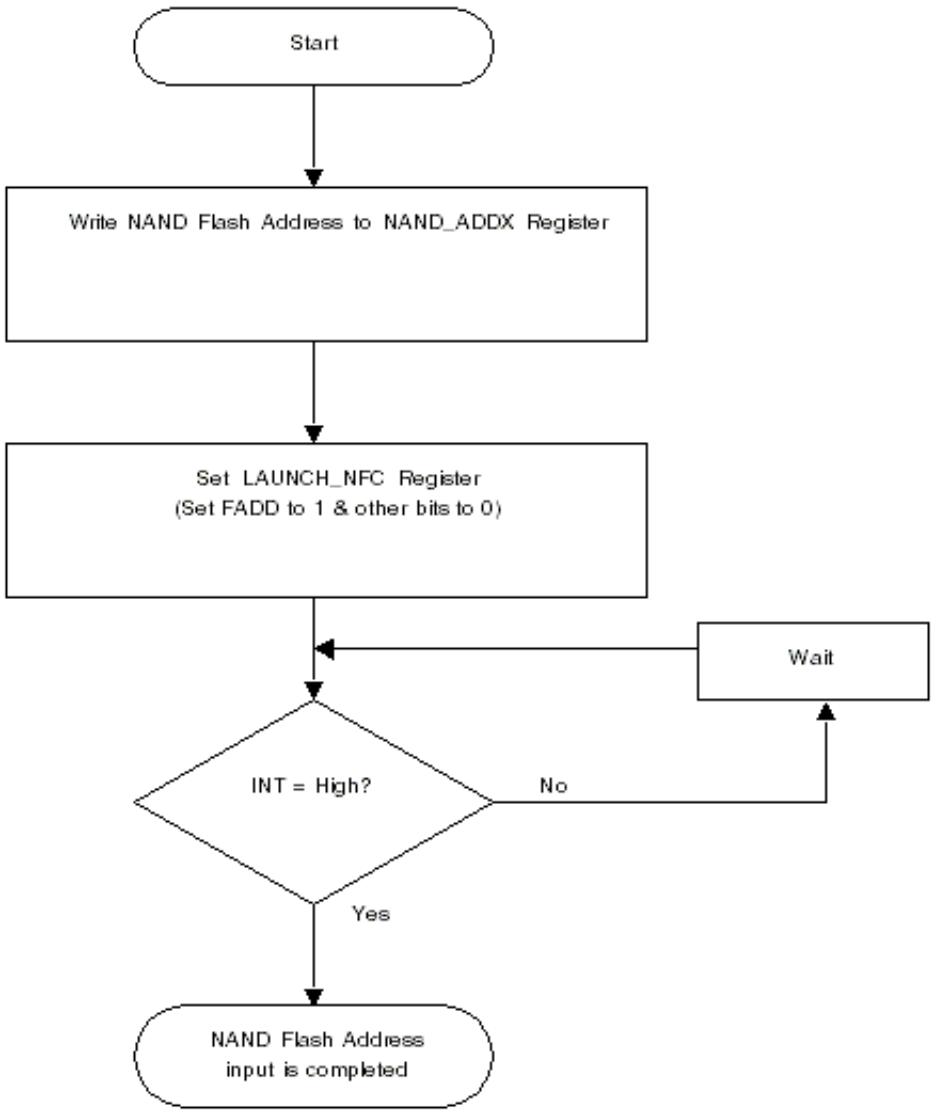


Figure 51-31. Flow Chart of NAND Flash Address Input Operation

51.11.2.4 NAND Flash Atomic Data Input Operation

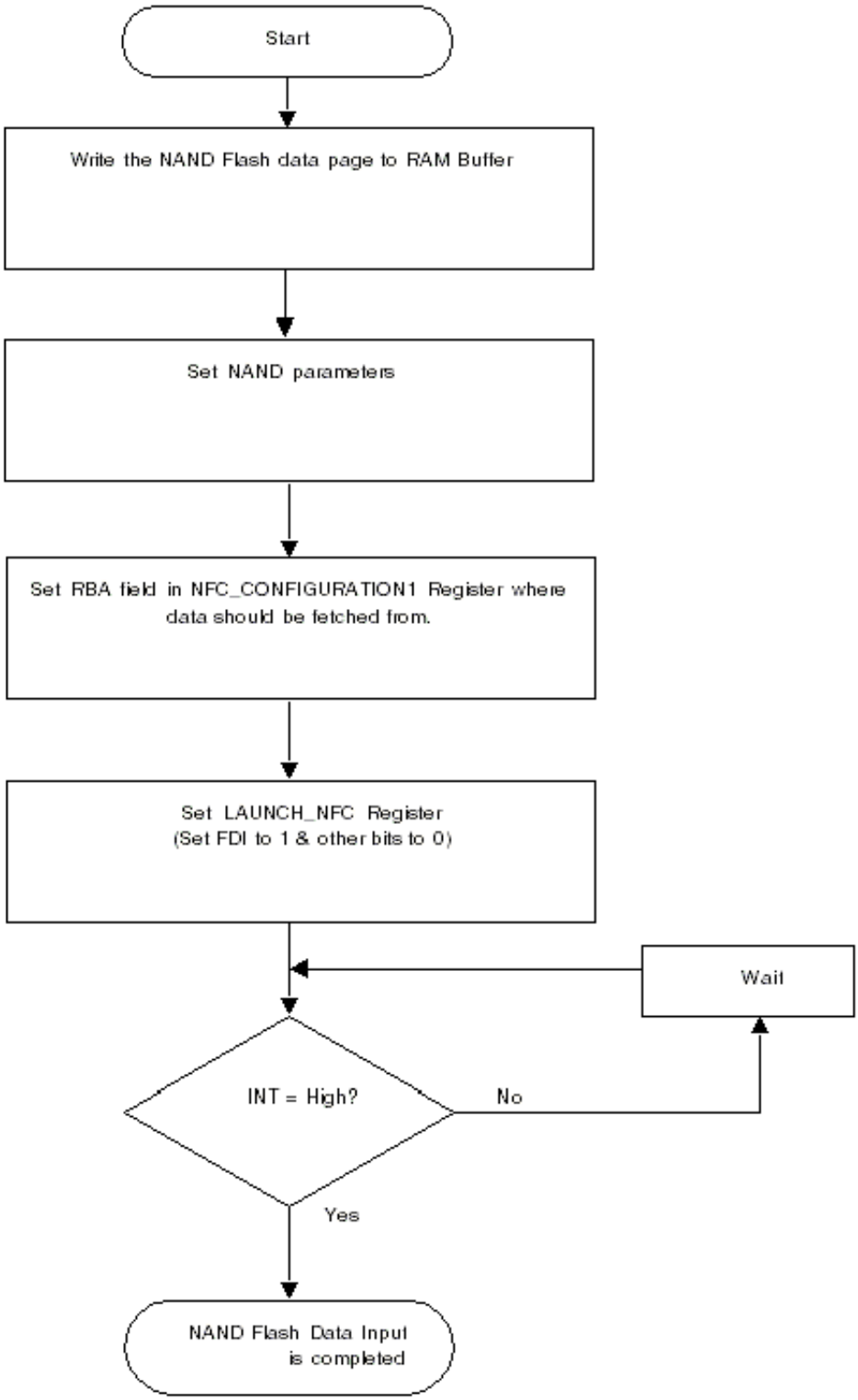


Figure 51-32. Flow Chart of NAND Flash Data Input Ope

51.11.2.5 NAND Flash Atomic Data Output Operation

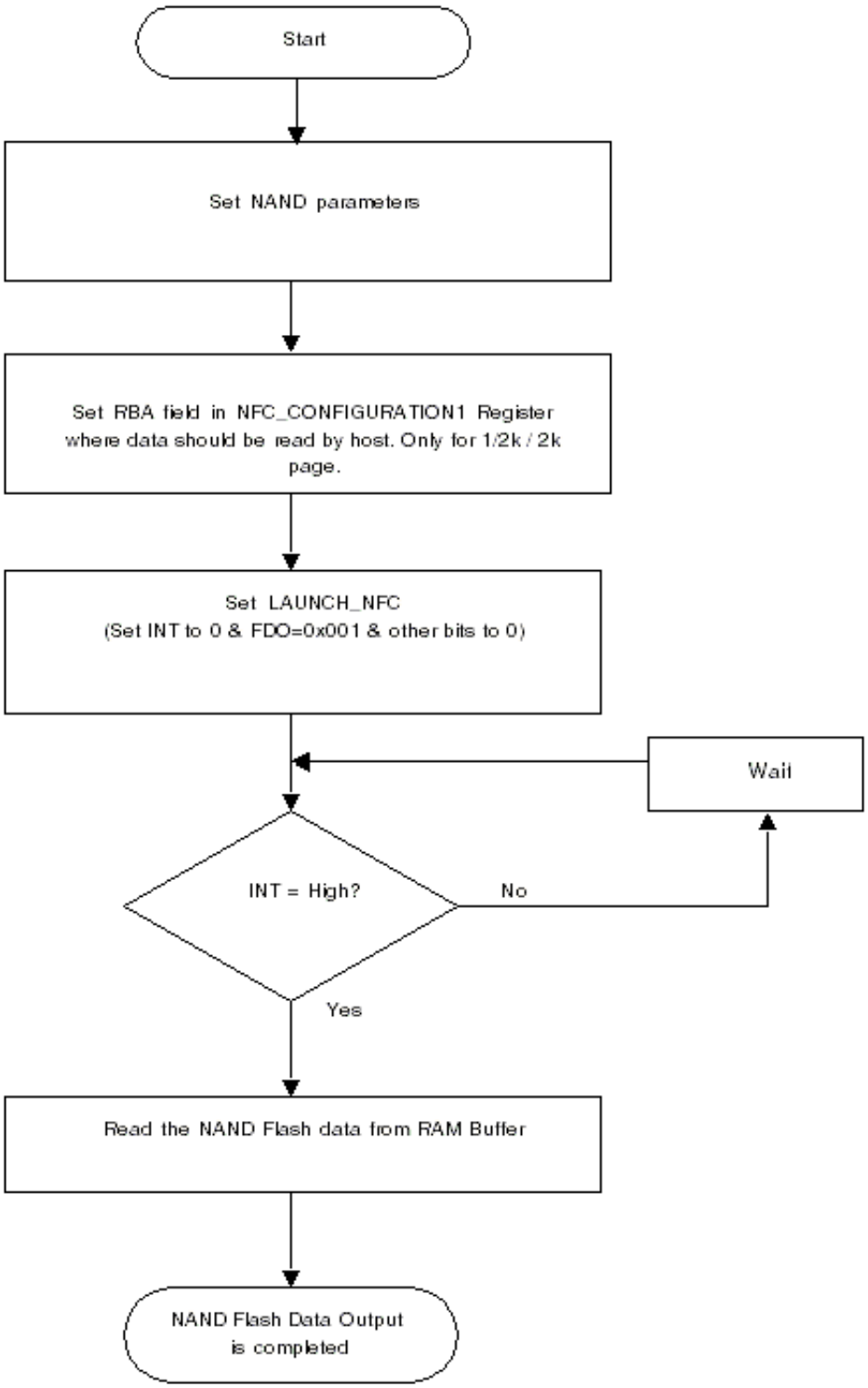


Figure 51-33. Flow Chart of NAND Flash Data Output Ope

51.11.2.6 Read NAND Flash Atomic ID Read Operation

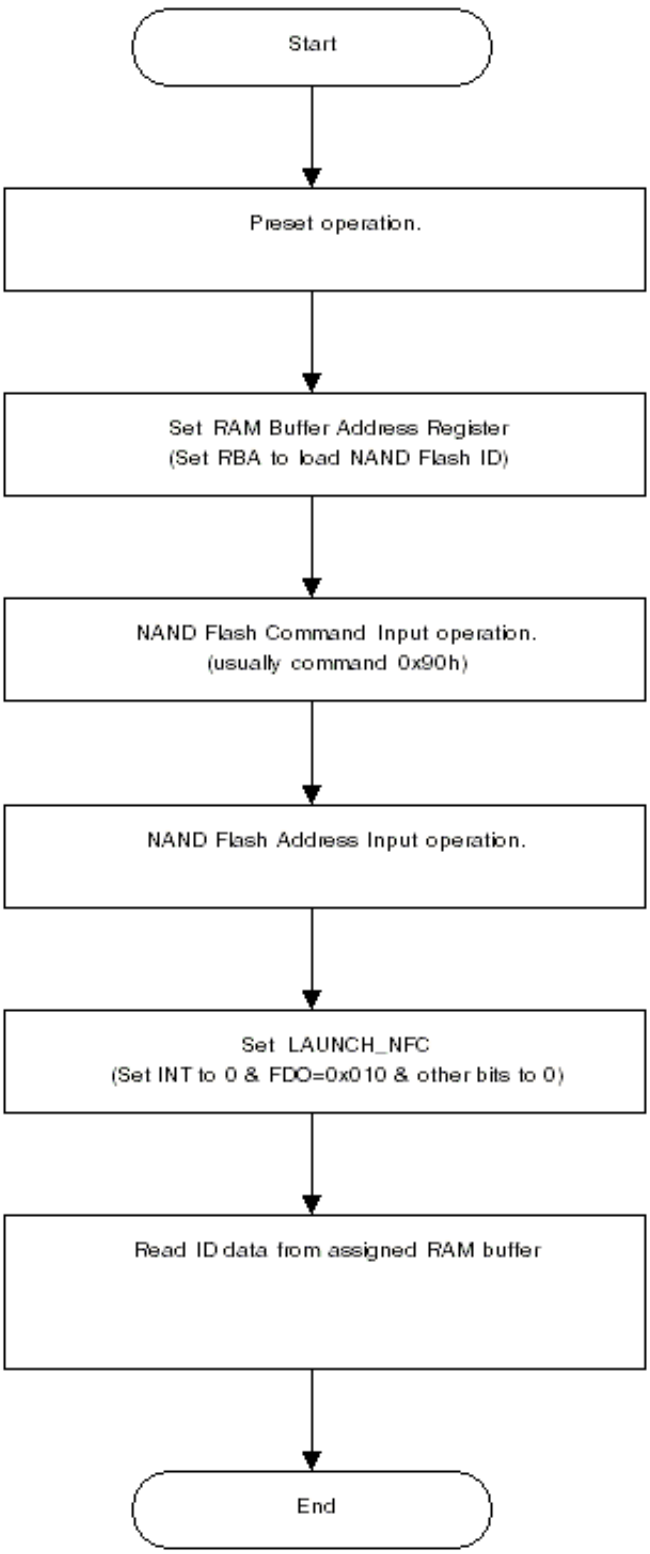


Figure 51-34. Flow Chart of Read NAND Flash ID Operation

NOTE

Read ID data from assigned RAM buffer (refer to NAND Flash ID Data Formats)

51.11.2.6.1 NAND Flash ID Data Formats

The format of NAND Flash ID data stored in the RAM buffer (for x8 NAND Flash) is shown in Table 51-33

Table 51-33. NAND Flash ID Data Format (x8)

RAM Buffer of RBA address	1st half-word				2nd half-word				3rd half-word						
	1st byte of ID		2nd byte of ID		3rd byte of ID		4th byte of ID		5th byte of ID		6th byte of ID				
	LSB			MSB											

The format of NAND Flash ID data stored in the RAM buffer (for x16 NAND Flash) is shown in Table 51-34.

RAM Buffer of RBA address	1st half-word				2nd half-word				3rd half-word						
	1st byte of ID		XXh		2nd byte of ID		XXh		3rd byte of ID		XXh				
	LSB			MSB											

Table 51-34. NAND Flash ID Data Format (x16)

RAM Buffer of RBA address	4th half-word				5th half-word				6th half-word						
	4th byte of ID		XXh		5th byte of ID		XXh		6th byte of ID		XXh				
	LSB			MSB											

51.11.2.7 NAND Flash Atomic Status Read Operation

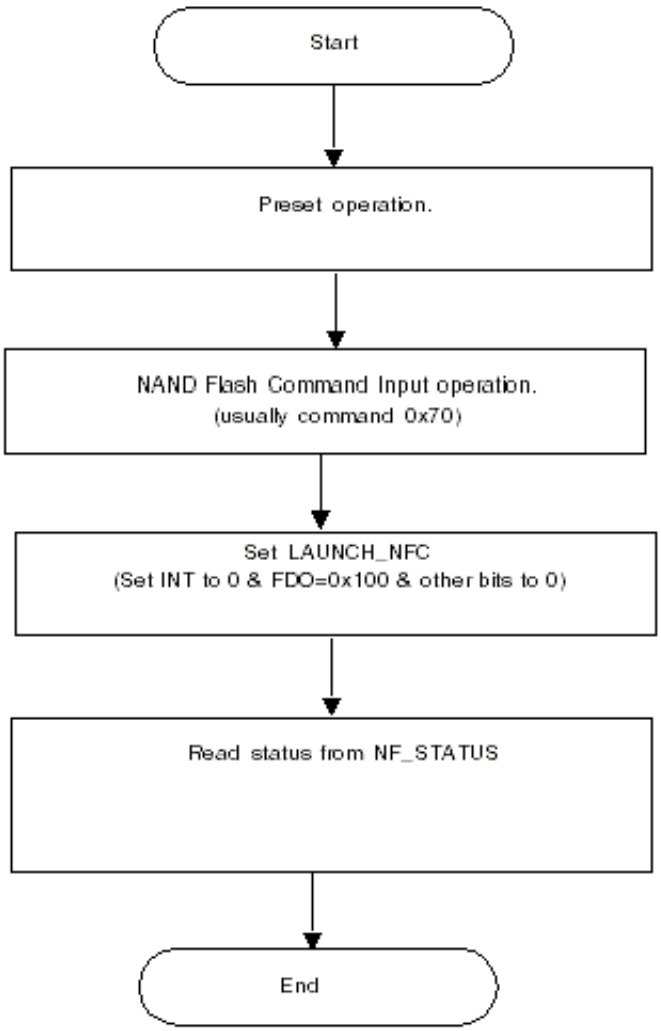


Figure 51-35. Flow Chart of Read NAND Flash Status Operation

51.11.3 Atomic Operations Sequence

51.11.3.1 Atomic Read Sequence Operation

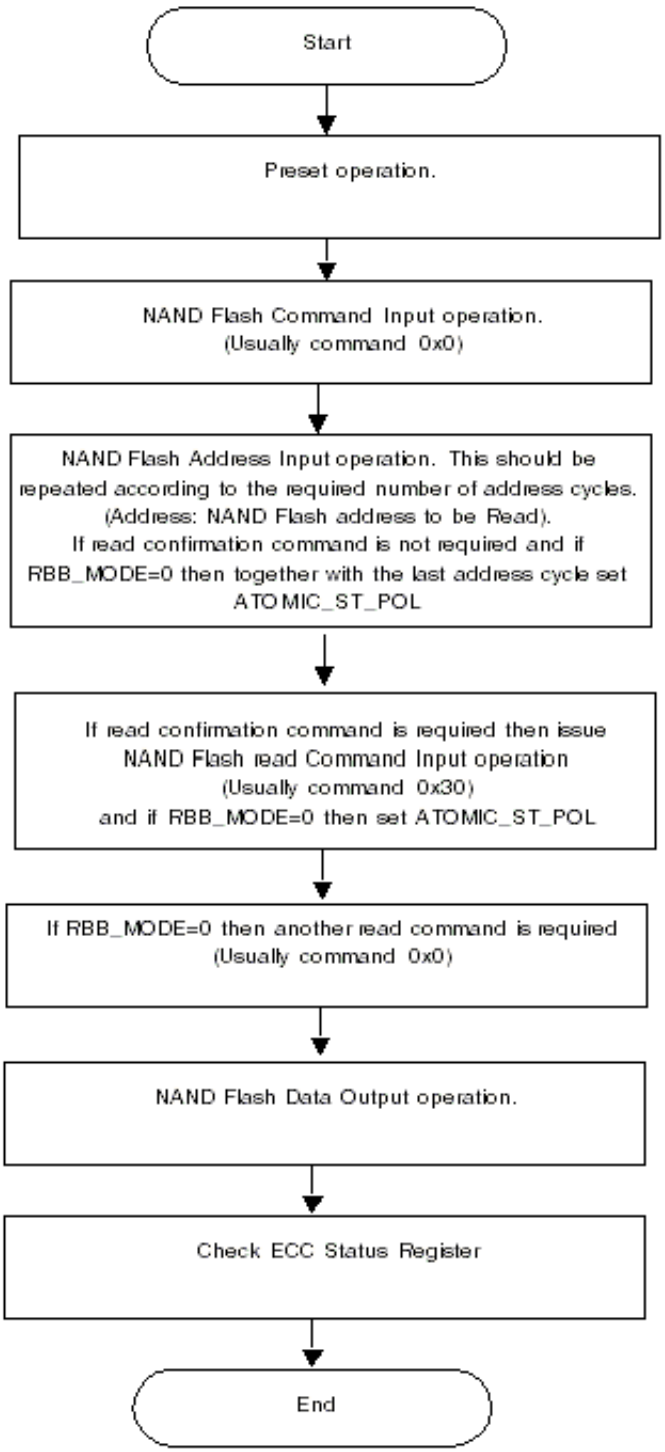


Figure 51-36. Flow Chart of Read NAND Flash Data Operation

51.11.3.2 Atomic Program Sequence Operation

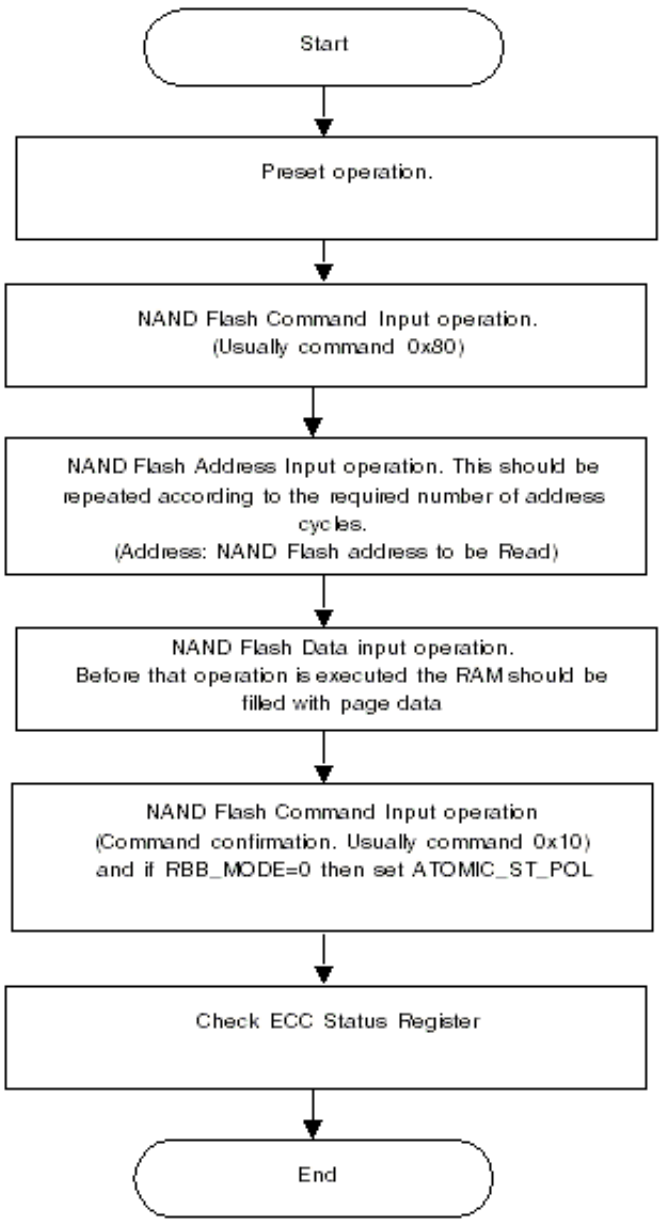


Figure 51-37. Flow Chart of Program NAND Flash Data Operation

51.11.3.3 Atomic Erase Sequence Operation

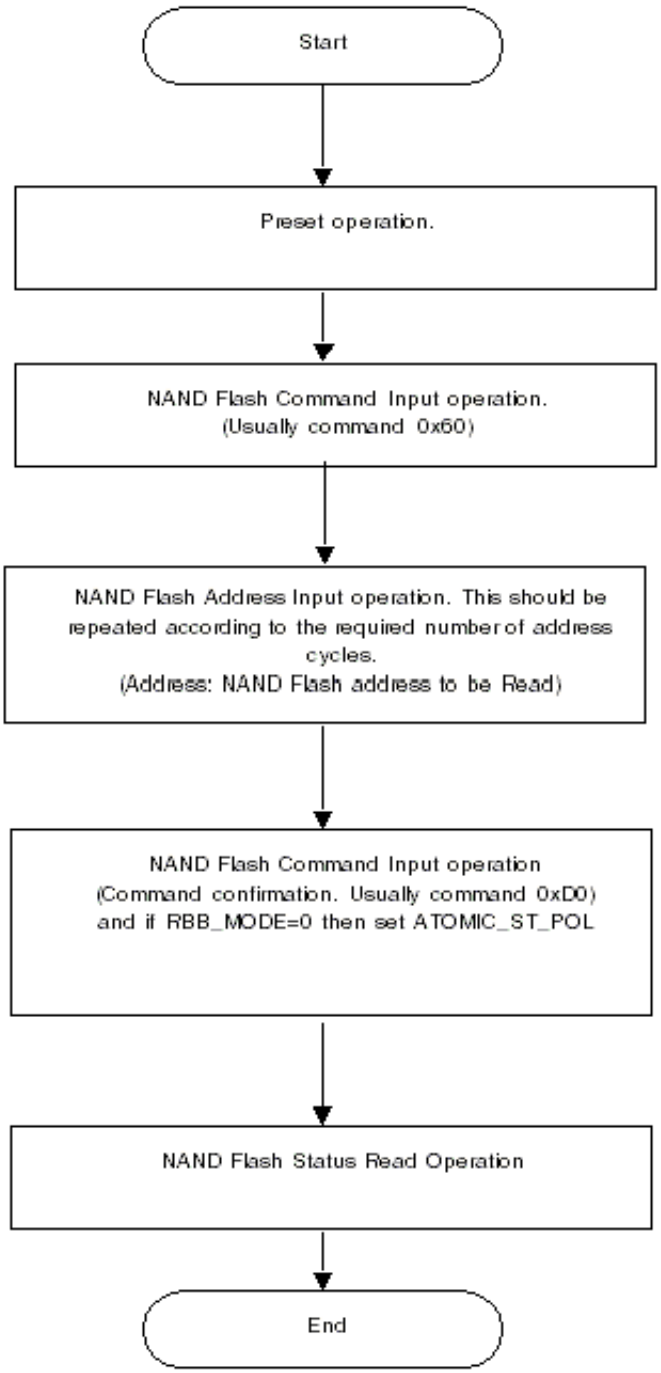


Figure 51-38. Flow Chart of Erase NAND Flash Operation

51.11.4 ECC Operation

51.11.4.1 ECC Normal Operation

The BCH ECC engine is responsible for detection and correction of up to 4/8/14/16 bits in each 528/538/548 byte section.

NFC divides the page into data sections of 1/2KB + spare area buffer. The ECC algorithm protects each data section and the associated spare area buffer.

- 4bits correction mode: In this mode, NFC protects 528B(512B of data section + 16B of spare area buffer). If the data section + spare area buffer are greater than 528B than these Bytes will not be protected by the ECC.
- 8/14bits correction mode: In this mode, NFC protects 538B(512B of data section + 26B of spare area buffer). If the section is greater than 538B than these Bytes will not be protected by the ECC.
- 16bits correction mode: In this mode the ECC will protect the whole data sections and the whole spare area buffers. If working with a page of 4K+218B, then the NFC will protect the first seven sections of 538B(512B of data section and 26B of spare area buffer) and at the last section it will protect 548B(512B of data section and 36B of spare area buffer).

When the NFC accesses the NAND Flash device for Program operation, it generates ECC code of 8/14/24/26 bytes. These bytes will override the corresponding spare area data that is written in the internal RAM refer to Table 51-5. These 8/14/24/26 bytes of ECC code are calculated based on the generator polynomial g(x) = x^13 + x^4 + x^3 + x + 1. When the NFC accesses the NAND Flash device for a Read operation, it performs a BCH decoding algorithm, and indicates how many bit errors were detected and corrected.

After a Read operation, the host can determine how many bit errors were found per 528/538/548 bytes by reading the status register (ECC_STATUS_RESULT). If the indication is uncorrectable error then it means that the number of bit errors that were detected (N) exceed the number of bit errors that can be corrected (T) according to ECC_MODE. Due to an algorithm weakness there might be a decoding error with a very low probability that the NFC will indicate that T number of bits error were detected and corrected instead of uncorrectable errors.

Upon program, the BCH ECC bytes are written directly to the NAND Flash device (and are not written back to the internal RAM). This means that if a user wants to read the ECC that was generated, he must read it from the NAND Flash device spare area.

There will be no error correction when performing a spare-only read from the nand flash device.

ECC is always calculated for 1/2KB+spare and not for 2KB/4KB + spare even if we work with a 2KB/4kB page memory. In case of 2KB/4KB page memory ECC will be calculated 4/8 times (once for every 1/2KB data section + associated spare).

That means that the 2k/4k page in the nandflash should be fill with 1/2k bytes data followed by 16/26 bytes of spare data 4/8 times:

512 main Byte + 16/26 spare byte + 512 main Byte + 16/26 spare byte + 512 main Byte + 16/26 spare byte + 512 main Byte + 16/26 spare byte...

The ECC operation can be bypassed using ECC_EN bit in NFC_CONFIGURATION2 register.

Program sequence:

1. Write a page of data to the internal RAM (main + spare area).
2. Set RBA to point to the section in which the data is in.
3. Configure the command and address sequence for program operation.
4. Program one page (main + spare) by writing "1" to FDI bit in LAUNCH_NFC register (BCH ECC code will override the corresponding spare area bytes)

Read sequence:

1. Set RBA to point to the section in the internal RAM Configure the command and address sequence for read operation.
2. read a page (main + spare) by writing "1" to FDO field in LAUNCH_NFC register (BCH ECC algorithm will automatically fix errors if there are any)

Table 51-35 shows the ECC code assignment of the NAND Flash spare area. This ECC code is updated by NFC automatically.

51.11.4.2 ECC Bypass Operation

In ECC bypass operation, the spare area is copied from NFC internal Ram Buffer to the nandflash device during program operation. During read operation the ECC detection/ correction mechanism will not work and the status register(ECC_STATUS_RESULT) will not be updated as well.

Table 51-35. ECC Code/Result Readability

operation	Read operation		Program operation		
	ECC Code from spare area buffer	ECC status register	ECC Code from spare area buffer	ECC status register	ECC content in NAND Flash device

Table continues on the next page...

Table 51-35. ECC Code/Result Readability (continued)

ECC operation	ECC code copied from nandflash device spare area	Valid	Invalid (old data [1]	-	ECC code generated by the NFC
ECC bypass	ECC code copied from nandflash device spare area	Valid	Invalid (old data)	-	Data from spare area of NFC internal Ram

1. Old data: ECC code is not updated to spare buffer, so ECC code placement of spare buffer remains old data.

51.11.4.3 How to Operate the ECC

In order to generate ECC and carry out the correction by the NFC,

- Program with ECC operation / Read with ECC operation

In order to generate ECC by the NFC and carry out the correction by the host,

- Program with ECC operation / Read without ECC operation

51.11.5 Symmetric/Asymmetric Mode Operation

In the default state of the NFC 1 flash clock cycle is used for toggling RE#/WE#.

In order not to work with lower frequency of flash clock, the SYM bit in NFC_CONFIGURATION2 register should be set to "0" and that will change the WE# and RE# period during read or program to be 2 flash clock cycles instead of 1.

Setting SYM bit also changes the duty cycle of RE# to be ~50% during read operation.

Latching the data during a read operation is done 1 flash_clk cycle + 1 aclk cycle after negedge of RE#.

The user can adjust the capturing timepoint by setting values to NFC_DEL_LINE register.

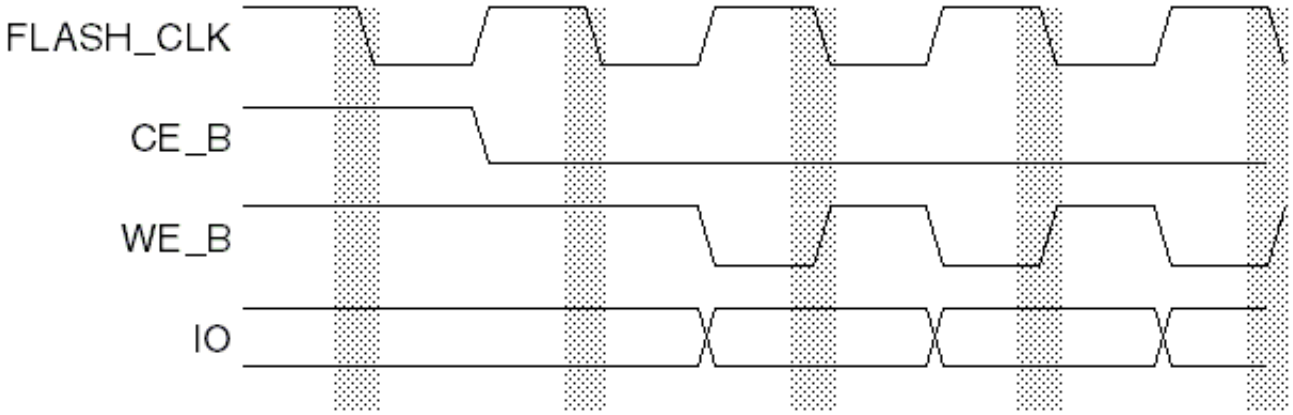


Figure 51-39. One Flash Clock Cycle Per Data Input (SYM bit =1)

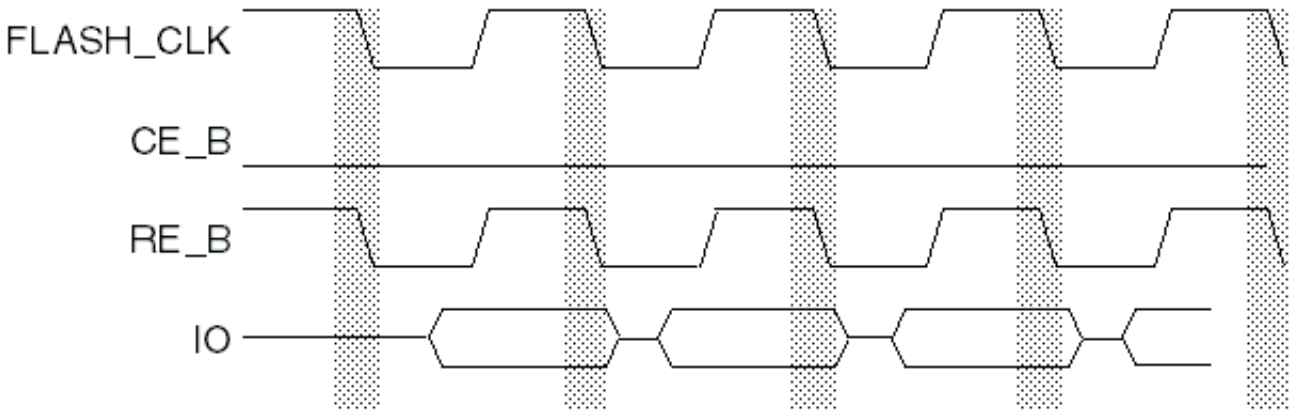


Figure 51-40. One Flash Clock Cycle Per Data Output (SYM bit =1)

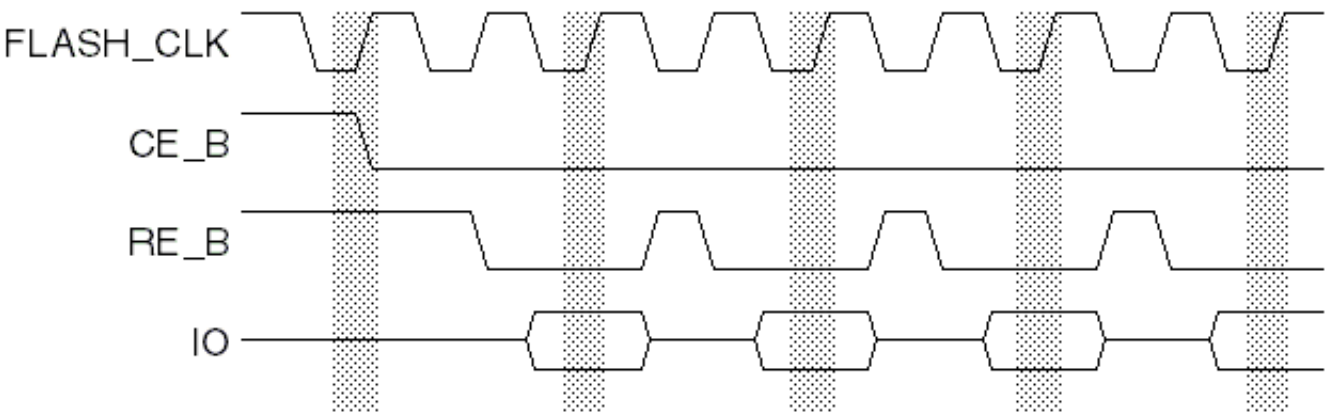


Figure 51-41. Two Flash Clock Cycles Per Data Output (SYM bit =0)

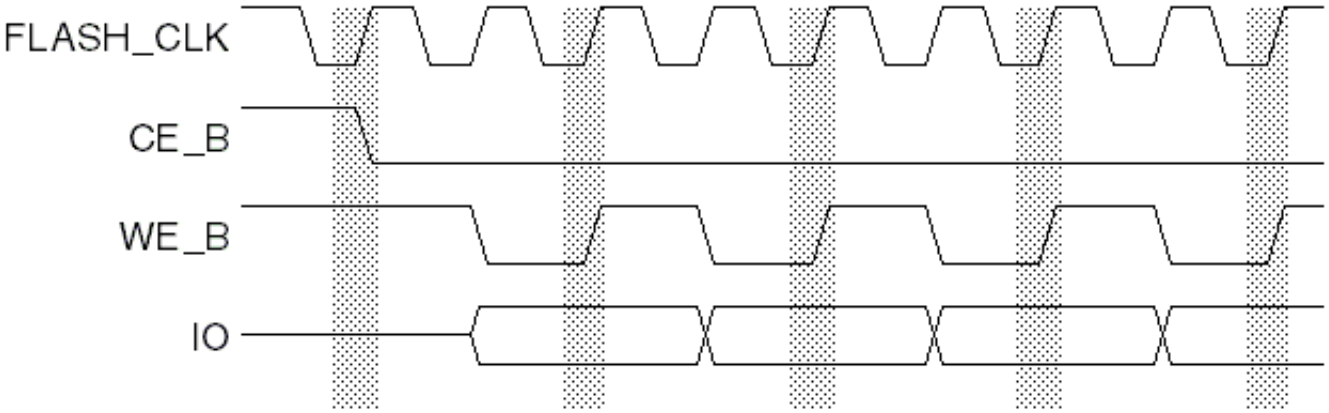


Figure 51-42. Two Flash Clock Cycles Per Data Input (SYM bit =0)

51.11.6 Delay Line Operation

51.11.6.1 Delay line background

When a read access is issued towards the NAND flash device then it takes a relatively long time from the assertion of the RE_B till the read data is latched by the NFC. The read data path starts with the deassertion of the RE_B (read enable) signal through the propagation in the I/O pads and till the RE_B port of the NAND flash device on the board. When the NAND flash device detects the deassertion of the RE_B signal it drives the associated read data back after Trea[ns] through the I/O pads and till being latched by the NFC. The NAND flash device is committed to keep driving the data for Trhoh [ns] after the posedge of RE_B. The delay line is used to compensate the long delay of the read data path and to adjust the sampling edge of the read data without any performance or frequency degradation.

The NFC generates an internal RE_B signal which is delayed by 1 flash clock cycle from the RE_B that is driven to the NAND device. The delayed RE_B passes through a delay line and the output of the delay line is used to latch the read data.

The delay line is composed from 4 delay line units. Each unit can delay the RE_B signal by a configurable time measured by axi slow clock cycle (fast clock of the NFC) fractions and up to 1/2 axi slow clock cycle. So the delay line can shift the RE_B signal by total of 2 axi slow clock cycles. Each fraction is 1/256 part of axi slow clock cycle and is configured in NFC_DELAY_LINE[NFC_ABS_DEL] register. This register is associated with a single delay line unit. By default the NFC latches the data after a fixed delay of 1 flash clock cycle + 1 axi slow clock cycle.

Figure 51-43 shows the read data path and Figure 51-44 shows the associated timing diagram.

51.11.6.2 Delay Line and Flash Clock Settings

The following will show how to set the flash clock cycle (2*Trp) and configure the desired value of the delay line. The calculations are based on parameters which depends on the NAND flash device, Freescale controller and the operating conditions.

- Assume the internal delay of the RE_B + the internal delay of the read data is D1[ns]
- Assume the board delay of the RE_B + board delay of the read data is D2[ns]
- Assume the delay of the RE_B caused by the delay line is Ddel[ns]. Note that the delay line has a minimal delay even if configured to zero delay
- Assume Trea[ns] is RE_B access time
- Assume Trp[ns] is RE_B pulse width
- Assume Trhoh[ns] is read data hold time from RE_B high

Note: The parameter D1 can be obtained from the AC characteristics of the Freescale device.

In order to sample the read data on time it is required that:

- 2Trp +Ddel > Trea + D1 + D2
- Trp + Ddel < Trhoh + D1 + D2
- Trp > Trp min defined by the NAND flash AC characteristics.

For example if:

- D1 = 10ns
- D2 = 2ns
- Delay line is configured to zero delay and its minimal delay is Ddel=3ns
- Trea = 20ns
- Trhoh = 15ns
- Trp min defined by the device is 12ns
- axi slow clock cycle is 7.5ns

In that case the pulse width of flash clock should be 14.5ns < Trp < 24ns or in other words the cycle of flash clock should be 29ns < Trc < 48ns (i.e flash clock frequency is 20.8Mhz < F< 34.4Mhz)

In case it required to work with Trp which is 12ns (i.e the minimal allowed by the NAND device) then it is possible by using the delay line as following. Setting a delay of 14.5 - 12 = 2.5ns means that we should delay the RE_B signal by 2.5/7.5 = 1/3 cycle of slow axi clock, which is approximately 85 parts of 256, So it is needed to configure NFC_DELAY_LINE[NFC_ABS_DEL] = 8'd85.

NOTE

For the above example the allowed ratio between axi slow clock cycle and flash clock cycle are 1:4,1:5 or1:6. If the cycle of the flash clock is set to 37.5ns (i.e ratio 1:4) and the total delay of the delay line is configured to 1 axi slow clock cycle (as the default value) then the latching edge of the read data will be delayed by 1 axi slow clock cycle, which is 7.5ns and it isn't guaranteed that the read data will be latched correctly.

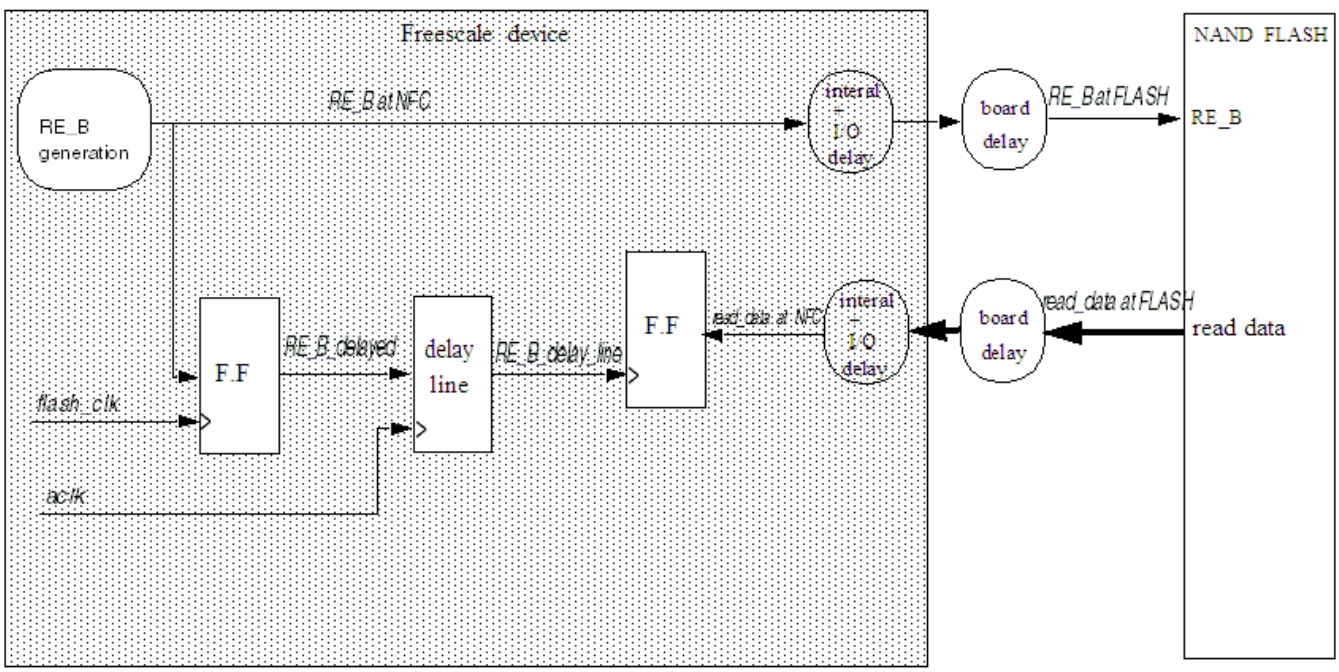


Figure 51-43. Read Data Path

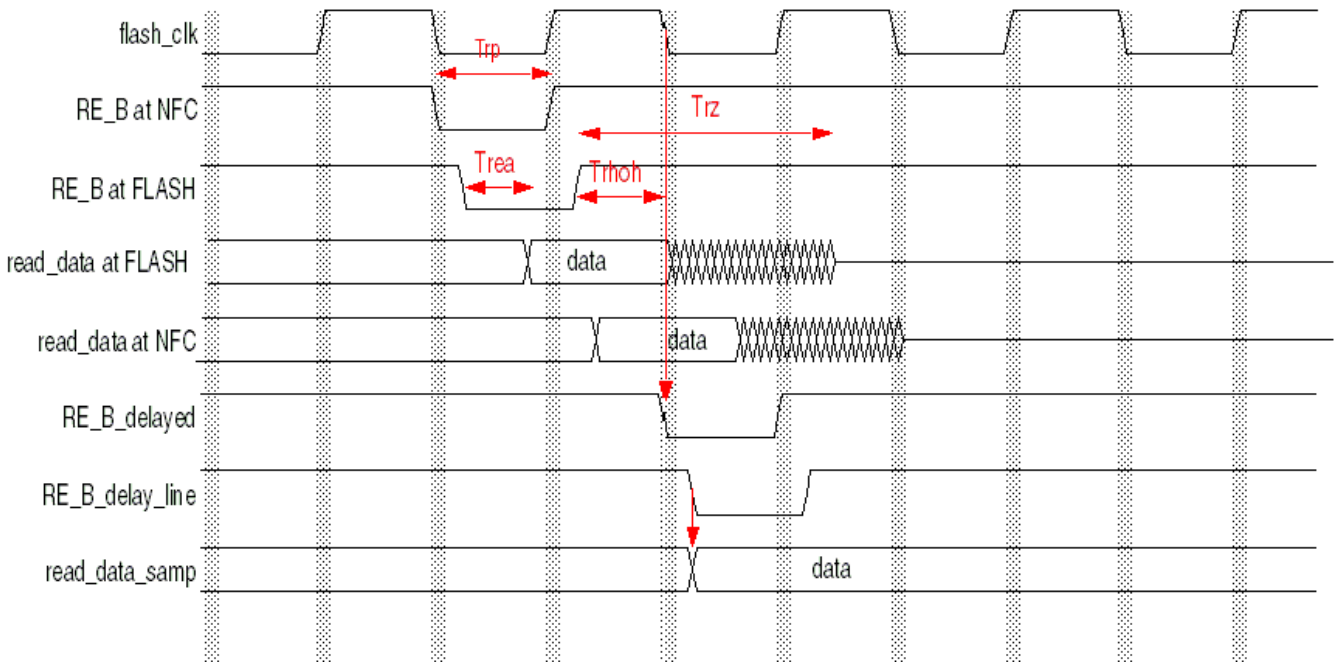


Figure 51-44. Read Data Path Timing

Figure 51-45 shows the block diagram of the delay line

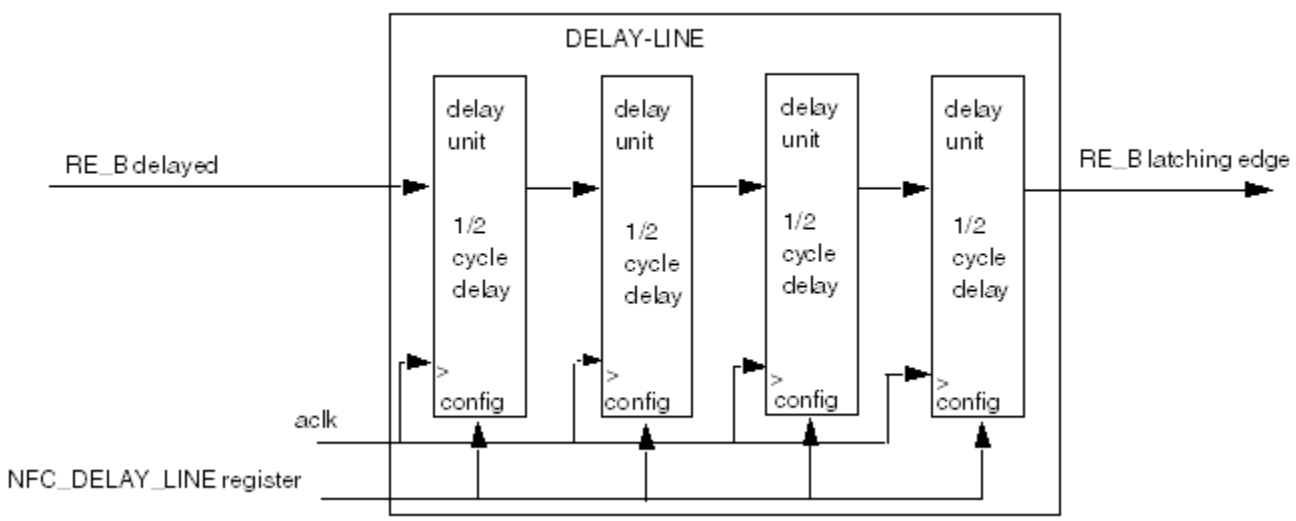


Figure 51-45. Delay Line block Diagram

51.12 Memory Connectivity Examples

Figure 51-46 and Figure 51-47 shows connectivity to 8-bits and 16-bits NAND device respectively

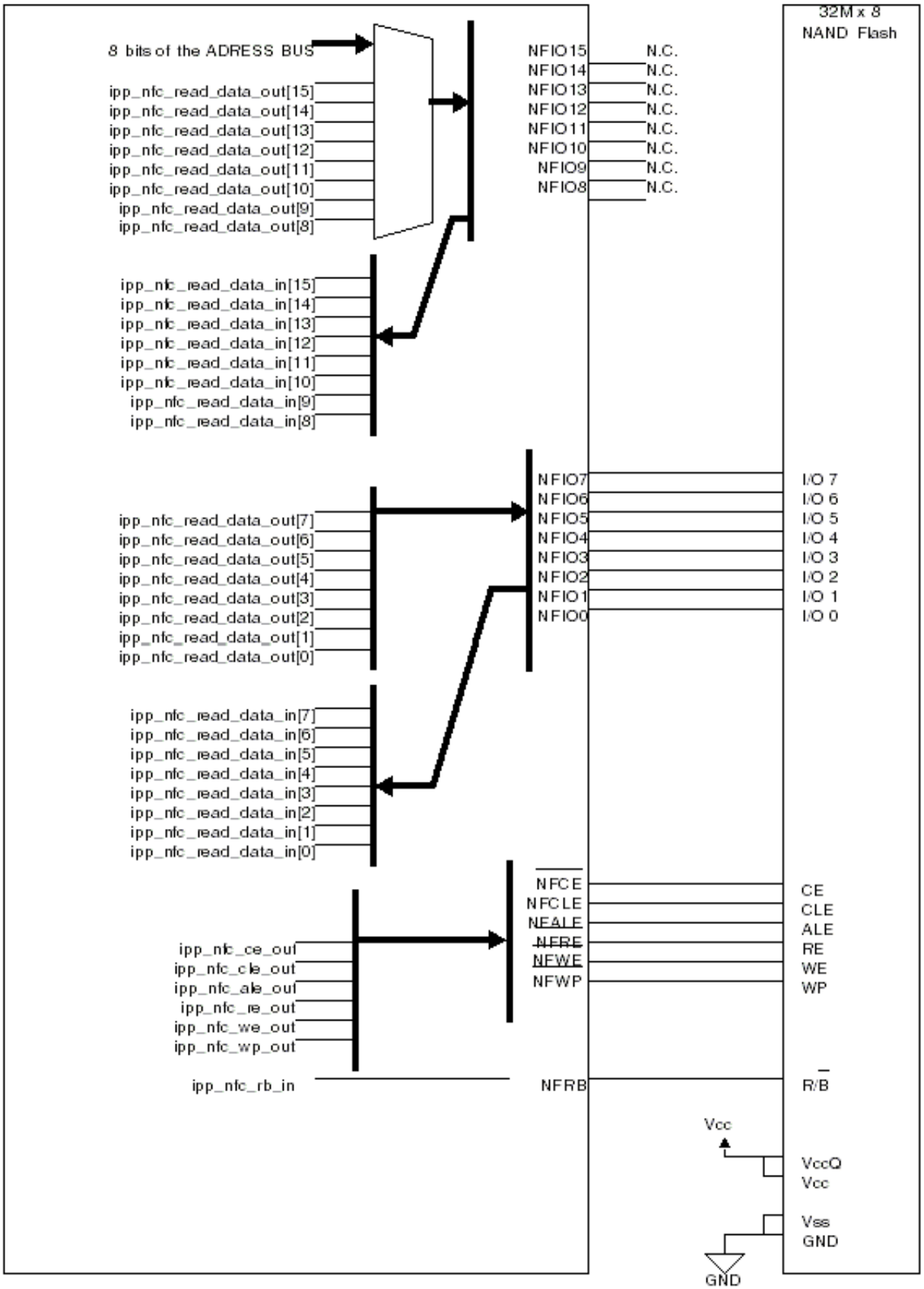


Figure 51-46. 8-bit NAND Flash Connectivity Diagram

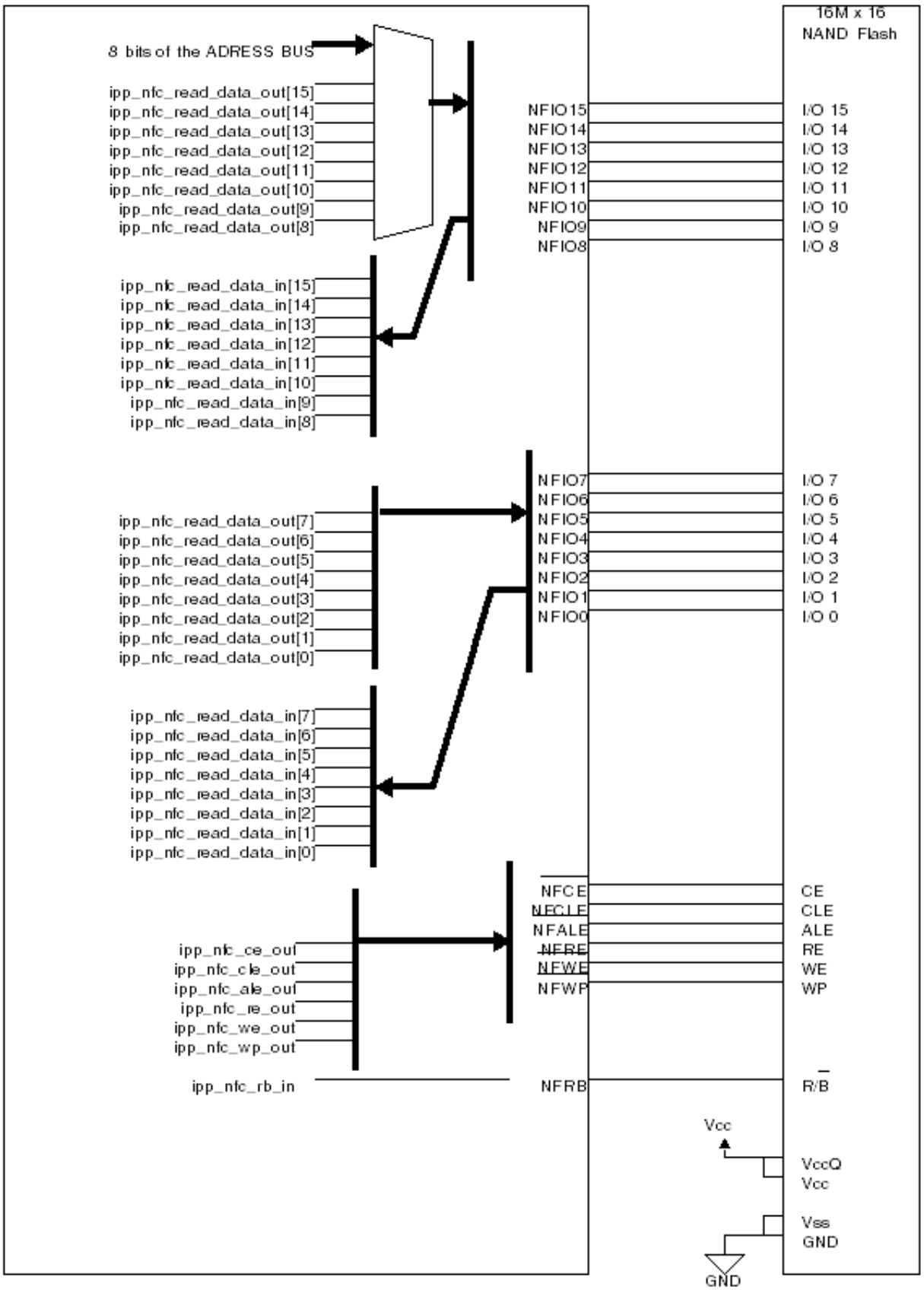


Figure 51-47. 16-bits NAND Flash Connectivity Diagr

NOTE

The NFC controller can support HS (High Speed) NAND Flash by supplying higher frequencies to the Flash Clock input.

51.13 Verified NAND Models

The following table lists the NAND models that where used for verifying the NFC, and are guaranteed to be supported.

Table 51-36. Verified NAND Models

Vendor	Part Number	Page Size	Bus Width
Samsung	k9f1g08u0m	2K	x8
	k9f1g16u0m	2K	x16
	k9f5608q0c	1/2K	x8
	k9k8g08u0m	2K	x8
	k9f5616q0c	1/2K	x16
	k9gag08u0m	4K	x8
Toshiba	tc58dvg02a1fti0	1/2K	x8
	tc58dvm82f1ft00	1/2K	x16
	tc58nvg0s3aft05	2K	x8
	tc58nyg0d9bxgj5	2K	816
	tc58nvg4d1dtg00	4K	x8
Hynix	hy27ss08121a	1/2K	x8
	hy27ss16121a	1/2K	x16
	hy27uf082g2m	2K	x8
	hy27uf162g2m	2K	x16
ST	nand02gw3b	2K	x8
	nand04gw3b	2K	x8
	nand512w3a2	1/2K	x8
Micron	mt29f2g08aab	2K	x8
	mt29f2g16aab	2K	x16
	mt29f32g08cbabawp	4K	x8

Chapter 52
On-Chip RAM Memory Controller (OCRAM)

52.1 Overview

A couple of options are provided to adding pipeline or wait-state in read/write access, in order to ensures flexible timing control at high/low frequency working conditions.

The internal block are shown in the figure below.

Figure 52-1. On-chip RAM Block Diagram

52.2 Basic Functions

52.2.1 Read/Write Arbitration

The detailed rules used in arbitration are as follows:

- If there is no granted read or write in the last cycle, and there is only read request or write request, the request will be granted.
- If there is no granted read or write in the last cycle, and there are both read or write requests coming in at the same time, the read request will be granted first.
- If a granted read/write transaction has just finished, the write/read request will have the higher priority in the next cycle.
- If the first read/write access request in a transaction is granted, all the data transfer in this burst will be finished before next arbitration begins, that is, the round-robin arbitration mechanism is based on AXI transaction, not data access.

52.2.2 TrustZone

TrustZone is also supported on this block.

When SECURE_ENBL bit in OCRAM.OCRAM_TZSECURE_REGION register is set, the STARTADDR and ENDADDR bit-fields in this register establish the region of OCRAM that can only be accessed (both read and write) according to the execution mode policy described in CSU chapter. If this bit is cleared to zero, the entire OCRAM can be accessed in either secure or non-secure mode.

NOTE

The ENDADDR is not configurable and its value is the last address of the OCRAM space. The STARTADDR granularity is of 4KB.

52.3 Advanced Features

Below are some advanced features designed to avoid timing issues when the on-chip RAM is working at high frequency.

52.3.1 Read Data Wait State

When the wait state is enabled, it will cost 2 cycles for each read access, (each beat of a read burst).

This can avoid the potential timing problem caused by the relatively longer memory access time at higher frequency.

When this feature is disabled, it only costs 1 clock cycle to finish a read transaction, that is, to get read data back in the next cycle of read request becomes valid on the bus.

52.3.2 Read Address Pipeline

When this feature is enabled, the read address from the AXI master is delayed 1 cycle before it can be accepted by the on-chip RAM.

This can avoid setup time issues for the read access on the memory cell at high frequency. Enabling this feature can cost, at most, 1 more clock cycle for each AXI read transaction, that is, at most 1 more clock cycle for each read burst with multiple beats of data.

When this feature is disabled, the read address from the AXI master can be accepted by the on-chip RAM without delay, and data can become ready for master at next clock cycle (if no other access and no read data wait).

52.3.3 Write Data Pipeline

When this feature is enabled, the write data from the AXI master would be delayed 1 cycle before it can be accepted by the on-chip RAM.

This can avoid setup time issue for the write access on the memory cell at high frequency. Enable this feature would cost at most 1 more clock cycle for each AXI write transaction, that is, at most 1 more clock cycle for each write burst with multiple beats of data.

When this feature is disabled, the write data from the AXI master can be accepted by the on-chip RAM without delay, and data can be written to memory at this cycle (if no other access and write address is also ready at this cycle).

52.3.4 Write Address Pipeline

When this feature is enabled, the write address from the AXI master would be delayed 1 cycle before it can be accepted by the on-chip RAM.

This can avoid setup time issue for the write access on the memory cell at high frequency. Enable this feature would cost at most 1 more clock cycle for each AXI write transaction, that is, at most 1 more clock cycle for each write burst with multiple beats of data.

When this feature is disabled, the write address from the AXI master can be accepted by the on-chip RAM without delay, and data can be written to memory at this cycle (if no other access and write data is also ready at this cycle).

52.4 Programmable Registers

There are no programmable registers in this block, however the TrustZone bits are implemented in the Software Controllable Signals (IIM.SCS1) register within the IIM No need to configure register SCS1, if TrustZone feature is not used. The bits description are shown below.

Table 52-1. TrustZone Control Register (IIM.SCS1)

0xIIM_BASE+0x030 (IIM.SCS1)					Access: Supervisor read/write			
	7	6	5	4	3	2	1	0
R	LOCK	STARTADDR					SECURE__ ENBL	
W								
RESET	0	0	0	0	0	0	0	0

Table 52-2. TrustZone Control Register (IIM.SCS1)

Field	Description
7 LOCK	Lock this register. 0 The register is not locked, it may be modified 1 The register is locked, all attempts to modify it are ignored
6-2 STARTADDR	STARTADDR - TiustZone Start Address The STARTADDR field holds the TrustZone start address in the OCRAM memory space. The start address affects the OCRAM address bits [16:12], hence the address granularity is 4KB. The ENDADDR is not configurable and is set to the end of OCRAM memory space.
1 SECURE_ENBL	SECURE_ENBL - Secure Enable 0 The TrustZone feature is disabled. Entire OCRAM space is available for all access types (secure/non-secure/user/supervisor). 1 The TrustZone feature is enabled. Access to address in the range specified by [ENDADDR:STARTADDR] follows the execution mode access policy described in CSU chapter.
0	Reserved

Chapter 53
1-Wire Block (OWIRE)

53.1 Overview

The OWIRE provides the communication link to a generic 1-Kbit add-only memory. The block sends or receives one bit at a time with an option for software to manage the data using bytes. The required protocol for accessing the generic OWIRE device is defined by Maxim-Dallas. The generic OWIRE device holds battery characteristics information.

The following figure shows a block diagram of the OWIRE.

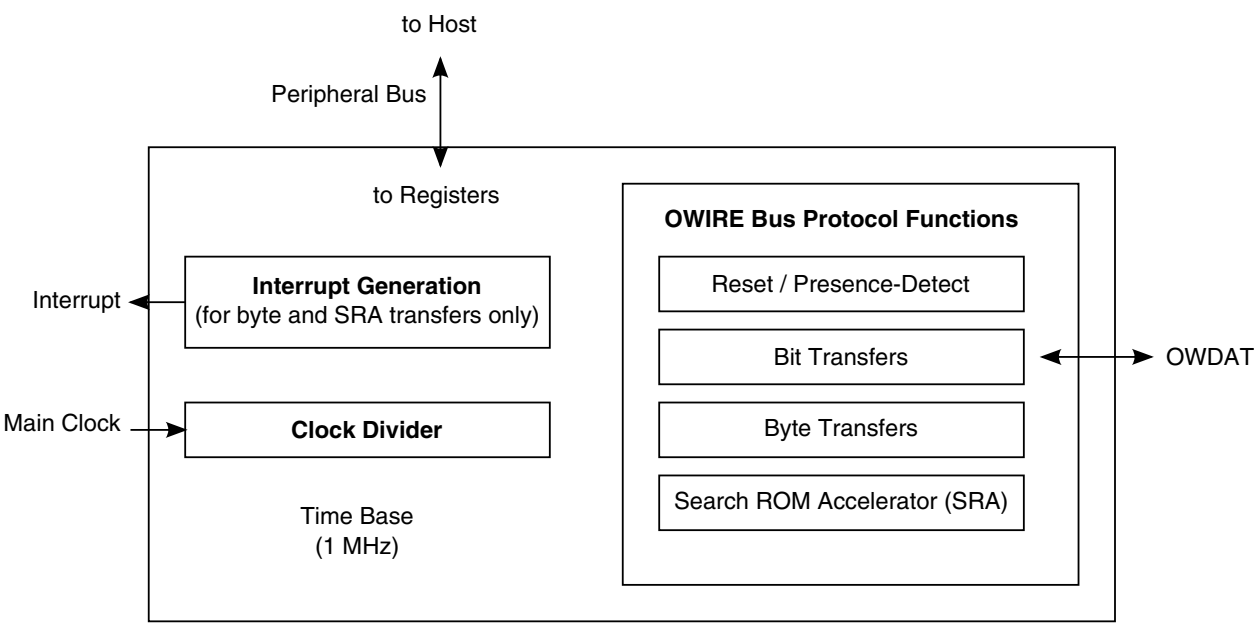


Figure 53-1. OWIRE Block Diagram

53.1.1 Features

The OWIRE includes the following features:

- Performs the OWIRE bus protocol to communicate with an external OWIRE device.
- Provides a clock divider to a generate a OWIRE bus reference clock (derived from the main clock provided internally to the block).
- Supports byte transfers with optional interrupts for more efficient programming.
- Provides a search ROM accelerator mode to speed the search ROM protocol.

53.1.2 Modes of Operation

The OWIRE supports the following operations:

- Normal Operating Modes (See Normal Operating Modes.)
  - Bit or Byte Transfers
  - Reset/Presence-detect Pulse
  - Search ROM Accelerator Mode
- Low Power Mode (See Low Power Mode.)

53.2 External Signals

Table 53-1 shows the signal that interfaces with a generic OWIRE device.

Table 53-1. OWIRE Block Signal

Signal	I/O	Function
OWDAT	I/O	One-Wire bus Requires an external pull-up resistor. The recommended resistor value is specified by the generic OWIRE device used in a given system.

53.3 Functional Description

The OWIRE interfaces with a generic 1-Kbit add-only memory, through a simple 1-bit bus. Software uses the OWIRE bus to program and read the 1-Kbyte memory.

The protocol involves first issuing one of four ROM function commands before the EPROM is accessible:

- Read ROM
- Match ROM
- Search ROM
- Skip ROM

Through the OWIRE bus, the host software interfaces with the generic OWIRE device and allows the required commands to be issued to control the EPROM of a generic OWIRE device. The host (through the OWIRE interface) is the bus master, and the generic OWIRE device(s) are the slave(s)

53.3.1 Normal Operating Modes

The OWIRE supports the following bus protocol functions:

- Reset/Presence-detect Pulse using the Control register (See Reset/Presence-detect Pulse.)
- Bit Transfers using the Control register (See Bit Transfers .)
- Byte Transfers using the TX/RX register (See Byte Transfers.)
- Search ROM Accelerator Mode using the Command register and the TX/RX register (See Search ROM Accelerator Mode.)

53.3.1.1 Reset/Presence-detect Pulse

The OWIRE provides for an automated initialization sequence for the OWIRE bus. Software initiates the initialization sequence by setting OWIRE_CONTROL[RPP]. The automated initialization sequence is as follows:

1. Generate a reset pulse.
2. Listen for a response from an external device by sampling for the OWIRE device presence bit.
3. After an amount of time determined by the OWIRE standard, latch the presence bit (true or false) in OWIRE_CONTROL[PST].

If an external device is detected (PST = 1), software can begin communications on the OWIRE bus.

The presence pulse is used by the OWIRE to determine if at least one generic OWIRE device is connected. Software determines if more than one generic OWIRE device exists.

53.3.1.2 Bit Transfers

After the initialization sequence (see Reset/Presence-detect Pulse), software can write and read one bit at a time using the Control register.

53.3.1.2.1 Write-0 Sequence

The Write-0 sequence writes a zero bit to the generic OWIRE device. Setting the OWIRE_CONTROL[WR0] bit initiates the Write-0 pulse sequence. Once the write is complete, the WR0 bit is automatically cleared.

53.3.1.2.2 Write-1 / Read Sequence

The Write-1 sequence writes a one bit to the generic OWIRE device. Setting the OWIRE_CONTROL[WR1] bit initiates the Write-1 pulse sequence. Once the write is complete, the WR1 bit is automatically cleared.

Because the Write-1 and Read timings are identical, this sequence also reads a bit from the bus. The sampled value is stored in the read status bit OWIRE_CONTROL[RDST] and is valid after the WR1 bit is self-cleared.

53.3.1.3 Byte Transfers

After the initialization sequence (see Reset/Presence-detect Pulse), software can transfer a byte at a time using the TX/RX register. Writing to the register connects to the Transmit buffer; reading to the register connects to the Receive buffer. See Figure 53-2.

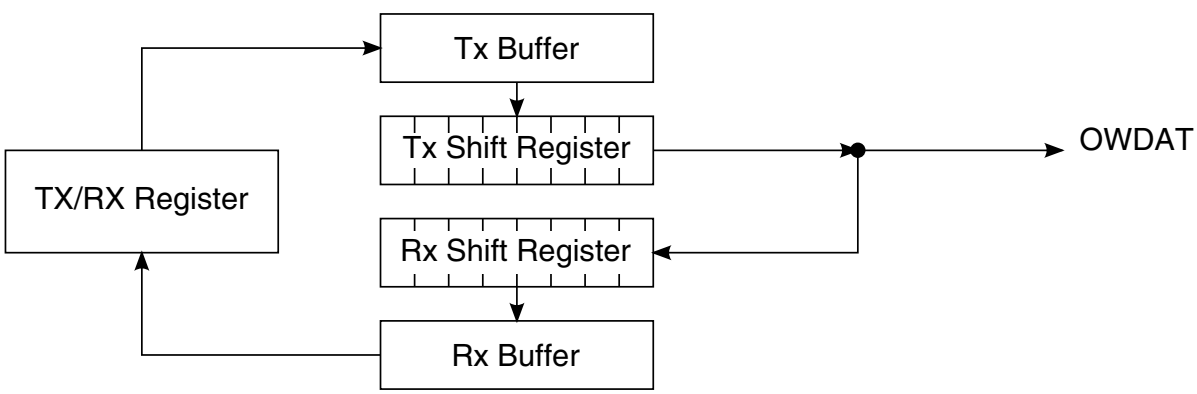


Figure 53-2. Byte Transfers

The Transmit buffer connects to an internal Transmit Shift Register where data is shifted serially onto the bus LSB first. Similarly, the Receive buffer connects to an internal Receive Shift Register where data is sampled serially from the bus.

Software can read a byte from the generic OWIRE device as follows:

1. Write 0xFF to the Transmit/Receive register location (connected to the Transmit buffer).

2. Wait for the receive-buffer-full (OWIRE_INTERRUPT[RBF]) interrupt (or poll the flag bit directly if the interrupt is disabled). During this time, the hardware is writing ones on the bus while sampling the wired-AND of the data from the device. The read data is shifted into the Receive Shift Register. When a byte is collected in the Receive Shift Register, the data is transferred to the Receive buffer, and the RBF flag is set.
3. Read from the Transmit/Receive register location (connected to the Receive buffer) upon receiving the RBF interrupt.

If the Receive buffer is full, new data is not shifted from the Receive Shift Register until the current data is read. To prevent the loss of data, software must read the TX/RX register to clear the receive-buffer-full flag (OWIRE_INTERRUPT[RBF]). This allows the Receive Shift Register to shift new data into the Receive buffer.

53.3.1.4 Search ROM Accelerator Mode

In *search ROM accelerator* mode the OWIRE relieves software from having to perform single-bit operations on the bus and helps determine if more than one generic OWIRE device exists.

The OWIRE enters search ROM accelerator mode when OWIRE_COMMAND[SRA] is set. This protocol specifies that the bus master read two bits (a bit and its complement), then writes a bit to specify which devices should remain on the bus for further processing.

Note that this mode requires that a reset followed by the search ROM command (0xF0) has already been issued on the OWIRE bus.

The OWIRE automatically exits search ROM accelerator mode if the OWIRE bus is re-initialized; see Reset/Presence-detect Pulse.

53.3.2 Low Power Mode

The OWIRE. automatically goes into low power mode whenever it is not communicating with a generic OWIRE device. The main clock is gated off in low power mode.

As soon as software writes to any register, the OWIRE exits low power mode.

53.3.3 Clocks

The OWIRE takes a main clock as a block input and passes it through a clock divider. (See the block diagram in Figure 53-1.) Software must program the divider factor to generate a 1-MHz clock that is used as an internal time base for the block, as given by the following equation:

time_base = main_clock ÷ (TIME_DIVIDER[DVDR] + 1)

For example, if the main clock frequency is 30 MHz, the value to write to the divider register is 29. If the main clock input frequency is not an integer, the programmer must ensure the time base frequency is within the range given by the following equation:

0.98 MHz ≤ time_base ≤ 1.02 MHz

NOTE

A main clock frequency below 10 MHz causes improper function of the block.

53.3.4 Reset

The OWIRE supports two levels of reset: hardware and software.

53.3.4.1 Hardware Reset

Whenever a device reset occurs, a hard reset is performed on the OWIRE, clearing all values written to all registers.

53.3.4.2 Software Reset

Software initiates a software reset by setting the reset bit OWIRE_RESET[RST]. A software reset clears all data written to the registers except for the Command and Interrupt registers (OWIRE_COMMAND, OWIRE_INTERRUPT).

NOTE

The reset register (OWIRE_RESET) itself is not cleared during a software reset. Software must clear the RST bit to release the software reset.

53.3.5 Interrupts

The OWIRE generates interrupts through the programming of the Interrupt Enable register; see Interrupt Enable Register (OWIRE_INTERRUPT_EN). The OWIRE can generate interrupts under the following conditions:

- Receive shift register or buffer full
- Transmit shift register or buffer empty
- Presence detect

Once any of these conditions are met, the Interrupt register (see Interrupt Register (OWIRE_INTERRUPT)) sets the corresponding bit and generates an interrupt if enabled in the Interrupt Enable register. The IAS bit within the Interrupt Enable register determines if the interrupt generated is active low, or active high. By default all interrupts are active high, and software should not modify IAS.

53.4 Programmable Registers

This section provides the block memory map and detailed descriptions of all registers.

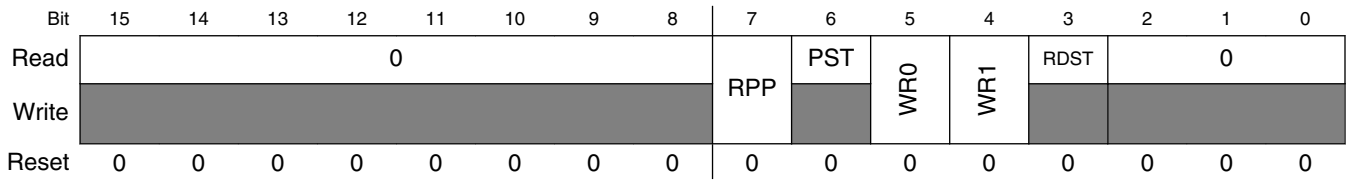
OWIRE memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FA_4000	Control register (OWIRE_CONTROL)	16	R/W	0000h	53.4.1/ 3658
63FA_4002	Time Divider register (OWIRE_TIME_DIVIDER)	16	R/W	0000h	53.4.2/ 3659
63FA_4004	Reset register (OWIRE_RESET)	16	R/W	0000h	53.4.3/ 3659
63FA_4006	Command Register (OWIRE_COMMAND)	16	R/W	0000h	53.4.4/ 3660
63FA_4008	Transmit/Receive Register (OWIRE_TX/RX)	16	R/W	0000h	53.4.5/ 3661
63FA_400A	Interrupt Register (OWIRE_INTERRUPT)	16	R	000Eh	53.4.6/ 3661
63FA_400C	Interrupt Enable Register (OWIRE_INTERRUPT_EN)	16	R/W	0000h	53.4.7/ 3663

53.4.1 Control register (OWIRE_CONTROL)

The control register is used to initiate the reset/presence-detect sequence and bit transfers. The register also provides the presence-detect status and bit-read status.

Address: OWIRE_CONTROL is 63FA_4000h base + 0h offset = 63FA_4000h



OWIRE_CONTROL field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7 RPP	Reset/Presence-detect Pulse. This bit is self-clearing and is cleared after the presence is determined. See Reset/Presence-detect Pulse. When writing: 0 Do nothing. 1 Generate Reset Pulse and sample the bus for the presence pulse from the external device. When reading: 0 Reset pulse complete. 1 Sequence not complete.
6 PST	Presence Status. This bit is valid after the RPP bit is self-cleared. 0 Device is not present. 1 Device is present.
5 WR0	Write 0. This bit is self-clearing and is cleared when the write of the bit is complete. See Write-0 Sequence. When writing: 0 Do nothing. 1 Write a 0 bit to the interface. When reading: 0 Write sequence complete. 1 Sequence not complete.
4 WR1	Write-1 / Read. This bit is self-clearing and is cleared when the write sequence is complete. See Write-1 / Read Sequence. When writing: 0 Do nothing 1 Write a 1 bit to the interface and sample the bus. When reading:

Table continues on the next page...

OWIRE_CONTROL field descriptions (continued)

Field	Description
	0 Sequence complete. 1 Sequence not complete.
3 RDST	Read Status. This bit is valid after the WR1 bit is self cleared. 0 A 0 has been sampled. 1 A 1 has been sampled.
2–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

53.4.2 Time Divider register (OWIRE_TIME_DIVIDER)

The time divider register is used for dividing the main clock input down to 1 MHz.

Address: OWIRE_TIME_DIVIDER is 63FA_4000h base + 2h offset = 63FA_4002h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								DVDR							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OWIRE_TIME_DIVIDER field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–0 DVDR	**Divider Factor.** The internal clock divider uses this field to generate the required time base for the block. See Clocks. 0x00 1 (default) 0x01 2 0xFF 256

53.4.3 Reset register (OWIRE_RESET)

The reset register is used to perform a software reset of the OWIRE.

Address: OWIRE_RESET is 63FA_4000h base + 4h offset = 63FA_4004h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0															RST
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OWIRE_RESET field descriptions

Field	Description
15–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 RST	Software Reset. See Software Reset. 0 Do not perform a software reset. 1 Initiate a software reset and hold the block in the software-reset state.

53.4.4 Command Register (OWIRE_COMMAND)

The OWIRE can be configured to run in Search ROM Accelerator mode using the command register. See Search ROM Accelerator Mode.

Address: OWIRE_COMMAND is 63FA_4000h base + 6h offset = 63FA_4006h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0														SRA	0
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OWIRE_COMMAND field descriptions

Field	Description
15–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1 SRA	**Search ROM Accelerator. This bit is cleared when the reset-presence-pulse bit CONTROL[RPP] is set.** 0 Deactivate the search ROM accelerator. 1 Switch to search ROM accelerator mode.
0 Reserved	This read-only field is reserved and always has the value zero. Reserved

53.4.5 Transmit/Receive Register (OWIRE_TX/RX)

Data sent and received from the OWIRE passes through the Transmit/Receive (TX/RX) register location. The OWIRE is double-buffered with separate transmit and receive buffers connected to the TX/RX register. See Byte Transfers.

Address: OWIRE_TX/RX is 63FA_4000h base + 8h offset = 63FA_4008h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								DATA							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OWIRE_TX/RX field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–0 DATA	Data byte. When writing: The data byte is written to the Transmit buffer. When reading: A data byte is read from the Receive buffer. The data is valid only when INTERRUPT[RBF] is set.

53.4.6 Interrupt Register (OWIRE_INTERRUPT)

Flags for the reset/presence-detect sequence and byte transfer operations are located in the Interrupt Register. These flags can generate an interrupt if the corresponding enable bit is set in the Interrupt Enable Register.

If interrupts are enabled, reading the Interrupt Register deactivates the interrupt even if all current flags are not cleared; therefore, the interrupt service routine should clear all pending flags during each routine call.

NOTE

When a byte is written to the Transmit/Receive Register, software then waits for a Transmit Shift Register Empty (TSRE) interrupt to occur. When the TSRE flag is set, the Receive Buffer Full (RBF) flag is also set. The RBF flag does not trigger an interrupt, assuming it is disabled. However, software should read the Transmit/Receive Register to clear the

RBF flag in order to give a proper status of the pending interrupts.

Address: OWIRE_INTERRUPT is 63FA_4000h base + Ah offset = 63FA_400Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0										RSRF	RBF	TSRE	TBE	PDR	PD
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

OWIRE_INTERRUPT field descriptions

Field	Description
15–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5 RSRF	Receive Shift Register Full. Hardware automatically clears this flag when data in the Receive Shift Register is transferred to the Receive buffer. 0 The Receive Shift Register is empty or currently receiving data. 1 A byte is waiting in the Receive Shift Register to be transferred to the Receive buffer.
4 RBF	Receive Buffer Full. This flag is cleared when software reads the byte from the TX/RX register. This flag prevents new data from being shifted into the Receive buffer from the Receive Shift Register. 0 No new data 1 A byte is waiting to be read from the TX/RX register.
3 TSRE	Transmit Shift Register Empty. Hardware automatically clears this flag when data in the Transmit buffer is transferred to the Transmit Shift Register. 0 Sending data 1 The Transmit Shift Register is empty and is ready to receive the next byte from the Transmit buffer.
2 TBE	Transmit Buffer Empty. This flag is cleared when software writes a byte to the TX/RX register. 0 The Transmit buffer is currently sending data to the Transmit Shift Register. 1 Nothing to transmit
1 PDR	Presence Detect Result. When a presence-detect (PD) interrupt occurs, this bit reflects the result of the presence-detect sequence. Note that this bit does not generate an interrupt. 0 Device found 1 Device not found
0 PD	Presence Detect. After an OWIRE reset has been issued, this flag is set after the appropriate amount of time for a presence detect pulse to have occurred. This flag is cleared when software reads the Interrupt Register. 0 A reset/presence-detect sequence has not been issued. 1 Reset/presence-detect sequence has completed. The result is provided in the PDR bit.

53.4.7 Interrupt Enable Register (OWIRE_INTERRUPT_EN)

The Interrupt Enable Register allows the system programmer to specify the source of interrupts. During a reset (hardware or software), all bits in this register are cleared, disabling all interrupt sources.

Address: OWIRE_INTERRUPT_EN is 63FA_4000h base + Ch offset = 63FA_400Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0										ERSF	ERBF	ETSE	ETBE	IAS	EPD
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OWIRE_INTERRUPT_EN field descriptions

Field	Description
15–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5 ERSF	Enable Receive Shift Register Full Interrupt. 0 Disable interrupt. 1 Enable interrupt.
4 ERBF	Enable Receive Buffer Full Interrupt. 0 Disable interrupt. 1 Enable interrupt.
3 ETSE	Enable Transmit Shift Register Empty Interrupt. 0 Disable interrupt. 1 Enable interrupt.
2 ETBE	Enable Transmit Buffer Empty Interrupt. 0 Disable interrupt. 1 Enable interrupt.
1 IAS	Interrupt Trigger Active State. This bit determines the polarity for all interrupts. Note that this bit is not an interrupt-enable bit. 0 Active high 1 Active low
0 EPD	Enable Presence Detect. 0 Disable interrupt. 1 Enable interrupt.

Chapter 54
Parallel Advanced Technology Attachment (PATA)

54.1 Overview

The Parallel Advanced Technology Attachment (PATA) block is an AT Attachment host interface, that is compliant with the ATA-5 standard. Its main use is to interface with IDE hard disc drives and ATAPI optical disc drives. It interfaces with the PATA device over a number of PATA signals.

See Figure 54-1 for the block diagram of the PATA interface.

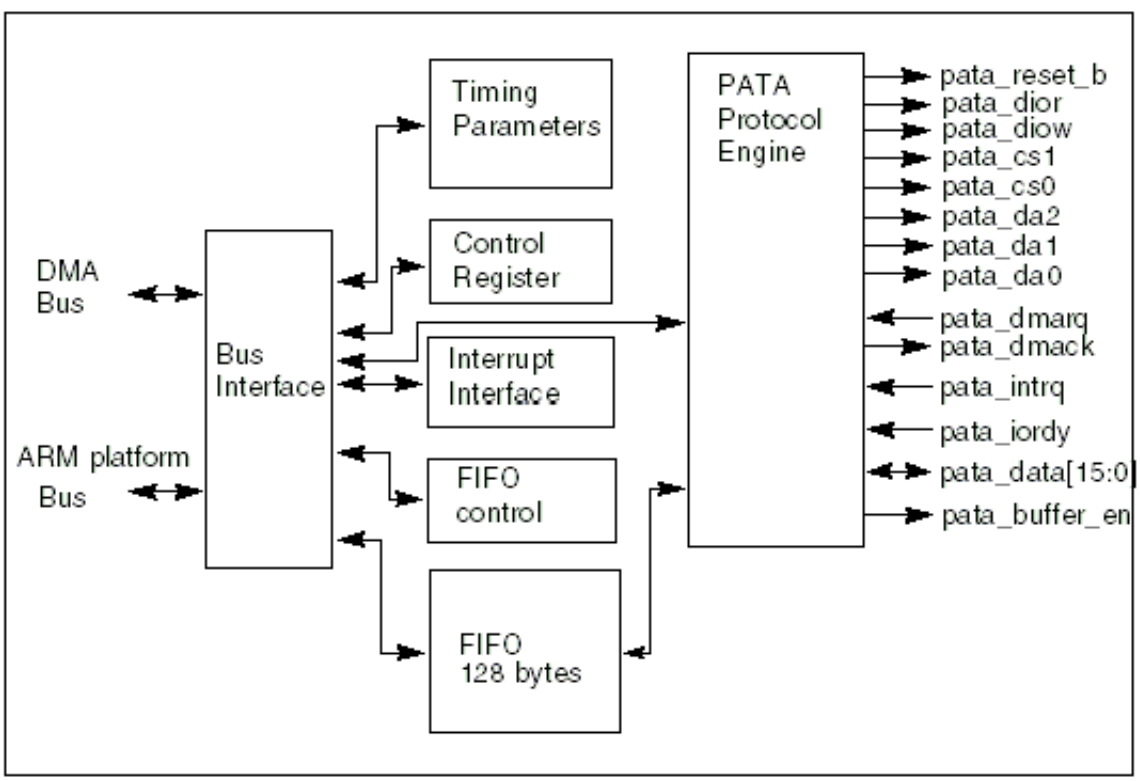


Figure 54-1. PATA Interface Block Diagram.

In Figure 54-1, the ARM platform and DMA buses communicate with the host processor and host DMA unit, respectively. All internal registers are visible from both the ARM and DMA buses, allowing smart DMA access to program the interface.

Two access modes are possible over the PATA bus: PIO mode and DMA mode. There are also two types of DMA mode: DMA slave mode (controlled by the host DMA) and DMA master mode (controlled by the internal DMA master). See Modes of Operation, for more information about these access modes.

54.1.1 Features

The PATA interface includes the following features:

- Programmable timing on the PATA bus. Works with a wide range of bus frequencies.
- Compliant with ATA-5 specification
  - Supports PIO modes 0-4
  - Supports multi-word DMA (MDMA) modes 0, 1, and 2
  - Supports ultra DMA (UDMA) modes 0-4 with a bus clock of at least 50 MHz, and the actual maximum supported speed is 66MHz
  - Supports ultra DMA (UDMA) mode 5 with a bus clock of maximum supported speed 66MHz
- 128-byte FIFO included within the interface
- FIFO receive, transmit, and end-of-transmission alarms to DMA unit
- Zero-wait cycles transfer between DMA bus and FIFO (enables fast FIFO reads/ writes)

54.1.2 Modes of Operation

The interface offers two alternative modes of operation, PIO mode and DMA mode. These modes are described in the following subsections.

54.1.2.1 PIO Mode

An access to the PATA bus in PIO mode occurs whenever an PATA PIO register is read or written by the host ARM platform or the host (smart) DMA unit. During a PIO transfer, the incoming IP Bus cycle is translated to an PATA PIO bus cycle by the PATA protocol engine. No buffering of data occurs, so the host ARM or host DMA cycle is stalled until the PATA bus read data is available on read, or is stalled until the IP Bus data can be put on the PATA bus during a write.

PIO accesses can be made at any time, even during a running PATA DMA transfer. In this case, the DMA transfer is paused, the PIO cycle is completed, and the DMA transfer is resumed.

54.1.2.2 DMA Mode

In DMA mode, data is transferred between the PATA bus and the FIFO. Two different DMA protocols are supported on the PATA bus: ultra DMA (UDMA) mode and multi-word DMA (MDMA) mode. These are described in more detail in the sections indicated below.

In DMA mode transfers, data is transferred between the PATA bus and the FIFO. Normally it is the task of the host smart DMA unit to write and read data to and from the FIFO to keep the transfer going.

The transfer pauses to avoid FIFO overflow and FIFO underflow. For this purpose, the fifo_rcv_alarm and fifo_tx_alarm signals are sent to the host DMA unit. An additional alarm signal (fifo_txfer_end_alarm) signals the end of the transfer to the smart DMA, which then completes the transfer and informs the ARM platform. Further details on these alarm signals are given in FIFO ALARM register (PATA_FIFO_ALARM).

The typical packet size is 32 bytes (8 32-bit words), but other packet sizes can also be handled.

Further details on DMA mode may be found in the following sections:

- Receiving Data from PATA Bus in DMA Mode
- Transmitting Data to PATA Bus in DMA Mode

54.2 External Signal Description

Table 54-1 lists the signals between this block and peripherals within the Tortola chip.

Table 54-1. Signal Properties

Name	Port	Function	Reset State	Type
pata_reset_b		PATA bus reset signal. Active low. If active, The PATA device is reset[1]	0	out
pata_dior		PATA bus read strobe	1	out
pata_diow		PATA bus write strobe	1	out
pata_cs1		PATA bus chip select 1	1	out
pata_cs0		PATA bus chip select 0	1	out

Table continues on the next page...

Table 54-1. Signal Properties (continued)

Name	Port	Function	Reset State	Type
pata_da2		PATA bus address line 2	0	out
pata_da1		PATA bus address line 1	0	out
pata_da0		PATA bus address line 0	0	out
pata_dmarq		PATA bus DMA request	-	in
pata_dmack		PATA bus DMA acknowledge	1	out
pata_intrq		PATA bus interrupt request	-	in
pata_iordy		PPATA bus iordy	-	out
pata_data[15:0]		PATA data bus (little-endian)	Hi-z	tristate in-out

1. This signal is a standard PATA bus signal. It conforms with the PATA specification.

The signals in Table 54-1 are described in Signal Descriptions.

54.2.1 Signal Descriptions

For a detailed description of the PATA bus signals, refer to the ATA-5 specification.

54.2.1.1 pata_reset_b (out)

When negated, the PATA reset signal indicates the PATA bus is in reset state. The PATA bus is in reset whenever the appropriate bit in the control register is cleared. After system reset, the PATA bus is reset.

54.2.1.2 pata_dior (out)

During PIO and MDMA transfers, the DIOR PATA signal functions as a read strobe. During UDMA data in bursts, it functions as HDMARDY. During UDMA data out burst mode, it functions as host strobe.

54.2.1.3 pata_diow (out)

During PIO and MDMA transfers, the DIOW PATA signal functions as a write strobe. During UDMA burst mode, it is used by the host terminate running UDMA transfers.

54.2.1.4 pata_cs0, pata_cs1, pata_da2, pata_da1, pata_da0 (out)

The address group of the PATA bus consists of the chip selects ata_cs0 and ata_cs1, and the address lines ata_da2, ata_da1, and ata_da0. All five lines follow the same timing.

54.2.1.5 pata_dmarq (in)

The PATA bus device DMA request is pulled high by the device if it wants to transfer data using MDMA or UDMA mode.

54.2.1.6 pata_dmack (out)

The PATA bus host DMA acknowledge is pulled low by the host when it grants the DMA request.

54.2.1.7 pata_intrq (in)

The PATA bus interrupt request is pulled high by the device whenever it wants to interrupt the host ARM platform.

54.2.1.8 pata_iordy (in)

The PATA bus IORDY line has three functions:

- IORDY-Active low, wait during PIO cycles
- DDMARDY-Active low, device ready during UDMA out-transfers
- DSTROBE-Device strobe during UDMA in-transfers

54.2.1.9 pata_data[15:0] (in/out-tristate)

The PATA data bus signal carries data to/from the PATA device.

54.2.1.10 pata_buffer_en

When the buffer direction control signal is asserted, data is driven outward to the device; when negated, data is driven inward to the host.

54.2.2 PATA Bus Timing

This section describes the PATA bus timing, and explains how to ensure that the PATA interface meets timing requirements. Timing diagrams and timing relation equations are provided.

54.2.2.1 Timing Parameters

Table 54-2 shows the PATA timing parameters, and their determining factors.Determining factors include the system design, the bus buffer used, the cable delay and the cable skew.

Table 54-2. Timing Parameters

Name	Meaning	Controlled by
T	Bus clock period	Tortola clock generator
ti_ds	Set-up time ata_data to ata_iordy edge (UDMA-in only)	top level design
ti_dh	hold time ata_iordy edge to ata_data (UDMA-in only)	top level design
tco	propagation delay bus clock L-to-H to ata_cs0, ata_cs1, ata_da2, ata_da1, ata_da0, ata_dior, ata_diow, ata_dmack, ata_data, ata_buffer_en	top level design
tsu	set-up time ata_data to bus clock L-to-H	top level design
tsui	set-up time ata_iordy to bus clock H-to-L	top level design
thi	hold time ata_iordy to bus clock H to L	top level design
tskew1	Max difference in propagation delay bus clock L-to-H to any of following signals ata_cs0, ata_cs1, ata_da2, ata_da1, ata_da0, ata_dior, ata_diow, ata_dmack, ata_data (write), ata_buffer_en	top level design
tskew2	Max difference in buffer propagation delay for any of following signals ata_cs0, ata_cs1, ata_da2, ata_da1, ata_da0, ata_dior, ata_diow, ata_dmack, ata_data (write), ata_buffer_en	transceiver
tskew3	Max difference in buffer propagation delay for any of following signals ata_iordy, ata_data (read)	transceiver
tbuf	Max buffer propagation delay	transceiver
tcable1	cable propagation delay for ata_data	cable
tcable2	cable propagation delay for control signals ata_dior, ata_diow, ata_iordy, ata_dmack	cable
tskew4	Max difference in cable propagation delay between ata_iordy and ata_data (read)	cable
tskew5	Max difference in cable propagation delay between (pata_dior, pata_diow, pata_dmack) and pata_cs0, pata_cs1, pata_da2, pata_da1, pata_da0, pata_data (write)	cable
tskew6	Max difference in cable propagation delay without accounting for ground bounce	cable

54.2.2.2 PIO Mode Timing

54.2.2.2.1 PIO Read Mode Timing

A timing diagram for PIO read mode is given in Figure 54-2.

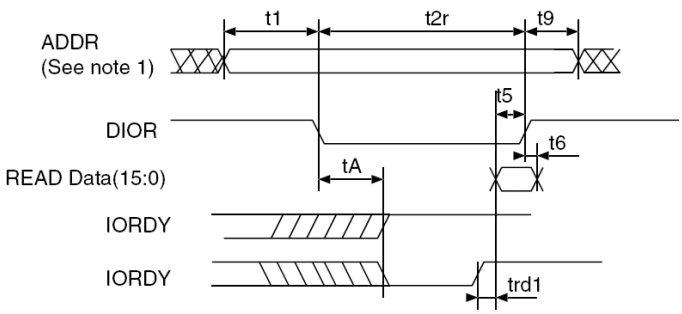


Figure 54-2. PIO Read Mode Timing

To meet timing requirements, a number of timing parameters must be controlled. Table 54-3 shows relations between different timing parameters, and identifies parameters in the PATA timing registers which may be programmed by the user to satisfy timing constraints (see Time Off register (PATA_TIME_OFF).

In Table 54-3 (and Table 54-4 below), the first column lists parameters from the PATA specification; the second column refers to timing parameters shown in Figure 54-2; the third column shows the relation between these timing parameters and programmable values; and the fourth column identifies the registers which may be programmed to meet the timing relations.

Table 54-3. Timing Parameter Relations for PIO Read

PATA parameter	PIO Read Mode Timing Parameter [1]	Relation	Programmable Register
t1	t1	t1(min) = time_1 * T - (tskew1 + tskew2 + tskew5)	time_1
t2 (read)	t2r	t2(min) = time_2r * T - (tskew1 + tskew2 + tskew5)	time_2r
t9	t9	t9(min) = time_9 * T - (tskew1 + tskew2 + tskew6)	time_9
t5	t5	t5(min) = tco + tsu + tbuf + tbuf+ tcable1 + tcable2	time_2 (affects tsu and tco)
tA	tA	tA(min) = (1.5 + time_ax) * T - (tco + tsui + tcable2 + tcable2 + 2*tbuf)	time_ax
trd	trd1	trd1(max) = (-trd)+ (tskew3 + tskew4) trd1(min) = (time_pio_rdx - 0.5)*T - (tsu + thi) (time_pio_rdx - 0.5) * T > tsu + thi + tskew3 + tskew4	time_pio_rdx
t0	-	t0(min) = (time_1 + time_2r+ time_9) * T	time_1, time_2r, time_9

1. See Figure 54-2 .

54.2.2.2.2 PIO Write Mode Timing

A timing diagram for PIO write mode is given in Figure 54-3.

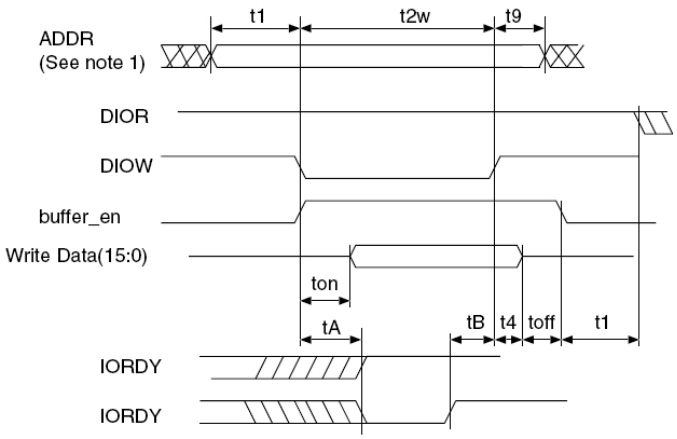


Figure 54-3. PIO Write Mode Timing

Table 54-4 shows relations between timing parameters, and identifies parameters in the PATA timing registers which may be programmed by the user to meet timing constraints (see Time Off register (PATA_TIME_OFF)).

Table 54-4. Timing Parameters Relations for PIO Write

PATA Parameter	PIO Write Mode Timing Parameter[1]	Relation	Programmable Value
t1	t1	t1(min) = time_1 * T - (tskew1 + tskew2 + tskew5)	time_1
t2 (write)	t2w	t2(min) = time_2w * T - (tskew1 + tskew2 + tskew5)	time_2w
t9	t9	t9(min) = time_9 * T - (tskew1 + tskew2 + tskew6)	time_9
t3	-	t3(min) = (time_2w - time_on)* T - (tskew1 + tskew2 +tskew5)	if not met, increase time_2w
t4	t4	t4(min) = time_4 * T - tskew1	time_4
tA	tA	tA = (1.5 + time_ax) * T - (tco + tsui + tcable2 + tcable2 + 2*tbuf)	time_ax
t0	-	t0(min) = (time_1 + time_2 + time_9) * T	time_1, time_2r, time_9
-	-	Avoid bus contention when switching buffer on by making ton long enough	-
-	-	Avoid bus contention when switching buffer off by making toff long enough	-

1. See Figure 54-3 .

54.2.2.3 Timing in Multi-word DMA (MDMA) Mode

Figure 54-4 and Figure 54-5 show read and write timings respectively for MDMA mode.

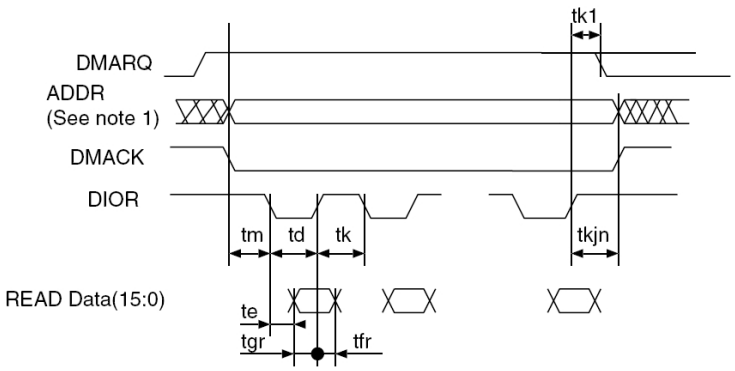


Figure 54-4. MDMA Read Timing

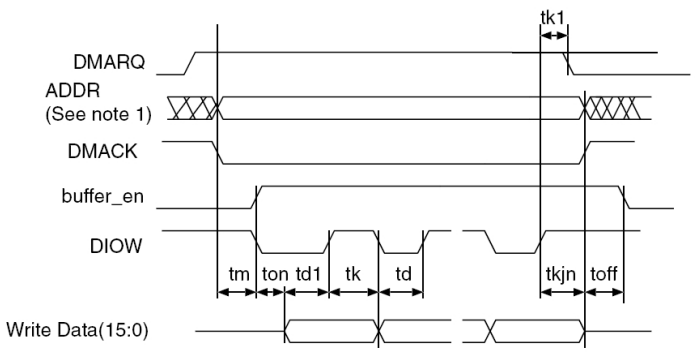


Figure 54-5. MDMA Write Timing

To meet timing requirements for MDMA mode, a number of timing parameters must be controlled. Table 54-5 shows the relations between different timing parameters, and identifies parameters in the PATA timing registers which may be programmed by the user to satisfy timing constraints (see Time Off register (PATA_TIME_OFF)).

In Table 54-5, the first column lists parameters from the PATA specification; the second column refers to timing parameters shown in Figure 54-2; the third column shows the relation between these timing parameters and programmable values; and the fourth column identifies the registers which may be programmed to meet timing constraints.

Table 54-5. Timing Parameter Relations for MDMA Read and Write

PATA Parameter	MDMA Read TIming[1] and MDMA Write TIming[2]	Relation	Programmable Value
tm, ti	tm	tm(min) = ti(min) = time_m * T - (tskew1 + tskew2 + tskew5)	time_m
td	td, td1	td1(min) = td(min) = time_d * T - (tskew1 + tskew2 + tskew6)	time_d
tk	tk[3]	tk(min) = time_k * T - (tskew1 + tskew2 + tskew6)	time_k
t0	-	t0(min) = (time_d + time_k) * T	time_d, time_k
tg (read)	tgr	tgr(min-read) = tco + tsu + tbuf + tbuf + tcable1 + tcable2 tgr(min-drive) = td - te(drive)	time_d
tf (read)	tfr	tfr(min) = 5 ns	(met by design)
tg(write)	-	tg(min-write) = time_d * T -(tskew1 + tskew2 + tskew5)	time_d
tf(write)	-	tf(min-write) = time_k * T - (tskew1 + tskew2 + tskew6)	time_k
tL	-	tL(max) = (time_d + time_k-2)*T - (tsu + tco + 2*tbuf + 2*tcable2)	time_d, time_k[4]
tn, tj	tkjn	tn= tj= tkjn = time_jn * T - (tskew1 + tskew2 + tskew6)	time_jn
-	ton toff	ton = time_on * T - tskew1 toff = time_off * T - tskew1	-

1. See Figure 54-4.
2. See Figure 54-5.
3. tk1 in the MDMA figures (Figure 54-4 and Figure 54-5) equals (tk -2*T)
4. tk1 in the MDMA figures equals (tk -2*T)

54.2.2.4 Timing for Ultra DMA (UDMA) Data In-Transfers

Figure 54-6 shows timing for the start of a UDMA data in-transfer.

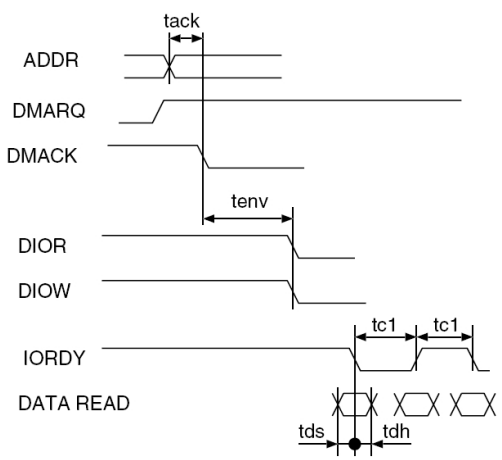


Figure 54-6. Timing for Start of UDMA Data In-transfer

Figure 54-7 shows timing for host termination of a UDMA data in-transfer.

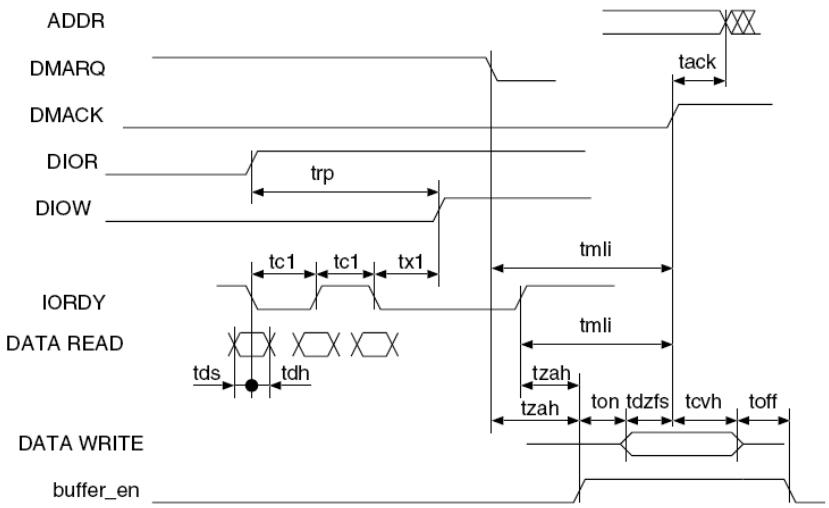


Figure 54-7. Timing for Host Termination of UDMA Transfer

Figure 54-8 shows timing for device termination of a UDMA data in-transfer.

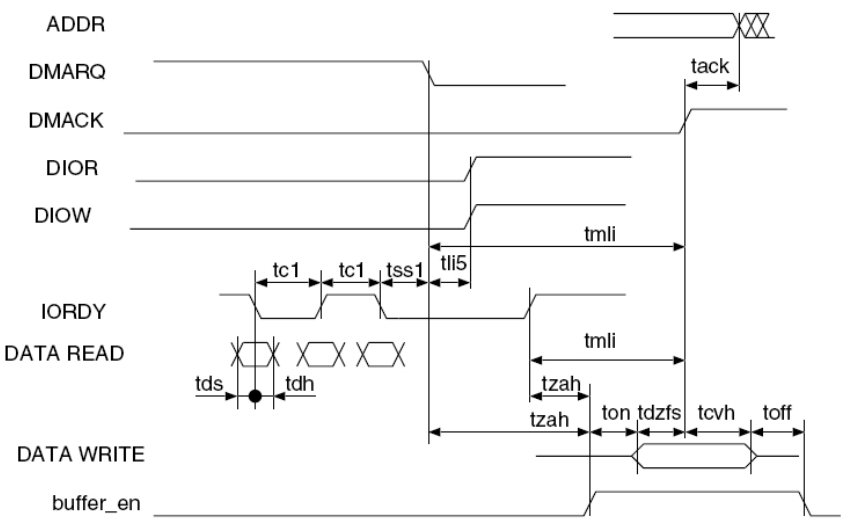


Figure 54-8. Timing for Device Termination of a UDMA Transfer

Timing parameters for UDMA data in-bursts are listed in Table 54-6. The UDMA in-burst timing parameters listed in the second column refer to Figure 54-6 through Figure 54-8.

Table 54-6. Timing Parameter Relations for UDMA Data In-bursts

PATA Parameter	UDMA In-burst Timing Parameter	Relation	Programmable Value
tack	tack	tack(min) = (time_ack * T) - (tskew1 + tskew2)	time_ack

Table continues on the next page...

Table 54-6. Timing Parameter Relations for UDMA Data In-bursts (continued)

PATA Parameter	UDMA In-burst Timing Parameter	Relation	Programmable Value
tenv	tenv	tenv(min) = (time_env * T) - (tskew1 + tskew2) tenv(max) = (time_env * T) + (tskew1 + tskew2)	time_env
trp	trp	trp(min) = time_rp * T - (tskew1 + tskew2 + tskew6)	time_rp
-	tx1[1]	(time_rp * T) - (tco + tsu + 3T + 2 *tbuf + 2*tcable2) > trfs (drive)	time_rp
tmli	tmli1	tmli1(min) = (time_mlix + 0.4) * T	time_mlix
tzah	tzah	tzah(min) = (time_zah + 0.4) * T	time_zah
tdzfs	tdzfs	tdzfs = (time_dzfs * T) - (tskew1 + tskew2)	time_dzfs
tcvh	tcvh	tcvh = (time_cvh *T) - (tskew1 + tskew2)	time_cvh
-	ton toff[2]	ton = time_on * T - tskew1 toff = time_off * T - tskew1	-

1. There is a special timing requirement in the PATA host that requires the internal DIOW to go only high three clocks after the last active edge on the DSTROBE signal. The equation given on this line tries to capture this constraint.
2. Make ton and toff big enough to avoid bus contention.

54.2.2.5 Timing for UDMA Data Out-transfers

Figure 54-9 shows the timing for the start of a UDMA data out-transfer.

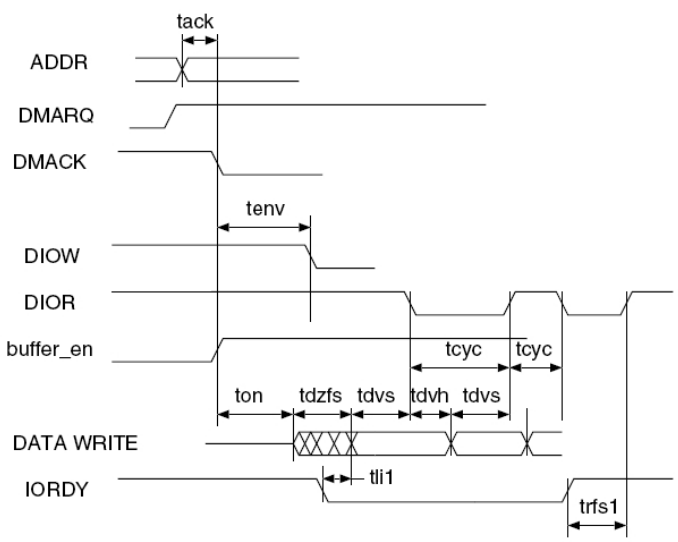


Figure 54-9. Timing of Start of UDMA Data Out-transfer

Figure 54-10 shows the timing for host termination of a UDMA data out-transfer.

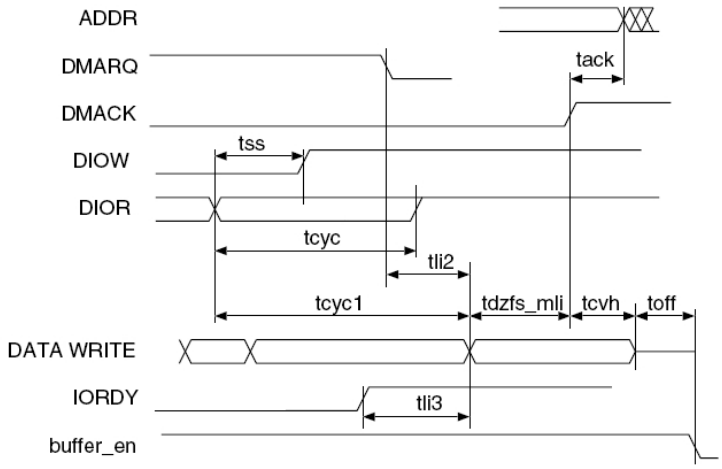


Figure 54-10. Timing of Host Termination of UDMA Data Out-transfer

Figure 54-11 shows timing for device termination of a UDMA data out-transfer.

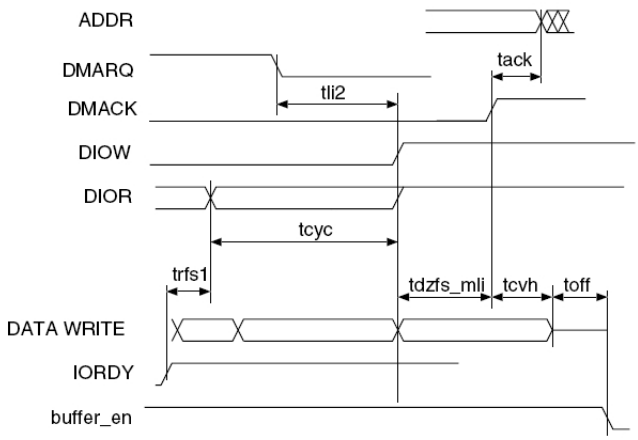


Figure 54-11. Timing for Device Termination of UDMA Data Out-transfer

Timing parameters and relations for UDMA data out-bursts are listed in Table 54-7. The UDMA out-burst timing parameters listed in the second column refer to Figure 54-9 through Figure 54-11.

Table 54-7. Timing Parameters and Relations for UDMA Out Burst

PATA parameter	UDMA Out-burst Timing Parameter	Timing Relation	Programmable Value
tack	tack	tack(min) = (time_ack * T) - (tskew1 + tskew2)	time_ack
tenv	tenv	tenv(min) = (time_env * T) - (tskew1 + tskew2) tenv(max) = (time_env * T) + (tskew1 + tskew2)	time_env
tdvs	tdvs	tdvs = (time_dvs * T) - (tskew1 + tskew2)	time_dvs

Table continues on the next page...

Table 54-7. Timing Parameters and Relations for UDMA Out Burst (continued)

PATA parameter	UDMA Out-burst Timing Parameter	Timing Relation	Programmable Value
tdvh	tdvh	tdvs = (time_dvh * T) - (tskew1 + tskew2)	time_dvh
tcyc	tcyc	tcyc = time_cyc * T - (tskew1 + tskew2)	time_cyc
t2cyc	-	t2cyc = time_cyc * 2 * T	time_cyc
trfs1	trfs	trfs = 1.6 * T + tsui + tco + tbuf + tbuf	-
-	tdzfs	tdzfs = time_dzfs * T - (tskew1)	time_dzfs
tss	tss	tss = time_ss * T - (tskew1 + tskew2)	time_ss
tmli	tdzfs_mli	tdzfs_mli =max(time_dzfs, time_mli) * T - (tskew1 + tskew2)	-
tli	tli1	tli1 > 0	-
tli	tli2	tli2 > 0	-
tli	tli3	tli3 > 0	-
tcvh	tcvh	tcvh = (time_cvh *T) - (tskew1 + tskew2)	time_cvh
-	ton toff	ton = time_on * T - tskew1 toff = time_off * T - tskew1	-

54.3 Functional Description

The PATA interface provides two alternative modes for communication with the PATA peripherals connected to the PATA bus:

- PIO mode read/write operations through the PATA bus
- DMA transfers through the PATA bus

The operation of the peripheral under these two modes is described in detail in subsequent sections.

54.3.1 Resetting the PATA Bus

The PATA bus reset ata_reset_b is asserted whenever bit 'ata_rst_b' (bit #6) of the PATA control register is cleared (see PATA interface control register (PATA_CONTROL)). Clearing the bit also resets the PATA protocol engine. When the bit is set, the reset is released.

54.3.2 Programming PATA Bus Timing and iordy_en

The timing of the PATA interface is programmable, using the 24 timing registers described in the Programmable Registers section. Programming the registers requires that the PATA bus be idle, so before reprogramming the user should make sure that:

- The dma_pending bit in the PATA control register is cleared (see PATA interface control register (PATA_CONTROL)).
- The controller_idle bit in the interrupt pending register is set (see Interrupt pending register (PATA_INTERRUPT_PENDING)).

These two conditions can be met by first clearing dma_pending, and then waiting until controller_idle is set before reprogramming the timing parameters. If dma_pending was set before the reprogramming started, it should be set again after the new timing is in effect to allow the drive to finish the current DMA transfer.

The bus timing should only be reprogrammed during an ongoing DMA transfer when the operating system requires a change in the bus clock (for example, in dynamic voltage frequency scaling).

The controller_idle bit must be set before reprogramming the bus timing: otherwise, the new timing values may affect a bus cycle that is still running and cause a error. PIO reads or writes to the PATA bus terminate after the bus cycle with the ARM platform has been terminated.

The iordy_en bit in the PATA control register determines whether the PATA interface responds to the drive's IORDY signal. Just as with timing registers, it should only be reprogrammed when dma_pending is cleared, and controller_idle is set.

54.3.3 Access to PATA Bus in PIO Mode

Before accessing the PATA bus in PIO mode, the user must:

1. Set the ata_rst_b bit in the PATA control register
2. Program the timing parameters

The drive is accessed in PIO mode simply by reading or writing to the correct drive register. The bus cycle is translated to an PATA cycle, and the drive is accessed.

Reads and writes to the drive are not possible while the PATA bus is in reset; the attempted operation fails, and is discarded.

54.3.4 Receiving Data from PATA Bus in DMA Mode

In DMA receive mode, the protocol engine transfers data from the drive to the FIFO using the multi-word DMA (MDMA) or ultra DMA (UMDMA) protocol. The transfer pauses when any of the following conditions occurs:

- The FIFO is full.
- The drive negates its DMA request signal ata_dmarq.
- The dma_pending bit in the ata_control register is cleared.

When the condition is removed, the transfer restarts. At the end of the transfer, the drive signals the host by asserting the ata_intrq signal. Alternatively the host can read the device status register, which also indicates when the transfer has ended.

The transfer of data from FIFO to memory is handled by the host system DMA. DMA data transfers from device to host are set up as follows:

1. Make sure the PATA bus is not in reset, and all timing registers are programmed.
2. Make sure the FIFO is empty by reading it until empty or by resetting it.
3. Initialize the DMA channel connected to fifo_rcv_alarm. Every time the fifo_rcv_alarm is high, the DMA should read <packetsize> 32-bit words from the FIFO, and store them in main memory (here 'packetsize' is defined as the number of 32-bit words in a packet: typically packetsize=8). This keeps the transfer going while avoiding FIFO overrun.
4. Write 2*<packetsize> to the FIFO alarm register (note that this setting corresponds to one packet of data, since the alarm threshold is specified in units of 16-bit halfwords). In this way, the FIFO notifies the DMA whenever there is at least one packet ready for transfer.
5. Ensure the FIFO is out of reset by setting the fifo_rst_b bit in the PATA control register (see PATA interface control register (PATA_CONTROL)).
6. Set the fifo_rcv_en bit in the PATA control register. This enables the DMA to empty the FIFO.
7. Set the dma_pending bit, and clear the dma_write bit. Also, program ultra_mode_selected=0 (for MDMA) or ultra_mode_selected=1 (for UDMA).

   Steps (1-7) have prepared the host side of the DMA.

8. Send commands to the drive in PIO mode that cause it to request DMA transfer on the PATA bus (consult the PATA specification for the specifics of this operation).
9. At this point, when the drive requests a DMA transfer by pulling ata_dmarq high, the PATA interface acknowledges with ata_dmack, and the transfer starts. Data is transferred automatically to the FIFO, and from there to the host memory.
10. During the transfer, the host can monitor for end of transfer by reading some device PATA registers. These reads cause the running DMA to pause; after the read is

completed, the DMA resumes. Alternatively, the host can wait unit the drive asserts ata_intrq, which also indicates end of transfer.

11. When the transfer ends, the host or host DMA should wait until the controller_idle bit is set in the PATA control register, then read the remaining halfwords from the FIFO and transfer these to memory. Note that there may be less than <packetsize> remaining 32-bit words, in which case the transfer is not be performed automatically by the DMA.

54.3.5 Transmitting Data to PATA Bus in DMA Mode

In DMA transmit mode, the protocol engine transfers data from the FIFO to the drive using the multi-word DMA (MDMA) or ultra DMA (UDMA) protocol. The transfer pauses when one of following conditions occurs:

- The FIFO is empty
- The drive negates its DMA request signal ata_dmarq
- The dma_pending bit in the ata_control register is cleared

When the condition is removed, the transfer restarts. At the end of the transfer, the drive signals the host by asserting the ata_intrq signal. Alternatively the host can read the device status register, which also indicates when the transfer has ended.

The transfer of data from FIFO to memory is handled by the host system DMA. DMA data transfers from host to device are set up as follows:

1. Make sure the PATA bus is not in reset, and all timing registers are programmed.
2. Make sure the FIFO is empty by reading it until empty, or by resetting it.
3. Initialize the DMA channel connected to fifo_tx_alarm. Every time the fifo_tx_alarm signal is high, the DMA should read <packetsize> 32-bit words from the main memory, and write them to the FIFO (typical packetsize is 8 32-bit words). Program the DMA such that it does not transfer more than <sectorsize> 32-bit words in total.
4. Write FIFO_SIZE - 2*<packetsize> to the FIFO alarm register. In this way, FIFO notifies the DMA when there is room for at least one additional packet. FIFO_SIZE is given in 16-bit halfwords: the typical value is 64 halfwords, or 128 bytes.
5. Prepare the PATA for a DMA transfer from host to device as follows:
   - Make sure the FIFO is out of reset by setting bit fifo_rst_b in the ata control register (see PATA interface control register (PATA_CONTROL)).
   - Program fifo_tx_en=1 in the PATA control register. This enables the FIFO to be filled by DMA.
   - Program dma_pending =1 and dma_write=1. Also, program ultra_mode_selected =0 (for MDMA) or ultra_mode_selected=1 (for UDMA).

Steps (1-5) have prepared the host side of the DMA.

6. Send commands to the drive in PIO mode that cause it to request DMA transfer on the PATA bus (consult the PATA specification for the specifics of this operation).
7. At this point, when the drive requests a DMA transfer by pulling ata_dmarq high, the PATA interface acknowledges with ata_dmack, and the transfer starts. Data is transferred automatically from the FIFO, and also from host memory to FIFO.
8. During the transfer, the host can monitor for end of transfer by reading some device PATA registers. These reads cause the running DMA to pause; after the read is completed, the DMA resumes. Alternatively, the host can wait unit the drive asserts ata_intrq, which also indicates end of transfer.

When the transfer ends, no additional FIFO manipulations are needed.

54.4 Initialization and Application of PATA

If the host asserts RESET the PATA device executes the hardware reset protocol, regardless of power management mode. For more details about the power-on and hardware reset protocol, refer to the PATA specification).

The host issues an IDENTIFY DEVICE command after the power-on reset or hardware reset protocol has been completed, in order to determine the current status of features implemented by the device.

54.5 Programmable Registers

This section contains the detailed register descriptions for the PATA registers.

The PATA interface works both in little-endian or big-endian mode. The addresses of all registers are the same in either mode. The 16-bit and 32-bit registers represent strings of 2 or 4 bytes. The byte order in the 16-bit or 32-bit register is dependent on the mode selection.

- Little-endian mode, 16- or 32-bit register:
  - bits [7:0]: byte 0
  - bits [15:8]: byte 1
  - bits [23:8]: byte 2
  - bits [31:24]: byte 3
- Big-endian mode, 32-bit register
  - bits [31:24]: byte 0

    - bits [23:16]: byte 1
    - bits [15:8]: byte 2
    - bits [7:0]: byte 3
- Big-endian, 16-bit register
    - bits [15:8]: byte 0
    - bits [7:0]: byte 1

The registers at offsets 0x00 through 0x17 contain timing parameters. These timing parameters control the timing on the PATA bus.

Every timing parameter is 8 bits wide and can assume valid values between 0x01 and 0xFF. The reset value for all registers is 0x01.

The three interrupt registers control the interrupt interface between the PATA block and the ARM platform/DMA. Two interrupts (ipbus_int and fifo_txfe_end_alarm) are controlled by these registers, as follows:

- The ipbus_int interrupt is controlled by bits 3,4,5 and 6 of the interrupt registers. It is asserted if one of the 4 bits is set in the interrupt_pending register, while the same bit is set in the interrupt_enable register. This interrupt goes to the ARM platform.
- The fifo_txfer_end_alarm interrupt is controlled by bit 7 of the interrupt registers. If ata_intrq1 is set in both the interrupt enable and interrupt pending register, then fifo_txfer_end_alarm is asserted. The purpose of this interrupt is to inform the DMA that the running data transfer has ended. This interrupt goes to the smart DMA.

PATA memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5003_0000	Time Off register (PATA_TIME_OFF)	8	R/W	01h	54.5.1/ 3685
5003_0001	Time On register (PATA_TIME_ON)	8	R/W	01h	54.5.2/ 3685
5003_0002	Time 1register (PATA_TIME_1)	8	R/W	01h	54.5.3/ 3686
5003_0003	Time 2W register (PATA_TIME_2W)	8	R/W	01h	54.5.4/ 3686
5003_0004	Time 2R register (PATA_TIME_2R)	8	R/W	01h	54.5.5/ 3686
5003_0005	Time AX register (PATA_TIME_AX)	8	R/W	01h	54.5.6/ 3687
5003_0006	Time PIO RDX register (PATA_TIME_PIO_RDX)	8	R/W	01h	54.5.7/ 3687
5003_0007	Time 4 register (PATA_TIME_4)	8	R/W	01h	54.5.8/ 3688

Table continues on the next page...

PATA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5003_0008	Time 9 register (PATA_TIME_9)	8	R/W	01h	54.5.9/ 3688
5003_0009	Time M register (PATA_TIME_M)	8	R/W	01h	54.5.10/ 3688
5003_000A	Time JN register (PATA_TIME_JN)	8	R/W	01h	54.5.11/ 3689
5003_000B	Time D register (PATA_TIME_D)	8	R/W	01h	54.5.12/ 3689
5003_000C	Time K register (PATA_TIME_K)	8	R/W	01h	54.5.13/ 3690
5003_000D	Time ACK register (PATA_TIME_ACK)	8	R/W	01h	54.5.14/ 3690
5003_000E	Time ENV register (PATA_TIME_ENV)	8	R/W	01h	54.5.15/ 3690
5003_000F	Time RPX register (PATA_TIME_RPX)	8	R/W	01h	54.5.16/ 3691
5003_0010	Time ZAH register (PATA_TIME_ZAH)	8	R/W	01h	54.5.17/ 3691
5003_0011	Time MLIX register (PATA_TIME_MLIX)	8	R/W	01h	54.5.18/ 3692
5003_0012	Time DVH register (PATA_TIME_DVH)	8	R/W	01h	54.5.19/ 3692
5003_0013	Time DZFS register (PATA_TIME_DZFS)	8	R/W	01h	54.5.20/ 3692
5003_0014	Time DVS register (PATA_TIME_DVS)	8	R/W	01h	54.5.21/ 3693
5003_0015	Time CVH register (PATA_TIME_CVH)	8	R/W	01h	54.5.22/ 3693
5003_0016	Time SS register (PATA_TIME_SS)	8	R/W	01h	54.5.23/ 3694
5003_0017	Time CYC register (PATA_TIME_CYC)	8	R/W	01h	54.5.24/ 3694
5003_0018	FIFO Data register 32-bit (PATA_FIFO_DATA_32)	32	R/W	0000_0000h	54.5.25/ 3694
5003_001C	FIFO Data register 16-bit (PATA_FIFO_DATA_16)	16	R/W	0000h	54.5.26/ 3695
5003_0020	FIFO FILL register (PATA_FIFO_FILL)	8	R	00h	54.5.27/ 3695
5003_0024	PATA interface control register (PATA_CONTROL)	16	R/W	0000h	54.5.28/ 3696

Table continues on the next page...

PATA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5003_0028	Interrupt pending register (PATA_INTERRUPT_PENDING)	8	R	10h	54.5.29/ 3697
5003_002C	Interrupt enable register (PATA_INTERRUPT_ENABLE)	8	R/W	00h	54.5.30/ 3698
5003_0030	Interrupt clear register (PATA_INTERRUPT_CLEAR)	8	W	00h	54.5.31/ 3698
5003_0034	FIFO ALARM register (PATA_FIFO_ALARM)	8	R/W	00h	54.5.32/ 3699

54.5.1 Time Off register (PATA_TIME_OFF)

Address: PATA_TIME_OFF is 5003_0000h base + 0h offset = 5003_0000h

Bit	7	6	5	4	3	2	1	0
Read / Write	TIME_OFF							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_OFF field descriptions

Field	Description
7–0 TIME_OFF	transceiver timing parameter. Controls toff

54.5.2 Time On register (PATA_TIME_ON)

Address: PATA_TIME_ON is 5003_0000h base + 1h offset = 5003_0001h

Bit	7	6	5	4	3	2	1	0
Read / Write	TIME_ON							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_ON field descriptions

Field	Description
7–0 TIME_ON	transceiver timing parameter. Controls ton

54.5.3 Time 1register (PATA_TIME_1)

Address: PATA_TIME_1 is 5003_0000h base + 2h offset = 5003_0002h

Bit	7	6	5	4	3	2	1	0
Read Write	TIME_1							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_1 field descriptions

Field	Description
7–0 TIME_1	PIO timing parameter. Controls t1

54.5.4 Time 2W register (PATA_TIME_2W)

Address: PATA_TIME_2W is 5003_0000h base + 3h offset = 5003_0003h

Bit	7	6	5	4	3	2	1	0
Read Write	TIME_2W							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_2W field descriptions

Field	Description
7–0 TIME_2W	PIO timing parameter. Controls t2 during write cycles

54.5.5 Time 2R register (PATA_TIME_2R)

Address: PATA_TIME_2R is 5003_0000h base + 4h offset = 5003_0004h

Bit	7	6	5	4	3	2	1	0
Read Write	TIME_2R							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_2R field descriptions

Field	Description
7–0 TIME_2R	PIO timing parameter. Controls t2 during read cycles

54.5.6 Time AX register (PATA_TIME_AX)

Address: PATA_TIME_AX is 5003_0000h base + 5h offset = 5003_0005h

Bit	7	6	5	4	3	2	1	0
Read Write	TIME_AX							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_AX field descriptions

Field	Description
7–0 TIME_AX	PIO timing parameter. Controls tA

54.5.7 Time PIO RDX register (PATA_TIME_PIO_RDX)

Address: PATA_TIME_PIO_RDX is 5003_0000h base + 6h offset = 5003_0006h

Bit	7	6	5	4	3	2	1	0
Read Write	TIME_RDX							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_PIO_RDX field descriptions

Field	Description
7–0 TIME_RDX	PIO timing parameter. Controls trd

54.5.8 Time 4 register (PATA_TIME_4)

Address: PATA_TIME_4 is 5003_0000h base + 7h offset = 5003_0007h

Bit	7	6	5	4	3	2	1	0
Read / Write	TIME_4							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_4 field descriptions

Field	Description
7–0 TIME_4	PIO timing parameter. Controls t4

54.5.9 Time 9 register (PATA_TIME_9)

Address: PATA_TIME_9 is 5003_0000h base + 8h offset = 5003_0008h

Bit	7	6	5	4	3	2	1	0
Read / Write	TIME_9							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_9 field descriptions

Field	Description
7–0 TIME_9	PIO timing parameter. Controls t9

54.5.10 Time M register (PATA_TIME_M)

Address: PATA_TIME_M is 5003_0000h base + 9h offset = 5003_0009h

Bit	7	6	5	4	3	2	1	0
Read / Write	TIME_M							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_M field descriptions

Field	Description
7–0 TIME_M	MDMA timing parameter. Controls tm

54.5.11 Time JN register (PATA_TIME_JN)

Address: PATA_TIME_JN is 5003_0000h base + Ah offset = 5003_000Ah

Bit	7	6	5	4	3	2	1	0
Read Write	TIME_JN							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_JN field descriptions

Field	Description
7–0 TIME_JN	MDMA timing parameter. Controls tn and tj

54.5.12 Time D register (PATA_TIME_D)

Address: PATA_TIME_D is 5003_0000h base + Bh offset = 5003_000Bh

Bit	7	6	5	4	3	2	1	0
Read Write	TIME_D							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_D field descriptions

Field	Description
7–0 TIME_D	MDMA timing parameter. Controls td

54.5.13 Time K register (PATA_TIME_K)

Address: PATA_TIME_K is 5003_0000h base + Ch offset = 5003_000Ch

Bit	7	6	5	4	3	2	1	0
Read / Write	TIME_K							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_K field descriptions

Field	Description
7–0 TIME_K	MDMA timing parameter. Controls tk

54.5.14 Time ACK register (PATA_TIME_ACK)

Address: PATA_TIME_ACK is 5003_0000h base + Dh offset = 5003_000Dh

Bit	7	6	5	4	3	2	1	0
Read / Write	TIME_ACK							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_ACK field descriptions

Field	Description
7–0 TIME_ACK	UDMA timing parameter. Controls tack

54.5.15 Time ENV register (PATA_TIME_ENV)

Address: PATA_TIME_ENV is 5003_0000h base + Eh offset = 5003_000Eh

Bit	7	6	5	4	3	2	1	0
Read / Write	TIME_ENV							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_ENV field descriptions

Field	Description
7–0 TIME_ENV	UDMA timing parameter. Controls tenv

54.5.16 Time RPX register (PATA_TIME_RPX)

Address: PATA_TIME_RPX is 5003_0000h base + Fh offset = 5003_000Fh

Bit	7	6	5	4	3	2	1	0
Read Write	TIME_RPX							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_RPX field descriptions

Field	Description
7–0 TIME_RPX	UDMA timing parameter. Controls trp

54.5.17 Time ZAH register (PATA_TIME_ZAH)

Address: PATA_TIME_ZAH is 5003_0000h base + 10h offset = 5003_0010h

Bit	7	6	5	4	3	2	1	0
Read Write	TIME_ZAH							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_ZAH field descriptions

Field	Description
7–0 TIME_ZAH	UDMA timing parameter. Controls tzah

54.5.18 Time MLIX register (PATA_TIME_MLIX)

Address: PATA_TIME_MLIX is 5003_0000h base + 11h offset = 5003_0011h

Bit	7	6	5	4	3	2	1	0
Read / Write	TIME_MLIX							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_MLIX field descriptions

Field	Description
7–0 TIME_MLIX	UDMA timing parameter. Controls tmli

54.5.19 Time DVH register (PATA_TIME_DVH)

Address: PATA_TIME_DVH is 5003_0000h base + 12h offset = 5003_0012h

Bit	7	6	5	4	3	2	1	0
Read / Write	TIME_DVH							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_DVH field descriptions

Field	Description
7–0 TIME_DVH	UDMA timing parameter. Controls tdvh

54.5.20 Time DZFS register (PATA_TIME_DZFS)

Address: PATA_TIME_DZFS is 5003_0000h base + 13h offset = 5003_0013h

Bit	7	6	5	4	3	2	1	0
Read / Write	TIME_DZFS							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_DZFS field descriptions

Field	Description
7–0 TIME_DZFS	UDMA timing parameter. Controls tdzfs

54.5.21 Time DVS register (PATA_TIME_DVS)

Address: PATA_TIME_DVS is 5003_0000h base + 14h offset = 5003_0014h

Bit	7	6	5	4	3	2	1	0
Read Write	TIME_DVS							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_DVS field descriptions

Field	Description
7–0 TIME_DVS	UDMA timing parameter. Controls tdvs

54.5.22 Time CVH register (PATA_TIME_CVH)

Address: PATA_TIME_CVH is 5003_0000h base + 15h offset = 5003_0015h

Bit	7	6	5	4	3	2	1	0
Read Write	TIME_CVH							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_CVH field descriptions

Field	Description
7–0 TIME_CVH	UDMA timing parameter. Controls tcvh

54.5.23 Time SS register (PATA_TIME_SS)

Address: PATA_TIME_SS is 5003_0000h base + 16h offset = 5003_0016h

Bit	7	6	5	4	3	2	1	0
Read / Write	TIME_SS							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_SS field descriptions

Field	Description
7–0 TIME_SS	UDMA timing parameter. Controls tss

54.5.24 Time CYC register (PATA_TIME_CYC)

Address: PATA_TIME_CYC is 5003_0000h base + 17h offset = 5003_0017h

Bit	7	6	5	4	3	2	1	0
Read / Write	TIME_CYC							
Reset	0	0	0	0	0	0	0	1

PATA_TIME_CYC field descriptions

Field	Description
7–0 TIME_CYC	UDMA timing parameter. Controls tcyc and t2cyc

54.5.25 FIFO Data register 32-bit (PATA_FIFO_DATA_32)

The FIFO_DATA register is used to read or write data to' the internal FIFO. It can be accessed as a 16-bit register or as a 32-bit register. Word writes (respectively long writes) put two bytes (respectively four bytes) into the FIFO. Word reads (respectively long reads) read two bytes (respectively four bytes) from the FIFO.

Address: PATA_FIFO_DATA_32 is 5003_0000h base + 18h offset = 5003_0018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R / W	FIFO_DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PATA_FIFO_DATA_32 field descriptions

Field	Description
31–0 FIFO_DATA	32 bit wide data port to/from FIFO

54.5.26 FIFO Data register 16-bit (PATA_FIFO_DATA_16)

The FIFO_DATA register is used to read or write data to the internal FIFO. It can be accessed as a 16-bit register or as a 32-bit register. Word writes (respectively long writes) put two bytes (respectively four bytes) into the FIFO. Word reads (respectively long reads) read two bytes (respectively four bytes) from the FIFO.

Address: PATA_FIFO_DATA_16 is 5003_0000h base + 1Ch offset = 5003_001Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	FIFO_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PATA_FIFO_DATA_16 field descriptions

Field	Description
15–0 FIFO_DATA	16 bit wide data port to/from FIFO

54.5.27 FIFO FILL register (PATA_FIFO_FILL)

FIFO_FILL is a read-only register. Any read returns the current number of halfwords present in the FIFO.

Address: PATA_FIFO_FILL is 5003_0000h base + 20h offset = 5003_0020h

Bit	7	6	5	4	3	2	1	0
Read	FIFO_FILL							
Write								
Reset	0	0	0	0	0	0	0	0

PATA_FIFO_FILL field descriptions

Field	Description
7–0 FIFO_FILL	FIFO filling in halfwords

54.5.28 PATA interface control register (PATA_CONTROL)

Address: PATA_CONTROL is 5003_0000h base + 24h offset = 5003_0024h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	-								fifo_rst_b	ata_rst_b	fifo_tx_en	fifo_rcv_en	dma_pending	dma_ultra_selected	dma_write	iordy_en
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PATA_CONTROL field descriptions

Field	Description
15–8 -	Reserved.
7 fifo_rst_b	This field controls the internal FIFO reset 0 FIFO reset 1 FIFO normal operation
6 ata_rst_b	This bit controls the level on the ata_reset_b pin, which controls the reset of the internal PATA protocol engine. 0 ata_reset_b = 0, PATA drive is reset, and internal protocol engine reset. 1 ata_reset_b = 1, PATA drive is not reset; internal protocol engine normal operation.
5 fifo_tx_en	FIFO transmit enable. This bit controls whether the FIFO makes transmit data requests to the DMA. If enabled, the FIFO requests the DMA to refill it whenever the FIFO level drops below the alarm threshold. 0 FIFO refill by DMA disabled 1 FIFO refill by DMA enabled
4 fifo_rcv_en	FIFO receive enable. This bit controls whether the FIFO makes receive data requests to the DMA. If enabled, the FIFO requests the DMA to empty it whenever the FIFO level meets or exceeds the alarm threshold. 0 FIFO empty by DMA disabled 1 FIFO empty by DMA enabled
3 dma_pending	DMA pending bit. This bit controls whether the PATA interface responds to a DMA request originating in the drive. If this bit is set, the PATA interface starts a MDMA or UDMA burst whenever the drive asserts the ata_dmarq signal. 0 PATA interface does not start DMA burst 1 PATA interface starts MDMA or UDMA burst whenever drive asserts dmarq
2 dma_ultra_selected	This bit determines the protocol (UDMA or MDMA) for any new DMA burst 1 UDMA protocol is used 0 MDMA protocol is used
1 dma_write	This bit determines the data direction on any new DMA burst

Table continues on the next page...

PATA_CONTROL field descriptions (continued)

Field	Description
	1 DMA out burst, PATA interface writes to drive 0 DMA in burst, PATA interface reads from drive
0 iordy_en	This bit determines whether ata_iordy handshake is used during PIO mode 1 IORDY handshake is used 0 IORDY is disregarded

54.5.29 Interrupt pending register (PATA_INTERRUPT_PENDING)

Address: PATA_INTERRUPT_PENDING is 5003_0000h base + 28h offset = 5003_0028h

Bit	7	6	5	4	3	2	1	0
Read	ata_intrq1	fifo_ underflow	fifo_overflow	controller_idle	ata_intrq2	Reserved		
Write								
Reset	0	0	0	1	0	0	0	0

PATA_INTERRUPT_PENDING field descriptions

Field	Description
7 ata_intrq1	PATA interrupt request 1. This bit reflects the value of the ata_intrq interrupt input. It is set in the when the drive interrupt is pending, and cleared otherwise. The interrupt clear register has no influence on this bit.
6 fifo_underflow	FIFO underflow. This bit reports FIFO underflow. Sticky bit. It is set when there is a FIFO underflow condition. It is cleared by writing a '1' to this bit in the interrupt clear register.
5 fifo_overflow	FIFO overflow. This bit reports FIFO overflow. Sticky bit. It is set when there is a FIFO overflow condition. It is cleared by writing a '1' to this bit in the interrupt clear register.
4 controller_idle	Controller Idle. This bit reports controller idle. It is set when the PATA protocol engine is idle, there is no activity on the PATA bus. It is cleared when there is activity on the PATA bus. The interrupt clear register has no influence on this bit.
3 ata_intrq2	PATA interrupt request 2. This bit reflects the value of the ata_intrq interrupt input. It is set when the drive interrupt is pending, and cleared otherwise. It has exactly same functioning as ata_intrq1, but this bit affects ipbus_int, while the other affects interrupt to the DMA.
2–0 Reserved	This field is reserved. -

54.5.30 Interrupt enable register (PATA_INTERRUPT_ENABLE)

Address: PATA_INTERRUPT_ENABLE is 5003_0000h base + 2Ch offset = 5003_002Ch

Bit	7	6	5	4	3	2	1	0
Read	ata_intrq1	fifo_ underflow	fifo_overflow	controller_idle	ata_intrq2	Reserved		
Write								
Reset	0	0	0	0	0	0	0	0

PATA_INTERRUPT_ENABLE field descriptions

Field	Description
7 ata_intrq1	PATA interrupt request 1. If this bit is set, then fifo_txfer_end_alarm is driven to the DMA when the corresponding bit is set in the interrupt enable register. This informs the DMA that the current transfer is finished. The interrupt clear register has no influence on this bit.
6 fifo_underflow	FIFO underflow. If this bit is set, then ipbus_int is driven to the ARM platform when the corresponding bit is set in the interrupt enable register.
5 fifo_overflow	FIFO overflow. If this bit is set, then ipbus_int is driven to the ARM platform when the same bit is set in the interrupt enable register.
4 controller_idle	Controller Idle. If this bit is set, then ipbus_int is driven to the ARM platform when the same bit is set in the interrupt enable register. The interrupt clear register has no influence on this bit.
3 ata_intrq2	PATA interrupt request 2. This bit reflects the value of the ata_intrq interrupt input. If this bit is set, then ipbus_int is driven to the ARM platform when the same bit is set in the interrupt enable register. This informs the ARM platform that the drive is requesting attention. The interrupt clear register has no influence on this bit.
2–0 Reserved	This field is reserved. -

54.5.31 Interrupt clear register (PATA_INTERRUPT_CLEAR)

Address: PATA_INTERRUPT_CLEAR is 5003_0000h base + 30h offset = 5003_0030h

Bit	7	6	5	4	3	2	1	0
Read	Reserved			Reserved				
Write		fifo_ underflow	fifo_overflow					
Reset	0	0	0	0	0	0	0	0

PATA_INTERRUPT_CLEAR field descriptions

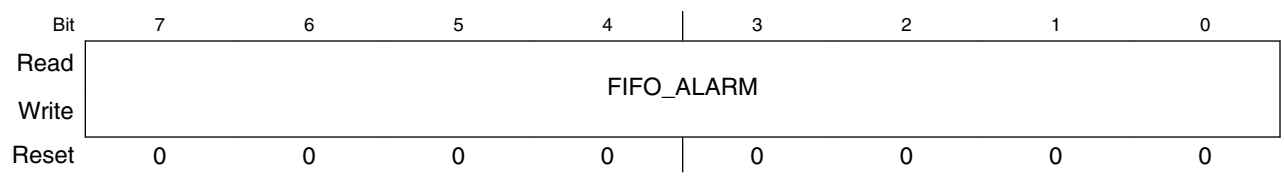
Field	Description
7 Reserved	This field is reserved. -
6 fifo_underflow	FIFO underflow. Writing '1' to this bit clears the corresponding bit in the interrupt pending register.
5 fifo_overflow	FIFO overflow. Writing '1' to this bit clears the corresponding bit in the interrupt pending register.
4–0 Reserved	This field is reserved. -

54.5.32 FIFO ALARM register (PATA_FIFO_ALARM)

This register contains the threshold (in 16-bit halfword units) which triggers fifo_rcv_alarm or fifo_tx_alarm signals to the DMA interface.

- If the fifo_rcv_en bit is set in the PATA control register and fifo_fill ≥ fifo_alarm, then the fifo_rcv_alarm signal is asserted to request the DMA to empty the FIFO.
- If the fifo_tx_en bit is set in the PATA control register and fifo_fill < fifo_alarm), then the fifo_tx_alarm signal is asserted to request the DMA to refill the FIFO.

Address: PATA_FIFO_ALARM is 5003_0000h base + 34h offset = 5003_0034h



PATA_FIFO_ALARM field descriptions

Field	Description
7–0 FIFO_ALARM	FIFO alarm threshold

54.5.33 Drive Registers Connected to PATA Bus

Some drive registers are addressable, but are not present in the PATA interface block. These are listed in Table 54-41. If a read or write access is made to one of these registers, the read or write is mapped to a PIO read or write cycle on the PATA bus, and the corresponding register in the device attached to the PATA bus is accessed. No description of these registers is given here-consult the PATA specification for information on these registers.

If the drive_dataregister is accessed while the PATA interface operates in big-endian mode, the bytes to/from the PATA bus are swapped. No swaps occur in little-endian mode, or for the other registers.

Table 54-41. Drive Registers connected to PATA Bus

Address	Name	Description	Access
Offset: A0 (DRIVE_DATA)	drive_data	Drive data register	R/W
Offset: A4 (DRIVE_FEATURES)	drive_features	Drive features register	R/W
Offset: A8 (DRIVE_SECTOR_ COUNT)	drive_sector_coun t	Drive sector count register	R/W
Offset: AC (DRIVE_SECTOR_ NUM)	drive_sector_num	Drive sector number register	R/W
Offset: B0 (DRIVE_CYL_LOW)	drive_cyl_low	Drive cylinder low register	R/W
Offset: B4 (DRIVE_CYL_HIGH)	drive_cyl_high	Drive cylinder high register	R/W
Offset: B8 (DRIVE_DEV_HEAD)	drive_dev_head	Drive device head register	R/W
Offset: BC (DRIVE_COMMAND)	drive_command	Drive command register	W
Offset: BC (DRIVE_STATUS)	drive_status	Drive status register	R
Offset: D8 (DRIVE_ALT_STATUS)	drive_alt_status	Drive alternate status register	R
Offset: D8 (DRIVE_CONTROL)	drive_control	Drive control register	W

Chapter 55
Power Fail Detector (PFD)

55.1 Introduction

Figure 55-1 shows the block diagram of PFD.

55.1.1 Overview

The PFD is a sub-block of the Secure Real Time Clock (SRTC). PFD circuit generates a reset signal till its supply vdd reaches a predetermined voltage level.The PFD circuit works only when the chip is in the Non-DSM mode. This means that when DSM is asserted high then there is no activity on the Power_Fail line and it is pulled high by the logic in the circuit. The back bone of the circuit is the resistance ladder and an inverter (shown as detector in Figure 55-1). The input to the inverter comes from one of the tap points in the resistance ladder.

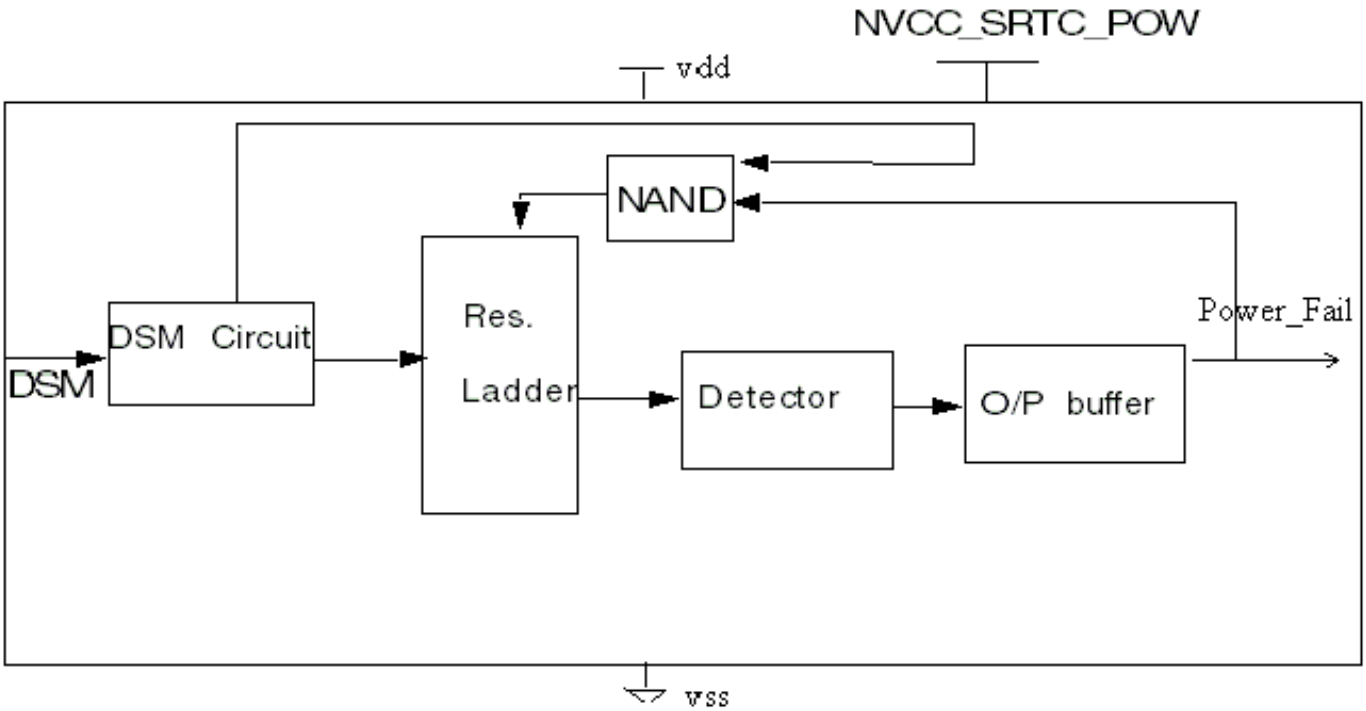


Figure 55-1. PFD Block Diagram

55.1.1.1 Assumptions

The NVCC_SRTC_POW voltage supply is nominally 1.2V. However, the PFD circuit has been simulated taking into account a 50mV variation in the NVCC_SRTC_POW voltage. With a 50mV variation on this supply, the threshold voltage levels at the same PVT corner exhibit a 35mV variation.

55.1.2 Features

The PFD has the following features:

- De-assertion voltage independent of vdd ramp-up rate
- Very Low current consumption
- Brown Out (BO) voltage independent of vdd ramp-down rate
- Negligible variation in threshold over Process, Voltage and Temperature (PVT) variations

55.1.3 Modes of Operation

There are three basic modes of operation: LPM, HPM and DSM. In the DSM mode, the PFD circuit does not function and the logic in the circuit ensures that the Power_Fail output is pulled high.

In the LPM and the HPM mode the PFD circuit functions as follows:

The PFD generates a Power_Fail active low pulse when the chip is powering up. When vdd ramps up and reaches a pre-determined threshold then the output signal Power_Fail goes low (meaning that there is no Power Failure at this time)

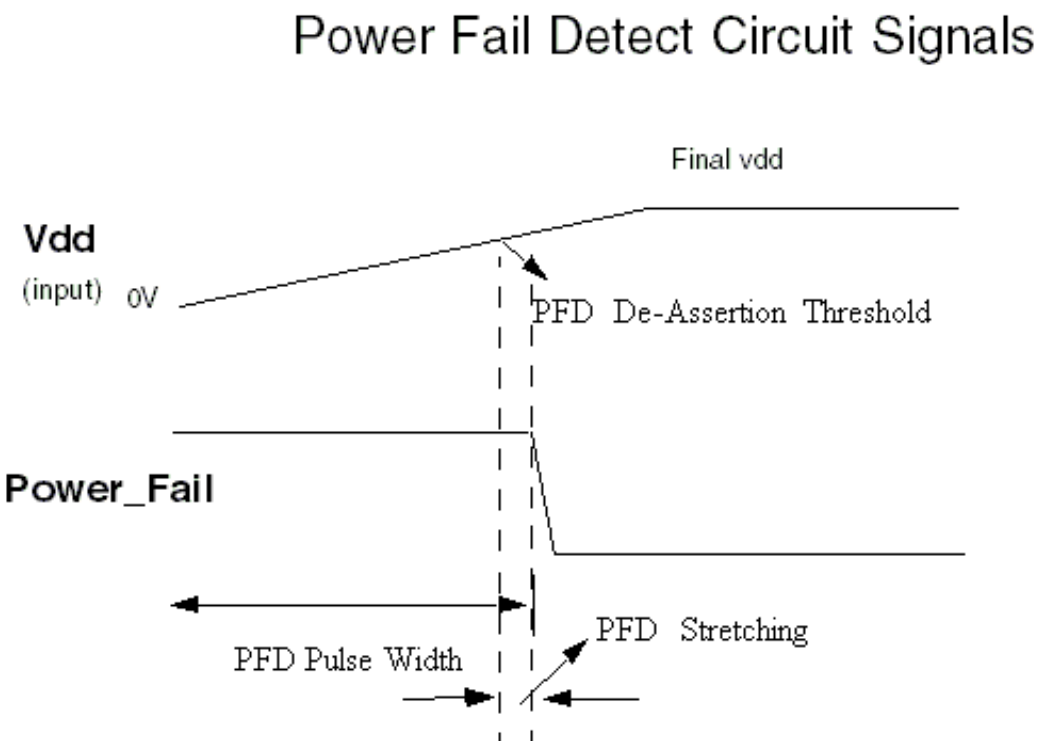


Figure 55-2. Power_Fail De-Assert Detection

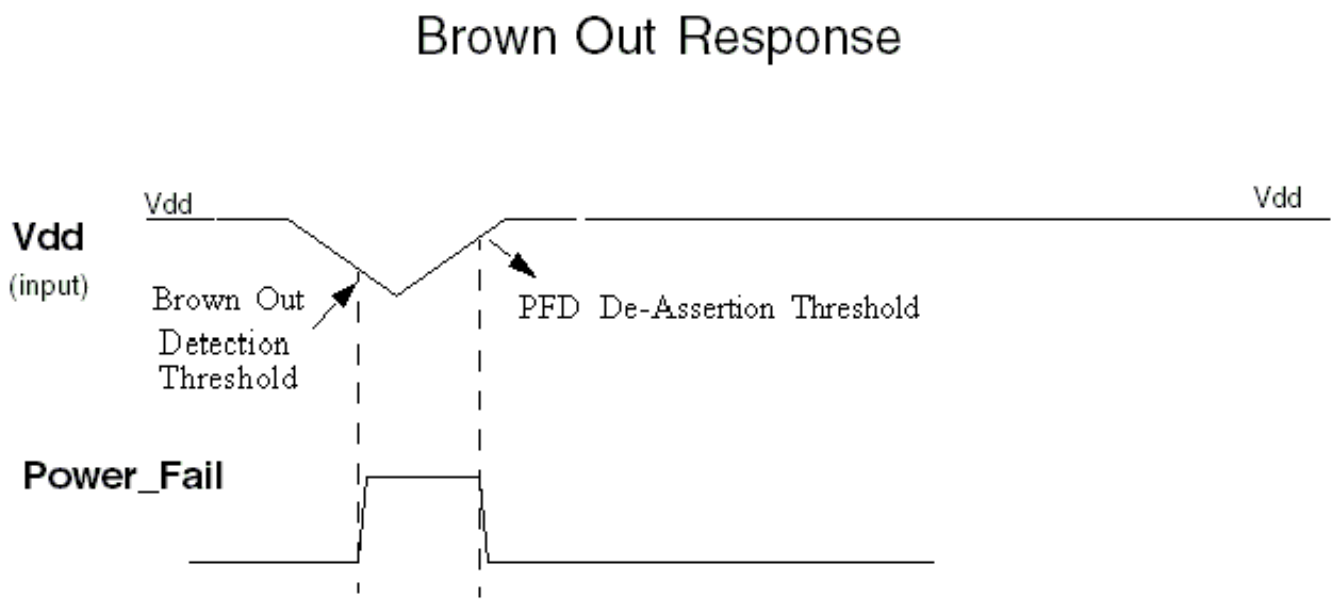


Figure 55-3. Power_Fail Brown-Out Detection

When a glitch appears in vdd and the supply goes below the Brown-Out (BO) voltage, then the output signal Power_Fail goes high.

The following table shows the various modes of operation of PFD and the behavior of Power_Fail output in these modes.

Table 55-1. Modes of Operation of PFD

DSM	LPM/HPM	vdd	Power_Fail
1	0	X	1
0	1	>0.8V	0
0	1	<0.8V	1

The PFD circuit design has been centred around the typical process corner, where the De-Assert threshold is 824mV and the BO threshold is 798mV. The De-Assert threshold at best process corner and worst process corners are 814mV and 835mVrespectively. Similarly the BO threshold at best process corner and worst process corners are 788mVand 809mVrespectively. Hence, the total variation in the De-Assert threshold around the mean value is 21mV, and that in the BO threshold is 21mV.

55.2 Functional Description

The PFD circuit will provide a LOW Power_Fail output upon power up. The Power Fail Detect Circuit has two power supplies namely, vdd and NVCC_SRTC_POW. The inverters and the buffers that are operational in the non-DSM mode work on the NVCC_SRTC_POW supply which is the always-ON supply for the chip. The 'vdd' supply has to be detected.

This circuit can be divided into the following parts:

- Resistance ladder
- A Detection inverter
- Brown-out detection
- Pulse Stretcher
- DSM circuitry

55.2.1 Resistance Ladder

The Resistance ladder is connected to vdd through two pmos switches and to vss directly. One of the two pmos switches is always kept on and hence there is a direct path current flowing from vdd to vss in all states of operation. During vdd ramping up, the voltages at the tapping points of the resistance ladder also start ramping up.

55.2.2 Detection Inverter

The detection inverter detects the voltage at one of the tapping points and changes its state when the vdd voltage has risen high enough to cause the tripping of this inverter. The sizing of this inverter and the inputs are selected so as to meet the "de-assert" threshold spec at all corners. Once the detection inverter trips, the output Power_fail is pulled low ensuring that we have reached a NO power failure state.

55.2.3 Brown-Out Detection

The Brown-Out is also detected through the same inverter that detects "de-assertion". Only difference is that now the second switch that connects to the resistance ladder also turns on, thus lowering the threshold for Brown-Out. This ensures that De-assert and Brown-Out have a minimum hysteresis between them.

55.2.4 Pulse Stretcher/Output Buffer

The buffer in the circuit ensures that the Power_Fail meets the specification for minimum pulse width at all corners.

55.2.5 DSM Circuitry

The NAND gate and the inverters that use vdd as their supply constitute the DSM circuitry. When DSM=High then the resistance ladder is OFF and input to the detection inverter is LOW and its output is HIGH. This ensures that Power_Fail is also HIGH and there is no current consumption through the PFD circuit.

55.3 Initialization/Application Information

The PFD does not contain nor access any memory map data or registers.

Chapter 56
PL301 4x1 AXI Arbiter (PLARB1)

56.1 Overview

This section provides an overview of the PL301_4x1 AXI Arbiter (PLARB1). The PL301 HPM (High Performance Matrix - by ARM Ltd.) is a configurable AXI arbiter between several interconnected masters and slaves, and is based on a standard configurable PrimeCell. The PLARB1 is designed such that many configuration options are selected at the HW design stage, determined by SoC characteristics and needs, while several other configuration options are SW controlled. This chapter covers the specific features enabled for this PLARB1 instance.

The top level diagram of PLARB1 is shown in the following block diagram.

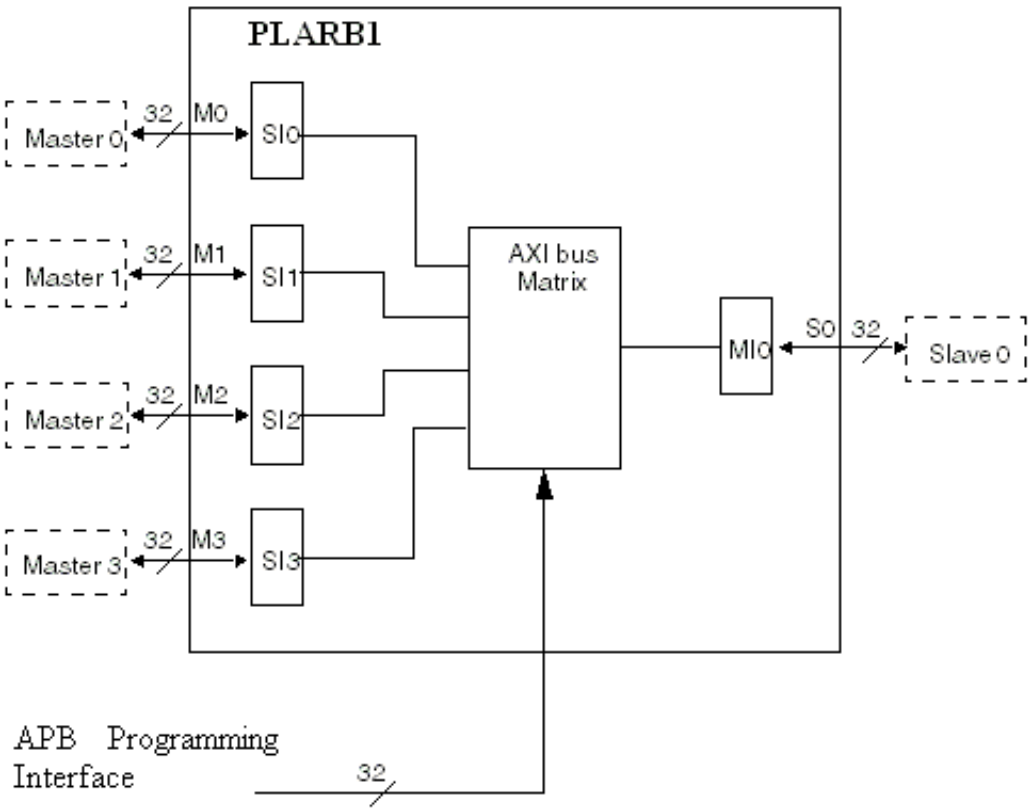


Figure 56-1. i.MX53 PLARB1 Block Diagram

56.1.1 System Connectivity

Masters and slaves of PLARB1:

- M0 - Coming from Burst-DMA port of SDMA
- M1 - FEC
- M2 - SATA
- M3 - MLB
- The S0 slave port of the PLARB is connected to M4 input of EMI (M4IF)

56.1.2 Features

Key features of the PLARB1 include:

- Support programmable Round Robin (Prog_RR) arbitration scheme with 16 slots on MI0 port
- All interfaces are 'Synchronous' with the core AXI arbiter

- "Single Slave" Cyclic Avoidance scheme
- APB interface for programming configuration / control registers
- Not supporting access interleaving
- Provide one buffer for Read and one for Write accesses on all slave interfaces, and a four (4) registers on the MI0 master interface. (The FIFO depth is one on all slave ports. Because the PLARB1 is designed for AHB masters arbitration, which could only have one active transaction at a time).

56.1.3 Modes and Operations

The PLARB1 supports a normal functional mode, as described in Normal Mode.

56.2 External Signals

There are no external I/O interfaces.

56.3 Programmable Registers

PLARB1 memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FE_0408	PLARB1 AR Programmable RR Arbitration Configuration for MI0 (PLARB1_RAC_MI0)	32	R/W	0000_0000h	56.3.1/ 3710
63FE_040C	PLARB1 AW Programmable RR Arbitration Configuration for MI0 (PLARB1_WAC_MI0)	32	R/W	0000_0000h	56.3.2/ 3711
63FE_0FC0	PLARB1 Configuration Register 0 (PLARB1_CR)	32	R	0000_0009h	56.3.3/ 3712
63FE_0FC4	PLARB1 Configuration Register 1 (PLARB1_CR)	32	R	0000_0004h	56.3.4/ 3713
63FE_0FC8	PLARB1 Configuration Register n (PLARB1_CR2)	32	R	0000_0000h	56.3.5/ 3713
63FE_0FCC	PLARB1 Configuration Register n (PLARB1_CR3)	32	R	0000_0000h	56.3.5/ 3713
63FE_0FE0	PLARB1 Peripheral ID Register 0 (PLARB1_PID0)	8	R	00h	56.3.6/ 3714

Table continues on the next page...

PLARB1 memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FE_0FE4	PLARB1 Peripheral ID Register 1 (PLARB1_PID1)	8	R	14h	56.3.7/ 3714
63FE_0FE8	PLARB1 Peripheral ID Register 2 (PLARB1_PID2)	8	R	13h	56.3.8/ 3715
63FE_0FEC	PLARB1 Peripheral ID Register 3 (PLARB1_PID3)	8	R	01h	56.3.9/ 3715
63FE_0FF0	PLARB1 ID Register 0 (PLARB1_ID0)	8	R	0Dh	56.3.10/ 3716
63FE_0FF4	PLARB1 ID Register 1 (PLARB1_ID1)	8	R	F0h	56.3.11/ 3716
63FE_0FF8	PLARB1 ID Register 2 (PLARB1_ID2)	8	R	05h	56.3.12/ 3716
63FE_0FFC	PLARB1 ID Register 3 (PLARB1_ID3)	8	R	B1h	56.3.13/ 3717

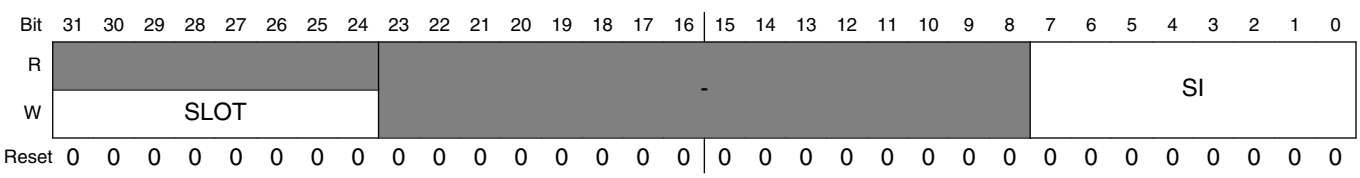
56.3.1 PLARB1 AR Programmable RR Arbitration Configuration for MI0 (PLARB1_RAC_MI0)

This register configures the slots that implement the Programmable Round Robin (prog_rr) scheme. See Programmable Round Robin (Prog_RR) Scheme for detailed algorithmic descriptions. The arbitration schemes are configured, programmed, and interrogated on a per-master-interface basis. The per-master-interface address space is at offset 0x400. The interfaces are spaced at 0x20 intervals from this base. The arbitration register address within the individual interface's space is further partitioned as:

a) the read channel, located at offset 0x8.

b) the write channel, at offset 0xC.

Address: PLARB1_RAC_MI0 is 63FE_0000h base + 408h offset = 63FE_0408h



PLARB1_RAC_MI0 field descriptions

Field	Description
31–24 SLOT	Contains the slot where data is to be written. For preliminary read operations, these bits are 0xFF. For rest of the read operations, their value is 0x00.

Table continues on the next page...

PLARB1_RAC_MI0 field descriptions (continued)

Field	Description
	NOTE: A write operation must be followed by a read operation to the same address. The write transfer sets the slot number to be read for that master interface. A read operation with 31:24 bits set to 0xFF returns the slave interface that is associated with that slot number.
23–8 -	Reserved.
7–0 SI	These bits set the slot number for the following read operation. It holds the encoded slave interface number associated with the addressed slot number. **NOTE:** The format of the read data returned has the slave interface number associated with the addressed slot in bits [7:0]. All other bits are set to zero. — — — 3 SATA

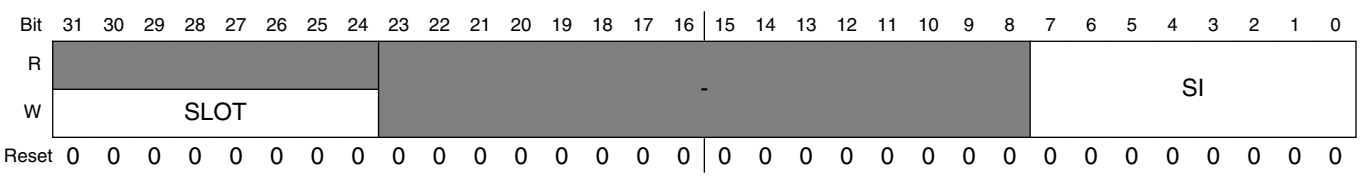
56.3.2 PLARB1 AW Programmable RR Arbitration Configuration for MI0 (PLARB1_WAC_MI0)

This register configures the slots that implement the Programmable Round Robin (prog_rr) scheme. See Programmable Round Robin (Prog_RR) Scheme for detailed algorithmic descriptions. The arbitration schemes are configured, programmed, and interrogated on a per-master-interface basis. The per-master-interface address space is at offset 0x400. The interfaces are spaced at 0x20 intervals from this base. The arbitration register address within the individual interface's space is further partitioned as:

a) the read channel, located at offset 0x8.

b) the write channel, at offset 0xC.

Address: PLARB1_WAC_MI0 is 63FE_0000h base + 40Ch offset = 63FE_040Ch



PLARB1_WAC_MI0 field descriptions

Field	Description
31–24 SLOT	The bits define the slot for which the data is to be written. There must be at least one slot per connected slave interface.When the programmable RR scheme is selected for a master interface, then, each of its arbitration slots can have the slave interface associated with it changed. For PLARB1 the slot configuration is as follow: • MI0 - 16 slots exists, default order of: 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3.

Table continues on the next page...

PLARB1_WAC_MI0 field descriptions (continued)

Field	Description
	NOTE: If these bits contain the master interface slot number, (while assigning the bits for writing Master Interface Channel Arbitration Values) then bits 7:0 contains the corresponding Slave Interface Number.
23–8 -	Reserved
7–0 SI	The encoded slave interface number. **NOTE:** No protection is imposed when you program the slots in the programmable RR arbitration scheme. So it is possible for you to remove a slave interface from all the slots which would make that slave interface inaccessible. — — — 3 SATA

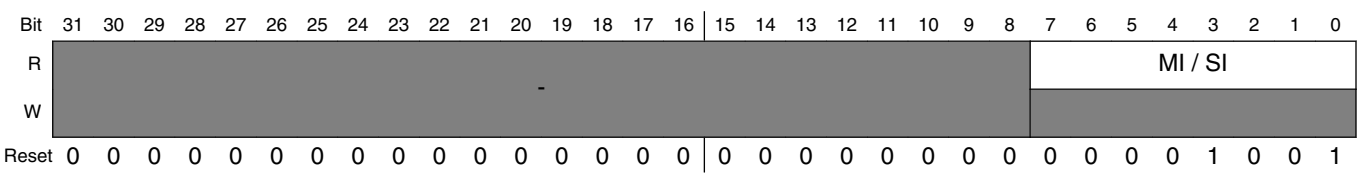
56.3.3 PLARB1 Configuration Register 0 (PLARB1_CR)

These are four 32-bit read-only registers at address 0xFC0-0xFCC. The register at:

- 0xFC0 is hard-coded and identifies how many SI's are configured in the PLARB1.
- 0xFC4 is hard-coded and identifies how many MI's are configured in the PLARB1.

The registers at 0xFC8 and 0xFCC read as zeros.

Address: PLARB1_CR is 63FE_0000h base + FC0h offset = 63FE_0FC0h



PLARB1_CR field descriptions

Field	Description
31–8 -	Reserved.
7–0 MI / SI	The total number of master/slave interfaces configured for the PLARB1. 0x0 Reserved 0x1 One MI / SI 0x2 Two MI's / SI's 0x3 Three MI's / SI's 0x20 32 MI's / SI's 0x21 0xFF Reserved

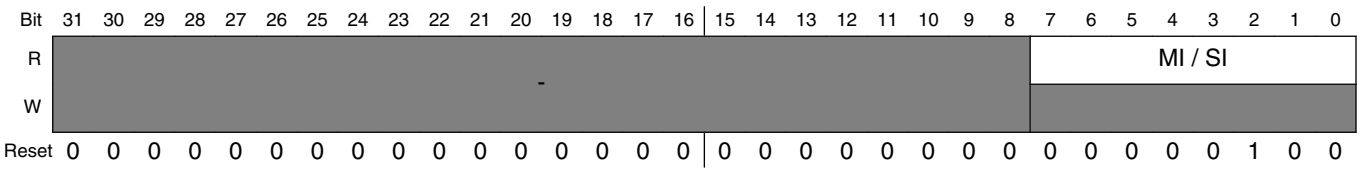
56.3.4 PLARB1 Configuration Register 1 (PLARB1_CR)

These are four 32-bit read-only registers at address 0xFC0-0xFCC. The register at:

- 0xFC0 is hard-coded and identifies how many SI's are configured in the PLARB1.
- 0xFC4 is hard-coded and identifies how many MI's are configured in the PLARB1.

The registers at 0xFC8 and 0xFCC read as zeros.

Address: PLARB1_CR is 63FE_0000h base + FC4h offset = 63FE_0FC4h



PLARB1_CR field descriptions

Field	Description
31–8 -	Reserved.
7–0 MI / SI	The total number of master/slave interfaces configured for the PLARB1. 0x0 Reserved 0x1 One MI / SI 0x2 Two MI's / SI's 0x3 Three MI's / SI's 0x20 32 MI's / SI's 0x21 0xFF Reserved

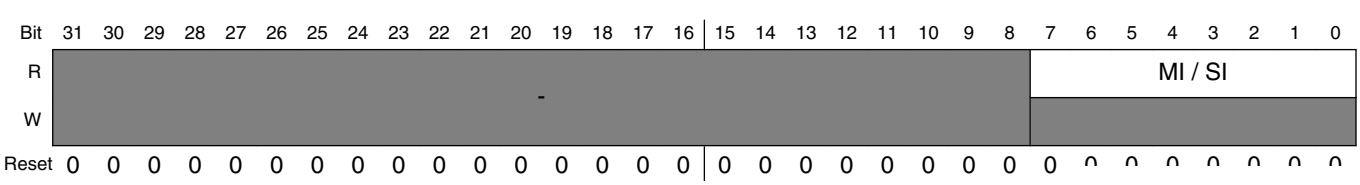
56.3.5 PLARB1 Configuration Register n (PLARB1_CR*n*)

These are four 32-bit read-only registers at address 0xFC0-0xFCC. The register at:

- 0xFC0 is hard-coded and identifies how many SI's are configured in the PLARB1.
- 0xFC4 is hard-coded and identifies how many MI's are configured in the PLARB1.

The registers at 0xFC8 and 0xFCC read as zeros.

Addresses: PLARB1_CR2 is 63FE_0000h base + FC8h offset = 63FE_0FC8h
PLARB1_CR3 is 63FE_0000h base + FCCh offset = 63FE_0FCCh



PLARB1_CR*n* field descriptions

Field	Description
31–8 -	Reserved.
7–0 MI / SI	The total number of master/slave interfaces configured for the PLARB1. 0x0 Reserved 0x1 One MI / SI 0x2 Two MI's / SI's 0x3 Three MI's / SI's 0x20 32 MI's / SI's 0x21 0xFF Reserved

56.3.6 PLARB1 Peripheral ID Register 0 (PLARB1_PID0)

The register is hard-coded.

Address: PLARB1_PID0 is 63FE_0000h base + FE0h offset = 63FE_0FE0h

Bit	7	6	5	4	3	2	1	0
Read	PART0							
Write								
Reset	0	0	0	0	0	0	0	0

PLARB1_PID0 field descriptions

Field	Description
7–0 PART0	These bits read back as 0x01.

56.3.7 PLARB1 Peripheral ID Register 1 (PLARB1_PID1)

The register is hard-coded.

Address: PLARB1_PID1 is 63FE_0000h base + FE4h offset = 63FE_0FE4h

Bit	7	6	5	4	3	2	1	0
Read	DES0				PART1			
Write								
Reset	0	0	0	1	0	1	0	0

PLARB1_PID1 field descriptions

Field	Description
7–4 DES0	These bits read back as 0x1.
3–0 PART1	These bits read back as 0x3.

56.3.8 PLARB1 Peripheral ID Register 2 (PLARB1_PID2)

The register is hard-coded.

Address: PLARB1_PID2 is 63FE_0000h base + FE8h offset = 63FE_0FE8h

Bit	7	6	5	4	3	2	1	0
Read	REV				PART1			
Write								
Reset	0	0	0	1	0	0	1	1

PLARB1_PID2 field descriptions

Field	Description
7–4 REV	These bits read back as the revision number. This can be between 0 and 15: 0x3 r1p2
3–0 PART1	These bits read back as 0x04.

56.3.9 PLARB1 Peripheral ID Register 3 (PLARB1_PID3)

The register is hard-coded.

Address: PLARB1_PID3 is 63FE_0000h base + FECh offset = 63FE_0FECh

Bit	7	6	5	4	3	2	1	0
Read	-				-			
Write								
Reset	0	0	0	0	0	0	0	1

PLARB1_PID3 field descriptions

Field	Description
7–4 -	Reserved
3–0 -	Reserved, read as 0x0

56.3.10 PLARB1 ID Register 0 (PLARB1_ID0)

The register is hard-coded.

Address: PLARB1_ID0 is 63FE_0000h base + FF0h offset = 63FE_0FF0h

Bit	7	6	5	4	3	2	1	0
Read	ID							
Write								
Reset	0	0	0	0	1	1	0	1

PLARB1_ID0 field descriptions

Field	Description
7–0 ID	These bits read back 0x0D

56.3.11 PLARB1 ID Register 1 (PLARB1_ID1)

The register is hard-coded.

Address: PLARB1_ID1 is 63FE_0000h base + FF4h offset = 63FE_0FF4h

Bit	7	6	5	4	3	2	1	0
Read	ID							
Write								
Reset	1	1	1	1	0	0	0	0

PLARB1_ID1 field descriptions

Field	Description
7–0 ID	These bits read back 0xF0

56.3.12 PLARB1 ID Register 2 (PLARB1_ID2)

The register is hard-coded.

Address: PLARB1_ID2 is 63FE_0000h base + FF8h offset = 63FE_0FF8h

Bit	7	6	5	4	3	2	1	0
Read	ID							
Write								
Reset	0	0	0	0	0	1	0	1

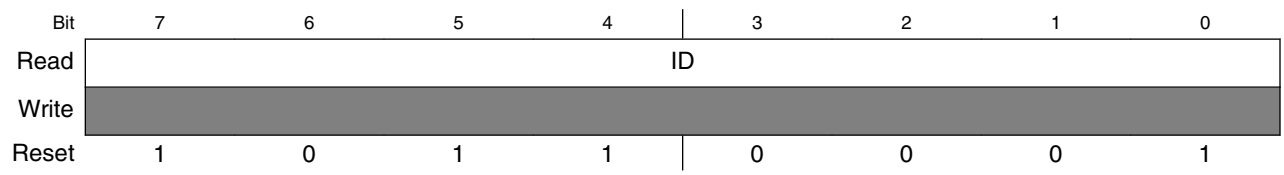
PLARB1_ID2 field descriptions

Field	Description
7–0 ID	These bits read back 0x05

56.3.13 PLARB1 ID Register 3 (PLARB1_ID3)

The register is hard-coded.

Address: PLARB1_ID3 is 63FE_0000h base + FFCh offset = 63FE_0FFCh



PLARB1_ID3 field descriptions

Field	Description
7–0 ID	These bits read back 0xB1

56.4 Functional Description

The PLARB1 is an interconnect consists of a high-performance AXI cross-bar switch and related auxiliary components that supports 4 external masters to a single slaves arbitration logic.

56.4.1 Normal Mode

The PLARB1 is an interconnect that provides the means for master transactions initiated by a master to reach the intended slave in a multi-master and multi-slave AXI environment.

56.4.2 Operations

The PLARB1 supports the "Single Slave" cyclic schemes to avoid possible deadlock. The scheme is determined at the hardware level, per PLARB1 salve interfaces (SI's) and cannot be changed by the software.

Please refer to Cyclic Dependency Avoidance Scheme (CDAS) below for details.

56.4.2.1 Sparse Connect

The PL301 HPM, provide means to specify "Sparse Connects", to specify just the needed connects between SI's to MI's.

Using Sparse Connects, reduces the overall gate count of the interconnect and can improve timing.

A sample sparse connection scheme is shown in Figure 56-17.

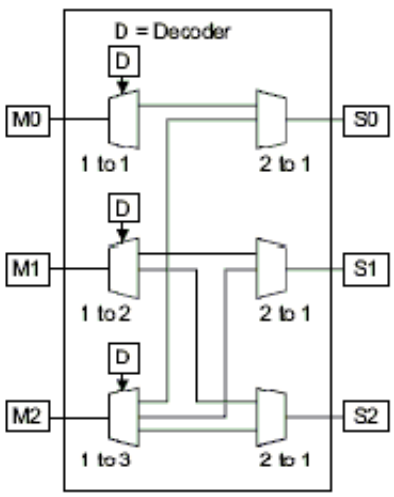


Figure 56-17. Sparse Connection Example

The PLARB1 specific "Sparse Connection" scheme is provided in Table 56-17. A "YES" denotes an existing path between masters and a "NO", a non-existing path between (SI's interfaces) to slaves (MI's interfaces).

NOTE

Because the PLARB1 has just one MI interface, there are no sparse connections.

Table 56-17. Sparse Connections

	MI0
SI0	YES
SI1	YES
SI2	YES
SI3	YES

56.4.2.2 Memory region mapping

Table 56-18 lists the memory map for each of the SIs in this example, based on the global memory map for the entire system. (Missing connections are marked by "-" in the table).

Table 56-18. PLARB1 Memory Regions

From Master	To Slave	Start Address	End Address
M0	S0	0x00000000	0xFFFFFFFF
M1	S0	0x00000000	0xFFFFFFFF
M2	S0	0x00000000	0xFFFFFFFF
M3	S0	0x00000000	0xFFFFFFFF

56.4.2.3 Cyclic Dependency Avoidance Scheme (CDAS)

In any AXI system, the interconnect must ensure that deadlock cannot occur when routing transactions concurrently to multiple slaves (because it is possible that any of the connected slaves would re-order the returning read /write data response). To remove any potential for deadlock, the HPM supports a variety of cyclic dependency avoidance schemes.

The PL301 HPM, supports various cyclic schemes, which are HW configurable, at SoC design time, and are not configurable by software. The schemes offer varying degrees of cyclic restriction and you can trade this off against timing performance.

The PLARB1 slave ports are configured to "Single Slave" scheme. See Single Slave Scheme section below for details.

56.4.2.3.1 Single Slave Scheme

This scheme enforces the rule that a master can have active transactions to only a single slave at any time. It is the simplest scheme, and has the lowest timing overhead, but it is also the most restrictive. Figure 56-18 shows the Single Slave scheme rules.

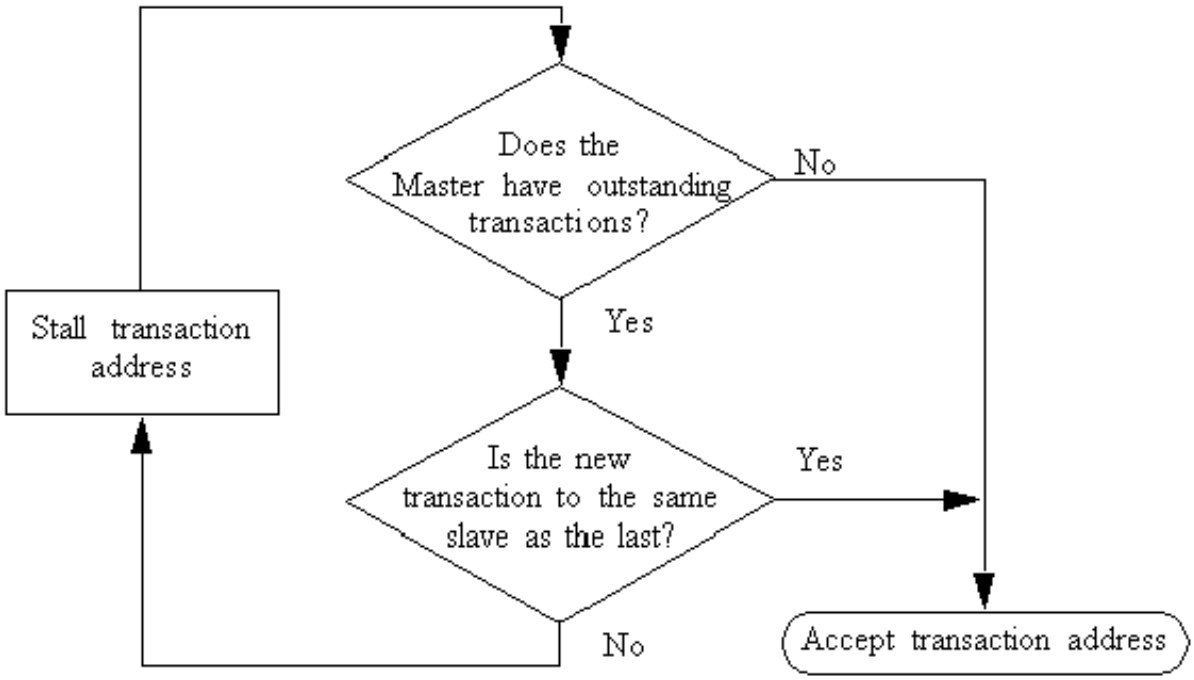


Figure 56-18. Single Slave Scheme Behavior

56.4.2.4 Arbitration Scheme

General arbitration scheme details of PL301 HPM:

Each MI can be configured separately to have an arbitration scheme that is either:

- A programmable or fixed Round-robin (RR) scheme
- A programmable Least Recently Granted (LRG) scheme

The AW and AR channels have separate arbiters and can be programmed, if applicable, and interrogated separately through the APB programming interface, but both AW and AR channels are configured identically. Because the AW and AR channels are arbitrated separately, an MI can permit simultaneous read and write transactions from different SIs.

The arbitration mechanism registers the arbitration decision for use in the subsequent cycle. An arbitration decision taken in the current cycle does not affect the current cycle. If no SIs are active, the arbiter adopts default arbitration, that is, the highest priority SI. If this occurs, and the highest priority interface becomes active in the same cycle as (or before) any other SI, then this does not constitute a grant to an active SI and the arbitration scheme does not change its state as a result of that transfer.

56.4.2.5 Arbitration Options Specific to the PLARB1 Arbiter

The Programmable RR arbitration scheme (Prog_RR) is used for MI0, with 16 slots, who's HW default slots is set to 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3 (25% for each master).

56.4.2.5.1 Programmable Round Robin (Prog_RR) Scheme

There must be at least one slot per connected SI and there can be up to 32 slots. By allocating multiple slots for a SI, you can allocate access to the slave, on average, in proportion to the number of slots. If the slots are appropriately ordered, this can also reduce the maximum time before a grant is guaranteed. The SI associated with a slot can be interrogated from the APB programming interface, but it cannot be changed.

Whenever arbitration is granted to an active SI, the slots are rotated so that the slot currently in the highest priority position becomes the lowest, and all other slots move to a higher priority but maintain their relative order, as shown in Figure 56-19. This means that if an SI is the highest priority active SI, but is not the highest priority interface, then it continues to win the arbitration until it becomes the highest priority interface, and then the lowest priority interface subsequently.

Because the arbitration value is registered, the arbitration decision made in this cycle is used in the next cycle. This means that if the SI that currently holds the arbitration is still the highest priority active SI in this cycle, then it wins the arbitration again regardless of whether or not it is active in the next cycle, as shown by the status of M3 in stages A, B, and C of Figure 56-19.

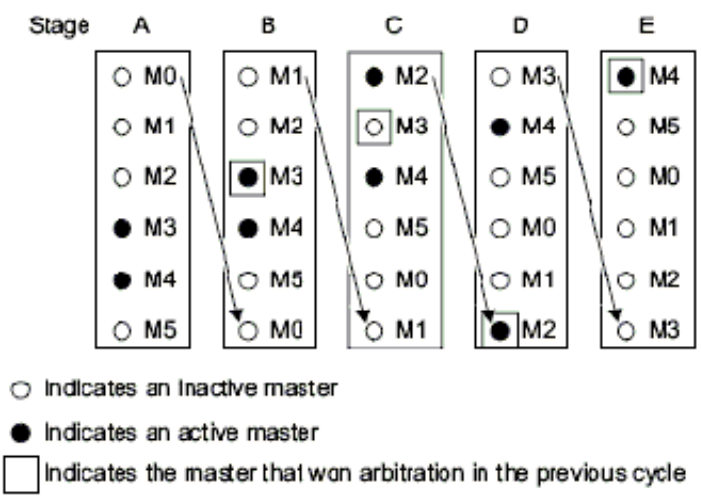


Figure 56-19. Operation of RR Arbitration Scheme

56.4.3 Various Configuration Options

The following options were selected at instantiation of PLARB1 during the design phase, and are hard-coded in the PLARB1 instance and therefore cannot be changed by SW.

For more details on the configuration parameters functionality, refer to *"PrimeCell High-Performance Matrix (PL301) Technical Reference Manual"* document by ARM.

56.4.3.1 SI Configuration

SI's ports settings:

- "read_acceptance_capability" = "1"
- "write_acceptance_capability" = "1"
- No registered I/O used. ("registered_io" = "0")
- "single_active_write" = "false".

56.4.3.2 MI Configuration

MI0 port settings:

- "write_interleave_capability" = "1"
- "write_issuing_capability" = "4"

56.4.4 Clocks

The PLARB1 supports multiple clock domains for several masters and slaves, as follows:

Table 56-19. PLARB1 Clock Domains

Port	Synchronization Type	Max Frequency[1]
PL301 Core	N/A	133 MHz
SI0	Synchronous to PL301 Core	133 MHz
SI1	Synchronous to PL301 Core	133 MHz
SI2	Synchronous to PL301 Core	133 MHz
SI3	Synchronous to PL301 Core	133 MHz
MI0	Synchronous to PL301 Core	133 MHz

1. As used in i.MX53.

56.4.5 Interrupts

This block does not generate interrupts.

Chapter 57
PL301 2x2 Arbiter (PLARB2)

57.1 Overview

This section provides an overview of the PL301 2x2 Arbiter (PLARB2). The PLARB2 HPM (High Performance Matrix - by ARM Ltd.) is a configurable AXI arbiter between several interconnected masters and slaves, and is based on a standard configurable PrimeCell. The PLARB2 is designed such that many configuration options are selected at the HW design stage, determined by SoC characteristics and needs, while several other configuration options are SW controlled. This chapter focuses largely on the features enabled for this PLARB2 instance.

The top level diagram of PLARB2 is shown in Figure 57-1.

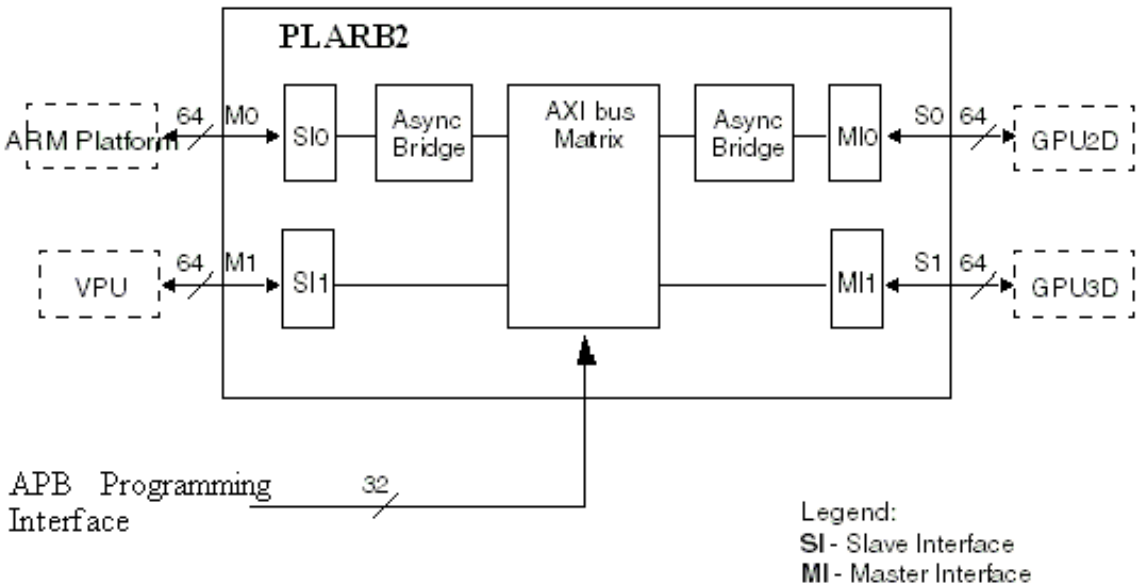


Figure 57-1. MX53 PLARB2 Block Diagram

57.1.1 Features

Key features of the PLARB2:

- Supports programmable Round Robin (Prog_RR) arbitration scheme on MI1 port (4x slots)
- Implements sparse connection, where path from SI0 to MI0 is omitted
- Supports cross-clocks domain synchronization on Master 0 and Slave 0 ports
- "Single Slave" Cyclic Avoidance scheme
- APB interface for programming configuration / control registers
- Provides FIFOs for both Read Write accesses on slave interfaces, and a single register on the MI0 master interface

57.1.2 Modes and Operations

The PLARB2 supports a normal functional mode, as described in Normal Mode.

57.2 External Signals

There are no external I/O interfaces.

57.3 Programmable Registers

PLARB2 memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0000_0408	PLARB2 AR Programmable RR Arbitration Configuration for MI0 (PLARB2_RAC_MI0)	32	R/W	0000_0000h	57.3.1/ 3728
0000_040C	PLARB2 AW Programmable RR Arbitration Configuration for MIn (PLARB2_WAC_MI0)	32	R/W	0000_0000h	57.3.2/ 3729
0000_0428	PLARB2 AR Programmable RR Arbitration Configuration for MI1 (PLARB2_RAC_MI1)	32	R/W	0000_0000h	57.3.3/ 3730
0000_042C	PLARB2 AW Programmable RR Arbitration Configuration for MIn (PLARB2_WAC_MI1)	32	R/W	0000_0000h	57.3.2/ 3729
0000_0FC0	PLARB2 Configuration Register 0 (PLARB2_CR0)	32	R	0000_0009h	57.3.4/ 3730
0000_0FC4	PLARB2 Configuration Register 1 (PLARB2_CR0)	32	R	0000_0004h	57.3.5/ 3731
0000_0FC8	PLARB2 Configuration Register n (PLARB2_CR2)	32	R	0000_0000h	57.3.6/ 3732
0000_0FCC	PLARB2 Configuration Register n (PLARB2_CR3)	32	R	0000_0000h	57.3.6/ 3732
0000_0FE0	PLARB2 Peripheral ID Register 0-3 (PLARB2_PID0-3)	32	R	0014_1301h	57.3.7/ 3733
0000_0FF0	PLARB2 ID Registers 0-3 (PLARB2_ID0-3)	32	R	0DF0_05B1h	57.3.8/ 3733

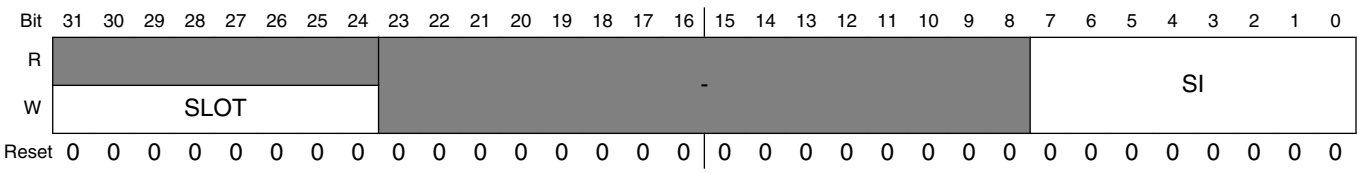
57.3.1 PLARB2 AR Programmable RR Arbitration Configuration for MI0 (PLARB2_RAC_MI0)

This register configures the slots that implement the Programmable Round Robin (prog_rr) scheme. See Programmable Round Robin (Prog_RR) Scheme for detailed algorithmic descriptions. The arbitration schemes are configured, programmed, and interrogated on a per-master-interface basis. The per-master-interface address space is at offset 0x400. The interfaces are spaced at 0x20 intervals from this base. The arbitration register address within the individual interface's space is further partitioned as:

a) the read channel, located at offset 0x8.

b) the write channel, at offset 0xC.

Address: PLARB2_RAC_MI0 is 0h base + 408h offset = 0000_0408h



PLARB2_RAC_MI0 field descriptions

Field	Description
31–24 SLOT	Contains the slot where data is to be written. For preliminary read operations, these bits are 0xFF. For rest of the read operations, their value is 0x00. **Note:** A write operation must be followed by a read operation to the same address. The write transfer sets the slot number to be read for that master interface. A read operation with 31:24 bits set to 0xFF returns the slave interface that is associated with that slot number.
23–8 -	Reserved.
7–0 SI	These bits set the slot number for the following read operation. It holds the encoded slave interface number associated with the addressed slot number. **Note:** The format of the read data returned has the slave interface number associated with the addressed slot in bits [7:0]. All other bits are set to zero. 2 - 7 - Not Used 0 256KB GPU 3D memory (GMEM). 1 On Chip 128KB RAM (via OCRAM controller)

57.3.2 PLARB2 AW Programmable RR Arbitration Configuration for MIn (PLARB2_WAC_MI*n*)

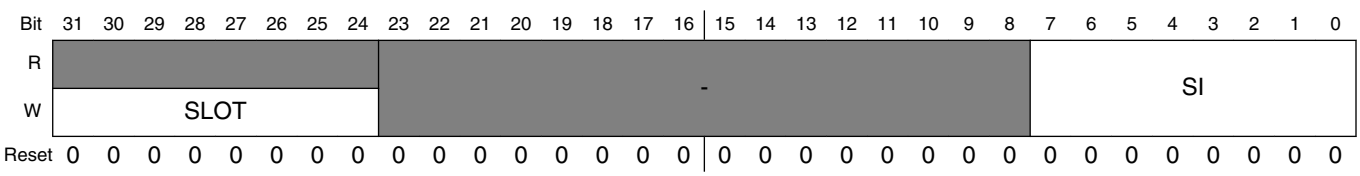
This register configures the slots that implement the Programmable Round Robin (prog_rr) scheme. See "Programmable Round Robin (Prog_RR) Scheme" for detailed algorithmic descriptions. The arbitration schemes are configured, programmed, and interrogated on a per-master-interface basis. The per-master-interface address space is at offset 0x400. The interfaces are spaced at 0x20 intervals from this base. The arbitration register address within the individual interface's space is further partitioned as:

a) the read channel, located at offset 0x8.

b) the write channel, at offset 0xC.

Addresses: PLARB2_WAC_MI0 is 0h base + 40Ch offset = 0000_040Ch

PLARB2_WAC_MI1 is 0h base + 42Ch offset = 0000_042Ch



PLARB2_WAC_MI*n* field descriptions

Field	Description
31–24 SLOT	The bits define the slot for which the data is to be written. There must be at least one slot per connected slave interface.When the programmable RR scheme is selected for a master interface, then, each of its arbitration slots can have the slave interface associated with it changed. For PLARB2, the slot configuration is as follow: • MI0 - One slot, fixed for SI1 (SI0 is sparse connection of MI0) MI1 - 4 slots exists, in order of: SI0, SI1, SI1, SI0. Note: If these bits contain the master interface slot number, (while assigning the bits for writing Master Interface Channel Arbitration Values) then bits 7:0 contains the corresponding Slave Interface Number.
23–8 -	Reserved.
7–0 SI	The encoded slave interface number. **NOTE:** No protection is imposed when you program the slots in the programmable RR arbitration scheme. So it is possible for you to remove a slave interface from all the slots which would make that slave interface inaccessible. 2 - 7 - Not Used 0 256KB GPU 3D memory (GMEM). 1 On Chip 128KB RAM (via OCRAM controller)

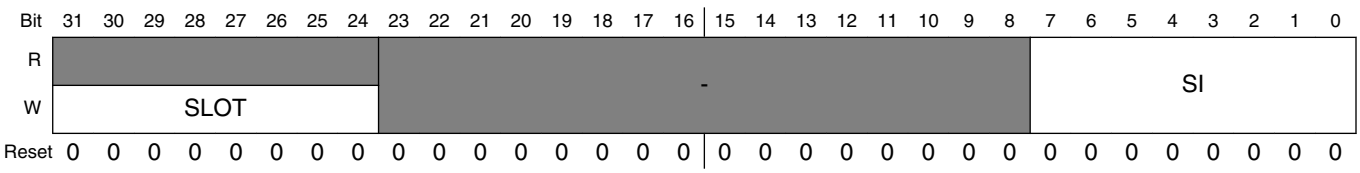
57.3.3 PLARB2 AR Programmable RR Arbitration Configuration for MI1 (PLARB2_RAC_MI1)

This register configures the slots that implement the Programmable Round Robin (prog_rr) scheme. See Programmable Round Robin (Prog_RR) Scheme for detailed algorithmic descriptions. The arbitration schemes are configured, programmed, and interrogated on a per-master-interface basis. The per-master-interface address space is at offset 0x400. The interfaces are spaced at 0x20 intervals from this base. The arbitration register address within the individual interface's space is further partitioned as:

a) the read channel, located at offset 0x8.

b) the write channel, at offset 0xC.

Address: PLARB2_RAC_MI1 is 0h base + 428h offset = 0000_0428h



PLARB2_RAC_MI1 field descriptions

Field	Description
31–24 SLOT	Contains the slot where data is to be written. For preliminary read operations, these bits are 0xFF. For rest of the read operations, their value is 0x00. **Note:** A write operation must be followed by a read operation to the same address. The write transfer sets the slot number to be read for that master interface. A read operation with 31:24 bits set to 0xFF returns the slave interface that is associated with that slot number.
23–8 -	Reserved.
7–0 SI	These bits set the slot number for the following read operation. It holds the encoded slave interface number associated with the addressed slot number. **Note:** The format of the read data returned has the slave interface number associated with the addressed slot in bits [7:0]. All other bits are set to zero. 2 - 7 - Not Used 0 256KB GPU 3D memory (GMEM). 1 On Chip 128KB RAM (via OCRAM controller)

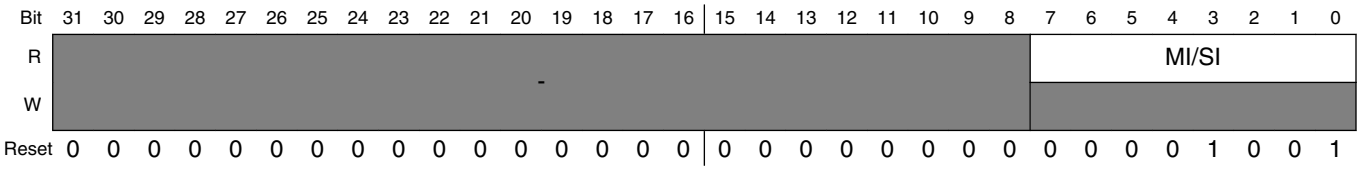
57.3.4 PLARB2 Configuration Register 0 (PLARB2_CR0)

These are four 32-bit read-only registers at address 0xFC0-0xFCC. The register at:

- 0xFC0 is hard-coded and identifies how many SI's are configured in the PLARB2.
- 0xFC4 is hard-coded and identifies how many MI's are configured in the PLARB2.

The registers at 0xFC8 and 0xFCC read as zeros.

Address: PLARB2_CR0 is 0h base + FC0h offset = 0000_0FC0h



PLARB2_CR0 field descriptions

Field	Description
31–8 -	Reserved
7–0 MI/SI	The total number of master/slave interfaces configured for the PLARB2. 0x0 Reserved 0x1 One MI / SI 0x2 Two MI's / SI's 0x3 Three MI's / SI's 0x20 32 MI's / SI's 0x21 0xFF Reserved

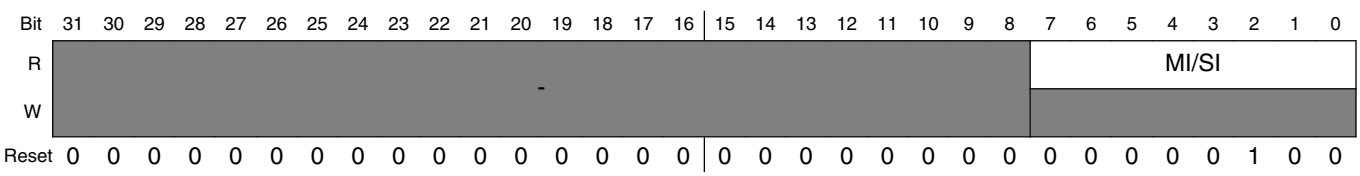
57.3.5 PLARB2 Configuration Register 1 (PLARB2_CR0)

These are four 32-bit read-only registers at address 0xFC0-0xFCC. The register at:

- 0xFC0 is hard-coded and identifies how many SI's are configured in the PLARB2.
- 0xFC4 is hard-coded and identifies how many MI's are configured in the PLARB2.

The registers at 0xFC8 and 0xFCC read as zeros.

Address: PLARB2_CR0 is 0h base + FC4h offset = 0000_0FC4h



PLARB2_CR0 field descriptions

Field	Description
31–8 -	Reserved

Table continues on the next page...

PLARB2_CR0 field descriptions (continued)

Field	Description
7–0 MI/SI	The total number of master/slave interfaces configured for the PLARB2. 0x0 Reserved 0x1 One MI / SI 0x2 Two MI's / SI's 0x3 Three MI's / SI's 0x20 32 MI's / SI's 0x21 0xFF Reserved

57.3.6 PLARB2 Configuration Register n (PLARB2_CR*n*)

These are four 32-bit read-only registers at address 0xFC0-0xFCC. The register at:

- 0xFC0 is hard-coded and identifies how many SI's are configured in the PLARB2.
- 0xFC4 is hard-coded and identifies how many MI's are configured in the PLARB2.

The registers at 0xFC8 and 0xFCC read as zeros.

Addresses: PLARB2_CR2 is 0h base + FC8h offset = 0000_0FC8h
PLARB2_CR3 is 0h base + FCCh offset = 0000_0FCCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																								MI/SI							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

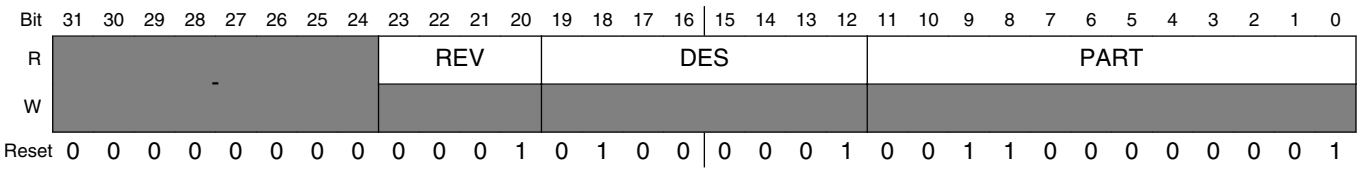
PLARB2_CR*n* field descriptions

Field	Description
31–8 -	Reserved
7–0 MI/SI	The total number of master/slave interfaces configured for the PLARB2. 0x0 Reserved 0x1 One MI / SI 0x2 Two MI's / SI's 0x3 Three MI's / SI's 0x20 32 MI's / SI's 0x21 0xFF Reserved

57.3.7 PLARB2 Peripheral ID Register 0-3 (PLARB2_PID0-3)

The registers are four eight-bit read-only registers and span address locations 0xFE0-0xFEC. The registers can be treated conceptually as a single register holding a 32-bit peripheral ID value. An external master reads them to determine the PLARB2 version of the device.

Address: PLARB2_PID0-3 is 0h base + FE0h offset = 0000_0FE0h



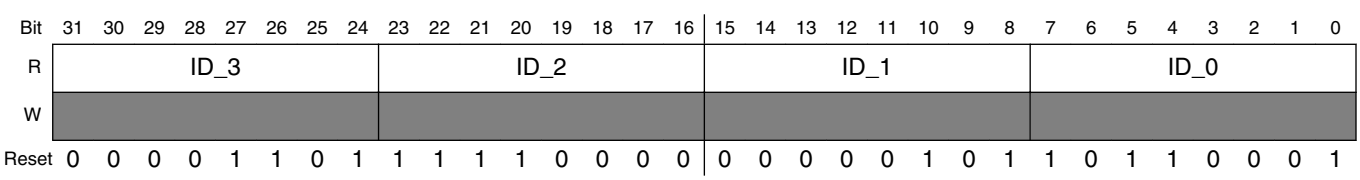
PLARB2_PID0-3 field descriptions

Field	Description
31–24 -	Reserved.
23–20 REV	The peripheral revision number is revision-dependent.
19–12 DES	Designer's ID number. This is 0x41 for ARM.
11–0 PART	The Identifies the peripheral. The part number for PLARB2 is 0x301.

57.3.8 PLARB2 ID Registers 0-3 (PLARB2_ID0-3)

The PrimeCell ID value is a 32-bit value. However, to ensure that it is accessible in all systems, the 32 bits are implemented as four 8-bit registers that can be accessed separately as the least significant eight bits of addresses 0xFF0, 0xFF4, 0xFF8, and 0xFFC.

Address: PLARB2_ID0-3 is 0h base + FF0h offset = 0000_0FF0h



PLARB2_ID0-3 field descriptions

Field	Description
31–24 ID_3	pcell_id_3 [7:0] - 0xB1
23–16 ID_2	pcell_id_2 [7:0] - 0x05
15–8 ID_1	pcell_id_1 [7:0] - 0xF0
7–0 ID_0	pcell_id_0 [7:0] - 0xD

57.4 Functional Description

The PLARB2 is an interconnect consists of a high-performance AXI cross-bar switch and related auxiliary components that supports 2 external masters and 2 external slaves.

57.4.1 Normal Mode

The PLARB2 is an interconnect that provides the means for master transactions initiated by a master to reach the intended slave in a multi-master and multi-slave AXI environment.

57.4.2 Low Power Modes

There is no special Low-Power mode for the PLARB2.

57.4.3 Operations

The PLARB2 supports the "Single Slave" cyclic schemes to avoid possible deadlock. The scheme is determined at the hardware level, per PLARB2 salve interfaces (SIs) and cannot be changed by the software.

Please refer to Cyclic Dependency Avoidance Scheme (CDAS) below for details.

57.4.3.1 Sparse Connect

The PLARB2 HPM provides means to specify "Sparse Connects", to specify just the needed connects between SIs to MIs.

Using Sparse Connects, reduces the overall gate count of the interconnect and can improve timing.

A sample sparse connection scheme is shown in Figure 57-14.

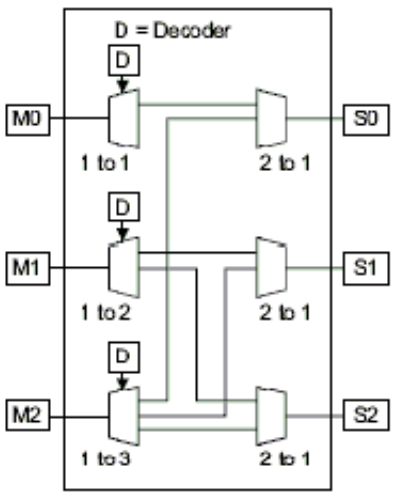


Figure 57-14. Sparse Connection Example

The PLARB2 specific "Sparse Connection" scheme is provided in Table 57-14. A "Yes", denotes an existing path between masters and a "NO" marks a non-existing path between (SI's interfaces) to slaves (MI's interfaces).

Table 57-14. Sparse Connections

	MI0	MI1
SI0	No	Yes
SI1	Yes	Yes

57.4.3.2 Memory Region Mapping

Table 57-15 lists the memory map for each of the SIs in this example, based on the global memory map for the entire system. (Missing connections are marked by "-" in the table).

Table 57-15. PLARB2 Memory Regions

From Master	To Slave	Start Address	End Address
M0	S0	-	-
M0	S1	0xF8000000	0xF801FFFF
M1	S0	0xF8020000	0xF805FFFF
M1	S1	0xF8000000	0xF801FFFF

57.4.3.3 Cyclic Dependency Avoidance Scheme (CDAS)

In any AXI system, the interconnect must ensure that deadlock cannot occur when routing transactions concurrently to multiple slaves (because it is possible that any of the connected slaves would re-order the returning read /write data response). To remove any potential for deadlock, the HPM supports a variety of cyclic dependency avoidance schemes.

The PLARB2 HPM, supports various cyclic schemes, which are HW configurable, at SoC design time, and are not configurable by software. The schemes offer varying degrees of cyclic restriction and you can trade this off against timing performance.

The PLARB2 slave ports are configured to "Single Slave" scheme. See Single Slave Scheme section below for details.

57.4.3.3.1 Single Slave Scheme

This scheme enforces the rule that a master can have active transactions to only a single slave at any time. It is the simplest scheme, and has the lowest timing overhead, but it is also the most restrictive. Figure 57-15 shows the Single Slave scheme rules.

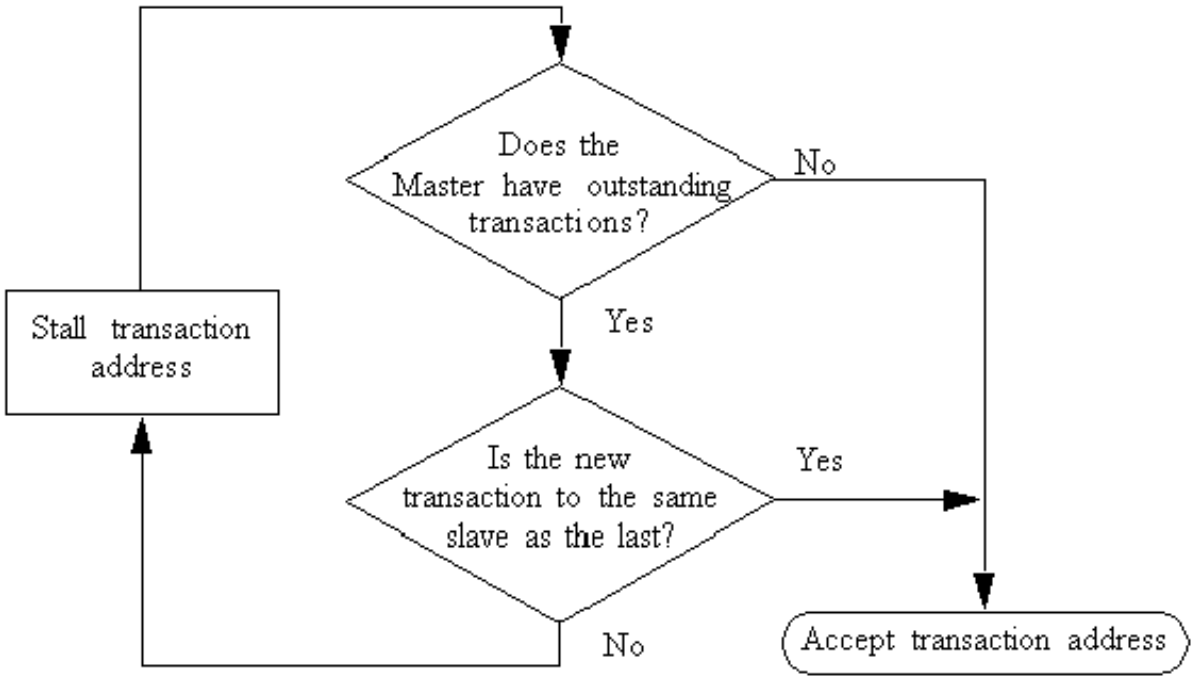


Figure 57-15. Single Slave Scheme Behavior

57.4.3.4 Arbitration Scheme

General arbitration scheme details of PLARB2 HPM:

Each MI can be configured separately to have an arbitration scheme that is either:

- A programmable or fixed Round-robin (RR) scheme
- A programmable Least Recently Granted (LRG) scheme.

The AW and AR channels have separate arbiters and can be programmed, if applicable, and interrogated separately through the APB programming interface, but both AW and AR channels are configured identically. Because the AW and AR channels are arbitrated separately, an MI can permit simultaneous read and write transactions from different SIs.

The arbitration mechanism registers the arbitration decision for use in the subsequent cycle. An arbitration decision taken in the current cycle does not affect the current cycle. If no SIs are active, the arbiter adopts default arbitration, that is, the highest priority SI. If this occurs and then the highest priority interface becomes active in the same cycle as, or before any other SI, then this does not constitute a grant to an active SI and the arbitration scheme does not change its state as a result of that transfer.

57.4.3.5 Arbitration Options Specific to the PLARB2 Arbiter

Because MI0 passes just SI1 traffic there is no need for any arbitration. A fixed RR arbitration is used with a single slot assigned to S1.

The Programmable RR arbitration scheme is used for MI1 with 4 slots. HW default SI slots is set to 0, 1, 1, 0. (50%/50%).

57.4.3.5.1 Programmable Round Robin (Prog_RR) Scheme

There must be at least one slot per connected SI and there can be up to 32 slots. By allocating multiple slots for a SI, you can allocate access to the slave, on average, in proportion to the number of slots. If the slots are appropriately ordered, this can also reduce the maximum time before a grant is guaranteed. The SI associated with a slot can be interrogated from the APB programming interface, but it cannot be changed.

Whenever arbitration is granted to an active SI, the slots are rotated so that the slot currently in the highest priority position becomes the lowest, and all other slots move to a higher priority but maintain their relative order, as Figure 57-16 shows. This means that if an SI is the highest priority active SI, but is not the highest priority interface, then it continues to win the arbitration until it becomes the highest priority interface, and then the lowest priority interface subsequently.

Because the arbitration value is registered, the arbitration decision made in this cycle is used in the next cycle. This means that if the SI that currently holds the arbitration is still the highest priority active SI in this cycle, wins the arbitration again regardless of whether or not it is active in the next cycle as shown by the status of M3 in stages A, B, and C of Figure 57-16.

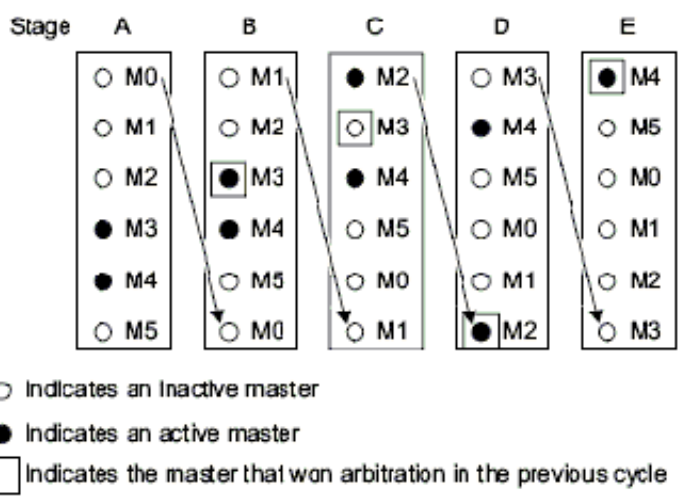


Figure 57-16. Operation of RR Arbitration Scheme

57.4.4 Various Configuration Options

The following options were selected at instanciation of the PLARB2, during the design phase, and are hard-coded in the PLARB2 instance. They cannot be changed by SW.

For more details on the configuration parameters functionality, refer to *"PrimeCell High-Performance Matrix (PL301) Technical Reference Manual"* document by ARM.

57.4.4.1 SI Configuration

SI's ports settings:

- "read_acceptance_capability" = 4
- "write_acceptance_capability" = "16" / "4" for SI0 and SI1 respectively.
- No registered IO used. ("registered_io" = "0")
- "single_active_write" = "false"

57.4.4.2 MI Configuration

MI's ports settings:

- "write_interleave_capability" = "1"
- "write_issuing_capability" = "1"

57.4.5 Clocks

The PLARB2 supports multiple clock domains for several masters and slaves, as follow:

Table 57-16. PLARB2 clock domains

Port	Synchronization Type	Max Frequency[1]
PLARB2 Core	N/A	133 MHz
SI0	Synchronous to PLARB2 Core	133 MHz
SI1	Asynchronous	200 MHz
MI0	Synchronous to PLARB2 Core	133 MHz
MI1	Asynchronous	200 MHz

1. As designed for i.MX53.

57.4.6 Interrupts

This block does not generate interrupts.

Chapter 58
Power On Reset (POR)

58.1 Overview

The POR is part of the Secure Real Time Clock (SRTC). It is designed for use in SoC applications which require low voltage, low power consumption. POR circuit generates a reset signal till its supply VDD reaches a predetermined voltage level.The POR circuit has five main sub-circuits: Recognition circuit, Pulse Latch, Brown-Out Detection Circuit, POR Stretcher and Output Buffer. The Recognition circuit detects and generates a signal till VDD crosses a predetermined voltage level called de-assertion voltage. The POR Pulse Stretcher widens the POR pulse. BO Detector detects brown out condition in VDD. Pulse Latch latches the POR output. Buffer is used to drive the on-chip load.

Figure 58-1 shows the block diagram of POR.

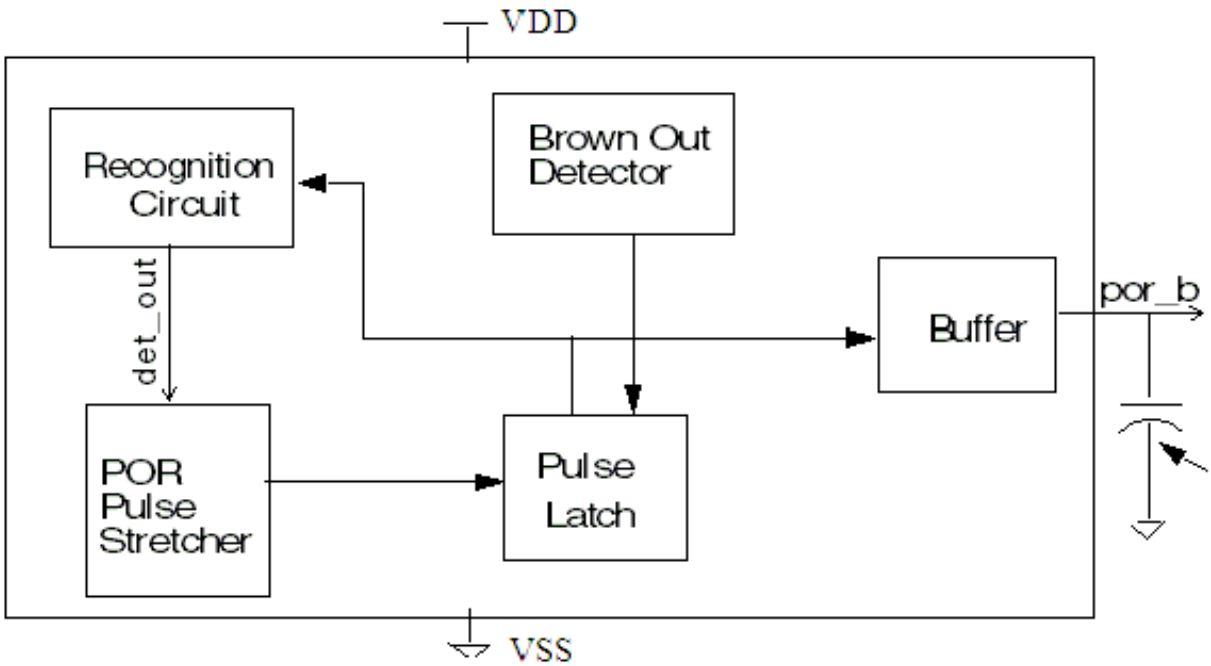


Figure 58-1. POR Block Diagram

58.2 Features

The POR has the following features:

- De-assertion voltage Independent of VDD ramp rate
- Almost zero steady state power consumption
- Brown Out Detection
- Voltage dependent Brown Out thresholds

58.3 Mode of Operation

The POR generates a por_b active low pulse when the chip is powering up.When the POR output gets de-asserted, the POR goes to low power state (POR circuit shuts off and output signals are latched). When a glitch appears in VDD and the supply goes below the Brown-Out (BO) detection threshold, the BO detection circuit pulls down the por_b signal and a reset is generated to indicate a BO condition. The por_b signal de-asserts only when VDD passes POR recognition threshold.

58.4 External Signal Description

58.4.1 Detailed Signal Descriptions

Table 58-1 lists all the pins in the POR.

Table 58-1. POR pins list

Name	Port	Function
por_b	Output	Power on Reset signal
vdd	Power	Power supply
vss	Power	Power ground

58.4.2 por_b - Power On Reset Signal

This signal is generated from the POR. This is the reset (active low) signal for the chip.

58.4.3 VDD - Power Supply

This signal is the power supply for the POR circuit.

58.4.4 VSS - Power Ground

This signal is the ground supply for the POR circuit.

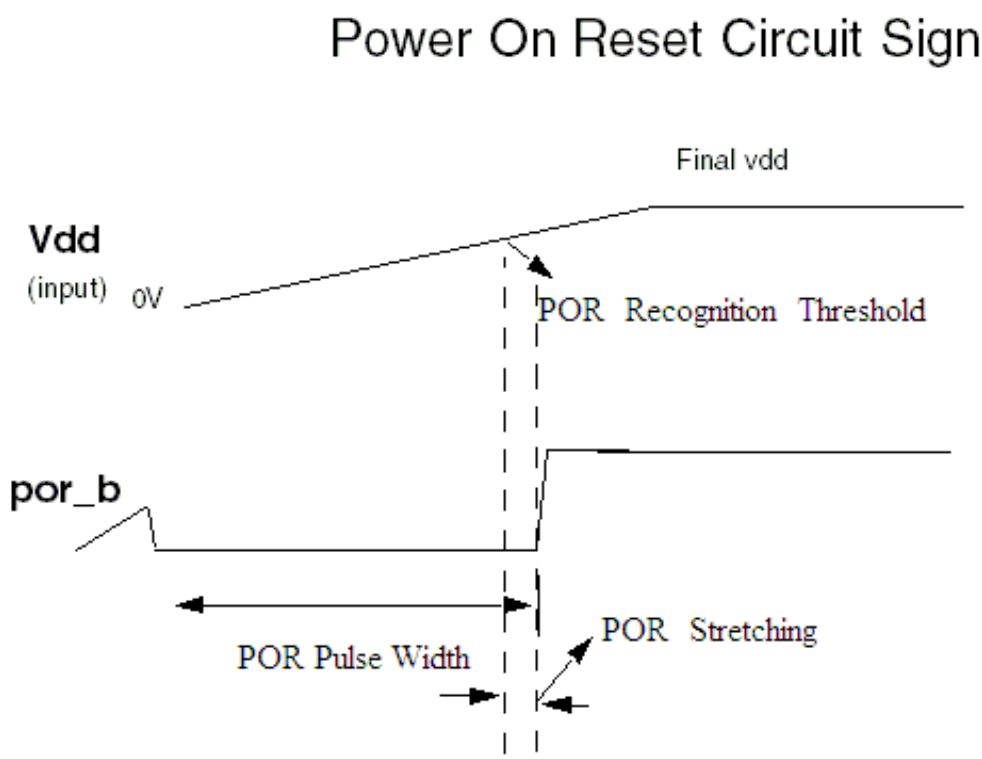


Figure 58-2. Power on sequence and RESET pulse generation

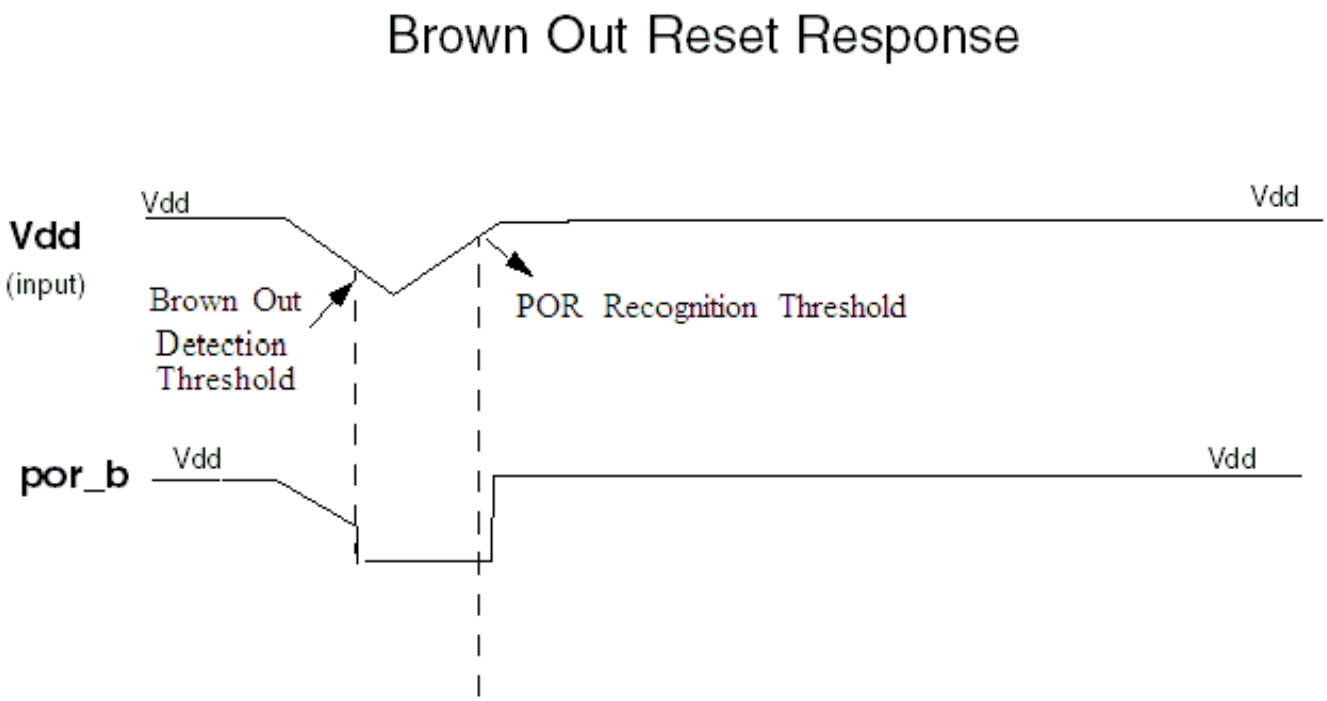


Figure 58-3. Brown-Out Detection

58.5 Functional Description

The POR does not contain nor access any memory map data or registers,

The POR circuit will provide a reset upon power up. The Power On Reset Circuit is connected to the on-chip core supply.

This circuit has five main elements:

- POR Recognition circuit
- A Pulse Latch circuit
- A Brown-Out detection Circuit
- POR Stretcher
- Output Buffer

58.5.1 Recognition Circuit

The Recognition circuit output is low when the supply is below a certain predetermined value. Below this value, known as the POR deassertion voltage, the POR output is asserted and causes the chip to reset as the supply continues to rise. On detecting this threshold, the recognition circuit output toggles, and subsequently the POR output is deasserted and the chip comes out of reset. This denotes the end of the POR operation. The recognition circuit output signal is stretched to ensure minimum assertion-width of POR reset signal. The delayed POR is used to disable the recognition circuit once its deasserted.

Thus the POR circuit goes into low-power state.

58.5.2 Pulse Latch

The Pulse Latch is used to latch the low-to high transition of the recognition circuit output, so that the recognition circuit can be powered down once the detection is over. It also ensures that POR output goes low initially during power-up.

58.5.3 Brown-Out Detector

The Brown-Out detection circuit detects conditions in supply when VDD dips below Brown-Out Detection threshold and begins to rise again. This threshold is supply dependent, meaning that they vary with the supply level from which VDD begins to drop in case of a Brown-Out event.

58.5.4 POR Pulse Stretcher

This comprises of a cascade of inverters to provide delay. It is used to stretch the POR pulse.

58.5.5 Output Buffer

These buffers are used to drive the on-chip capacitive load at the output of the signal por_b.

58.6 Initialization/Application Information

The POR does not contain nor access any memory map data or registers.

Chapter 59
Pulse Width Modulation (PWM)

59.1 Overview

The Pulse Width Modulation (PWM) has a 16-bit counter, and is optimized to generate sound from stored sample audio images and it can also generate tones. It uses 16-bit resolution and a 4 x 16 data FIFO.

This section presents an overview of the PWM. The figure below illustrates the PWM diagram.

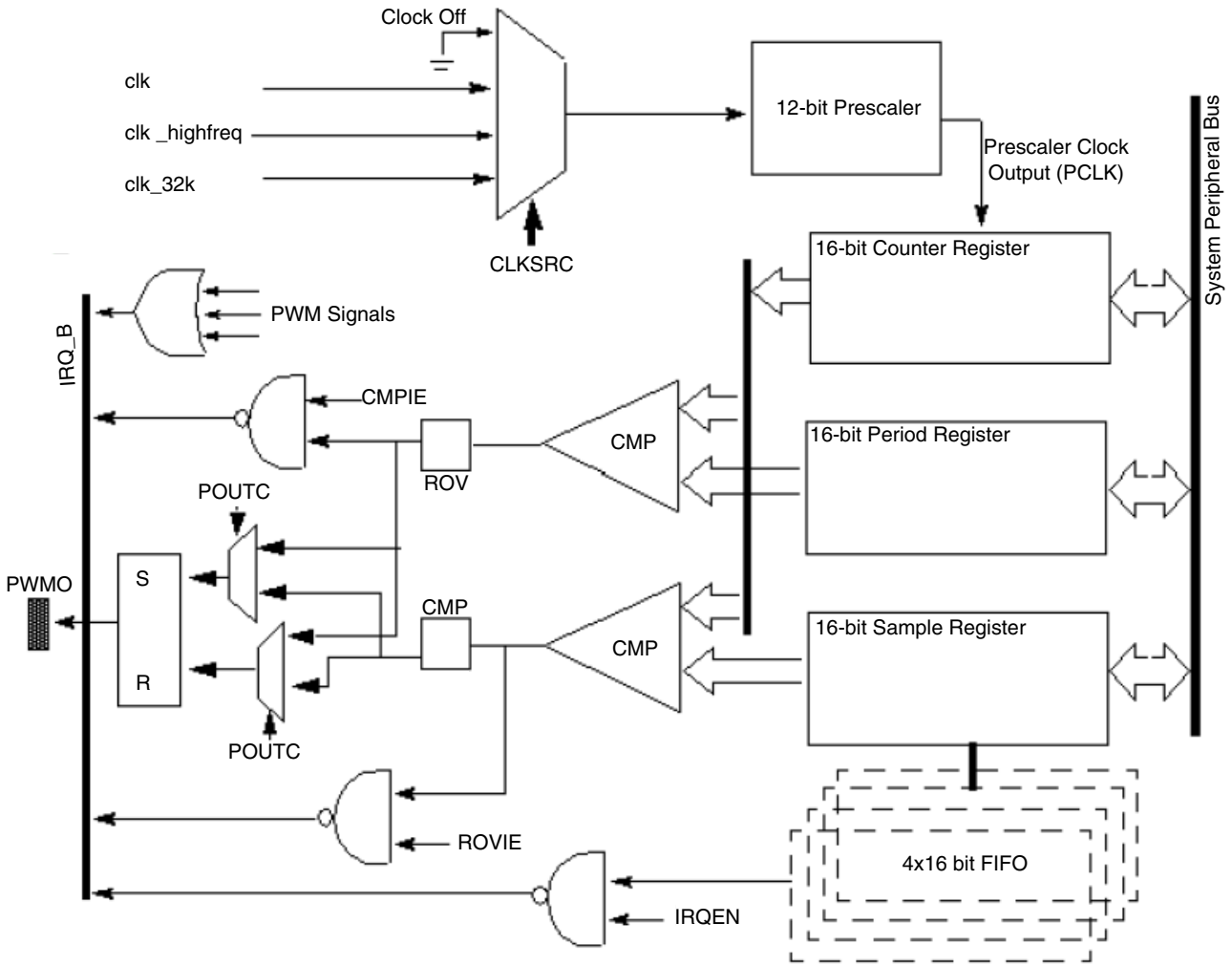


Figure 59-1. Pulse-Width Modulator Block Diagram

The following features characterize the PWM:

- 16-bit up-counter with clock source selection
- 4 x 16 FIFO to minimize interrupt overhead
- 12-bit prescaler for division of clock
- Sound and melody generation
- Active high or active low configured output
- Can be programmed to be active in low-power mode
- Can be programmed to be active in debug mode
- Interrupts at compare and rollover

59.2 Signal Description

The PWM follows IP Bus protocol when interfacing with the processor core. It does not have any interface signals with any other block inside the chip except for clock and reset inputs from the Clock Control Module (CCM), System Reset Controller (SRC), and interrupt signals to the processor interrupt handler. There is a single output signal going outside the chip boundary.

59.2.1 External Signals

PWM has a single output signal to the chip boundary named ipp_do_pwmo.

The following table outlines the external signals.

Table 59-1. External Signals

Name	Direction	Function	Reset State	Pull up
PWMxO	Output	This is the functional output of the PWM. The modulated signal of the block is observed at this pin. It can be viewed as a clock signal whose period and duty cycle can be varied with different settings of the PWM. The smallest period can be two ipg_clk clock periods with duty cycle of 50%.	0	-

59.3 Functional Description

The following sections detail the PWM operation and function.

59.3.1 Operation

The output of the PWM is a toggling signal whose frequency and duty cycle can be modulated by programming the appropriate registers. It has a 16-bit up counter which counts from 0x0000 until the counter value equals the [Value in Period register] + 1. After this match occurs the Counter is reset to 0x0000.

At the beginning of a count period cycle, the PWMO pin is set to one (default) and the counter begins counting up from 0x0000. The sample value in the sample FIFO is compared on each count of prescaler clock. When the sample and count values match, the PWMO signal is cleared to zero (default). The counter continues counting until the period match occurs and subsequently another period cycle begins.

When the PWM is enabled the counter starts running and generates an output with the reset values in the period and sample registers. It is recommended that the programming of these registers be done before PWM is enabled.

A hardware reset results in all the PWM count and sample registers begin cleared and the FIFO being flushed. The control register shows that FIFO is empty and it can be written into, and the PWM is disabled. A software reset has the same results, however the state of the STOPEN, DOZEN, WAITEN, and DBGEN bits in the control register are not affected. Software reset can be asserted even when the PWM is in disabled state.

59.3.1.1 Clocks

The clock that feeds the prescaler can be selected from:

- High frequency Clock (ipg_clk_highfreq) pat_ref or CKIH

  This is a high frequency clock, provided by the Clock Control Module (CCM). This clock should be in the low power mode when the ipg_clk is turned off. Thus, the PWM can be run on this clock in the low power mode.

- Low Reference Clock (ipg_clk_32k) CKIL

  This is the 32 KHz low reference clock which is provided by the CCM. This clock should be on in the low power mode when ipg_clk is turned off. Thus, PWM can be run on this clock in the low power mode.

- Global Functional Clock (ipg_clk)

  This clock should be on in normal operations. In low power modes it can be switched off.

The clock input source is determined by the CLKSRC field of the PWM control register. The CLKSRC value should only be changed when the PWM is disabled.

A change in the value of the PRESCALER field of the control register is immediately reflected on its output clock frequency.

59.3.1.2 FIFO

Digital sample values can be loaded into the pulse-width modulator as 16-bit words. The endianess can be changed using the BCTR and HCTR bits of the control register. A 4-word (16-bit) FIFO minimizes interrupt overhead. A maskable interrupt is generated when the number of data words fall below the water level set by the FWM field in the control register.

A write in the sample register results in the value being stored into the FIFO if it is not full. A write when the FIFO is full sets FWE (FIFO write error) bit in the status register and the FIFO contents remain unchanged. The FIFO can be written at any time, but can be read only when the PWM is enabled. The FIFOAV field shows how many data words are currently contained in the FIFO and whether or not it can be written into.

A read on the sample register yields the current FIFO value that is being used, or will be used, by the PWM for generation on the output signal. Therefore, a write and a subsequent read on the sample register may result in different values being obtained.

59.3.1.3 Rollover and Compare Event

The counter is reset to 0x0000 after its value equals the PERIOD + 1 and resumes counting thereafter. This event is referred to as a rollover. When PERIOD = 0x0000, the counter is reset after count reaches 0x0001. Therefore PERIOD = 0xffff or 0xfffe results in the counter value being reset after count till 0xffff. During a rollover event the output is either set (default), reset or has no effect according to the programming of the POUTC field in the control register. This event can also generate an interrupt if the respective interrupt enable bit is set in the control register.

When the counter value reaches the sample value, the output of the PWM is reset (default), set or has no effect according to the programming of the POUTC field of control register. This event is referred to as a compare event. This event can also generate an interrupt if the respective interrupt enable bit is set in the control register.

If the rollover event sets the PWM output signal, the compare event will reset it and vice versa for a particular programming configuration of POUTC field.

59.3.1.4 Low Power Mode Behavior

In low power modes, if the clock from the selected clock source is available, the PWM counter continues to run and an output is produced, depending on whether the control bit for that mode is set or not. In the absence of the clock itself, or if the corresponding low power bit in the control register is 0, the counter is reset and resumes counting when it exits the low power mode.

59.3.1.5 Debug Mode Behavior

In debug mode, PWM has the option of continuing to run or be halted. If the DBGEN bit is not set in the PWM_PWMCR, the PWM is halted. If the DBGEN bit is set, then the PWM will continue to run in the debug mode.

59.4 Programmable Registers

The PWM includes six user-accessible 32-bit registers.

PWM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FB_4000	PWM Control Register (PWM-1_PWMCR)	32	R/W	0000_0000h	59.4.1/ 3753
53FB_4004	PWM Status Register (PWM-1_PWMSR)	32	w1c	0000_0008h	59.4.2/ 3755
53FB_4008	PWM Interrupt Register (PWM-1_PWMIR)	32	R/W	0000_0000h	59.4.3/ 3756
53FB_400C	PWM Sample Register (PWM-1_PWMSAR)	32	R/W	0000_0000h	59.4.4/ 3757
53FB_4010	PWM Period Register (PWM-1_PWMPR)	32	R/W	0000_FFFEh	59.4.5/ 3758
53FB_4014	PWM Counter Register (PWM-1_PWMCNR)	32	R	0000_0000h	59.4.6/ 3758
53FB_8000	PWM Control Register (PWM-2_PWMCR)	32	R/W	0000_0000h	59.4.1/ 3753
53FB_8004	PWM Status Register (PWM-2_PWMSR)	32	w1c	0000_0008h	59.4.2/ 3755
53FB_8008	PWM Interrupt Register (PWM-2_PWMIR)	32	R/W	0000_0000h	59.4.3/ 3756

Table continues on the next page...

PWM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FB_800C	PWM Sample Register (PWM-2_PWMSAR)	32	R/W	0000_0000h	59.4.4/ 3757
53FB_8010	PWM Period Register (PWM-2_PWMPR)	32	R/W	0000_FFFEh	59.4.5/ 3758
53FB_8014	PWM Counter Register (PWM-2_PWMCNR)	32	R	0000_0000h	59.4.6/ 3758

59.4.1 PWM Control Register (PWMx_PWMCR)

The PWM control register (PWM_PWMCR) is used to configure the operating settings of the PWM. It contains the prescaler for the clock division.

Addresses: PWM-1_PWMCR is 53FB_4000h base + 0h offset = 53FB_4000h

PWM-2_PWMCR is 53FB_8000h base + 0h offset = 53FB_8000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				FWM		STOPEN	DOZEN	WAITEN	DBGEN	BCTR	HCTR	POUTC		CLKSRC	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PRESCALER												SWR	REPEAT		EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWMx_PWMCR field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved. These reserved bits are always read as zero.
27–26 FWM	FIFO Water Mark. These bits are used to set the data level at which the FIFO empty flag will be set and the corresponding interrupt generated 00 FIFO empty flag is set when there are more than or equal to 1 empty slots in FIFO 01 FIFO empty flag is set when there are more than or equal to 2 empty slots in FIFO 10 FIFO empty flag is set when there are more than or equal to 3 empty slots in FIFO 11 FIFO empty flag is set when there are more than or equal to 4 empty slots in FIFO
25 STOPEN	Stop Mode Enable. This bit keeps the PWM functional while in stop mode. When this bit is cleared, the input clock is gated off in stop mode. This bit is not affected by software reset. It is cleared by hardware reset.

Table continues on the next page...

PWMx_PWMCR field descriptions (continued)

Field	Description
	0 Inactive in stop mode 1 Active in stop mode
24 DOZEN	Doze Mode Enable. This bit keeps the PWM functional in doze mode. When this bit is cleared, the input clock is gated off in doze mode. This bit is not affected by software reset. It is cleared by hardware reset. 0 Inactive in doze mode 1 Active in doze mode
23 WAITEN	Wait Mode Enable. This bit keeps the PWM functional in wait mode. When this bit is cleared, the input clock is gated off in wait mode. This bit is not affected by software reset. It is cleared by hardware reset. 0 Inactive in wait mode 1 Active in wait mode
22 DBGEN	Debug Mode Enable. This bit keeps the PWM functional in debug mode. When this bit is cleared, the input clock is gated off in debug mode. This bit is not affected by software reset. It is cleared by hardware reset. 0 Inactive in debug mode 1 Active in debug mode
21 BCTR	Byte Data Swap Control. This bit determines the byte ordering of the 16-bit data when it goes into the FIFO from the sample register. 0 byte ordering remains the same 1 byte ordering is reversed
20 HCTR	Half-word Data Swap Control. This bit determines which half word data from the 32-bit IP Bus interface is written into the lower 16 bits of the sample register. 0 Half word swapping does not take place 1 Half words from write data bus are swapped
19–18 POUTC	PWM Output Configuration. This bit field determines the mode of PWM output on the output pin. 00 Output pin is set at rollover and cleared at comparison 01 Output pin is cleared at rollover and set at comparison 10 PWM output is disconnected 11 PWM output is disconnected
17–16 CLKSRC	Select Clock Source. These bits determine which clock input will be selected for running the counter. After reset the system functional clock is selected. The input clock can also be turned off if these bits are set to 00. This field value should only be changed when the PWM is disabled 00 Clock is off 01 ipg_clk 10 ipg_clk_highfreq 11 ipg_clk_32k
15–4 PRESCALER	Counter Clock Prescaler Value. This bit field determines the value by which the clock will be divided before it goes to the counter. 0x000 Divide by 1 0x001 Divide by 2 0xfff Divide by 4096

Table continues on the next page...

PWMx_PWMCR field descriptions (continued)

Field	Description
3 SWR	Software Reset. PWM is reset when this bit is set to 1. It is a self clearing bit. A write 1 to this bit is a single wait state write cycle. When the block is in reset state this bit is set and is cleared when the reset procedure is over. Setting this bit resets all the registers to their reset values except for the STOPEN, DOZEN, WAITEN, and DBGEN bits in this control register. 0 PWM is out of reset 1 PWM is undergoing reset
2–1 REPEAT	Sample Repeat. This bit field determines the number of times each sample from the FIFO is to be used. 00 Use each sample once 01 Use each sample twice 10 Use each sample four times 11 Use each sample eight times
0 EN	PWM Enable. This bit enables the PWM. If this bit is not enabled, the clock prescaler and the counter is reset. When the PWM is enabled, it begins a new period, the output pin is set to start a new period while the prescaler and counter are released and counting begins. 0 PWM disabled 1 PWM enabled

59.4.2 PWM Status Register (PWMx_PWMSR)

The PWM status register (PWM_PWMSR) contains seven bits which display the state of the FIFO and the occurrence of rollover and compare events. The FIFOAV bit is read-only but the other four bits can be cleared by writing 1 to them. The FE, ROV, and CMP bits are associated with FIFO-Empty, Roll-over, and Compare interrupts, respectively.

Addresses: PWM-1_PWMSR is 53FB_4000h base + 4h offset = 53FB_4004h

PWM-2_PWMSR is 53FB_8000h base + 4h offset = 53FB_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									FWE	CMP	ROV	FE	FIFOAV		
W										w1c	w1c	w1c	w1c			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

PWMx_PWMSR field descriptions

Field	Description
31–7 Reserved	This read-only field is reserved and always has the value zero. Reserved. These reserved bits are always read as zero.

Table continues on the next page...

PWMx_PWMSR field descriptions (continued)

Field	Description
6 FWE	FIFO Write Error Status. This bit shows that an attempt has been made to write FIFO when it is full. 0 FIFO write error not occurred 1 FIFO write error occurred
5 CMP	Compare Status. This bit shows that a compare event has occurred. 0 Compare event not occurred 1 Compare event occurred
4 ROV	Roll-over Status. This bit shows that a roll-over event has occurred. 0 Roll-over event not occurred 1 Roll-over event occurred
3 FE	FIFO Empty Status Bit. This bit indicates the FIFO data level in comparison to the water level set by FWM field in the control register. 0 Data level is above water mark 1 When the data level falls below the mark set by FWM field
2–0 FIFOAV	FIFO Available. These read-only bits indicate the data level remaining in the FIFO. An attempted write to these bits will not affect their value and no transfer error is generated. 000 No data available 001 1 word of data in FIFO 010 2 words of data in FIFO 011 3 words of data in FIFO 100 4 words of data in FIFO 101 unused 110 unused 111 unused

59.4.3 PWM Interrupt Register (PWMx_PWMIR)

The PWM Interrupt register (PWM_PWMIR) contains three bits which control the generation of the compare, rollover and FIFO empty interrupts.

Addresses: PWM-1_PWMIR is 53FB_4000h base + 8h offset = 53FB_4008h

PWM-2_PWMIR is 53FB_8000h base + 8h offset = 53FB_8008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																													CIE	RIE	FIE
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWMx_PWMIR field descriptions

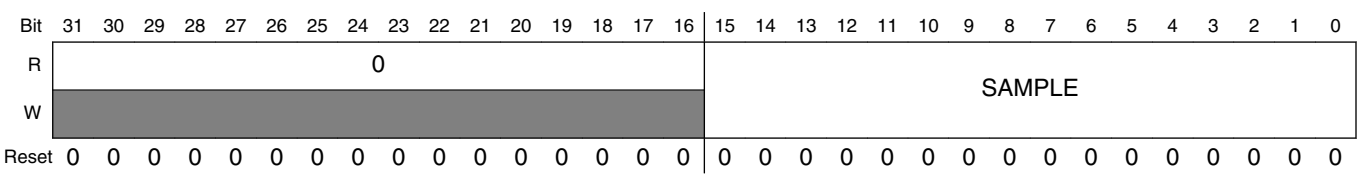
Field	Description
31–3 Reserved	This read-only field is reserved and always has the value zero. Reserved. These reserved bits are always read as zero.
2 CIE	Compare Interrupt Enable. This bit controls the generation of the Compare interrupt. 0 Compare Interrupt not enabled 1 Compare Interrupt enabled
1 RIE	Roll-over Interrupt Enable. This bit controls the generation of the Rollover interrupt. 0 Roll-over interrupt not enabled 1 Roll-over Interrupt enabled
0 FIE	FIFO Empty Interrupt Enable. This bit controls the generation of the FIFO Empty interrupt. 0 FIFO Empty interrupt disabled 1 FIFO Empty interrupt enabled

59.4.4 PWM Sample Register (PWMx_PWMSAR)

The PWM sample register (PWM_PWMSAR) is the input to the FIFO. 16-bit words are loaded into the FIFO. The FIFO can be written at any time, but can be read only when the PWM is enabled. The PWM will run at the last set duty-cycle setting if all the values of the FIFO has been utilized, until the FIFO is reloaded or the PWM is disabled. When a new value is written, the duty cycle changes after the current period is over.

A value of zero in the sample register will result in the ipp_pwm_pwmo output signal always being low/high (POUTC = 00 it will be low and POUTC = 01 it will be high), and no output waveform will be produced. If the value in this register is higher than the PERIOD + 1, the output will never be set/reset depending on POUTC value.

Addresses: PWM-1_PWMSAR is 53FB_4000h base + Ch offset = 53FB_400Ch
PWM-2_PWMSAR is 53FB_8000h base + Ch offset = 53FB_800Ch



PWMx_PWMSAR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. These are reserved bits and writing a value will not affect the functionality of PWM and are always read as zero.
15–0 SAMPLE	Sample Value. This is the input to the 4x16 FIFO. The value in this register denotes the value of the sample being currently used.

59.4.5 PWM Period Register (PWMx_PWMPR)

The PWM period register (PWM_PWMPR) determines the period of the PWM output signal. After the counter value matches PERIOD + 1, the counter is reset to start another period.

PWMO (Hz) = PCLK(Hz) / (period +2)

A value of zero in the PWM_PWMPR will result in a period of two clock cycles for the output signal. Writing 0xFFFF to this register will achieve the same result as writing 0xFFFE.

A change in the period value due to a write in PWM_PWMPR results in the counter being reset to zero and the start of a new count period.

Addresses: PWM-1_PWMPR is 53FB_4000h base + 10h offset = 53FB_4010h

PWM-2_PWMPR is 53FB_8000h base + 10h offset = 53FB_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																PERIOD															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

PWMx_PWMPR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. These are reserved bits and writing a value will not affect the functionality of PWM and are always read as zero.
15–0 PERIOD	Period Value. These bits determine the Period of the count cycle. The counter counts up to [Period Value] +1 and is then reset to 0x0000.

59.4.6 PWM Counter Register (PWMx_PWMCNR)

The read-only pulse-width modulator counter register (PWM_PWMCNR) contains the current count value and can be read at any time without disturbing the counter.

Addresses: PWM-1_PWMCNR is 53FB_4000h base + 14h offset = 53FB_4014h

PWM-2_PWMCNR is 53FB_8000h base + 14h offset = 53FB_8014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWMx_PWMCNR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. These are reserved bits and writing a value will not affect the functionality of PWM and are always read as zero.
15–0 COUNT	Counter Value. These bits are the counter register value and denotes the current count state the counter register is in.

Chapter 60
ROM Controller with Patch (ROMC)

60.1 Introduction

60.1.1 Overview

The Read Only Memory Controller with ROM Patch (ROMC) acts as an interface between the ARM advanced high-performance bus (AHB - Lite) and the Read Only Memory. The ROMC consists of a ROM Controller and a ROM Patch. The ROM Patch is used to either patch code routines or fix data tables in the ROM area. There is an IP Bus interface to access the ROM Patch Registers and. The figure below depicts the main functional sub-blocks of the ROMC.

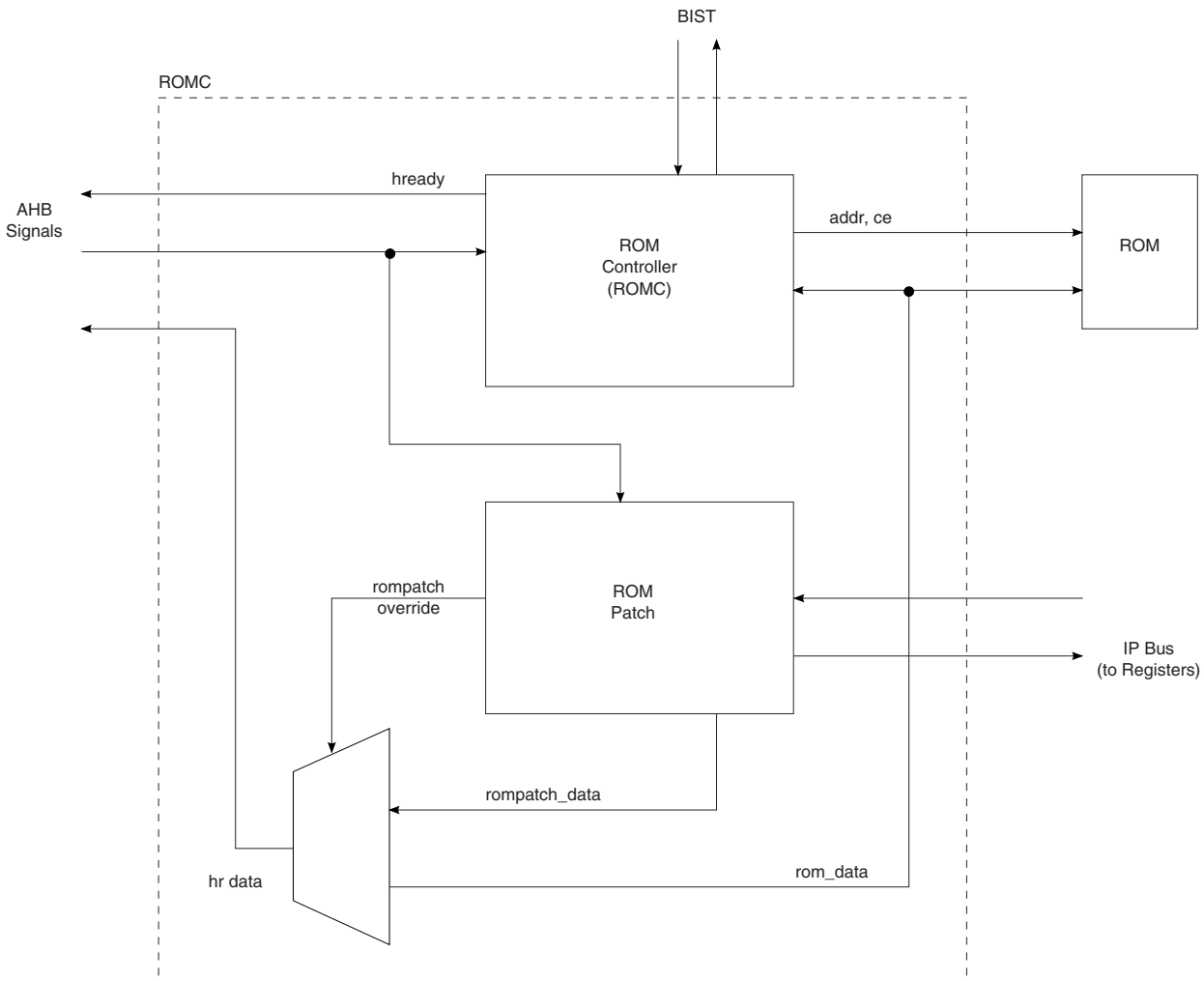


Figure 60-1. ROMC Block Diagram

60.1.2 Features

- Supports ROM size ranges from 16 Kbyte up to 4 Mbyte with increments of 1 Kbyte
- Supports opcode patching for a maximum of 16 different addresses in 4 Mbytes of ROM space
- Supports one-word data fixes for a max of 8 memory locations in 4 Mbytes of ROM space
- Supports patching of the Reset Vector (at 0x0000_0000) to allow external booting

60.1.3 Modes of Operation

There are two modes of operation: normal mode and BIST mode.

In normal mode (ipt_bist_en = 0), the ROMC ensures correct reads from the ROM, assuming the memory complies with the characteristics and requirements for which the ROMC was designed. The ROMC also performs hole decoding, aborting accesses into memory holes and converting the non-continuous ROM memory map into a continuous one for physical memory accesses.

60.1.3.1 Low Power Modes

There are two clock enables that are used to switch off parts of the ROMC logic when inactive. The first clock enable is used to disable the ROM Controller when the master connected to the AHB interface is not initiating a read to the ROM. The second clock enable is used to disable the registers used to program the ROM patch feature when the registers are not being accessed.

60.2 Memory Map

The figure below shows a sample memory map.

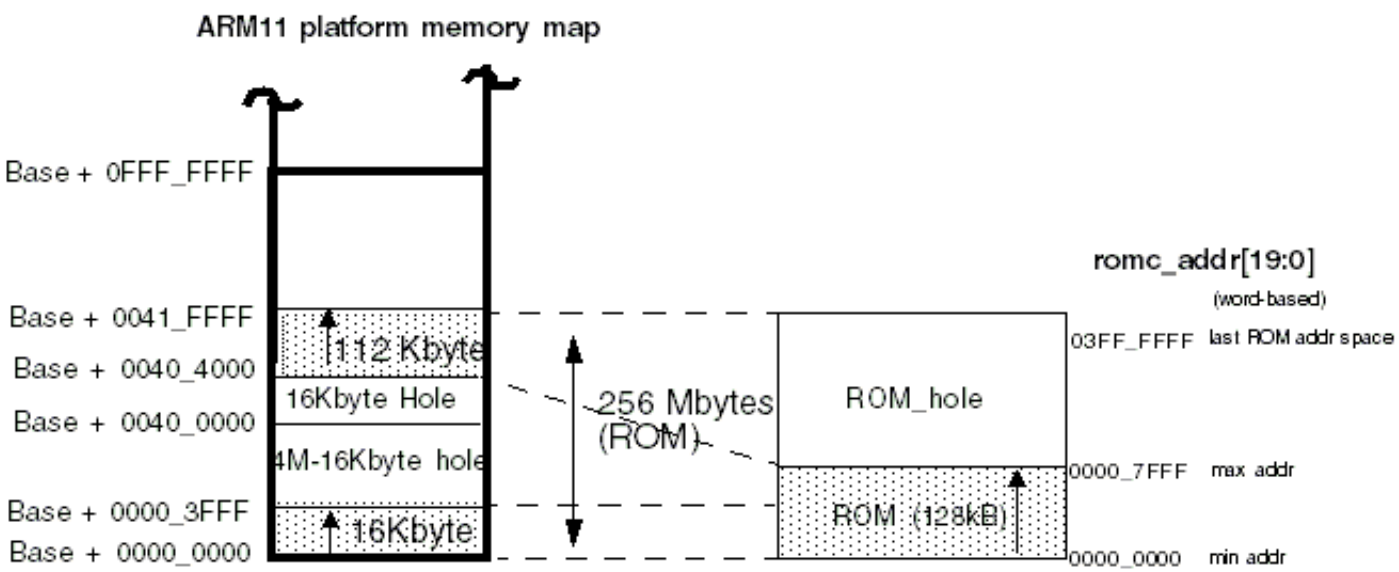


Figure 60-2. Memory Map-Example 128K ROM Connected

60.2.1 ROM Memory Map in detail

The ROMC supports ROM sizes with a range of 16 Kbyte to 4 Mbyte with an increment of 1 Kbyte. The 16 Kbyte lower limit was chosen because the minimum size of security code on an ARM platform is approximately 16 Kbyte of code, which is only accessible in supervisor mode. Note that it is the MMU that controls whether any region of memory is secure.

The exception vectors must be secured as well, and must be put in the same area as the security code. Since they must reside at address 0x0000_0000, the entire 16 Kbyte of ROM which can only be accessible in supervisor mode is located at the very beginning of the platform memory map.

If the user chooses not to use the security code, a memory size smaller than 16 Kbyte can be connected to the platform (minimum of 1 Kbyte). The MMU can be programmed to allow any kind of access into this memory. However, if the ROM size is less than 16 Kbyte, memory aliasing will occur for all invalid addresses greater than the memory size but within the 16 Kbyte of space.

For ROM sizes bigger than 16 Kbyte, the rest of its physical size resides at the address starting at 0x0040_4000 (4 M+16 Kbyte) going up to [0x0040_4000 + (mem. size - 16Kbyte)]. For examples on power-of-two memories please see Figure 60-3.

The reason for the big memory hole is because the ROMC is designed to be as flexible as possible to comply with demands of different operating systems. For example, the requirement of Windows CE OS is to be able to manage memory regions through the MMU page tables (to control accesses to privileged, cacheable, or bufferable areas) which requires regions to be separated into 1 Mbyte page tables. Since it is preferable to offer the flexibility of having the first 16 Kbyte as a secure region with vectors and security code, but allowing the rest of the ROM to be user-mode accessible, two separate 1Mbyte page tables are used.

To start a potentially insecure region directly on a 1 Mbyte boundary (starting on address 0010_0000) requires additional address translation within the ROMC. That causes an extra delay in the address path through the complicated shifting logic in an already critical design timing path. To prevent the extra delay, the first bit above the maximum address of the maximum ROM size that can be connected to the platform is used to select the region. After entering the ROMC this bit is only used in the ROM hole decode, and is not passed to the ROM on the address bus. Therefore, no translation is necessary and the address timing path is unaffected by the mapping scheme.

The maximum size of ROM connectable to the ROMC is 4 Mbyte. To address it the address bits haddr[21:2] are needed - as a consequence haddr[22] is the bit selecting between the upper or lower ROM in cases where the ROM is bigger than 16 Kbyte. That creates a 4 Mbyte hole between the two regions, which is greater than the required 1 Mbyte boundary, but acceptable. The branch instruction of the ARM code is capable of jumping between these two separate regions. It also implies that the second (potentially unprotected) region always starts at address 0x0040_4000 (4 M-16 Kbyte) and goes up to the maximum size of the memory connected to the platform (- 16 Kbyte). If an address outside the given regions is accessed, the ROMC will respond with an AHB-lite two-cycle error response.

60.3 Functional Description

This section is divided up into the ROM Controller Functional Description and the ROMC functional description.

60.3.1 ROM Controller (ROMC) Functional Description

60.3.1.1 Functionality overview

The ROMC serves two main functions. First, as an interface between the AHB-Lite bus on an ARM platform and the ROM. Second, it drives and receives several signals for the BIST engine. In normal mode of operation, the ROMC monitors the AHB-Lite for memory access requests and performs the memory operation to the ROM.

The ROMC includes the option to wait state all accesses from either the ARM or non-ARM masters to ROM in the event that timing requirements will not allow single hclk clock cycle reads. If a wait state is required, the static inputs rom_wait_arm or rom_wait_alt_mstr can be set to 1 and accesses will take two hclk clock cycles. If wait states are not required, rom_wait_arm or rom_wait_alt_mstr can be set to 0 and accesses will take one hclk clock cycle to complete.

60.3.1.2 ROMC Architecture Diagram

A simple block diagram illustrating the internal architecture of the ROMC is shown in the figure below. The basic logic sub-blocks of the ROMC include an AHB-Lite interface, a ROM memory interface, a BIST sub-block and a sub-block for muxing between BIST mode and normal mode.

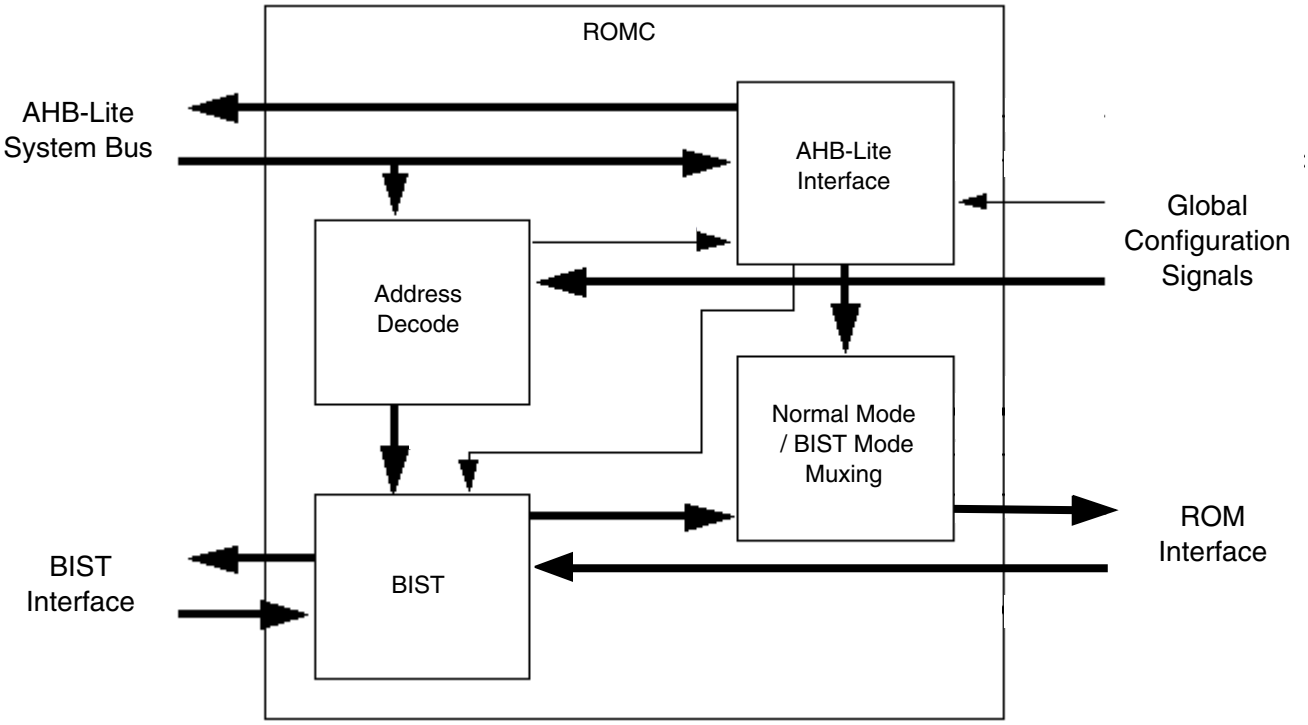


Figure 60-3. ROMC Internal Architecture

60.3.2 ROMC Functional Description

60.3.2.1 ROMC Disabling

All the bits in the ROMC_ROMPATCHENL register are cleared on Reset, disabling all the address comparators. Once the comparators have been enabled, the ROMC functions of data fixing and opcode patching can be quickly disabled by setting the DIS bit in the ROMC_ROMPATCHCNTL register. This bit is used to enable secure operations in which patching functions need to be disabled. This bit is cleared on Reset.

60.3.2.2 ROMC Event Priority

The ROMC has a total of 16 address comparators. The first 8 (0 through 7) comparators can be programmed for the data fixing function (through the 8 data fix enable bits in the ROMC_ROMPATCHCNTL register) while the rest are for opcode patching by default. This allows for potential multiple matching events involving both data fixing and opcode patch types. In these cases the ROMC assigns the highest priority to a data fixing event.

For example, if the ROMC is set up to data fix a certain address with comparator 4 and also opcode patch the same address with comparator 7, it will let comparator 4 have higher priority in indicating a match, and data from ROMC_ROMPATCHD4 will be put on the rompatch_romc_hrdata bus as the override value.

If multiple address matches of the same type level occur concurrently, then the ROMC will choose the source number based on the one with the highest source number. For example, the ROMC is setup to data fix the same location with address comparators 4 and 7, then address comparator 7 will have higher priority in indicating a match, and the value from ROMC_ROMPATCHD7 will be put on the rompatch_romc_hrdata bus as the orverride value. The same priority applies for an opcode patch event, except the override data is in the form of an SWI instruction with the comment field set to the source number with the highest priority.

60.3.2.3 Data Fixing

The data fixing feature allows ROM data to be updated by direct replacement when it is being read. This data usually originates from data tables, but can include ARM instructions. To enable data fixing on a certain address, this address value is written in to one of the first eight (0 through 7) of ROMC_ROMPATCHAxx registers and the same numbered bit set in the ROMC_ROMPATCHENL and ROMC_ROMPATCHCNTL registers. The data to be used for replacement is placed in the corresponding ROMC_ROMPATCHDxx.

The ROMC looks for a read access to ROM (either code fetch or data load) by snooping the AHB interface for read transactions. The address is compared with the values stored in the ROMC_ROMPATCHAxx[22:2] registers. If a match occurs from one of the comparators, the ROMC places the value in the corresponding ROMC_ROMPATCHDxx register on the read data bus by overriding the read data coming from the actual ROM (see the mux in Figure 60-1). The value on the read data bus is maintained until hready is asserted to terminate the access. In data fixing, the entire word is replaced so if a byte or half-word access occurs on a "data fix" location, the entire data word is replaced. The word being replaced is word aligned. (The two LSBs of the matching ROMC_ROMPATCHAxx are ignored in the data fix operation.)

60.3.2.4 Opcode Patching

The opcode patch feature provides the ARM core a mechanism to fetch updated versions of code routines that were originally programmed in ROM. This patching mechanism makes use of the SWI (software interrupt instruction) and a table of function pointers residing in writable memory. The opcode being patched is replaced with a SWI

instruction by the ROMC. Subsequent processing of the SWI reads from a function pointers table to obtain the address of the replacement code. Execution resumes with this code patch.

To enable opcode patching of a certain address, this address value is written into one of the ROMPATCHAxx registers and the corresponding bit set in the ROMPATCHENL to enable the associated comparator. The register's LSB (ROMC_ROMPATCHxx[0]) should be set if THUMB mode patching is in effect for this address. The ROMC identifies a ROM read access by snooping the AHB interface. The address is compared with the values stored in the ROMC_ROMPATCHAxx[22:2] registers. If a match occurs from one of the comparators, the ROMC generates the opcode of a software interrupt (SWI) instruction with the comment field containing the number of the matching address comparator. This opcode and comment is placed on the read data bus until hready is asserted by the ROM controller to terminate the read access.

The type of SWI generated, (that is, either ARM or THUMB), is determined by the LSB of the ROMC_ROMPATCHAxx register associated with the opcode patch. This bit is cleared for ARM mode (32 bits). The ROMC generates a 32-bit SWI (opcode field is 0xEF, occupying bits [31:24] of the word), with the least significant 5 bits of the 24-bit comment field (bits [23:0]) containing the number of the matching address comparator. The rest of the comment field is filled with zeros. This means that the ROMC will use 16 of the 16777216 possible software interrupts. The ROMC overrides the read data from the ROM.

If the LSB of the matching ROMC_ROMPATCHAxx register is set, the opcode patch is in THUMB mode (16 bits or half word). The ROMC generates a 16-bit SWI instruction (opcode field is 0xDF, occupying bits [15:8] of the half word) with the least significant 5 bits of the 8-bit comment field containing with the source number of the address comparator. The rest of the comments field is filled with zeros. This means that the ROMC will use 16 of the 256 possible software interrupts. The ROMC puts this 16 bit SWI instruction value on the proper half of the rompatch_romc_hrdata bus. The other half is zeroed out. Which half of the bus contains the SWI opcode and comment depends on the mode (Big Endian or Little Endian) and the bit 1 of the matching ROMC_ROMPATCHAxx register. In Little Endian mode, the lower half is bits {15:0] and the upper half is bits {31:16]. The order is reversed in Big Endian mode.

In Little Endian mode (bigend signal negated), if bit 1 of the matching ROMC_ROMPATCHAxx is cleared (lower half word selected) then the SWI instruction is put on the lower 16 bits of the read data bus and the upper 16 bits are zeroed out. Only the lower 16 bits of the read data bus is overwritten by the ROMC data. If ROMC_ROMPATCHAxx[1] is set (upper half word selected), the SWI instruction is put on the upper 16 bits of the read data bus and the lower 16 bits are zeroed out. Only the upper 16 bits of the read data bus is overwritten.

In Big Endian mode (bigend asserted), if bit 1 of the matching ROMC_ROMPATCHAxx is cleared (lower half word selected) then the SWI instruction is put on the upper 16 bits of the read data bus while the lower 16 bits are zeroed out. Only the upper 16 bits of the read data bus is overwritten. If ROMC_ROMPATCHAxx[1] is set (upper word selected), the SWI instruction is put on the lower 16 bits and the upper 16 bits are zeroed out. Only the lower 16 bits of the read data bus is overwritten.

The eventual execution of the SWI causes the ARM to save the CPSR in SPSR_SVC, the address of the next instruction after the SWI in R14_SVC, enter Supervisor mode, and fetch the SWI vector at 0x8, which then takes it to a handler for further processing as described in the next section.

60.3.2.4.1 Typical Software Response to Opcode Patch

When the SWI handler executes it needs to determine whether the SWI was generated by the ROMC. This is done by loading the SWI instruction and extracting its comment field. The state of the ARM core (ARM or THUMB) when the SWI was executed dictates whether to load the instruction word (ARM) or half word (THUMB). This state information can be determined by testing the T bit (bit 5) of the SPSR. If it's set, the execution was in THUMB mode.

By convention, if the comment field of the SWI is greater than 16, the software interrupt was initiated by software (i.e. an operating system call), and a branch is taken to the appropriate handler routine for further processing. If the comment field is less than 16, the SWI was generated by the ROMC performing a code patch operation. In this case, the software then reads from a table of function pointers, using the value in the SWI comment field as the index into the table. The value that is read is the address of the code patch. This value is loaded into the PC to begin the execution of the code patch. The following code segment illustrates a typical handling of the SWI.

```
stmfd        sp!, {r0-r1,lr}       @ push register onto SWI stack
mrs         r0, spsr        @ get saved status register
tst          r0, #0x20     @ check if call was in THUMB mode
ldrneh      r0, [lr,#-2]        @ yes: load opcode half-word and
bicne       r0, r0, #0xff00   @ yes: extract THUMB comment
ldreq       r0, [lr,#-4]        @ no: load opcode word and
biceq       r0, r0, #0xff000000 @ no: extract ARM comment
                    @ now r0 has comment field
cmp         r0, #16      @ compare to 16 (maximum for ROMC)
ldrlt          lr, =rompatch_tbl_ptr @ < 16: get top of current ROMC
                    @ table; global variable which is
                    @ changeable per context
ldrlt          r1, [lr, r0, lsl #2]   @ < 16: read function pointer from
                    @ table assumed an array of pointers
                    @ patch functions
strlt           r1, [sp, #8]     @ < 16: store function pointer onto
                    @ stack in position of link register
ldmltfd        sp!, {r0-r1,pc}^      @ < 16: "fake" return from SWI, will
                    @ vector core to appropriate patch
                    @ function and set core back to previous
                    @ mode of operating
ldr          r1, =swi_hdler   @ >= 16: pointer to standard SWI
```

```
                        @ handler
mov         lr, pc          @ >= 16: set link register
bx          r1            @ >= 16: jump to standard SWI
                        @ handler
ldmfd        sp!, {r0-r1,pc}^      @ >= 16: pop registers from stack
```

60.3.2.5 External Boot Feature

Following a Reset event, the ARM issues an instruction fetch of the Reset Vector from address 0x0. This instruction, normally residing in ROM is usually a branch to a Reset handler or boot code which also normally resides in ROM. The ROMC external boot feature allows the bypassing of this code, using a different boot code residing perhaps in external memory.

This feature uses the data fix mechanism and works as follows: if the boot_int signal is negated when a Reset event occurred, the ROMC will perform a data fix of the Reset Vector at 0x0 with the following instruction (opcode 0xE59FF00C):

```
ldr         pc,[pc,#12]      @ read 0x0000_0014 for reset_vector
```

The value of PC when this instruction is executed is 8 so that a PC relative offset of 12 makes the source address 20 or 0x14. When this instruction executes, the ARM core reads from address 0x0000_0014, triggering a ROMC data fix operation which places the value taken from the external boot address on the read data bus, with the two LSBs zeroed out. This value is returned to the ARM to be placed in the PC causing code fetch and execution to start from that address.

60.3.2.6 Alternate Masters and ROMC

The ROMC sits on the AHB bus of the internal ROM (ROMC). This means that the ROMC can modify values on the read data bus going to the master. Therefore, any master which reads an opcode patched or data patched location will read patched data.

60.4 Programmable Registers

All registers are accessible through an IP Bus and can only be accessed in privileged mode. These registers can only be written with 32-bits stores and are clocked by hclk_reg.

The ROMC register placement was originated from the AWPT design used in the ARM7 platform of Neptune

ROMC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FB_80D4	ROMC Data Registers (ROMC_ROMPATCH0D)	32	R/W	0000_0000h	60.4.1/ 3772
63FB_80D8	ROMC Data Registers (ROMC_ROMPATCH1D)	32	R/W	0000_0000h	60.4.1/ 3772
63FB_80DC	ROMC Data Registers (ROMC_ROMPATCH2D)	32	R/W	0000_0000h	60.4.1/ 3772
63FB_80E0	ROMC Data Registers (ROMC_ROMPATCH3D)	32	R/W	0000_0000h	60.4.1/ 3772
63FB_80E4	ROMC Data Registers (ROMC_ROMPATCH4D)	32	R/W	0000_0000h	60.4.1/ 3772
63FB_80E8	ROMC Data Registers (ROMC_ROMPATCH5D)	32	R/W	0000_0000h	60.4.1/ 3772
63FB_80EC	ROMC Data Registers (ROMC_ROMPATCH6D)	32	R/W	0000_0000h	60.4.1/ 3772
63FB_80F0	ROMC Data Registers (ROMC_ROMPATCH7D)	32	R/W	0000_0000h	60.4.1/ 3772
63FB_80F4	ROMC Control Register (ROMC_ROMPATCHCNTL)	32	R/W	0840_0000h	60.4.2/ 3773
63FB_80F8	ROMC Enable Register High (ROMC_ROMPATCHENH)	32	R	0000_0000h	60.4.3/ 3774
63FB_80FC	ROMC Enable Register Low (ROMC_ROMPATCHENL)	32	R/W	0000_0000h	60.4.4/ 3775
63FB_8100	ROMC Address Registers (ROMC_ROMPATCH0A)	32	R/W	0000_0000h	60.4.5/ 3776
63FB_8104	ROMC Address Registers (ROMC_ROMPATCH1A)	32	R/W	0000_0000h	60.4.5/ 3776
63FB_8108	ROMC Address Registers (ROMC_ROMPATCH2A)	32	R/W	0000_0000h	60.4.5/ 3776
63FB_810C	ROMC Address Registers (ROMC_ROMPATCH3A)	32	R/W	0000_0000h	60.4.5/ 3776
63FB_8110	ROMC Address Registers (ROMC_ROMPATCH4A)	32	R/W	0000_0000h	60.4.5/ 3776
63FB_8114	ROMC Address Registers (ROMC_ROMPATCH5A)	32	R/W	0000_0000h	60.4.5/ 3776
63FB_8118	ROMC Address Registers (ROMC_ROMPATCH6A)	32	R/W	0000_0000h	60.4.5/ 3776
63FB_811C	ROMC Address Registers (ROMC_ROMPATCH7A)	32	R/W	0000_0000h	60.4.5/ 3776
63FB_8120	ROMC Address Registers (ROMC_ROMPATCH8A)	32	R/W	0000_0000h	60.4.5/ 3776

Table continues on the next page...

ROMC memory map (continued)

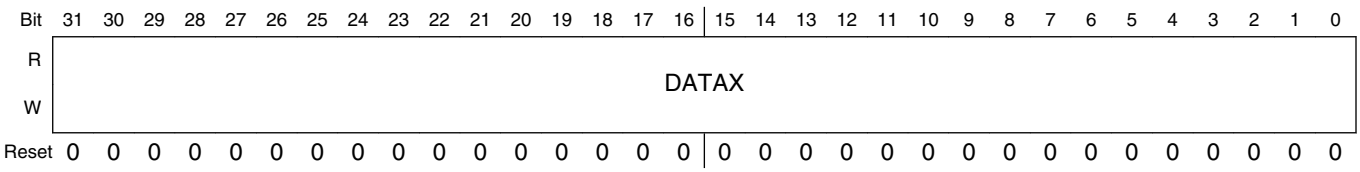
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FB_8124	ROMC Address Registers (ROMC_ROMPATCH9A)	32	R/W	0000_0000h	60.4.5/ 3776
63FB_8128	ROMC Address Registers (ROMC_ROMPATCH10A)	32	R/W	0000_0000h	60.4.5/ 3776
63FB_812C	ROMC Address Registers (ROMC_ROMPATCH11A)	32	R/W	0000_0000h	60.4.5/ 3776
63FB_8130	ROMC Address Registers (ROMC_ROMPATCH12A)	32	R/W	0000_0000h	60.4.5/ 3776
63FB_8134	ROMC Address Registers (ROMC_ROMPATCH13A)	32	R/W	0000_0000h	60.4.5/ 3776
63FB_8138	ROMC Address Registers (ROMC_ROMPATCH14A)	32	R/W	0000_0000h	60.4.5/ 3776
63FB_813C	ROMC Address Registers (ROMC_ROMPATCH15A)	32	R/W	0000_0000h	60.4.5/ 3776
63FB_8208	ROMC Status Register (ROMC_ROMPATCHSR)	32	w1c	0000_0000h	60.4.6/ 3777

60.4.1 ROMC Data Registers (ROMC_ROMPATCHD)

The ROMC data registers (ROMC_ROMPATCHD7 through ROMC_ROMPATCHD0) store the data to use for the 8 1-word data fix events. Each register is associated with an address comparator (7 through 0). When a data fixing event occurs, the value in the data register corresponding to the comparator that has the address match is put on the romc_hrdata[31:0] bus until romc_hready is asserted by the ROM controller to terminate the access. A MUX external to the ROMC will select this data over that of romc_hrdata[31:0] in returning read data to the ARM core. The selection is done with the control bus rompatch_romc_hrdata_ovr[1:0] with both bits asserted by the ROMC.

If more than one address comparators match, the highest-numbered one takes precedence, and the value in corresponding data register is used for the patching event.

Addresses: ROMC_ROMPATCH0D is 63FB_8000h base + D4h offset = 63FB_80D4h
ROMC_ROMPATCH1D is 63FB_8000h base + D8h offset = 63FB_80D8h
ROMC_ROMPATCH2D is 63FB_8000h base + DCh offset = 63FB_80DCh
ROMC_ROMPATCH3D is 63FB_8000h base + E0h offset = 63FB_80E0h
ROMC_ROMPATCH4D is 63FB_8000h base + E4h offset = 63FB_80E4h
ROMC_ROMPATCH5D is 63FB_8000h base + E8h offset = 63FB_80E8h
ROMC_ROMPATCH6D is 63FB_8000h base + ECh offset = 63FB_80ECh
ROMC_ROMPATCH7D is 63FB_8000h base + F0h offset = 63FB_80F0h



ROMC_ROMPATCH*n*D field descriptions

Field	Description
31–0 DATAX	Data Fix Registers - Stores the data used for 1-word data fix operations. The values stored within these registers do not affect the writes to the memory system. They are selected over the read data from ROM when a data fix event occurs. If any part of the 1-word data fix is read, then the entire word is replaced. Therefore, a byte or half-word read will cause the ROMC to replace the entire word. The word is word address aligned.

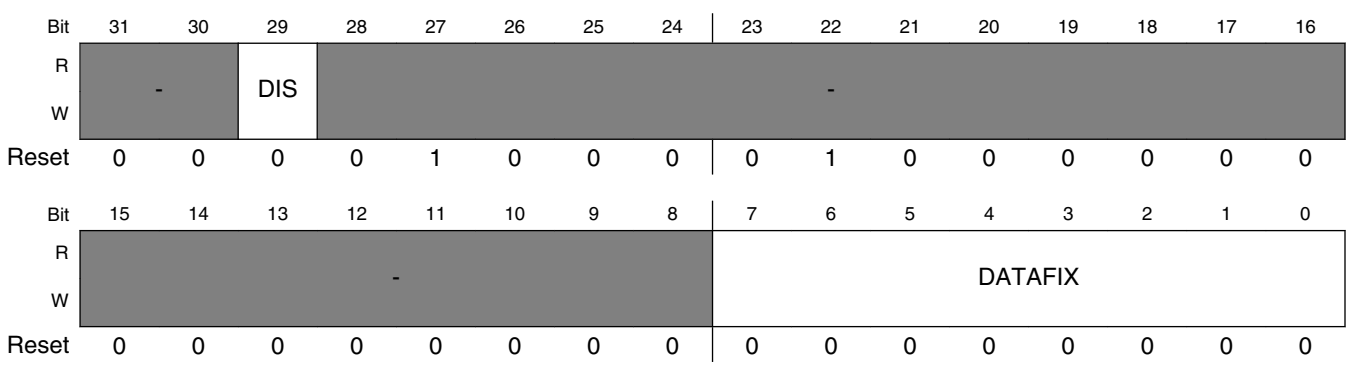
60.4.2 ROMC Control Register (ROMC_ROMPATCHCNTL)

The ROMC control register (ROMC_ROMPATCHCNTL) contains the block disable bit and the data fix enable bits. The block disable bit provides a means to disable the ROMC data fix and opcode patching functions, even when the address comparators are enabled. The External Boot feature is not affected by this bit. The eight data fix enable bits (0 through 7), when set, assign the associated address comparators to data fix operations

NOTE

Bits 27 and 22 always read as 1s.

Address: ROMC_ROMPATCHCNTL is 63FB_8000h base + F4h offset = 63FB_80F4h



ROMC_ROMPATCHCNTL field descriptions

Field	Description
31–30 -	Reserved
29 DIS	ROMC Disable -- This bit, when set, disables all ROMC operations. This bit is used to enable secure operations. 0 Does not affect any ROMC functions (default) 1 Disable all ROMC functions: data fixing, and opcode patching
28–8 -	Reserved
7–0 DATAFIX	**Data Fix Enable - Controls the use of the first 8 address comparators for 1-word data fix or for code patch routine.** 0 Address comparator triggers a opcode patch 1 Address comparator triggers a data fix

60.4.3 ROMC Enable Register High (ROMC_ROMPATCHENH)

The ROMC enable register high (ROMC_ROMPATCHENH) and ROMC enable register low (ROMC_ROMPATCHENL) control whether or not the associated address comparator can trigger a opcode patch or data fix event. This implementation of the ROMC only has 16 comparators, therefore ROMC_ROMPATCHENH and the upper half of ROMC_OMPATCHENL are read-only. ROMC_ROMPATCHENL[15:0] are associated with comparators 15 through 0. ROMC_ROMPATCHENLH[31:0] would have been associated with comparators 63 through 32.

Address: ROMC_ROMPATCHENH is 63FB_8000h base + F8h offset = 63FB_80F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ROMC_ROMPATCHENH field descriptions

Field	Description
31–0 -	Reserved

60.4.4 ROMC Enable Register Low (ROMC_ROMPATCHENL)

The ROMC enable register high (ROMC_ROMPATCHENH) and ROMC enable register low (ROMC_ROMPATCHENL) control whether or not the associated address comparator can trigger a opcode patch or data fix event. This implementation of the ROMC only has 16 comparators, therefore ROMC_ROMPATCHENH and the upper half of ROMC_ROMPATCHENL are read-only. ROMC_ROMPATCHENL[15:0] are associated with comparators 15 through 0. ROMC_ROMPATCHENLH[31:0] would have been associated with comparators 63 through 32.

Address: ROMC_ROMPATCHENL is 63FB_8000h base + FCh offset = 63FB_80FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																ENABLE																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

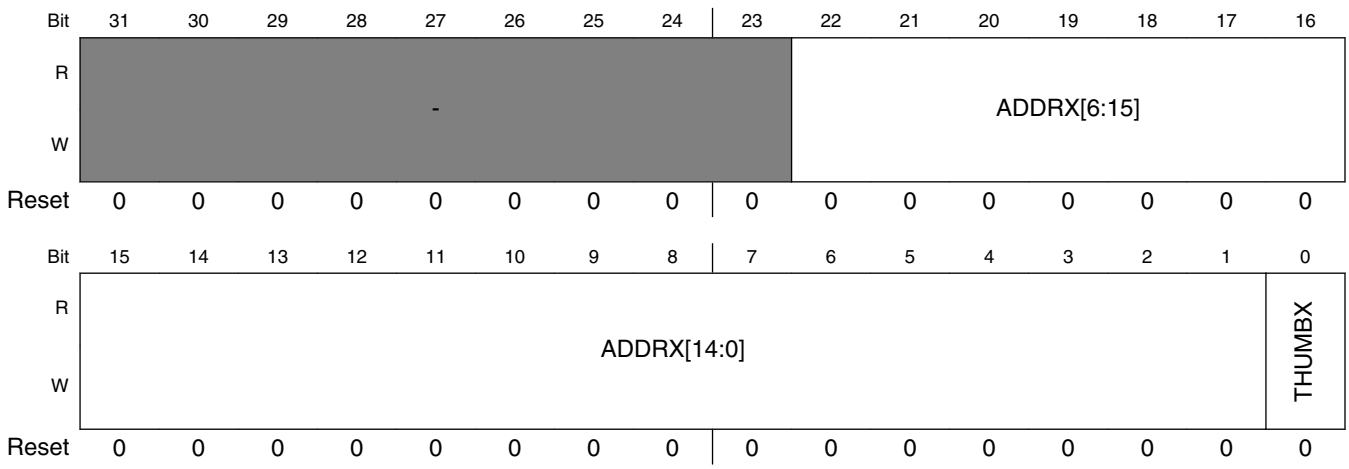
ROMC_ROMPATCHENL field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved 0 Address comparator disabled 1 Address comparator enabled, ROMC will trigger a opcode patch or data fix event upon matching of the associated address
15–0 ENABLE	**Enable** Address **Comparator -** This bit enables the corresponding address comparator to trigger an event. 0 Address comparator disabled 1 Address comparator enabled, ROMC will trigger a opcode patch or data fix event upon matching of the associated address

60.4.5 ROMC Address Registers (ROMC_ROMPATCHA)

The ROMC address registers (ROMC_ROMPATCHA0 through ROMC_ROMPATCHA15) store the memory addresses where opcode patching begins and data fixing occurs. The address registers ROMC_ROMPATCHA0 through ROMC_ROMPATCHA15 are each 21 bits wide and dedicated to one 4 Mbyte memory space. Bits 21 through 2 are address bits, to be compared with romc_haddr[21:2] for a match; bit 1 is also an address bit used for half word selection. Bit 0 is the mode bit (set to 1 for THUMB mode). 1-word data fixing can only be used on the first 8 of the address comparators. ROMC_ROMPATCHA0 through ROMC_ROMPATCHA15 are associated each with address comparators 0 through 15.

Addresses: 63FB_8000h base + 100h offset + (4d × *n*), where *n* = 0d to 15d



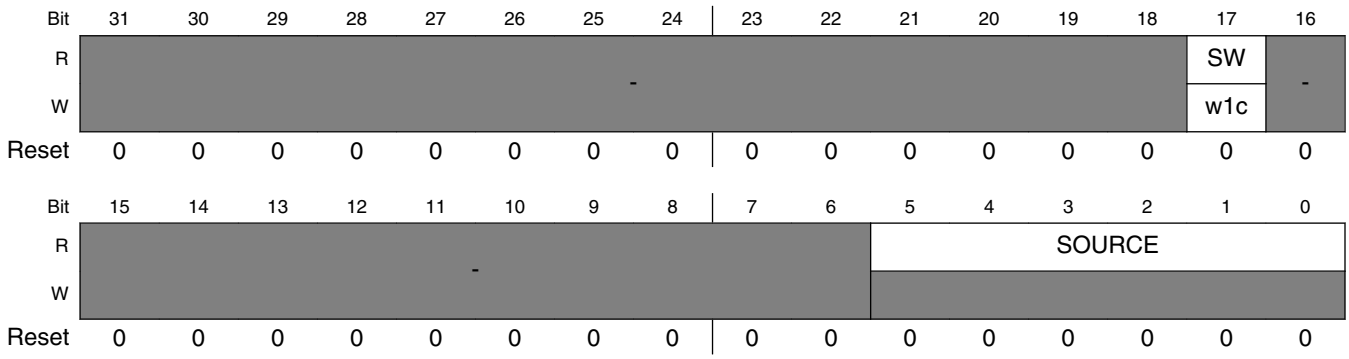
ROMC_ROMPATCH*n*A field descriptions

Field	Description
31–23 -	Reserved
22–1 ADDRX	Address Comparator Registers - Indicates the memory address to be watched. All 16 registers can be used for code patch address comparison. Only the first 8 registers can be used for a 1-word data fix address comparison. Bit 1 is ignored if data fix. Only used in code patch
0 THUMBX	THUMB Comparator Select - Indicates that this address will trigger a THUMB opcode patch or an ARM opcode patch. If this watchpoint is selected to be a data fix, then this bit is ignored as all data fixes are 1-word data fixes. 0 ARM patch 1 THUMB patch (ignore if data fix)

60.4.6 ROMC Status Register (ROMC_ROMPATCHSR)

The ROMC status register (ROMC_ROMPATCHSR) indicates the current state of the ROMC and the source number of the most recent address comparator event.

Address: ROMC_ROMPATCHSR is 63FB_8000h base + 208h offset = 63FB_8208h



ROMC_ROMPATCHSR field descriptions

Field	Description
31–18 -	Reserved
17 SW	ROMC AHB Multiple Address Comparator matches Indicator - Indicates that multiple address comparator matches occurred. Writing a 1 to this bit will clear this it. 0 no event or comparator collisions 1 a collision has occurred
16–6 -	Reserved
5–0 SOURCE	ROMC Source Number - Binary encoding of the number of the address comparator which has an address match in the most recent patch event on ROMC AHB. If multiple matches occurred, the highest priority source number is used. 0 Address Comparator 0 matched 1 Address Comparator 1 matched 15 Address Comparator 15 matched

Chapter 61
Run-Time Integrity Checker (RTIC)

61.1 Overview

The Run-Time Integrity Checker (RTIC) is part of the Platform Independent Security Architecture (PISA) family of platform security components. Its function is to ensure the integrity of the peripheral memory contents and assist with boot authentication. The RTIC has the ability to verify the memory contents during system boot and during run-time execution. If the memory contents at runtime fail to match the hash signature, an error in the security monitor is triggered.

61.2 Features

The RTIC offers the following features:

- SHA-1 and SHA-256 message authentication
- Input DMA (AMBA-AHB Lite bus master) interface
- Segmented data gathering to support non-contiguous data blocks in memory (up to 2 segments per)
- Works with High Assurance Boot process
- Secure-scan DFT security
- Programmable DMA bus duty cycle timer
- Power-saving clock gating logic
- Full word memory reads (word-aligned addresses, multiple of 32-bit lengths)

61.2.1 Modes of Operation

The RTIC operates in two primary modes:

- One time hash mode

- Used during high assurance boot for code authentication or one time integrity checking
- Stores hash result internally and signals interrupt to host

- Continuous hash mode
  - Used at run time to continuously verify integrity of memory contents
  - Checks re-generated hash against internally stored values and interrupts host only if error occurs

Chapter 62
SAHARA Security Accelerator (SAHARA)

62.1 Overview

The Symmetric/Asymmetric Hashing and Random Accelerator (SAHARA) is a security coprocessor.

SAHARA implements block encryption algorithms, (AES, DES, and 3DES), hashing algorithms (MD5, SHA-1, SHA-224, and SHA-256), a stream cipher algorithm (ARC4), and a hardware random number generator. It has a slave IP bus interface for the host to write configuration and command information, and to read status information. SAHARA also has a DMA controller with an AHB bus interface to reduce the burden on the host to move the required data to and from memory.

See the figure below for the block diagram of SAHARA.

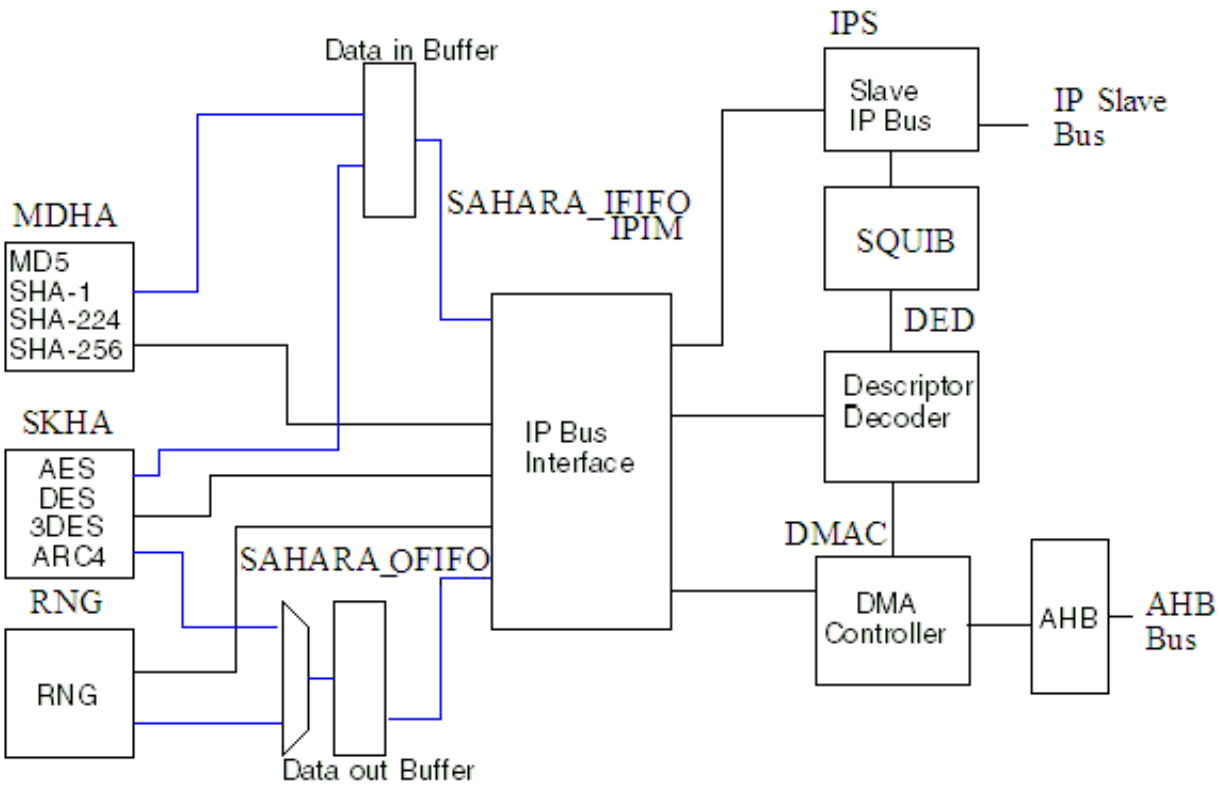


Figure 62-1. Block Diagram of SAHARA

62.2 Features

SAHARA accelerates the following security functions:

- AES encryption/decryption
  - ECB, CBC, CTR, and CCM modes
  - 128-bit key
- DES/3DES
  - ECB, CBC and CTR modes
  - 56-bit key with or without parity (DES)
  - 112-bit or 168-bit key with parity (3DES)
- ARC4 (RC4-compatible cipher)
  - 5-16 byte key
  - Host accessible S-box
- MD5, SHA-1, SHA-224 and SHA-256 hashing algorithms.
  - Messages lengths that are multiples of bytes (with auto padding).
  - Auto padding supported.

    - HMAC (support for IPAD and OPAD via Descriptors).
    - Up to 2^{32} byte message length (with auto padding).
- Pseudo-Random number generator
    - Entropy is generated via independent free running ring oscillators.

SAHARA also provides the following features:

- Descriptor based processing to reduce communication between host processor and SAHARA
- Multiple Master Interface (SQUIB), allows two masters to share SAHARA transparently.
- ARM TrustZone support via additional AHB and IPBus signalling
- Low power design
    - Automatic power down of individual blocks when not in use
    - Clock gating on registers
    - RNG sleep mode
- Restricted access to potentially sensitive information
    - Internal registers are cleared after a Descriptor chain has completed processing in BATCH mode.
    - SCC Security Monitor can cause data to be cleared.
    - Scan reset and scan exit signals prevent data being scanned out.
- Optimization for Maximum of 8-word AHB Burst.

62.2.1 Modes of Operation

SAHARA can be used in three modes. The mode bits can be changed only when SAHARA is idle. Otherwise, the mode bits are not written and no other indication is given.

62.2.1.1 BATCH Mode-Overview

The BATCH mode is the default operating mode. After a chain of Descriptors has been processed, the internal functional blocks are initialized, and an interrupt is generated. Initializing SAHARA prevents a task from being able to access the left over data from a Descriptor chain of a different task.

62.2.1.2 DEDICATED Mode-Overview

The DEDICATED mode is used when SAHARA is dedicated to a single task. In the DEDICATED mode, SAHARA does not reset the internal state information at the end of a Descriptor chain. This allows a task to improve performance by not having to save and restore its context information between each set of Descriptor chains. This is useful when a task does not have access to all of the data at one time, and must therefore present its Descriptors at different times.

62.2.1.3 DEBUG Mode-Overview

In direct access DEBUG mode, the host writes directly to the address space of the internal hardware blocks. SAHARA single-steps Descriptors only when in DEBUG mode. This is intended only for interactive debugging, and is disabled when the SCC Security Monitor is in the Secure state .

In the DEBUG mode, the host processor can read and write the internal encryption engine's registers. No interrupts are generated while SAHARA is in DEBUG mode.

NOTE

Detailed information about SAHARA is provided in the Securety Reference Manual. Contact your Freescale representative for information about obtaining this document.

Chapter 63
Serial Advanced Technology Attachment Controller (SATA)

63.1 Introduction

The chip includes an integrated Serial Advanced Technology Attachment (SATA) Controller that is compatible with the Advanced Host Controller Interface (AHCI) specification.

The SATA Controller block (SATA) along with integrated physical link hardware (SATA PHY) provide one SATA port for the attachment of external SATA compliant storage devices.

The following section introduces the block features and architecture.

63.1.1 Features

The SATA block supports the following features:

- Compliant with the following specifications:
  - Serial ATA 2.6
  - AHCI Revision 1.3 (except FIS-based switching)
  - AMBA 2.0 from ARM
- SATA 1.5 Gb/s speed.
- eSATA (external analog logic also needs to support eSATA)
- RX data buffer for recovered clock systems
- Data alignment circuitry when RX data buffer is also included
- OOB signaling detection and generation
- 8b/10b encoding/decoding
- Asynchronous signal recovery, including retry polling
- Power management features including automatic partial-to-slumber transition
- BIST loopback modes

- Supports one SATA device(port0)
- Configurable AMBA AHB interface (one master and one slave for each interface)
- Internal DMA engine
- Hardware-assisted Native Command Queuing for up to 32 entries
- Port Multiplier with command-based switching
- Disabling RX and TX Data clocks during power down modes

63.1.2 System Overview

SATA is an AHCI-compliant Host Bus Adapter (HBA). Together with the corresponding physical layer (PHY), it forms a complete AHCI HBA interface.

Note that only Port 0, PHY0, and bus interface are implemented in this product.

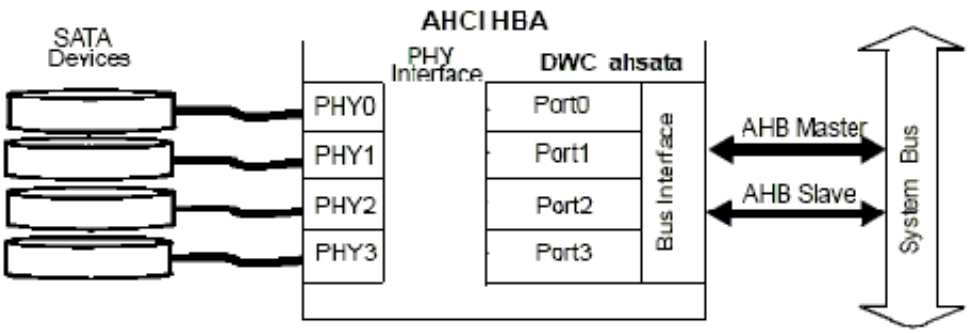


Figure 63-1. SATA AHCI System Block Diagram (Shown with Four Ports)

63.2 Block Overview

This section describes the functional sub-blocks of the SATA block

63.2.1 Block Diagram

The following figure provides a high-level view of the SATA architecture, followed by a brief description. The diagram shows additional ports and PHY interfaces not implemented in this product.

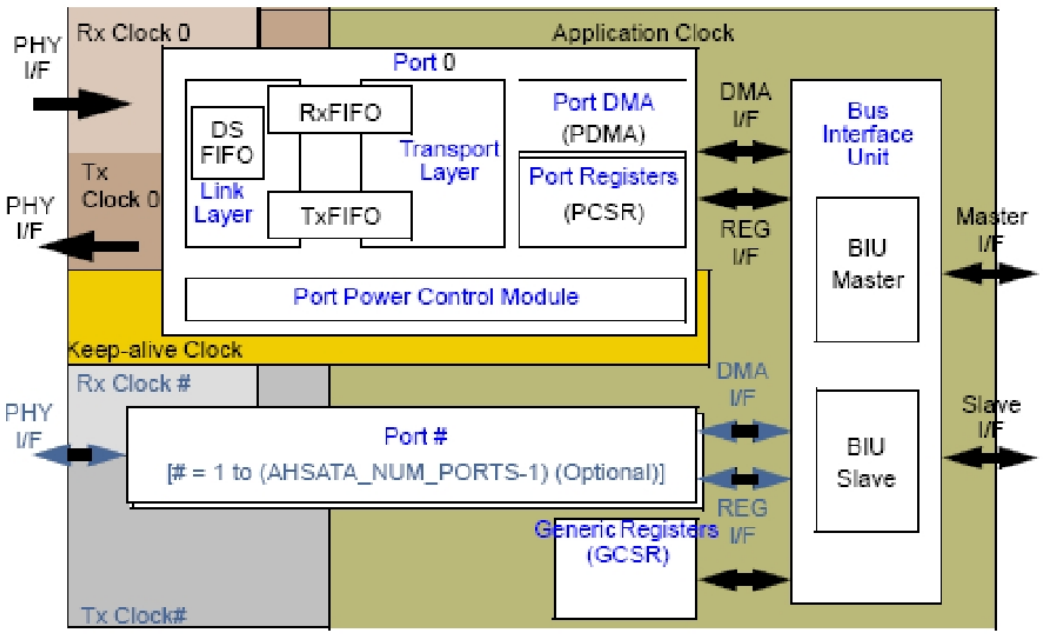


Figure 63-2. SATA Block Diagram

The SATA block consists of three main sub-blocks:

- Bus Interface Unit (BIU)
- Generic Registers (GCSR) (GCSR)
- Port

The system bus provides the application clock (hclk). BIU, GCSR and part of the Port operate in the application clock domain. Rx (when present), and Tx clocks are generated in the PHY and depend on the interface speed and PHY data width, as shown in the table below. The Rx OOB clock frequency is 50 or 60 MHz depending on PHY's reference clock (external or internal respectively).

Table 63-1. PHY-Supplied Clock Frequencies (clk rbc and clk asic)

Interface Speed (GB/s)	8/10 bits (Mhz)	16/20 bits (Mhz)	32/40 bits (Mhz)
1.5	150	75	37.5

Port logic contains its own DMA engine that implements AHCI functionality and initiates all of the data transfers between the external SATA device and system memory.

The Port PHY interface operates in three or four clock domains: RxOOB clock (clk_rxoob), Rx clock (clk_rbc) when present, keep-alive clock (clk_pmalive) and Tx clock (clk_asic). Most of the Link Layer (both receive and transmit data paths), and part of the Transport Layer always operate in the Tx clock domain. The Rx clock is a clock

recovered from the PHY, and is used for clocking data into the SATA block. The Port contains a Power Control Module operating in the always-alive pmalive clock domain. This Power Control Module facilitates disabling Rx and Tx clocks in power down modes.

All SATA block clocks are asynchronous to each other in general. The BIU connects the Port to the system bus. AHB master interface is used to transfer data between the Port and the system memory, while the AHB slave interface is used by the software (driver) to access SATA block registers. All the SATA global registers are implemented in the Generic Register module.

63.2.2 SATA Block Transfer Hierarchy

The following figure shows the SATA block transfer hierarchy from the system bus point of view.

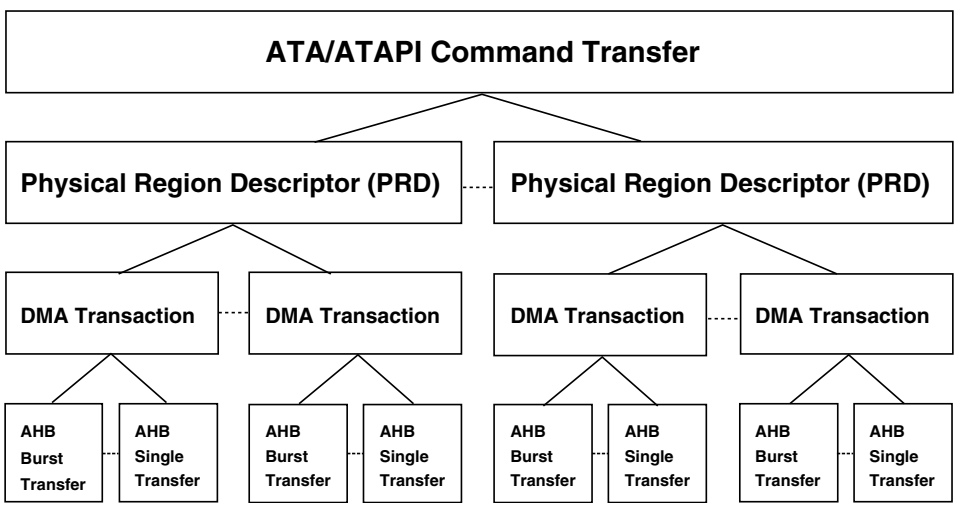


Figure 63-3. SATA Block Transfer Hierarchy

Data for an ATA/ATAPI command resides in the system memory in the form of one or more Physical Region Descriptors (PRD). PRD entries describe the physical location and length of data to be transferred.

Each PRD entry is read by the SATA Port DMA from the system memory before transferring the block of data associated with this entry.

The Port DMA transfers the PRD data block between the system memory and its FIFOs using one or more DMA transactions. DMA transaction size can be set by software.

Finally, the BIU Master generates one or more AHB burst or single bus transfers based on the DMA transaction request from the PDMA. The BIU Master tries to generate a burst equal to the DMA transaction size, but it might break it into several burst/single transfers due to various bus conditions (for example, early burst termination, 1-KB address boundary crossing, etc.).

63.2.3 Standards Compliance

1. The Serial ATA specifications can be found at the following website:

   http://sata-io.org

2. The AMBA Specification can be found at the following website:

   http://www.arm.com/products/solutions/AMBA_Spec.html

3. The AHCI specification can be found at the following website:

   http://www.intel.com/technology/serialata/ahci.htm

In the event of any conflict between the information in this chapter and the AHCI specification, the AHCI specification prevails.

63.3 Architecture

This section describes the SATA block interfaces, protocols, functionality, and implementation.

63.3.1 Architecture Overview

The SATA block diagram is shown in the following figure. While this diagram shows multiple ports, note that only one port is implemented.

The functional blocks that make up SATA are described in the following sub-sections.

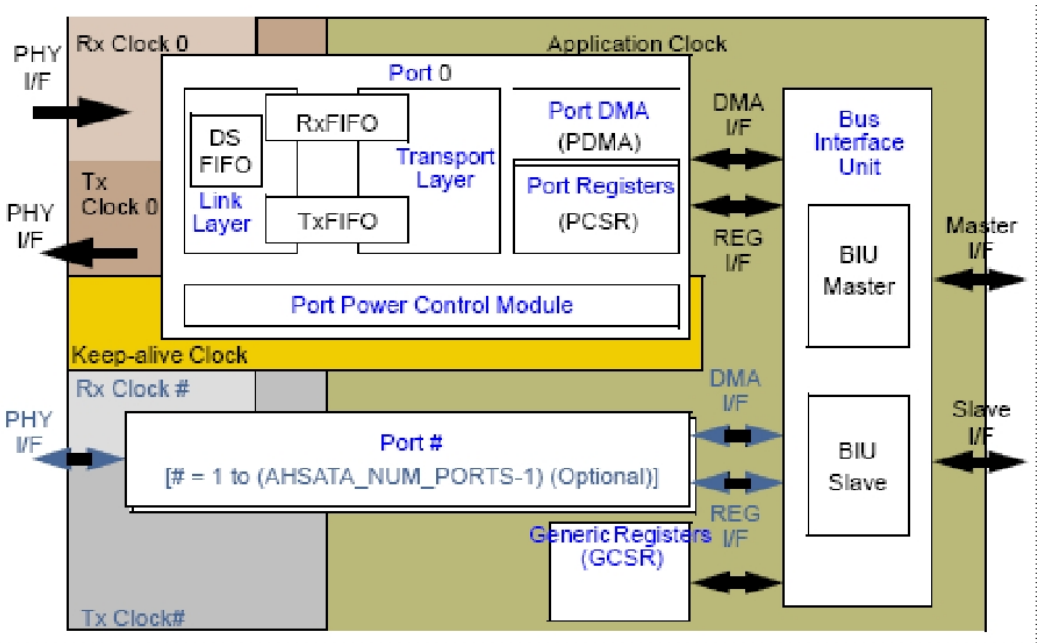


Figure 63-4. SATA Block Functional Diagram

63.3.2 Bus Interface Unit

The Bus Interface Unit (BIU) connects the AHB Master to the internal Port DMA controllers, and connects the AHB Slave to the internal Port registers.

The BIU is comprised of the following:

- AHB Slave Bus/GIF Interface (AHB Slave) and mapping logic.
- Register Read MUX, mapped to the AHB Slave read-response channel.
- AHB Master Bus/GIF Interface (AHB Master) and mapping logic.
- DMA Arbiter, mapped to the AHB Master.

The following figure displays a block diagram of the Bus Interface Unit.

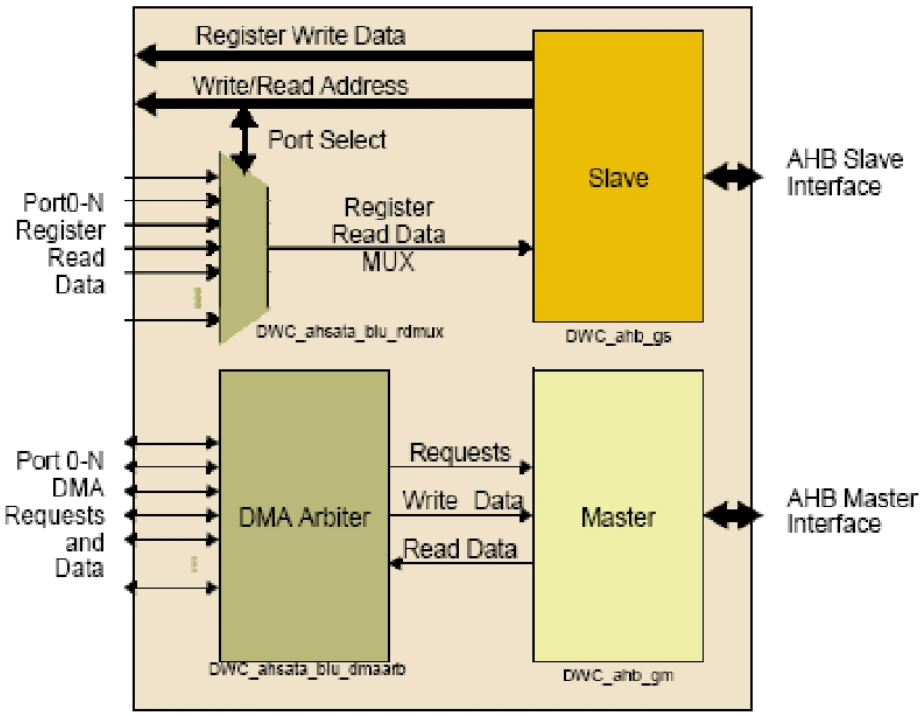


Figure 63-5. SATA Block Bus Interface Unit Block Diagram

63.3.2.1 AHB Slave Bus GIF Interface

The AHB Slave Bus GIF Interface (AHB Slave) converts AHB bus cycles to internal read/write request and response channel signals, which are mapped to Generic and Port control registers (GCSR and PCSR).

Characteristics of this unit include the following:

- Fully AMBA 2.0-Compliant AHB slave interface
- Supports single transfer type
- Supports INCR burst type
- Configured for little-endian byte ordering
- Supports 32-bit data bus width and 32-bit address width
- Master supports 32-bit (burst/single) and 16-bit (single only) transfer sizes, slave supports 8, 16, and 32-bit transfer sizes using SINGLE, INCR4, INCR8, and INCR16 burst types.
- Supports master BUSY and early burst termination
- Generates OKAY or ERROR response (SPLIT and RETRY are not supported).

The wait state for different accesses are:

- 0 wait state for any NSEQ or SEQ write access, 1 wait state for any NSEQread access.
- 1 wait state for any read after write decode to the same address.

AHB Slave generates ERROR response on s_hresp output when it detects any of the following illegal accesses:

- Transfer size is not 8, 16, or 32-bit.
- Address is not 32-bit or greater aligned.

Each Port decodes the csr_waddr address to select the required register when AHB Slave external mapping asserts corresponding p0_csr_sel signal during register write access. When a Port is not configured, then its register locations should be treated as reserved: read accesses return zeros, write accesses have no effect.

Figure 3-3 shows various AHB Slave accesses.

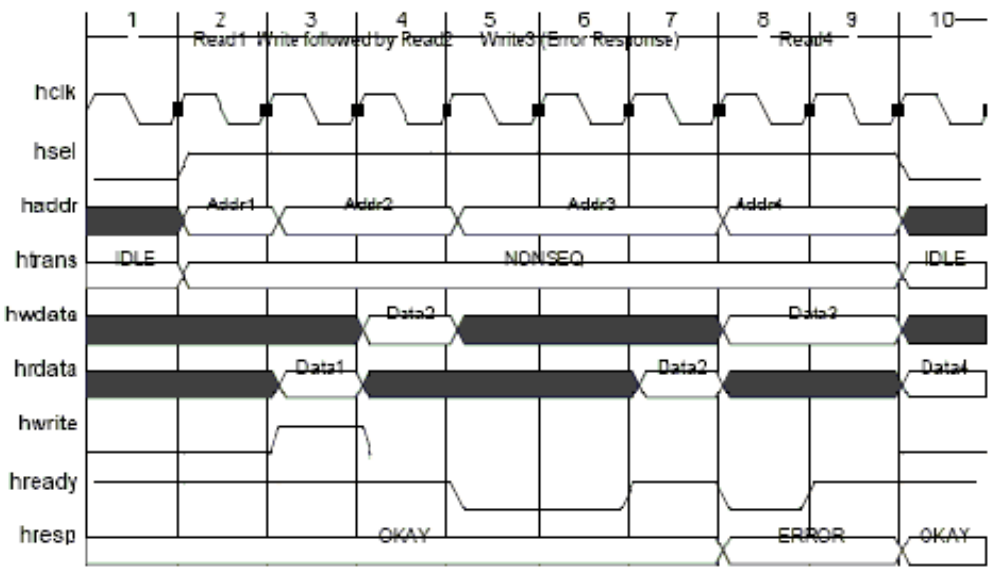


Figure 63-6. AHB Single Transfer

63.3.2.2 Register Read Multiplexer

The Register Read Multiplexer simply multiplexes all of the Port registers to the AHB Slave read response channel.

The AHB Slave read address selects which registers to read.

63.3.2.3 AHB Master Bus/GIF Interface

The AHB Master Bus/GIF Interface (AHB Master) converts Port DMA requests, from the DMA Arbiter, into AHB Master requests.

The AHB Master implements the following function:

- Converts Port DMA requests into AHB cycles

Characteristics of this unit include the following:

- Fully AMBA 2.0-compliant AHB master
- SINGLE, INCR4, INCR8 & INCR16 burst types supported on the Port DMA request:
- Handles AHB SPLIT, RETRY, ERROR conditions
- Supports early burst termination via requesting remainder of burst
- Configured for little-endian byte ordering
- Handles AHB 1-KB boundary breaking
- All AHB transfers are 16- or 32-bit aligned
- Does not support locked transfers (m_hlock=0) and wrapped burst transfer (WRAP type).

The following sequence of events describes basic AHB Master operation:

1. When a Port becomes active, that is has data in the RxFIFO to transfer to system memory, or needs data to transfer to the Device, it asserts dma_req signal to the AHB Master to request DMA transaction in the direction indicated by the dma_read signal (0 - Port to AHB, 1- AHB to Port). Transaction size is indicated by the dma_count signal. Since all configured Ports operate independently, several Ports may assert dma_req at the same time.
2. The DMA Arbiter arbitrates between multiple active Ports using round-robin equal-priority scheme, selects a Port, and provides the request to the AHB Master. The AHB Master then latches starting address and transaction size. Arbitration is done on a per-transaction basis, that is, the Port "owns" the bus until all data of the given transaction size has been transferred or the transfer is terminated due to error.
3. AHB Master asserts either dma_rxpop signal to request data from the Port during AHB bus write transfer, or dma_txpush signal along with valid data to the Port during AHB bus read transfer.
4. AHB Master generates AHB transfer and tries to complete the data transfer either in one AHB burst (ideally), or it may break the transaction into multiple burst transfers, for example, due to the 1-KB address boundary crossing, or early burst termination condition.
5. AHB Master completes the AHB transfers when all the data of the requested transaction size has been successfully transferred, which allows the DMA Arbiter to

switch to the next active Port. The following figure shows various BIU Master accesses.

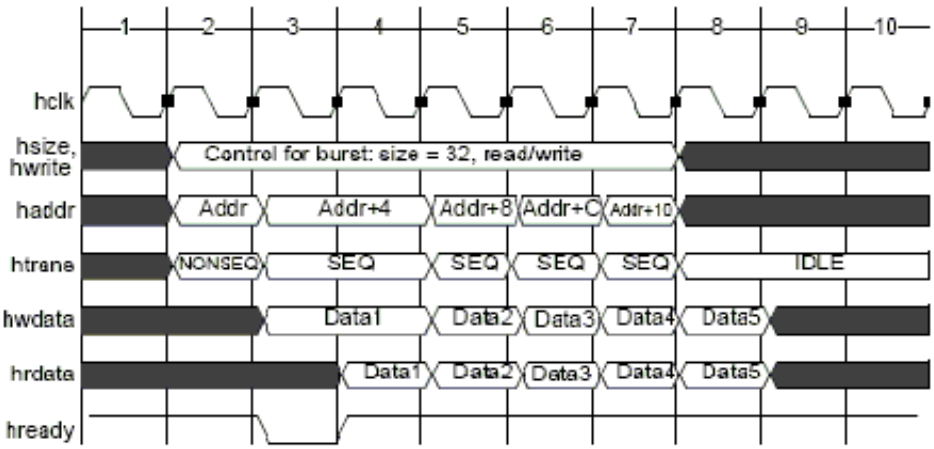


Figure 63-7. AHB Burst Transfer

63.3.2.4 DMA Arbiter

The DMA Arbiter simply monitors individual Port DMA requests and presents them to the AHB Master based on a round robin priority scheme.

Requests of larger burst length inherently gain more AHB bus priority than requests of shorter burst length. Software can use this approach to give Ports different priority.

63.3.3 Generic Registers (GCSR)

This module implements all SATA global registers and provides the following functions:

- Generic configuration and control;
- Global interrupt support;
- BIST operation (implementation-specific registers).

Refer to "Register Descriptions" for details on register operation.

63.3.4 Port

The Port instantiates the following modules:

- Port DMA

- Port Registers
- Transport Layer
- Link Layer
- Port Power Control Module

63.3.4.1 Port DMA

The Port DMA (PDMA) module implements the following functions:

- Monitors commands posted by system software using SATA_P 0CI register. When any of the command slots becomes active, PDMA downloads the corresponding Register FIS from the Command List structure and passes it to the Transport Layer TxFIFO for transmission to the Device.
- Controls data transfer between the Transport Layer FIFOs and system memory using Physical Region Descriptor Tables (PRDT).
  - During Data FIS reception, PDMA requests AHB write transfer of SATA_P 0DMACR[RXTS] size from the BIU Master when RxFIFO contains data of at least this size.
  - During Data FIS transmission, PDMA requests AHB read transfer of SATA_P 0DMACR[TXTS] size from the BIU Master when TxFIFO contains space of at least this size.
- Transfers non-Data FISes received from the device to system memory using Received FIS Structure.

Most of the communication between the PDMA and software is done using two system memory descriptors that are constructed by software prior to initiating the transfer: FIS descriptor, which contains FISes received from the device, and the other is the Command List, which contains a list of 1 to 32 commands available for the Port to execute and the pointers for data transfers.

Some additional communication is done via registers located in the GCSR and PCSR modules.

System memory structures are described in the SATA AHCI specification and are not repeated in this document for the sake of brevity.

The PDMA module operates in the application clock (hclk) domain and has 32-bit-wide data path. It supports 32-bit addressing.

The following figure shows a high-level state diagram of the Port DMA operation. Refer to the SATA AHCI specification for a more detailed Port state diagram.

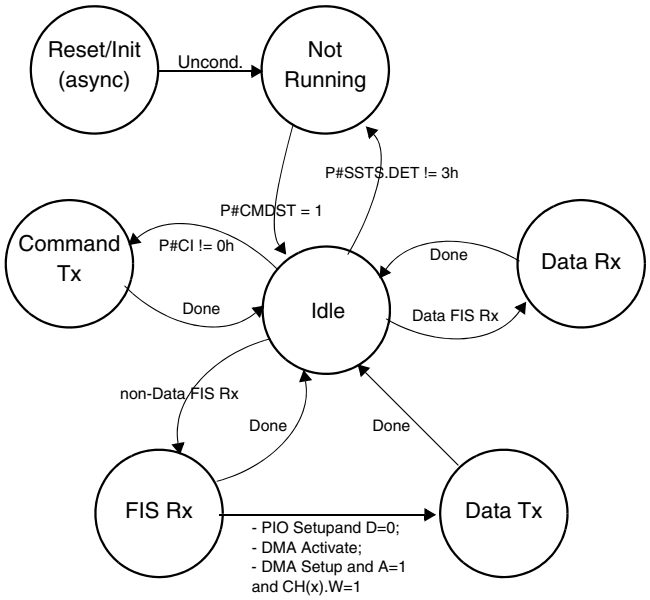


Figure 63-8. Port DMA State Diagram

63.3.4.2 Port Registers

The Port registers (PCSR) module implements all Port-specific registers:

- Command List and FIS Base address
- Interrupt Status/ Enable
- Port Command/ Status
- Task File Data/ Signature/ Serial ATA
- DMA status/control (implementation-specific)
- PHY status/control (implementation-specific)

Refer to "Register Descriptions" for details on register operation.

63.3.4.3 Transport Layer

The Transport Layer constructs Frame Information Structures (FIS) for transmission and decomposes received FISes.

The following list describes how an FIS is constructed:

- Receives FIS content from the Port DMA or Port Register modules;
- Notifies the Link Layer of required frame transmission and passes FIS to Link;
- Manages TxFIFO flow, notifies Link of required flow control via TxFIFO flags;

- Receives frame receipt acknowledge from Link;
- Reports good transmission or errors to requesting higher layer.

The following list describes how an FIS is decomposed:

- Receives the FIS from the Link Layer;
- Determines FIS type and checks for PHY/Link/Transport errors;
- Provides good/bad FIS acknowledge to the Link Layer;
- Distributes the FIS content to the locations indicated by the FIS type;
- Reports good reception or errors to the Port DMA/Registers.

The Transport Layer functional block diagram is shown in the figure below. The Transport Layer consists of the following five main modules:

- Receive FIFO (RxFIFO)
- Transmit FIFO (TxFIFO)
- Transport Check module (TCHK)
- Transport State Machine module (TSM)
- Synchronization module (APP_ASIC)

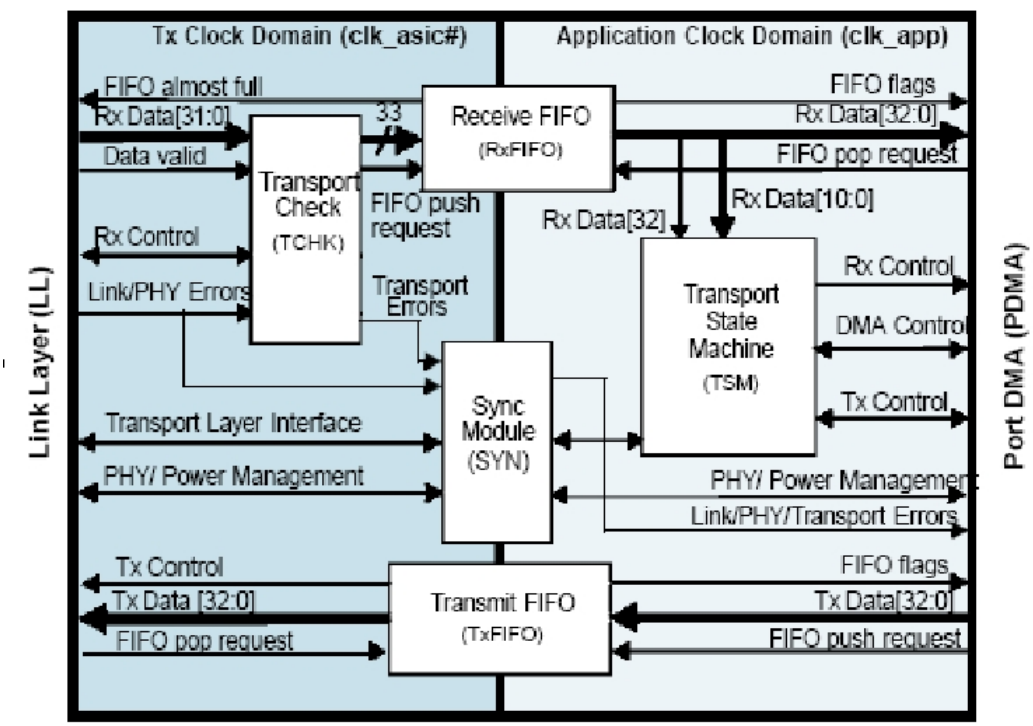


Figure 63-9. Transport Layer Functional Block Diagram

The Transport Layer operates in two clock domains: transmit and application. Transmit clock is generated in the PHY and depends on the Link Layer data path width (valid frequency values are: 37.5 MHz, 75 MHz,150 MHz, and 300 MHz). The application clock is sourced from the system bus and depends on the software. Both transmit and receive data paths are 32-bits wide.

The Transport Layer block provides FIS reception and transmission functions of the SATA Transport Layer. During reception the Transport Layer receives a new FIS from the Link Layer through the RxFIFO, decodes the FIS type, and instructs the PDMA to route the FIS payload data to the appropriate location in system memory. During transmission the Transport Layer instructs the PDMA to construct the appropriate FIS, and then passes it to the Link Layer through the TxFIFO. The Transport Layer block receives all the PHY/Link errors from the Link Layer, detects Transport errors, and passes them to the PCSR for setting the corresponding error bits.

The Transport Layer processes one FIS at time on the transmit side, meaning only one FIS is allowed in the TxFIFO at a time. On the receive side, RxFIFO can potentially contain more than one FIS at a time. For example, when the device transmits several DMA Data FISs back-to-back with minimal delay, RxFIFO might still have the previous Data FIS while the next FIS is being received.

63.3.4.3.1 Transport Layer FIS Reception

The FIS reception process is described as follows:

- The Link Layer starts frame reception and passes FIS content to the Transport Layer THCK. RxFIFO "almost full" flag notifies the Link Layer to send HOLDp to the device to prevent RxFIFO overflow. Upon detecting EOFp, the Link Layer asserts an "End status" signal to indicate the end of the FIS. All Link Layer/PHY errors are valid at this time.
- THCK module checks for Transport Layer protocol errors, passes FIS data to the RxFIFO, then appends "End Status" DWORD at the end with all the Link/PHY and Transport errors.
- TSM module receives the FIS from the RxFIFO and passes it to the PDMA/PCSR. When any of the Link/PHY/Transport errors is detected, then the FIS is either ignored (when non-Data FIS) or the transfer is aborted (when Data FIS) and the corresponding bits are set in the SATA_P 0SERR register.

63.3.4.3.2 Transport Layer FIS Transmission

The FIS transmission process is described as follows:

- The PDMA detects a request from the system software and notifies the TSM to enter a transmit state. The DMA data transmission is activated by the TSM after it receives DMA Activate FIS from the device.
- The PDMA receives the appropriate FIS from BIU Master and pushes it into the TxFIFO. The following FIS types are supported:
  - Register FIS - Control or Command type.
  - Data FIS - PIO or DMA type.
  - BIST Activate FIS
- The Link Layer uses negation of the TxFIFO "empty" flag to generate SOFp and begin frame transmission. Bit 32 of the TxFIFO is used to indicate the FIS "last DWORD" to the Link Layer. When the Link Layer sees this bit valid, it closes the frame with CRC and EOFp.
- The TSM waits for either positive or negative frame transmission acknowledgement from the Link Layer (Link Layer "handshake" error). Both of these conditions are passed from Link to TSM in the "End Status" DWORD. Negative acknowledgement is generated when the device detects an error during the frame reception and signals it to the host Link Layer. In this case any non-data FIS is resent to the device using Transport Layer retry logic. When the error is detected during Data FIS transmission, then this transfer is aborted and the FIS is not resent.

NOTE

When neither positive nor negative acknowledgement is received from the Link Layer following frame transmission, host s/w times-out and resets the interface.

63.3.4.3.3 Error Handling

All SATA errors are summarized in SATA_POSERRPort0 Serial ATA Error Register. The Link Layer accumulates all Link Layer/PHY errors during frame reception and presents them to the Transport Layer TCHK module with "End Status" signal asserted when it detects EOFp. "PHY Not READY" and "Link Illegal Transition/Sequence" errors terminate the current frame reception/transmission in progress and cause PHY/Link Layer reset and "End Status" assertion. During frame transmission, Link Layer detects R_ERRp/R_OKp from the device and passes this condition in the "End Status" DWORD.

The TCHK module checks for Transport Layer errors in the received FIS when no PHY/Link Layer errors were detected. All errors from the PHY, Link Layer, and Transport Layer TCHK module are passed to the Transport Layer TSM module by way of RxFIFO in the "End status" DWORD (with bit 32 set).

Some errors such as "Non-recovered/Recovered Data Integrity" are detected in the Transport Layer TSM module. "Internal Host Adapter" error is detected in the BIU.

All PHY, link, and transport errors are also passed to the PCSR module through the APP_ASIC module directly for setting the appropriate bits in the SATA_P 0SERR register. This is done to insure that error bits are reliably set in various error conditions. For example, when Data FIS is corrupted, this may result in potential DMA lock-up unless error that caused this condition is passed to the PCSR module and software can act on it.

PHY internal errors are filtered out outside the FIS. Errors related only to the current receive FIS cause SATA_P 0SERR[DIAG_I] bit to be set. To enable errors inside the FIS or outside the FIS to be reflected in the SATA_P 0SERR register, software must set SATA_BISTCR[ERREN] bit for the corresponding Port selected by the SATA_TESTR[PSEL] field.

63.3.4.3.4 Receive/Transmit FIFO (Rx/TxFIFO)

Both receive and transmit FIFOs are used as temporary FIS buffers and for clock domain crossing.

The RxFIFO width is 33 bits: 32 bits are used to transfer data and the 33rd bit is used to indicate the "End- Status" DWORD so the Transport Layer can detect the end of the previous FIS and the start of the next FIS in the situation when more than one FIS is in the RxFIFO.

The TxFIFO is 33-bits wide: 32 bits are used to transfer data and 33rd bit is used to indicate the "last" FIS DWORD to the Link Layer. Both FIFOs are reset on power-up either by the system bus reset signal, by software setting SControl register (P 0SCTL) DET field bit 0 (which results in the SATA block performing an interface initialization sequence COMRESET), or by the COMINIT condition.

NOTE

Both RX and TX FIFO RAM width is (M_HDATA_WIDTH + M_HDATA_WIDTH/32 + 1) bits.

You can select FIFO capacity in DWORDS for each Port separately, based on the system bus software requirements, such as number of masters, burst size, utilization, and so on.

The RX FIFO “almost full” level is set to the value indicated in the following table, to prevent an RX FIFO overflow regardless of the total Host and Device HOLD-HOLDA latency. When the RX FIFO “almost full” flag is asserted, the Link layer starts sending a HOLD status to the Device.

NOTE

Both RX and TX FIFO RAM width is (M_HDATA_WIDTH + M_HDATA_WIDTH/32 + 1) bits.

Table 63-2. RX FIFO "Almost Full" Level

RX FIFO Capacity (DWORDS)	M_HDATA_WIDTH (Bits)	"Almost Full" Level (DWORDs)
64	32 64 128	54 52 52
128	32 64 128 256	62 60 56 64
256	32 64 128 256	64
512	32 64 128 256	64
1024	32 64 128 256	64
2048	32 64 128 256	64

NOTE

It is very important that the total HOLDp/HOLDAp latency for both host and device does not exceed the number of RX FIFO almost full-level DWORDs as indicated in Table 63-2, otherwise, RX FIFO overflow (fatal error) occurs.

NOTE

Host latency must be calculated by adding the Link layer latency to the Host PHY latency. Some SATA configurations are close to 20 DWORDs leaving no margin for the PHY. It is recommended that such configurations be used only for FPGA validation/testing and not for actual product because exceeding 20 DWORD latency requirement may result in Device RX FIFO overflow.

63.3.4.4 Transport Check (TCHK)

The TCHK module provides the following functions:

- Detects new FIS reception by the Link Layer based on the received control signals.
- Decodes the FIS type located in the least-significant byte of the first DWORD and checks its validity. The following FIS types are supported:
  - Register FIS
  - Set Device Bits FIS
  - PIO Setup FIS
  - DMA Activate FIS
  - DMA Setup FIS
  - Data FIS
  - BIST Activate FIS
  - Unknown FIS (length is less than or equal to 64 bytes)

- Checks for all the Transport Layer errors (for example, unrecognized FIS, protocol, or transition).
- Detects an "End Status" signal assertion indicating the end of the current FIS from the Link Layer and passes all Link Layer/PHY/Transport Layer errors to the RxFIFO and to the PCSR module.
- The TCHK provides "Good FIS/Bad FIS" status acknowledgement to the Link Layer at the end of the received FIS.

The TCHK module receives 32-bit FIS DWORD data from the Link Layer and adds one bit (bit 32) before writing it to the RxFIFO. This bit indicates either FIS data, when cleared, or "End Status" DWORD, when set. The following Transport Layer errors are checked in the TCHK (assuming no errors were detected in the Link/PHY):

1. FIS length:
   - Non-data FIS according to the FIS type
   - Data FIS should be between 2 and 2049 DWORDs
   - Unknown FIS should be between 1 and 16 DWORDs
2. PIO Setup FIS transfer count - should be non-zero and even byte count and not exceed 8192 bytes
3. PIO Data FIS following the PIO Setup FIS with D=1 (PIO read) DWORD count - should match the transfer count
4. PIO read protocol FIS sequence - only Data FIS or end status when error are expected after the PIO Setup FIS with D=1, any other FIS would be negatively acknowledged to the Link Layer
5. DMA Setup FIS buffer offset - bits 0 and 1 should be cleared and transfer count should be an even (not zero) number
6. First Party DMA read protocol - DMA Setup FIS with D=1 is followed either by Data FIS or Set Device Bits FIS or end status when error
7. First Party DMA write protocol - DMA Setup FIS with D=0 is followed by DMA Activate FIS (when A=0) or Set Device Bits FIS or end status (when A=1)
8. BIST Activate FIS is supported type only (see BIST Operation for details)
9. RxFIFO push error for Data FIS - detected when Link has valid data and RxFIFO is "full" (for example, device violates HOLD latency requirement)

The Transport Transition Error SATA_P 0SERR[DIAG_T] bit is set when errors 1-8 are detected. The Unknown FIS SATA_P 0SERR[DIAG_F] bit is set when the Unknown FIS length does not exceed 64 bytes. The Protocol Error SATA_P 0SERR[ERR_P] bit is set on detection of error 9.

63.3.4.4.1 Transport State Machine (TSM)

The TSM module provides the following functions:

- Implements the host Transport Layer state machine according to the SATA spec with the exception of the FIS checking and error handling functions.
- Decodes the FIS type by reading the least-significant-byte of the first DWORD of the FIS.
- Detects the "End status" DWORD and checks for any Link Layer/PHY/Transport Layer errors. When any of the errors is detected:
  - On a non-data FIS, the received FIS is discarded, the transmitted FIS is retried indefinitely, and the corresponding SATA_P 0SERR register ERR_I bit is set.
  - On a data FIS, it can be passed to the system memory before the final status is reflected in the SATA_P 0SERR register ERR_T bit.
- Generates/receives the appropriate control signals to/from the PDMA based on the received FIS and its state.
- Handles transfer termination requests originated from the Link Layer or PDMA module.

63.3.4.4.2 Sync Module (APP_ASIC)

This module is used to synchronize several control signals between the Link Layer and the Transport Layer clock domains.

63.3.4.5 Link Layer

The Link Layer performs the following functions:

- Controls initialization between the Link Layer, PHY, and a connected device
- Generates and detects OOB signaling
- Transmits and receives frames
- Transmits primitives based on control signals from the Transport Layer and PHY
- Receives primitives from the PHY layer that are used to control the Transport and Link Layers
- Controls power management via the Port Power Control Module

Hot-plugging a device is digitally supported in the Link Layer when Tx OOB sequences are generated by the Link Layer via normal initialization polling. When Tx OOB sequences are generated in the PHY, it is up to the PHY to digitally support hot-plugging a device.

The Link Layer functional block diagram is shown in the following figure.

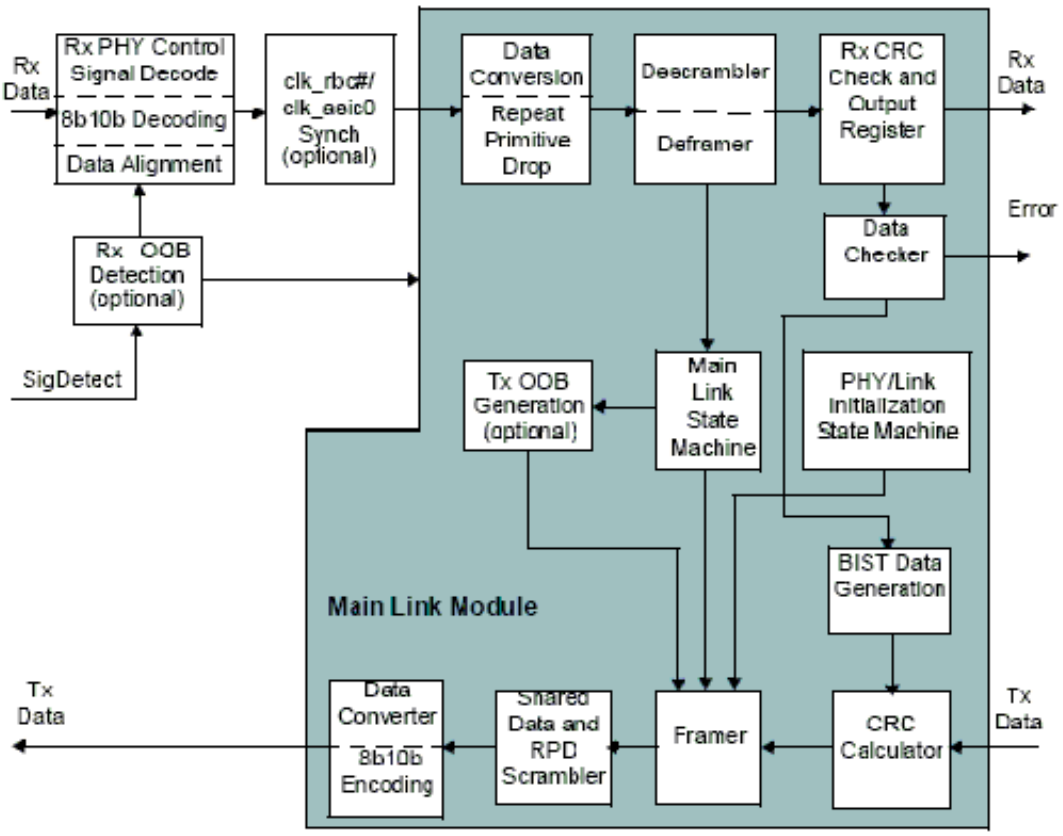


Figure 63-10. Link Layer Functional Block Diagram

On power-up, system reset or device hot-plug, the following sequence occurs:

1. The Link Layer transmits sequences of control data and ALIGN Primitives to the PHY. .
2. They are then forwarded to a device PHY as OOB signaling.
3. In addition, the Link Layer detects OOB sequences.

These OOB sequences bring the host controller, PHY, and device to an initialized condition. Once this occurs:

1. The Link Layer passes a PHY Ready status to the Transport Layer and normal communication begins.
2. The Link Layer receives requests from the Transport Layer to transmit data, in the form of a Frame Information Structure (FIS) comprised of DWORDs, to a device via the local PHY.
3. The Link Layer in turn transmits the FIS by inserting Primitives, scrambling and optionally encoding the data, sending it to the PHY and waiting for status.

4. When a status FIS is received, the Link Layer optionally decodes, aligns and descrambles the data, removes Primitives and forwards the data to the Transport Layer.
5. The Link Layer then notifies the Transport Layer of the ending transfer status. The Link Layer has no notion of the FIS content, other than its beginning and end points and CRC.
6. Data alignment is performed on received FIS data via ALIGN Primitives. Flow control is also achieved on FIS going in either direction via HOLD Primitives.
7. In addition, the Link Layer receives requests from the Transport and PHY Layers to go into and out of power management modes.

Power management is achieved by notifying the PHY of a partial or slumber condition and then disabling normal data transmission on PHY Rx and Tx interfaces until a wake-up request from Transport Layer or remote device via the PHY is seen from the Power Control Module. Power management is controlled via Partial and Slumber requests as described in the SATA specifications.

The Initialization State Machine controls the Link Layer, PHY and device system initialization. The main Link Layer State Machine controls FIS traffic, flow control, and error detection and status reporting. FIS traffic is generated and disassembled via Framer and Deframer modules. The Link Layer also performs CRC calculations on FIS, as well as scrambling and optionally encoding the data.

- Decoding of received FIS is performed in the clk_rbc0 clock domain due to the fact that the incoming FIS is on an asynchronous, but frequency locked clock of the same rate as the clk_asic0 clock domain.
- 8b/10b encoding and decoding are performed in the Link Layer.

The Link Layer receives data on either clk_rbc0, recovered from the incoming data stream by the PHY, or on clk_asic0. This single receive clock is then used in this module to decode data and control signals from the PHY and pass it to the rest of the Link Layer. Data is passed through a synchronizing Datastream FIFO. ALIGN Primitives are also detected and dropped in the front end of the receiver as a means of guaranteeing no Datastream FIFO overruns, when a Datastream FIFO is included. ALIGN Primitives are also used to synchronize to the data stream in the PHY by triggering data realignment where necessary.

Finally, ALIGNs are required by the Tx OOB initialization state machine to complete initialization, following the SATA specifications. For this reason, the PHY must indicate the presence of at least two ALIGNs after the Link Layer detects the release of COMWAKE. Otherwise the Link Layer is not able to complete initialization and begin normal operation. This is required regardless whether the PHY drops ALIGNs at any other time.

NOTE

Even if the PHY drops ALIGNs, data indicating the comma character must be present on phy_rx_data, in the corresponding phy_comma_det slot. This is required to invalidate comma characters before they are stable.

This Databook attempts to avoid redundancy with the *High Speed Serialized ATA* specification. Please refer to the latter for standard specifications and descriptions. This discussion pertains to specific implementation-related operation and details.

63.3.4.5.1 Link Layer Features

The SATA block Link Layer features are as follows:

- Rx Data Buffer for recovered clock systems
- OOB signaling and system Initialization
- Frame negotiation and arbitration
- Envelope framing/deframing
- CRC calculating, insertion and checking
- 8b/10b encoding/decoding
- Flow control
- Frame acknowledgement and status reporting
- Data width conversions
- Data scrambling/descrambling for EMI reduction
- Repeat Primitive data transmission and reception handling
- ALIGN Primitive detection, dropping and data alignment
- Power management support

63.3.4.5.2 User-Defined Status and Control

There are up to 32 bits each of optional user-defined status and control Ports available in the SATA block. These are used to obtain information from the PHY, and control PHY functions that might differ across particular PHYs and which are not defined in the SATA specifications.

Each of these variable bit width, user defined Ports map to a register that can be read and/or written by the system via the Bus Interface. When no user-defined Ports are included in the design, they are absent from the netlist, along with the registers that would have been allocated for their use. See Figure 63-8 for the top-level I/O diagram.

NOTE

There are clock crossing restrictions on these Ports. Refer to "Signal Descriptions".

63.3.4.5.3 PHY Initialization Details

Please refer to the "Out of band signaling" section of the SATA specifications for the following descriptions.

The PHY/device signaling from SATA block is dependent on whether the Link Layer generates and detects the proper OOB sequences or whether the PHY generates and detects them.

NOTE

For PHY Initialization and general operation related to the Link Layer, all SATA block inputs and outputs are sampled and generated, respectively, in the following clock domains. Note that the arrangement of signals facilitates there being no clk_rbc present until the clock is recovered from incoming data.

- Rx Data Inputs
  - clk_asic
  - clk_rbc
    - phy_rx_err
    - phy_comma_det
    - phy_rx_data_vld
    - phy_rx_data
    - phy_rx_kch_det
    - phy_rx_dec_err
    - phy_rx_disp_err
- Tx Data Outputs
  - clk_asic
  - phy_tx_data_vld
  - phy_tx_data
- Control Outputs
  - clk_asic
    - phy_rx_enable
    - phy_tx_enable
    - phy_spdsel
    - phy_nearafelb
    - phy_farafelb
    - phy_reset
- Control Outputs
  - clk_pmalive
    - phy_partial
    - phy_slumber

- Control Inputs - Fixed, multiple and/or configuration-dependent clock domains
  - phy_sig_det - Sampled in all of these clock domains
    - clk_rbc
    - clk_rxoob
    - clk_pmalive
    - clk_asic
- phy_calibrated Sampled in all of these clock domains
  - clk_rbc
  - clk_asic
- phy_spdmode
  - clk_asic

NOTE

phy_rx_data_vld should not be delayed after OOB initialization (held inactive LOW), while phy_comma_det is indicating COMMA characters are being received, or normal operation may not begin. Either phy_comma_det must be ANDed with phy_rx_data_vld, or all phy_rx_data_vld bits must be tied permanently HIGH.

However, in order to tie phy_rx_data_vld permanently HIGH, a phy_sig_det must be present and indicate valid signal detection. For example, phy_sig_det cannot be tied HIGH when phy_rx_data_vld is also tied HIGH. Normal operation can fail due to missing the first non-ALIGN Primitives when they are in the form of Repeat Primitive Data, before phy_rx_data_vld becomes active. A PHY that always drops all ALIGN data cannot delay phy_rx_data_vld at all after initialization. Finally, phy_rx_data_vld and phy_comm_det should not become high until the PHY has completely locked onto ALIGNs and data is error free, to avoid problems transitioning between OOB and normal operation.

63.3.4.5.3.1 Link Layer Tx OOB Initialization Sequence Details

NOTE

In order for the Link Layer to generate the Tx OOB initialization sequences, at least two ALIGN Primitives must be indicated and flagged to the Link Layer by the PHY via the phy_comma_det signal. This must occur after the release of COMWAKE, per the SATA initialization specifications. Otherwise the Link layer is not able to complete initialization

and begin normal operation. This is required regardless of whether the PHY drops ALIGNs at any other time. In addition, if data can ever become misaligned from the PHY, ALIGN Primitives are required in order for the Link Layer to realign the data. Finally, ALIGNs must be returned to the SATA block for all BIST modes to function properly.

The Link Layer Tx OOB initialization sequence is depicted in the figure below. Not all details are shown, but the following sequence summarizes the flow:

1. PHY Reset state is entered during a system asynchronous-reset or if a Port reset (COMRESET) has been detected. This causes the phy_reset output signal to become active and can be used to synchronously clear the PHY of any errors, when the PHY has that capability. The PHY reset is active for 12 1.5 Gb/s rate DWORDs of time, or 320 ns.

NOTE

clk_asic0 cannot be removed while phy_reset is asserted, because clk_asic0 is required to negate phy_reset.

Signal phy_reset is intended only to clear non SATA errors in this configuration, and must not be used to asynchronously reset the PHY.

2. Upon exiting PHY Reset state, the PHY speed is always set to 1.5 Gb/s speed, regardless of the speed being negotiated for.
3. After a PHY reset, and anytime an unsolicited COMINIT has been detected, the PHY Wait state is entered in order to check and wait for PHY calibration.
4. After the PHY Wait state, communication with a device is attempted by transmitting a COMRESET sequence, followed by waiting for a COMINIT sequence from the device. This takes place in a polling routine. During the Wait Cominit state, when a COMINIT has not been detected in 14ms, the state machine loops back and restarts from the PHY Wait state indefinitely.
5. Once a COMINIT has been detected, normal OOB sequences then take place, interrupted only by an unsolicited COMINIT; a COMINIT that was not expected.
6. After completion of the OOB sequence, when 6.0 Gb/s or 3.0 Gb/s are enabled and have not already been attempted and failed, phy_spd_sel is asserted to the PHY to change clk_asic0 rate to the highest supported speed. The host then begins waiting for ALIGNs.
7. At the beginning of awaiting ALIGNs, the speed is changed back to the highest speed to be negotiated. Otherwise the speed stays at Gen1. During the Await ALIGN state, when an ALIGN has not been detected by the time 32K 1.5 Gb/s rate DWORDS of D10.2 characters have been transmitted, the Link Layer moves to the

PHY Wait state. In addition, when a higher speed fails negotiation, an internal flag is set to force the host to stay at the current speed after the next OOB sequence has completed, unless one of the following occurs:

- Asynchronous system reset
- Port reset
- An unsolicited COMINIT has been detected from the device
- 6.0 and 3.0 speed disabled and re-enabled (programmed to 1.5 Gb/s speed, then back to 3.0 Gb/s or 6.0 Gb/s, followed by a host COMRESET)

8. When an ALIGN has been detected before 32K 1.5 Gb/s rate DWORDS of D10.2 have been transmitted, the Link Layer moves to the Send ALIGN state, followed by the Init Complete state, once three non-ALIGN Primitives have been detected. Note that while the Link Layer is generating D10.2 characters at 3.0 Gb/s or 6.0 Gb/s, 1.5 Gb/s rate D10.2 characters are emulated by doubling or quadrupling the transmitted data, i.e., 1100110011, as opposed to 1010101010.
9. Once initialization is complete, the Ready state is entered and the Main Link state machine takes control.
10. Finally, the Partial and Slumber states are only entered after a Power Mode has been set. This disables the Tx interface until either the host or device requests a wake-up. For further information, refer to Power Management Operations.

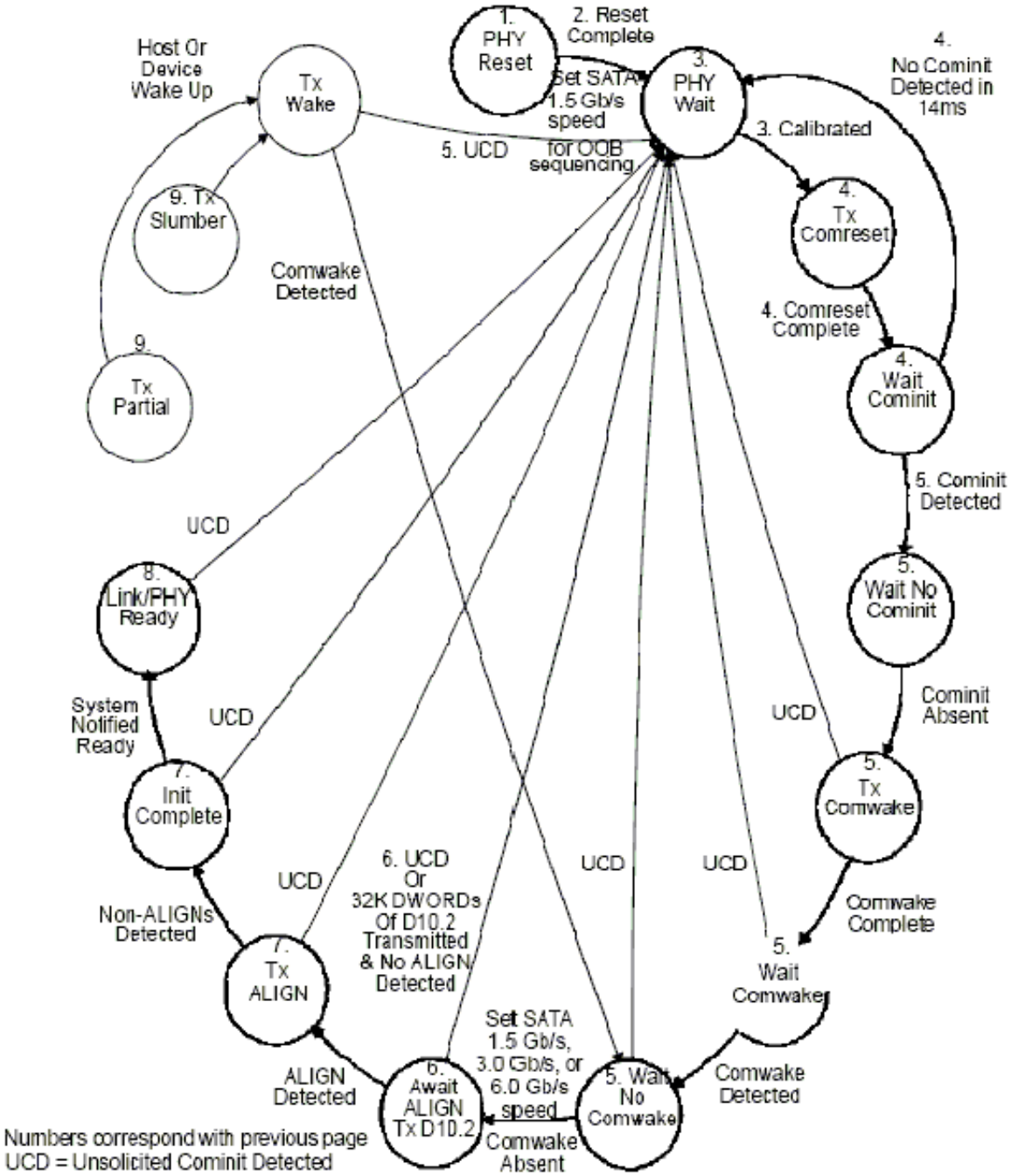


Figure 63-11. Tx OOB Initialization Sequence

63.3.4.5.3.2 Link Layer Tx OOB Sequence Generation

When the Link Layer generates the Tx OOB sequences , the Tx interface connections are different than when the PHY generates them.

Figure 63-11 depicts a small portion of an initialization phase, which includes the first part of a COMRESET condition, followed by a portion of a COMWAKE. The actual OOB sequences are created by sending ALIGN Primitives in conjunction with de-assertion of the phy_tx_enable signal. This causes the Tx OOB data and NULL sequences required by the SATA PHY specifications. Note that this diagram is not the complete initialization sequence. In a complete initialization sequence, there are multiple bursts of COMRESTs and COMWAKEs, followed by a d10.2 stream, followed by an ALIGN stream and then finally normal data. The phy_tx_data_vld is only used for qualifying data in systems that require it, but is unrelated to the OOB sequencing itself.

NOTE

Even when TX OOB signaling is generated in the PHY, the Link Layer still needs to see two ALIGNs, followed by three non-ALIGN primitives to transition from initialization to normal operation.

This feature allows the Link Layer to transition to normal operation without requiring the two ALIGNs. However, this also requires all data from the PHY to be properly aligned and to stay that way. The Link layer does still require three non-ALIGN Primitives, so they must be passed to the core before the incoming data turns into CONT 'Repeat Primitive Data'. This feature does not work when ALIGN_MODE = Misaligned. For more details, see BISTCR.QPHYINIT (bit 14) on page 162.

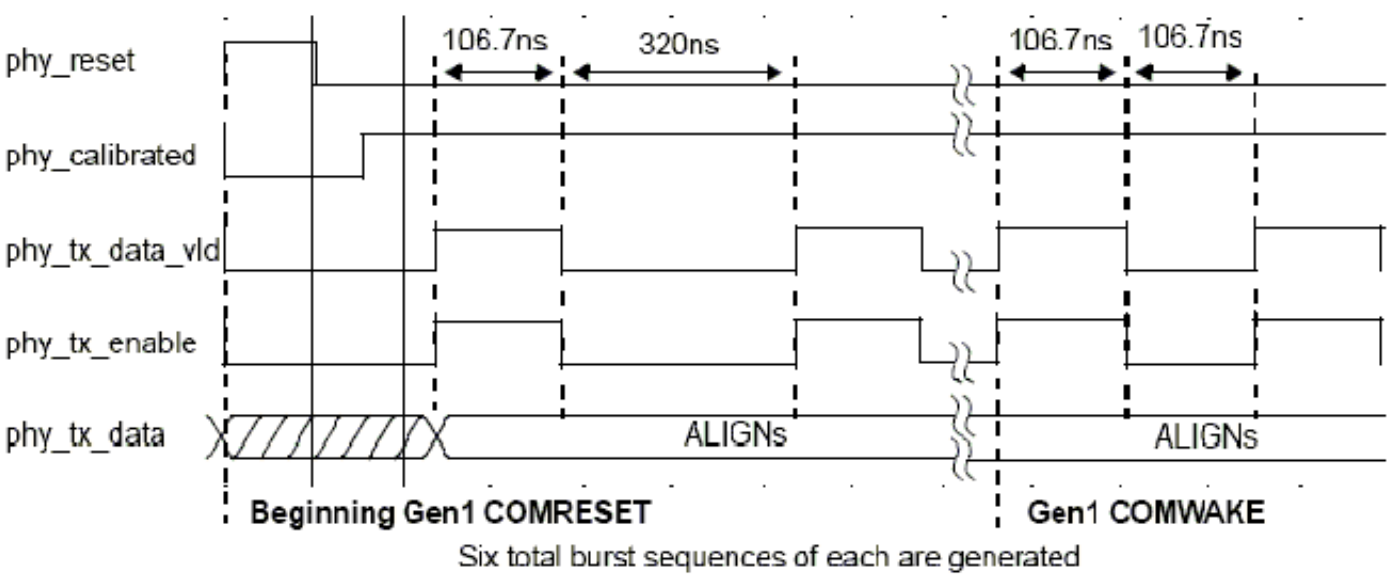


Figure 63-12. Link Layer Generated Tx OOB Signaling

63.3.4.5.3.3 Link Layer Rx OOB Sequence Detection

When the Link Layer detects the Rx OOB sequences, the Rx interface connections are different than when the PHY detects them.

Figure 63-12 depicts a small portion of an initialization phase, which includes the first part of a COMINIT condition, followed by a portion of a COMWAKE.

The actual OOB sequences are comprised of specifically timed ALIGN Primitives in combination with negation of the phy_sig_det signal (indicating NULL on the differential pair). However, only the phy_sig_det signal is used for qualifying Rx OOB sequences by the Link Layer.

NOTE

The PHY must “condition” (filter and delay) phy_sig_det so that the OOB detector accurately detects SATA sequences and prevents the Link Layer from accidentally misreading the Device as disconnected, thus avoiding reset. A filter is required to prevent phy_sig_det from detecting transient ‘loss of signal’ due to bits on the Rxp/Rxn wires crossing the zero voltage level.

To ensure that power modes function properly, phy_sig_det should be timed to the data delay through the PHY so that it does not go low until at least four PMACKs are passed completely to the core. This delay must also account for the existence of an ‘elasticity buffer’ located in the PHY. Both the filtering and the delay must be ‘symmetric’ (phy_sig_det rise filter/delay equal to the fall filter/delay) so that errors are not introduced in OOB signal detection or impact OOB compliance.

Figure 63-12 does not show the complete initialization sequence. In a complete initialization sequence, the host receives multiple bursts of COMINITs and COMWAKEs, followed by an ALIGN stream, and then finally normal data.

The Link Layer qualifies valid COMINIT and COMWAKE sequences according to the SATA specifications, using calculated comparator values that always fall within specification-compliant ranges. The SATA- specified valid ranges of sequence spacing are shown in Figure 63-12. The actual COMINIT and COMWAKE sequences are calculated and qualified by the Link Layer as described in the following subsections. See the application note and "Example Calculated COMWAKE, COMINIT and Data Burst Lengths" for important information.

To accurately qualify OOB spacing inside the compliant range required by the SATA specifications, the OOB sampling clock, clk_rxoob, must be no lower than 60 MHz.

The phy_rx_data_vld input is used only to qualify data in systems that require it, but is unrelated to the OOB sequencing itself. Systems with this configuration that do not use phy_rx_data_vld (data valid every clock cycle), should tie the bits HIGH. This does not affect OOB detection, because data is qualified with phy_sig_det for this purpose.

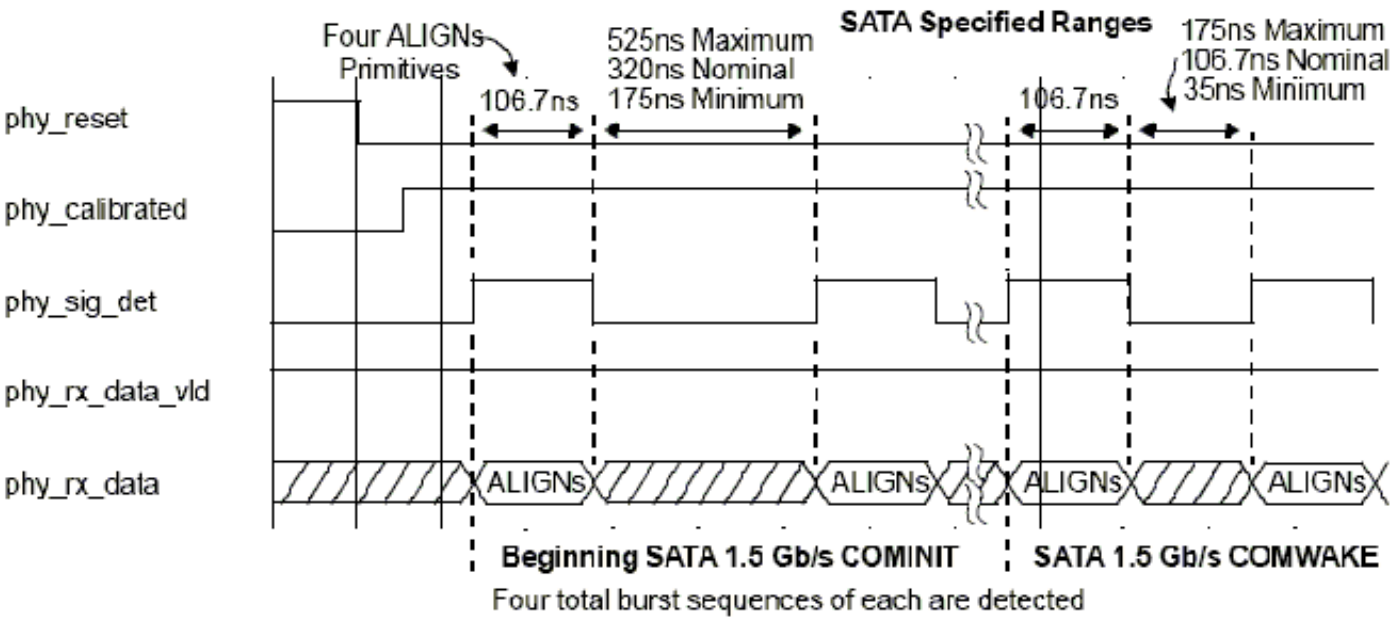


Figure 63-13. Link Layer Rx OOB Detection

63.3.4.5.4 Link Layer Power Management Details

Power management OOB detection and generation sequences occur in the same signaling format as described in PHY Initialization Details. However, they follow the required sequencing specified in the SATA Power management specifications.

In the SATA block, all data on the Tx channel is disabled until the host or device requests a wake-up condition. In addition, Rx data is not allowed to propagate in the SATA block Rx data path until one of these conditions has been detected and the PHY and device have been fully initialized per requirements.

A power down and wake up is depicted in Figure 63-14 and Figure 63-15, excluding actual OOB signaling, which are dependent on OOB Modes. Note that 'OOB' on Tx and Rx Data indicates that OOB sequences are present.

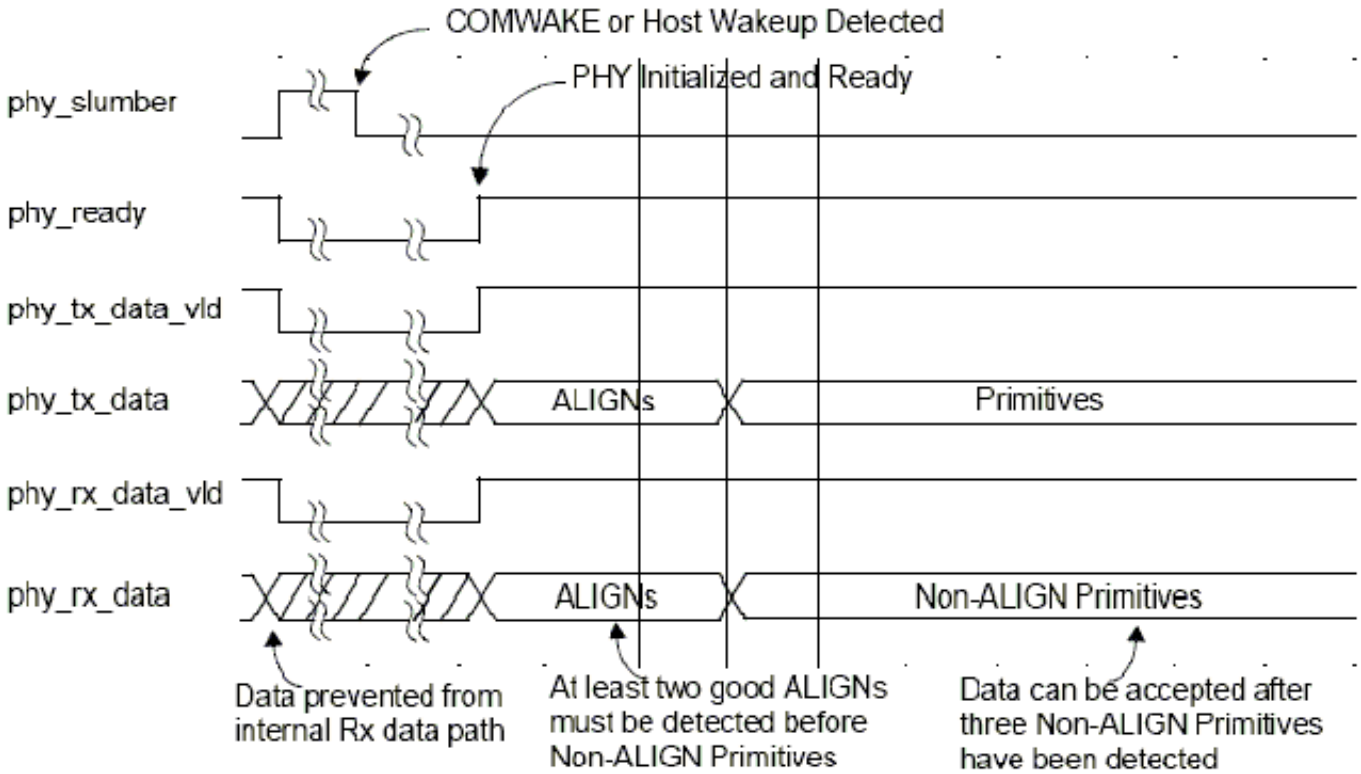


Figure 63-14. Power Mode Example: Rx and Tx OOB In PHY

Seperator Text

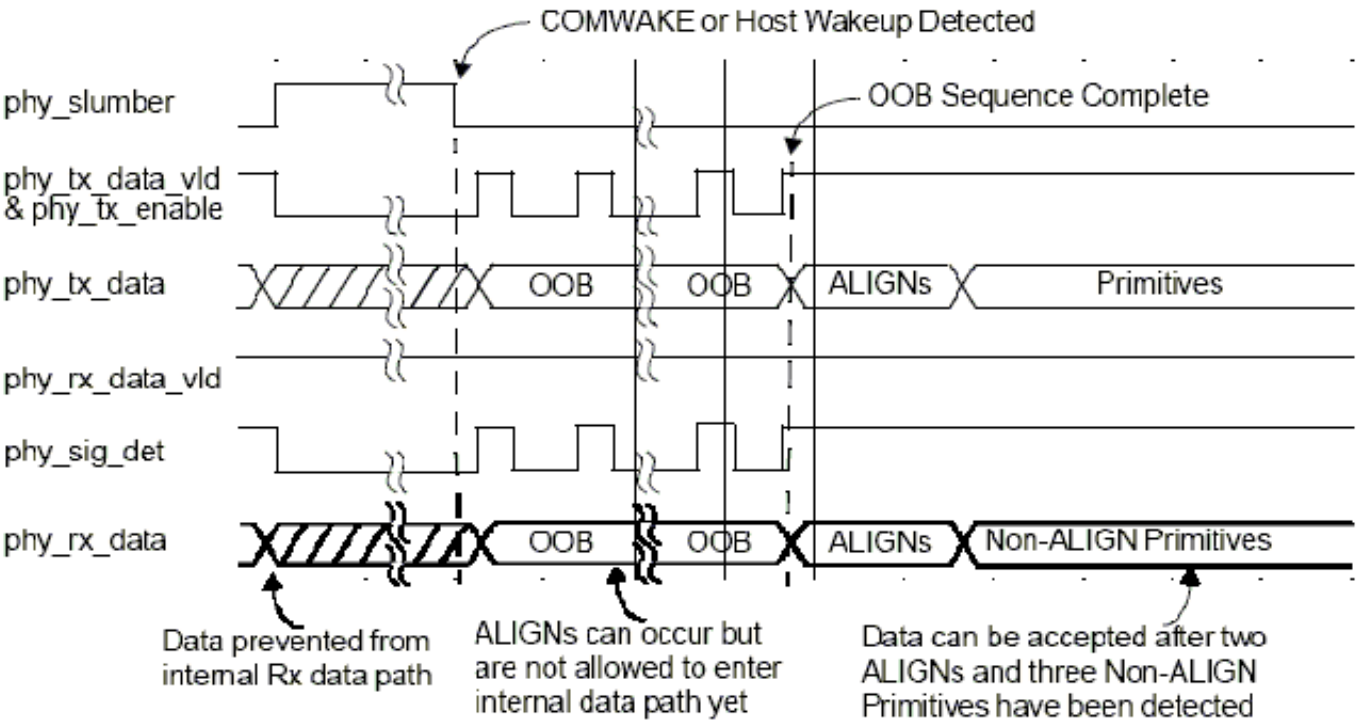


Figure 63-15. Power Mode Example: Rx and Tx In Link

63.3.4.6 Port Power Control Module

The Port Power Control Module (PCM) implements the following functions:

- Monitors Transport, Link and PHY ready/not ready conditions, as well as Device and Host power requests.
- Systematically controls the Link and Transport Layer transitions into and out of offline conditions (system reset, COMRESET and power modes).
- Allows clk_asic0 and clk_rbc0 to be stopped during Slumber and Partial power modes.

The PCM main function is to allow disabling clk_asic0 and clk_rbc0 in SATA power down modes.

CAUTION

Clocks supplied to the core should never glitch at any time, including before, during, and after initialization and power modes. Clock glitch protection should be performed outside the core for any clocks that might glitch.

In addition, inputs to the core should remain static during the time external logic is removing external clock glitches.

For more information, refer to the SATA specifications related to power mode exit time requirements, or contact Synopsys support.

NOTE

If clk_asic0 or clk_rbc0 are stopped, Near End Analog Loopback mode is not supported when a device is connected to the system. Therefore, it is recommended to only stop clocks in Slumber mode, in order to support Near End Analog Loopback mode when a device is connected.

NOTE

In order to support Host-initiated power modes where clk_rbc and clk_asic are removed, the PMACK received from the Device must be able to make it through the clk_rbc clock domain, synchronization, and the Link Layer clk_asic domain Rx Data path to the Link state machine, before the clocks can be removed. The SATA specifications allow a Device to transmit 4 to 16 PMACKs before going into power down. While 16 PMACKs are enough to guarantee receipt by the Link state machine, 4 are not. In the cases where a Device does not send enough PMACKs, the clocks will need to be kept running long enough for the Link state machine to detect the PMACK, or the Host will not go completely into power down mode and a Host COMRESET would be required to exit the failed power mode.

Figure 63-16 shows a high-level state diagram of the Power Control Module. Some signals are omitted for clarity.

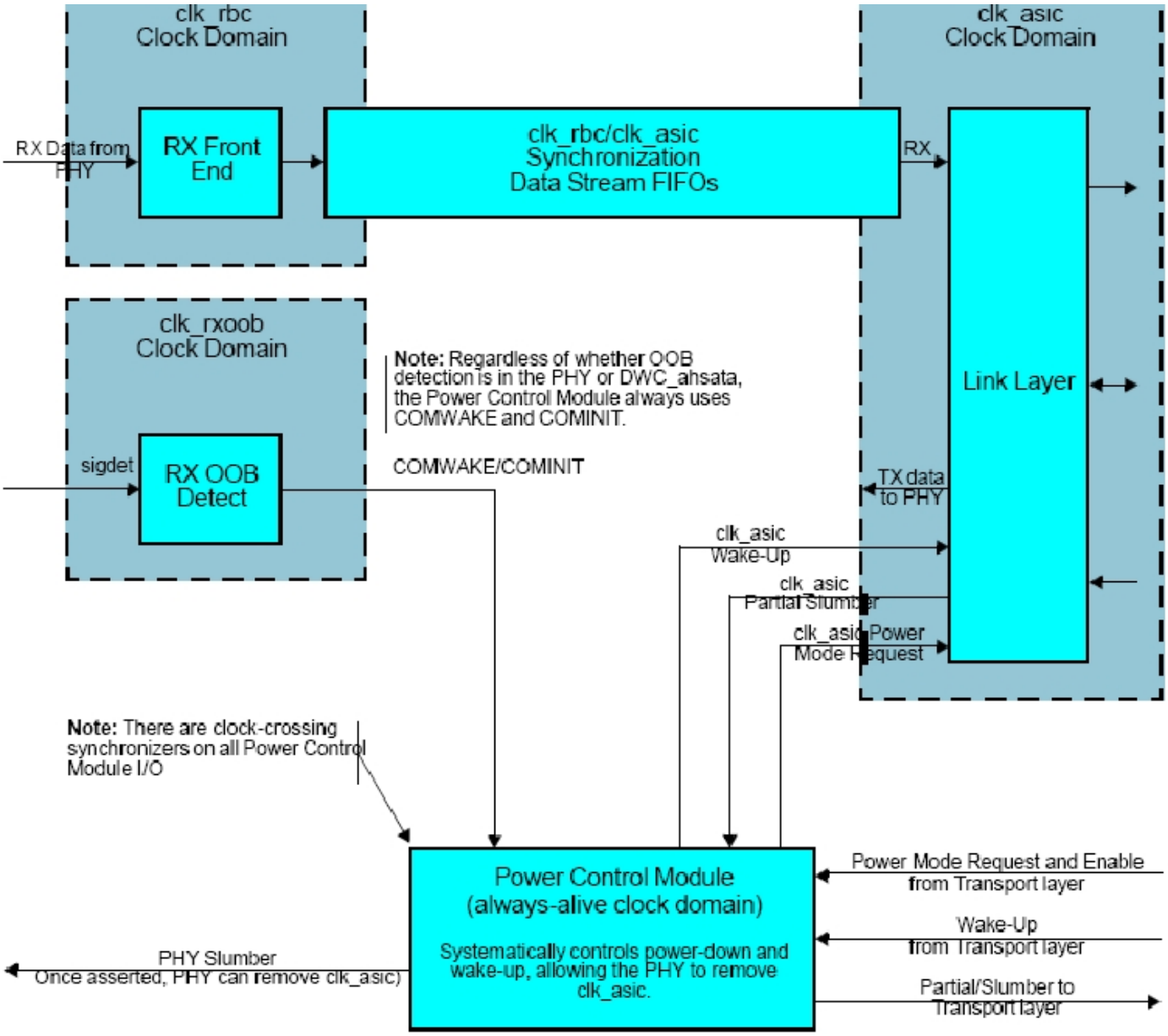


Figure 63-16. Port Power Control Module Diagram

The Power Control Module exists in the 'always alive' clk_pmalive clock domain. The clk_pmalive must always be present and must never change frequency. All signals into and out of the PCM are synchronized between the clk_pmalive, clk_asic0, clk_rbc0, and hclk clock domains with one or more of the synchronizers described in "Cross Clock Synchronization". The Power Control Module serves to assure all SATA block Layers and the PHY move correctly between inactive and active states in unison.

NOTE

Within the core there is no difference between going into and out of Partial and Slumber power modes, even if a system

disables clk_asic0 and clk_rbc0 in one mode, but not the other. Clocks do not have to be removed in either mode.

An example Slumber Power Mode Control with clocks stopped follows:

Example 1: Host Initiated Slumber Power Mode 1

1. System host requests Slumber Power Mode and asserts a Slumber request to the Power Control Module, which then forwards the request to the Link Layer.
2. If the Link Layer was in an IDLE state, then the Link Layer sends PMREQ_Sp primitives to the device. If the device responds with PMACKp primitives, the Link Layer asserts a Slumber signal back to the Power Control Module. If the Link Layer was not IDLE (or already in a power state), or the device denies the Slumber request, the Link Layer sends a power mode abort signal to the Power Control Module, which forwards it to the Transport Layer, and normal operation continues.
3. Provided the Link did not abort the Slumber request, and once the power control module receives sees the Slumber signal from the Link Layer, phy_slumber is asserted to the PHY and clk_asic and clk_rbc can be stopped.
4. At the same time, a slumber signal is sent to the Transport Layer; if no activity or wake up has been detected since the original request was made, the Transport Layer asserts an acknowledge signal back to the power control module. If activity has been detected in the Transport Layer, it responds with a wake up signal back to the power control module and the power mode is cancelled to the PHY. If the Transport Layer allowed the power mode to take place, it records the power down status in register SATA_P 0SSTS[IPM]. Note that the power mode to the PHY must be asserted if the Link Layer had accepted it, given the Tx Data will have been stopped at that point. Therefore, to avoid a power mode assertion to the PHY that is too short, the phy_slumber will be asserted for a minimum of 16 clk_pmalive clock cycles before the signal is asserted to the Transport Layer. This guarantees the power mode is not cancelled before a minimum of 16 cycles of assertion.
5. Once the PHY sees the Slumber signal, p0_phy_slumber, then clk_asic0 and clk_rbc0 can be stopped.

Example 2: Wake Up From Slumber Power Mode

1. The Slumber power mode is stopped by either detection of a COMRESET/ COMWAKE from the device, or software cancelling the power mode. When software has cancelled the power mode, an internal wake up signal is sent from the Transport Layer to the Power Control Module. Whenever the PCM detects the a host "wake up", or a COMRESET or COMWAKE, the Slumber signal to the PHY p0_phy_slumber, is immediately negated, which should result in the PHY restarting clk_asic0 and clk_rbc.

2. At the same time the Slumber is negated to the PHY, Slumber is negated to the Transport Layer and all power requests to the Link Layer are cancelled and normal operation resumes. Because the Link

Layer uses rising edge detection on power requests, and the Transport Layer uses rising edge detects on power mode assertion, unwanted power modes do not occur.

63.3.5 Operation Details

63.3.5.1 Data Transfer

The DMA engine for PORT0 is implemented in the PDMA module. Its operation is based on the AHCI specification Port State Machine.

The following sections outline examples of the ATA DMA read and write transfers.

NOTE

Optional prefetching of ATAPI commands, PRD entries or data, as indicated by the 'P'-bit in the Command Header, is not supported in the current release.

63.3.5.1.1 ATA DMA Read

This is the DMA read sequence:

1. Software finds a free command slot by reading the SATA_P 0CI register, then builds a DMA read command in the Command List for the Port to execute, and sets the bit corresponding to this command slot in the SATA_P 0CI register.
2. The PDMA fetches the Command Header from system memory.
3. The PDMA fetches the command Register FIS from system memory and transfers it to the device.
4. Since this was a DMA read command, the device responds with a number of Data FISes. When they arrive, PDMA performs the following operations:
   - Fetches the first PRD from system memory.
   - Transfers data from the RxFIFO to system memory until the byte count for this PRD is satisfied.
   - Continues to fetch PRDs and transfer data until the byte count for the command is satisfied.
5. Device sends D2H Register FIS with the command ending status and when the I-bit is set, interrupt is generated. D2H Register FIS is posted to the Received FIS memory structure.

6. When this is the last command, and the SATA block was enabled for aggressive power management, the PDMA requests the Link Layer to enter either Partial or Slumber state.

63.3.5.1.2 ATA DMA Write

This is the DMA write sequence:

1. Software finds a free command slot by reading the SATA_P 0CI register, then builds a DMA write command in the Command List for the Port to execute and sets the bit corresponding to this command slot in the SATA_P 0CI register.
2. The PDMA fetches the Command Header from system memory.
3. The PDMA fetches the command Register FIS from system memory and transfers it to the device.
4. Device responds with DMA Activate FIS. When it arrives, PDMA performs the following operations:
   - Fetches the first PRD from system memory;
   - Transfers data from system memory to TxFIFO until the PRD byte count is satisfied or 8-KB FIS boundary is reached. The Link layer sends Data FIS to the device. If more than one FIS is needed, device sends DMA Activate FIS for each Data FIS;
   - Continues to fetch PRDs and transfer data until the byte count for the command is satisfied.
5. Device sends D2H Register FIS with the command ending status and when the I-bit is set, interrupt is generated. D2H Register FIS is posted to the Received FIS memory structure.
6. When this is the last command, and the SATA block was enabled for aggressive power management, the PDMA requests the Link Layer to enter either Partial or Slumber state.

63.3.5.1.3 Native Queued Command (NCQ) Transfers

The SATA block supports NCQ feature (READ/WRITE FPDMA QUEUED commands). Data transfers are activated via the DMA Setup FIS, and command completion is performed via the Set Device Bits FIS.

NOTE

- Device must also support NCQ, otherwise it aborts READ/ WRITE FPDMA QUEUED commands. The non-zero buffer offset feature should be disabled in the device.
- ATA/ATAPI-7 queued feature set (legacy queueing) is not supported by the AHCI spec.

63.3.5.1.4 PIO Transfer

The SATA block supports multiple DRQ block PIO operation (SATA_CAP[PMD]=1).

From the SATA block point of view, PIO transfer looks like a DMA transfer: a command table is set up, and the data is transferred from or to system memory by the PDMA module.

63.3.5.1.5 Transfer Size

Normally, all SATA block AHB master transfers are done with 32-bit transfer size (m_hsize==3'b010) and 32-bit-aligned addresses.

When the transfer count for a command has odd number of 16-bit words (for example, 257 words, or 514 bytes), then on device reads the device would pad bits [31:16] of the last Data FIS dword with zeroes, and on device writes, SATA block would clear bits [31:16] of the last Data FIS dword sent to the device.

Odd-word transfer count implies creating a PRD with odd-word byte count and possibly 16-bit-aligned data address.

Setting PRD Data Base Address (DBA) or Byte Count (DBC) to a 16-bit-aligned value (for example, 2, 6, 10, etc.) results in 16-bit single transfers on AHB master interface (m_hsize==3'b001) regardless of the SATA_P 0DMACR[RXTS] or SATA_P 0DMACR[TXTS] values. This is due to the fact that SATA block operation is optimized for 32-bit operation when both DBA and DBC are 32-bit-aligned (for example, 4, 8, and so forth).

NOTE

To achieve maximum performance with odd-word transfer, it is recommended that the PRD with the odd-word byte count is the last and the smallest PRD of all the PRDs for the command, and the rest of the PRDs are all 32-bit-aligned. This assumes that the starting address of the data is 32-bit- aligned

63.3.5.2 Power Management Operations

The SATA block Port power management states (PARTIAL or SLUMBER) can be initiated by the software, the Port itself, or by the device.

The power state machine is implemented in the Link Layer power management module (refer to Link Layer Power Management Details for details). It asserts corresponding signal (p0_phy_partial or p0_phy_slumber) to the PHY to enter the power management state.

Software requests transition to either PARTIAL or SLUMBER state using the SATA_P 0CMD[ICC] field, however, the Port acts on it when the Link Layer is currently in the L_IDLE state, otherwise this request is ignored.

The device requests power management state by transmitting PMREQ_Pp or PMREQ_Sp primitives to the Port. Software can disable transition to power management states using the SATA_P 0SCTL[IPM field].

The SATA block supports aggressive power management states that allow the Port to initiate an interface power management state as soon as there are no commands outstanding to the device. The SATA_P 0CMD[ALPE] bit defines whether the feature is enabled and the SATA_P 0CMD[ASP] field controls whether PARTIAL or SLUMBER state is initiated by the Port when enabled. When SATA_P 0CMD[ALPE] is set, if the Port recognizes that there are no commands to process, the Port transitions to PARTIAL or SLUMBER state based upon the SATA_P 0CMD[ASP] field setting.

The Port recognizes no commands to transmit as either:

- SATA_P 0SACT is cleared to 0h, and the SATA_P 0CI is updated from a non-zero value to 0h.
- SATA_P 0CI is cleared to 0h, and a Set Device Bits FIS is received that updates SATA_P 0SACT from a non-zero value to 0h.

SATA block supports automatic PARTIAL to SLUMBER power state transition feature. This is enabled by the software setting the bit SATA_P 0CMD[APSTE]. When SATA_P 0CMD[APSTE] is set and the SATA block Link is in PARTIAL state, the core automatically transitions to SLUMBER state regardless whether it was host software-, Port (aggressive)-, or device-initiated.

The power management state is terminated when either one of the following conditions becomes true:

- Software requests transition to active state by writing SATA_P 0CMD[ICC] = 1h
- Device requests interface wakeup by transmitting COMWAKE OOB sequence

The state of the interface (active, PARTIAL, or SLUMBER power management) is reflected in the SATA_P 0SSTS[IPM] field.

63.3.5.3 Hot Plug

This topic covers the following descriptions:

- Native Hot Plug

63.3.5.3.1 Native Hot Plug

The SATA block supports native SATA hot-plug through the use of the SATA_P#0SERR[DIAG_X] and SATA_P#0IS[PCS] bits. It is set every time the Link detects a COMINIT sequence, indicating hot plug insertion event or system power-up.

Hot plug removal is detected by a change in the state of the Port internal "PHY READY" signal. This event is reflected in the SATA_P#0SERR[DIAG_N] and SATA_P#0IS[PRCS] bits.

63.3.5.4 Port Multiplier Support

The SATA block supports Port Multiplier functionality with command-based switching. When a Port is connected to a Port Multiplier, software must first enumerate it by issuing software reset to Port 0Fh (control Port) on the Port Multiplier.

When the signature returned corresponds to a Port Multiplier, then a Port Multiplier is attached. When the signature returned corresponds to another device type, then a Port Multiplier is not attached.

The SATA block provides command list override feature (as indicated by the SATA_CAP[SCLO] = 1) via SATA_P 0CMD[CLO] to help software reliably enumerate the Port Multiplier:

1. Software ensures that SATA_P 0CMD[ST] bit is '0';
2. Software constructs the two Register FISes required for a software reset in the command list, where the PM Port field value in the Register FIS is cleared to 0Fh;
3. Software sets SATA_P 0CMD[CLO] to '1' to force the BSY and DRQ bits in the SATA_P 0TFD register to be cleared;
4. Software sets SATA_P 0CMD[ST] bit to '1' and set appropriate SATA_P 0CI bits in order to begin execution of the software reset command.

63.3.5.5 Interrupts

The SATA block uses a two-tiered interrupt structure defined in the AHCI specification.

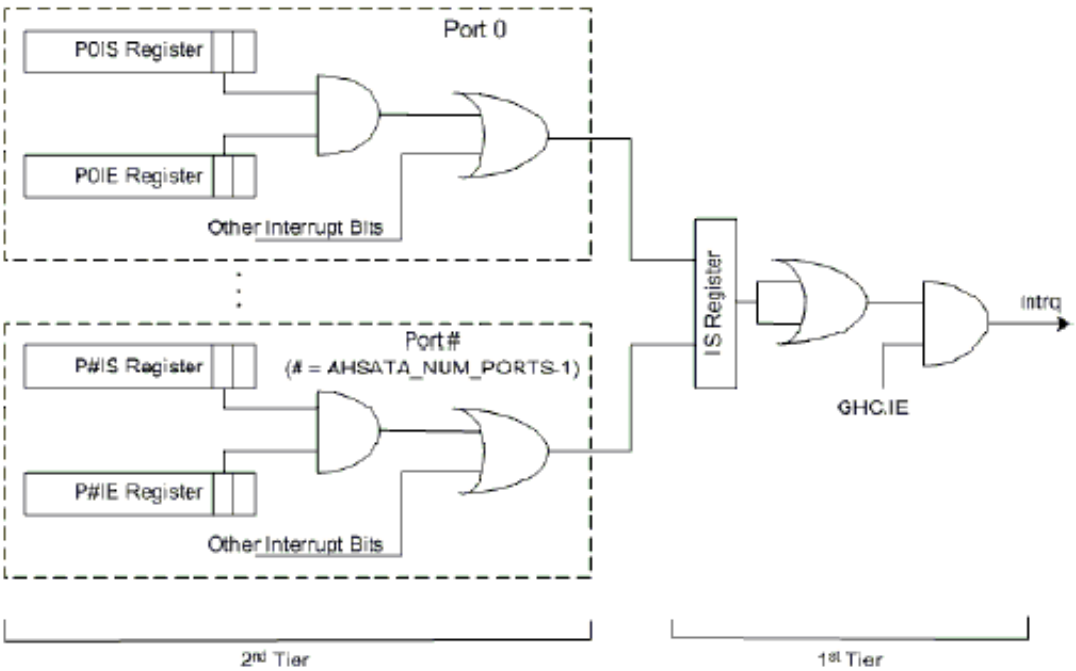


Figure 63-17. SATA block Interrupt Tiers

63.3.5.5.1 First Tier (SATA_IS Register)

The first tier is identified by the IS and SATA_GHC registers. SATA_GHC[IE] bit enables interrupts for the entire SATA block: when it is cleared, intrq output is not asserted regardless of any bits set in the SATA_IS register.

SATA_GHC[IE] bit acts as a mask and does not affect the setting of any interrupt status bits.

The 32-bit SATA_IS register reports the SATA Port has an interrupt pending.

Command Completion Coalescing (CCC) logic generates interrupt (if enabled by software) by setting the IS.IPS[INT] bit, where INT= AHSATA_NUM_PORTS. For example, if DWC_ahsata is configured with eight ports (AHSATA_NUM_PORTS = 8), then the ports use IS bits [7:0], while the CCC interrupt uses IS bit 8.

63.3.5.5.2 Second Tier (SATA_P 0IS Registers)

The second tier is identified in each Port through the SATA_P 0IS (status) and SATA_P 0IE (interrupt enable) register. The SATA_P 0IS register has various interrupt bits that can be individually enabled or disabled by setting the corresponding bit in the SATA_P0IE.

The status bit in the SATA_P 0IS is always set regardless of the setting of the corresponding SATA_P 0IE bit.

63.3.5.6 PHY and Link Control

The SATA block provides two registers - SATA_P0PHYCR and SATA_P0PHYSR for PHY control and status respectively (both are located in the Port PCSR module).

SATA_P0PHYCR register is mapped to the corresponding bits of the p0_phy_ctrl output bus and the SATA_P0PHYSR register to the phy_status input bus.

The Port Link Layer features (scrambler, descrambler, repeat drop) are controlled by the SATA_BISTCR[LLC] field (SATA_BISTCR is located in the GCSR module) and can be disabled in normal operation, such as for testing purposes, by clearing the corresponding bits. The required Port is selected using SATA_TESTR[PSEL] register.

SATA_BISTCR[LLC] bits are set on power up enabling scrambler, descrambler, RPD functions by default. To disable these functions, software must:

1. set SATA_P0SCTL[DET]
2. clear the required SATA_BISTCR[LLC] bits
3. clear SATA_P0SCTL[DET]

63.3.5.7 Reset Conditions

63.3.5.7.1 System Reset

System bus resets SATA block by asserting hresetn=0 (asynchronous bus reset). It is usually initiated on power-up or during system bus failure. All components of the SATA block are initialized, including Ports, Generic registers, BIU.

63.3.5.8 Global Reset

Software may globally reset SATA block by setting SATA_GHC[HR]. When software sets the SATA_GHC[HR] bit, the SATA block performs an internal reset action, then clears this bit when the reset is complete.

Writing 0 to SATA_GHC[HR] has no effect.

NOTE

This reset clears the field SATA_P 0SCTL[SPD]. All Port communication is restarted with the maximum allowable speed, as set by the p0_phy_spdmode input.

To perform the Global reset, software sets SATA_GHC[HR] to '1' and may poll until this bit is read to be '0', indicating the reset completion. The SATA block is initialized as follows:

- SATA_GHC[AE], SATA_GHC[IE], the SATA_IS register, and all Port register fields (except SATA_P 0FB and SATA_P 0CLB) that are not HwInit in the register memory space are reset;
- All other global registers/bits and any HwInit bits in the Port-specific registers are not affected by setting SATA_GHC[HR] to '1';
- The Port-specific registers SATA_P 0FB, and SATA_P 0CLB are not affected by setting SATA_GHC[HR] to '1';
- SATA_P 0CMD[SUD] bit is reset to '0'; software is responsible for setting the SATA_P 0CMD[SUD] and SATA_P 0SCTL[DET] fields appropriately such that communication can be established on the SATA link.

63.3.5.8.1 Port Reset (COMRESET)

Software causes a Port reset by writing 1h to the SATA_P 0SCTL[DET] field to invoke COMRESET on the interface and start a re-establishment of the PHY Layer communication.

Software should wait at least 1ms before clearing SATA_P 0SCTL[DET] to 0h; this ensures that at least one COMRESET signal is sent over the interface. After clearing SATA_P 0SCTL[DET] to 0h, software should wait for communication to be re-established as indicated by bit 0 of SATA_P 0SSTS[DET] being set to '1'. Then software should write all ones to the SATA_P 0SERR register to clear any bits that were set as part of the Port reset.

NOTE

SATA asserts the corresponding p 0_phy_reset(_n) output when P 0SCTL.DET=0x1. It negates p 0_phy_reset(_n) and sends a COMRESET OOB sequence when P 0SCTL.DET=0x0

63.3.5.8.2 Software Reset

Software builds two H2D Register FISes in the command list.

The first Register FIS has the SRST bit set to '1' in the Control field of the Register FIS, the 'C' bit is cleared to '0' in the Register FIS, and the command table has the CH[R] (reset) and CH[C] (clear BSY on R_OK) bits set to '1'. The CH[R] (reset) bit causes the Port to perform a SYNC escape when necessary to put the device into an idle condition before sending the software reset. The CH[C] (clear BSY on R_OK) bit needs to be set for the first Register FIS to clear the BSY bit and proceed to issue the next Register FIS since the device does not send a response to the first Register FIS in a software reset sequence. The second Register FIS has SRST='0' in the Control field of the Register FIS, the 'C' bit is cleared to '0'in the Register FIS, and the command table has the CH[R] (reset) and CH[C] (clear BSY on R_OK) bits cleared to '0'. When issuing a software reset sequence, there should not be other commands in the command list. Before issuing the software reset, software must clear SATA_P 0CMD[ST], wait for the Port to be idle (SATA_P 0CMD[CR]='0'), and then set SATA_P 0CMD[ST] bit again. SATA_P 0TFD[STS] BSY bit and SATA_P 0TFD[STS] DRQ bit must be cleared prior to issuing the reset. When SATA_P 0TFD[STS] BSY bit or the SATA_P 0TFD[STS] DRQ bit is still set based on the failed command, then a Port reset should be attempted or command list override (SATA_P 0CMD[CLO]) should be used.

NOTE

A Port reset (COMRESET) is the preferred mechanism for error recovery and should be used in place of software reset.

63.3.5.9 Interface Speed Support

The SATA block supports 1.5 Gb/s and 3 Gb/s interface speeds as indicated by the SATA_CAP[ISS]=2h. Software can limit a Port's speed to 1.5 Gb/s by setting SATA_P 0SCTL[SPD] field to 1h.

63.3.5.10 Staggered Spin-up

The SATA block supports staggered spin-up operation when SATA_CAP[SSS]='1'. This feature is used to individually spin-up attached devices, thus reducing power supply requirements for multiple devices power-up.

NOTE

In order for a system to support staggered spin-up, the devices and BIOS/driver software must also support staggered spin-up

The SATA_P 0CMD[SUD] bit is used to manipulate the PHY behavior. SATA_P 0SCTL[DET] and SATA_P 0CMD[SUD] must be set correctly in order to avoid illegal combinations of the two values.

The following table describes interaction between the SATA_P 0CMD[SUD] and SATA_P 0SCTL[DET] bits.

Table 63-3. SATA_P 0CMD[SUD] and SATA_P 0SCTL[DET] Interaction

SATA_P0SCTL[DET]	SATA_P0CMD[SUD]	Mode	Behavior
0h	0	Listen	Interface is in a reduced power state. When COMINIT is received then SATA_P 0SERR[DIAG_X] is set and no response (OOB signal) is sent to the device COMWAKE is ignored. The applcation must place the Port into this state only when no device is detected as connected to this Port. In this mode, the Port forces the PHY into a low power state without requesting a SLUMBER transition on the link.
0h	0 -> 1	Spin-Up	Port sends COMRESET, begins initialization sequence.
0h	1	Normal	Normal operating state when the Port is performing data transfers.
1h	0	(not allowed)	This combination is prohibited in hardware. SATA_P 0CMD[SUD] can not be cleared when SATA_P0 SCTL[DET]=1h, and SATA_P0 SCTL[DET] can not be set to 1h when SATA_P0 CMD[SUD}=0.
1h	1	Reset	Port continuously transmits COMRESET and does not listen for COMINIT. When COMINIT is received in this state, the SATA_P0 SERR[DIAG_X] bit is set.
1h-> 0h	1	Initialize	Port stops sending COMRESET, begins initialization sequence.
4h	N/A	Off	Port PHY is off.

Software must only clear SATA_P 0CMD[SUD] when it believes that no device is attached. In Listen Mode (SATA_P 0SCTL[DET]=0h and SATA_P 0CMD[SUD]=0), the Port PHY enters a reduced power state, equivalent to the SLUMBER power management state. The Port PHY enters this state without negotiating a transition to SLUMBER on the link, as asking for a transition to SLUMBER when no device is attached fails, and therefore the PHY remains in a high power state. To avoid this software should ensure that SATA_P 0SSTS[DET]=0h indicating that no device is present before clearing SATA_P 0CMD[SUD].

In order to spin up the devices attached to the SATA block Ports, software should perform the procedure outlined in Firmware Specific Initialization.

63.3.5.11 Asynchronous Notification

The SATA block supports asynchronous notification feature as indicated by the SATA_CAP[SSNT]=1. This feature allows an ATAPI device to send a signal to the host when media is inserted or removed and avoids polling the device for media changes.

The signal sent to the host is a Set Device Bits FIS with the 'I' (interrupt) and'N' (notification) bits set to '1'.

To use asynchronous notification, software should set the SATA_P 0IS[SDBS] bit to enable interrupt notification on a Set Device Bits FIS. When accesses to the ATAPI device are idle, software should place the device in a low power state. When the device has a media change, it signals this to the block's SATA Port with a Set Device Bits FIS. In response to receiving a P 0IS[SDBS] interrupt on an idle Port, software should interrogate the device to determine the cause of the interrupt.

The first DWORD of any FIS received by the host contains a 4-bit Port Multiplier Port (PM Port) field. The second DWORD of the BIST Activate FIS least significant byte (bits [7:0]) is stored in the SATA_BISTAFR[NCP] field. The PM Port field indicates which Port/target behind the Port Multiplier issued the FIS to the block's SATA Port. When a Set Device Bits FIS is received by the block's SATA Port and the 'N' (notification) bit is set, the bit position in the SATA_P 0SNTF register corresponding to the PM Port field is set. The block's SATA Port sets the SATA_P 0IS[SDBS] field when the 'I' (interrupt) bit is set in the Set Device Bits FIS. This causes an interrupt to be generated when that interrupt is enabled.

NOTE

When a Port Multiplier is not present, the PM Port field in the Set Device Bits FIS is 0h, causing bit 0 of the SATA_P 0SNTF register to be set.

63.3.5.12 BIST Operation

The SATA Port can be put into one of the BIST loopback modes described in the following subsections.

NOTE

Scrambler and Descrambler are bypassed (disabled) in the Port Link layer in all BIST modes by default. When needed, Scrambler and Descrambler can be enabled through software, by clearing the bits SATA_BISTCR[SCRAM] and SATA_BISTCR[DESCRAM] (SCRAM and DESCRAM are

bits within the SATA_BISTCR[LLC] field) prior to entering BIST mode.

63.3.5.13 Loopback Responder

Software must ensure that the Port is in idle state and there are no outstanding commands by checking SATA_P 0CI

and SATA_P 0SACT registers are both cleared, SATA_P 0TFD[STS] register BSY, DRQ and ERR bits are all cleared.

The block's SATA Port enters one of the BIST loopback responder modes when a corresponding BIST Activate FIS is successfully received from the device and is supported by the SATA block. SATA_P 0SSTS[DET] field returns 4h when read. Since BIST registers' locations are shared between all the active Ports, software must first select the Port for BIST operation by writing the Port number to the SATA_TESTR[PSEL] field before accessing SATA_BISTAFR (Port0 is selected in the SATA_TESTR[PSEL] on power-on (system) or global SATA block reset).

NOTE

When the device sends a BIST Activate FIS with a request to enter a non-supported loopback mode, SATA block responds with R_ERRp response upon reception of the FIS.

The following loopback responder modes are supported by the SATA block:

- Far-end retimed
  - The Port receives BIST Activate FIS with Pattern Definition field (bits [23:16] of the first DWORD) = 0x10 from the RxFIFO and stores it in the SATA_BISTAFR[PD] field
  - All the data received from the device in the form of a SATA-compliant pattern is retimed in the Link Layer and transmitted back to the device.
  - Alternately, this mode can be initiated with device disconnected from the Port PHY (Link is in NOCOMM state) when software writes SATA_BISTCR[FERLB]=1. After the device is connected to the SATA block, the device must transmit the number of ALIGNs required for the PHY to sync.
- Far-end analog (Port PHY must support this mode)

- The Port receives BIST Activate FIS with Pattern Definition field (bits [23:16] of the first DWORD) = 0x08 from the RxFIFO and stores it in the SATA_BISTAFR[PD] field
- The Port asserts p0_phy_farafelb signal to the PHY to put it to the Far-end analog loopback mode. The PHY receives and retransmits the raw data without retiming

- Far-end transmit only
  - The Port receives BIST Activate FIS with Pattern Definition field (bits [23:16] of the first DWORD) = 0x80 (scrambling is enabled) or 0xA0 (scrambling is bypassed) from the RxFIFO The Port stores it in the SATA_BISTAFR[PD]. The second DWORD of the BIST Activate FIS least significant byte (bits [7:0]) is stored in the SATA_BISTAFR[NCP] field.
  - The Port transmits corresponding a SATA non-compliant test pattern to the device based on the SATA_BISTAFR[NCP] value:
    - 0xF1: Low transition density pattern (LTDP)
    - 0xB5: High transition density pattern (HTDP)
    - 0xAB: Low frequency spectral component pattern (LFSCP)
    - 0x7F: Simultaneous switching outputs pattern (SSOP)
    - 0x8B: Lone bit pattern (LBP)
    - 0x78: Mid frequency test pattern (MFTP)
    - 0x4A: High frequency test pattern (HFTP)
    - 0x7E: Low frequency test pattern (LFTP)
  - Alternately, this mode can be initiated with device disconnected from the Port PHY (Link NOCOMM state) when software writes a one to the SATA_BISTCR[TXO] bit. SATA block transmits non-compliant BIST pattern defined by the value in the SATA_BISTCR[PATTERN] field.

Loopback responder BIST modes can be exited either when the device signals COMINIT OOB condition, or when the software initiates Port reset (COMRESET).

63.3.5.13.1 Loopback Initiator

The software first selects the Port for BIST operation by writing the Port number to the SATA_TESTR[PSEL] field, then the required pattern by writing to the SATA_BISTCR[PATTERN] field.

The software builds a BIST FIS with the required mode in the commands list and sets CTBAz[B]. Once a BIST command is placed into the list, software is not allowed to build any more commands until it clears SATA_P 0CMD[ST]. After the Port successfully transmits this FIS, it enters this mode and generates/receives the compliant test pattern selected by the SATA_BISTCR[PATTERN] field.

SATA_P 0SSTS[DET] returns 4h when read. SATA_BISTSR and SATA_BISTFCTR registers are updated with error/FIS count information for each received BIST FIS.

NOTE

The device must support either PARTIAL or SLUMBER power modes for near-end analog loopback mode, otherwise it should be initiated with the device disconnected from the Port PHY. It is not clear how the device responds when it does not support the requested BIST mode - with R_OKp and ignoring the request or with R_ERRp. In the former case, the host assumes the device has entered the BIST mode and starts the test that fails.

The following BIST initiator modes can be requested by the software:

- "Far-end retimed"
- "Far-end analog"
- "Near-end analog"
- "Far-end transmit only"

63.3.5.13.1.1 Far-end retimed

The host software writes the SATA_BISTCR[PATTERN] field to select one of the SATA-defined compliant patterns:

- 0000b: Simultaneous switching outputs pattern (SSOP)
- 0001b: High transition density pattern (HTDP)
- 0010b: Low transition density pattern (LTDP)
- 0011b: Low frequency spectral component pattern (LFSCP)
- 0100b: Composite pattern (COMP)
- 0101b: Lone bit pattern (LBP)
- 0110b: Mid frequency test pattern (MFTP)
- 0111b: High frequency test pattern (HFTP)
- 1000b: Low frequency test pattern (LFTP)

The software prepares BIST Activate FIS with bits [23:16]=10h of the first DWORD (Pattern Definition field) in the command list. The Port sends this BIST Activate FIS to the device.

After successful transmission of the BIST Activate FIS (device acknowledges the FIS with R_OKp) the Port generates the requested compliant pattern in the form of BIST frames continuously and checks for errors on the receive side.

SATA_BISTFCTR register is updated with the received BIST frame count and SATA_BISTSR with frame/burst error count. . SATA_P 0SERR register is updated with CRC, disparity, and 10B8B errors for each frame. SATA_BISTFCTR, and SATA_BISTSR registers can be cleared by writing '1' to the SATA_BISTCR[CNTCLR] bit.

To change the pattern, the software issues a Port Reset (COMRESET), writes to the SATA_BISTCR[PATTERN] field to select a new pattern and re-issues the command by setting the SATA_P 0CI bit.

63.3.5.13.1.2 Far-end analog

The software prepares BIST Activate FIS with bits [23:16]=08h of the first DWORD (Pattern Definition field) in the command list. The Port sends this BIST Activate FIS to the device.

The operation proceeds as described in the far-end retimed test above.

63.3.5.13.1.3 Near-end analog

NOTE

Port PHY must support this mode, plus both clk_asic0 and clk_rbc0 must be running in the selected power management states or when the device is disconnected. When this is not true, this loopback mode is not supported by the SATA block.

This mode can be initiated either in one of the power management modes (PARTIAL or SLUMBER) or with the device disconnected from the Port PHY (Link NOCOMM state). The software issues a PARTIAL or SLUMBER power state request to the device via SATA_P 0CMD[ICC] field and sets the SATA_BISTCR[NEALB] bit. The SATA_BISTCR[PATTERN] field selects the required BIST pattern.

The Port asserts p 0_phy_nearafelb to the PHY. The PHY loops the data from its transmitter to its receiver and ignores any data coming from the device.

The operation proceeds as described in the far-end retimed test above, except BIST Activate FIS is not sent to the device.

63.3.5.13.1.4 Far-end transmit only

The software prepares BIST Activate FIS in the command list with the second and third DWORDs containing the required pattern, and the first DWORD - with the Pattern Definition (bits [23:16]) value corresponding to the required mode - Bit 23 (T) is set, bits 20 (L), 19 (F), 17 (R) cleared and bits 22, 21 and 18 are used to enable the following options:

- Bit 22 (A) is set - Bypass ALIGN
- Bit 21 (S) is set - Bypass scrambling
- Bit 18 (P) is set - Primitive bit (refer to the SATA specification for more details)

The Port sends this BIST Activate FIS to the device. After the device acknowledges the reception of this FIS with R_OKp, the Port disables the PHY receiver and transmitter (any received data is ignored by the Link Layer, transmitter is idle and maintains common mode bias per SATA specification).

Loopback initiator BIST modes can be terminated either by the device when it signals COMINIT OOB condition (except the near-end analog mode), or when the software initiates Port reset (COMRESET).

63.3.5.14 Command Completion Coalescing

A Command Completion Coalescing (CCC) feature is used to reduce the interrupt and command completion overhead in a heavily-loaded system.

The SATA block core generates an interrupt to allow software to process completed commands when either of these conditions is true:

- A software-specified number of commands have completed
- A software-specified time-out has expired

This feature applies to all Ports selected to be in the CCC set by software via the SATA_CCC_PORTS register.

CCC logic uses SATA block-specific register SATA_TIMER1MS to generate 1ms interval based on the AHB clock frequency. Software must load this register with the required value before enabling the CCC feature:

- Fhclk * 1000, where Fhclk = AHB clock frequency, in MHz

Additional Command Completion Coalescing details and examples can be found in the AHCI specification.

NOTE

CCC logic can be removed from the SATA core if it is not needed by setting the CCC_SUPPORT configuration parameter to “Exclude” (0). In this case, all CCC-related registers become “reserved” (returns 0 on read).

63.4 Programming

The SATA block software initialization consists of two independent phases: a firmware phase (platform BIOS) and a system software phase.

This section contains the following sub-sections:

- Firmware Specific Initialization
- System software Specific Initialization

63.4.1 Firmware Specific Initialization

The firmware initialization is done on power-up.

The following registers should be initialized to values that reflect the capabilities supported by the platform:

- SATA_CAP[SSS] - support for staggered spin-up
- SATA_CAP[SMPS] - support for mechanical presence switches
- SATA_PI - Ports implemented
- SATA_P 0CMD[HPCP] - whether the Port is hot plug capable. The SATA_P 0CMD[HPCP] should be set to 1 when SATA_P 0CMD[MPSP] or SATA_P 0CMD[CPD] is set to 1 for the Port.
- SATA_P 0CMD[MPSP] - whether mechanical presence switch is attached to the Port.
- SATA_P 0CMD[CPD] - whether cold presence detect logic is attached to the Port.

NOTE

Firmware should initialize the HPCP, MPSP, and CPD bits for each Port implemented on the platform as defined by the SATA_PI register.

After firmware has initialized the above mentioned registers, it should then perform the following steps to complete the staggered spin-up process (when applicable to the platform) on each Port implemented (as indicated by the SATA_PI register):

1. Ensure that SATA_P 0CMD[ST]=0, SATA_P 0CMD[CR]=0, SATA_P 0CMD[FRE]=0, SATA_P 0CMD[FR]=0, and SATA_P 0SCTL[DET]=0.
2. Allocate memory for the command list and the FIS receive area. Set SATA_P 0CLB 0 to the physical address of the allocated command list. Set SATA_P 0FB to the physical address of the allocated FIS receive area. Then set P 0CMD[FRE] to 1.
3. Initiate a spin-up of the SATA drive attached to the Port by setting P 0CMD[SUD] to 1
4. Wait for a positive indication that a device is attached to the Port (the maximum time to wait for presence indication is specified in the Serial ATA specification). This is done by polling SATA_P 0SSTS[DET]. When SATA_P 0SSTS[DET] returns a value of 1h or 3h when read, then the firmware should continue to the next step, otherwise when polling process times out, it moves to the next implemented Port and returns to Step 1.
5. Clear the SATA_P 0SERR register by writing ones to each implemented bit location.
6. Wait for indication that SATA drive is ready. This is determined through examination of SATA_P 0TFD[STS]. When SATA_P 0TFD[STS] BSY bit, SATA_P 0TFD[STS] DRQ bit, and SATA_P 0TFD[STS] ERR bit are all 0, prior to the maximum allowed time as specified in the ATA/ATAPI-7 specification, the device is ready.

63.4.2 System software Specific Initialization

Software may perform the SATA block global reset prior to initializing by setting SATA_GHC[HR] to 1 when desired.

When firmware (BIOS) already allocated memory and initialized the appropriate registers for the command list and FIS receive area, the software may skip this step in the process.

Following is the list of steps for system software to place the SATA block into a minimally initialized state:

1. Determine which Ports are implemented by the SATA block, by reading the PI register. This bit map value aids the software to determine how many Ports are available and which Port registers need to be initialized.
2. Ensure that the SATA block is not in the running state by reading and examining each implemented Port's SATA_P 0CMD register. When SATA_P 0CMD[ST], SATA_P 0CMD[CR], SATA_P 0CMD[FRE] and SATA_P 0CMD[FR] are all cleared, the Port is in an idle state. Otherwise, the Port is not idle and should be placed in the idle state prior to manipulating the SATA block global and Port specific register. System software places a Port into the idle state by clearing SATA_P 0CMD[ST] and waiting for SATA_P 0CMD[CR] to return 0 when read. Software should wait at least 500ms for this to occur. When SATA_P 0CMD[FRE] is set to 1

software should clear it to 0 and wait at least 500ms for SATA_P 0CMD[FR] to return 0 when read. When SATA_P 0CMD[CR] or SATA_P 0CMD[FR] do not clear to 0 correctly, then software may attempt a Port reset or a global reset to recover.

3. Determine how many command slots the HBA supports, by reading SATA_CAP[NCS].
4. For each implemented Port, system software should allocate memory for and program:
   - SATA_P 0CLB
   - SATA_P 0FB

NOTE

It is good practice for system software to zero-out the memory allocated and referenced by SATA_P 0CLB and SATA_P 0FB. After setting SATA_P 0FB to the physical address of the FIS receive area, system software should set SATA_P 0CMD[FRE] to 1.

5. For each implemented Port, clear the SATA_P 0SERR register, by writing ones to each implemented bit location.
6. Determine which events should cause an interrupt and set the appropriate enable bits of the SATA_P 0IE register. To enable the SATA block to generate interrupts, system software must also set SATA_GHC[IE] to 1.

NOTE

Due to the multi-tiered nature of the SATA block interrupt architecture, system software must always ensure that the SATA_P 0IS (clear this first) and SATA_IS[IPS] (clear this second) register are cleared to '0' before programming the SATA_P 0IE and SATA_GHC[IE] registers. This prevents any residual bits set in these registers from causing an interrupt to be asserted.

Software should not set SATA_P 0CMD[ST] to 1 until it is determined that a functional device is present on the Port as determined by SATA_P 0TFD[STS] BSY bit, SATA_P 0TFD[STS] DRQ bit, and SATA_P 0TFD[STS] ERR bit all cleared, and SATA_P 0SSTS[DET]=3h. To enable the SATA_P 0TFD register to be updated with the initial Register FIS for a Port, the SATA_P 0SERR[DIAG_X] bit must be cleared to 0.

63.5 Software Manipulation of Port DMA

This section contains the following topics:

- Start (SATA_P 0CMD[ST])
- FIS Receive Enable (SATA_P 0CMD[FRE])

63.5.1 Start (SATA_P 0CMD[ST])

When SATA_P 0CMD[ST] is set to 1, software is not allowed to perform the following actions:

- Manipulate SATA_P 0CMD[POD] to power on or off a device through cold presence detect logic (when supported by the platform and enabled in the SATA block);
- Manipulate SATA_P 0SCTL[DET] to change the PHY state;
- Manipulate SATA_P 0CMD[SUD] to spin-up the device (when supported by the platform)

The above actions are only allowed while the Port is in the Not Running state, indicated by both SATA_P 0CMD[ST] and SATA_P 0CMD[CR] being 0.

Software should set SATA_P 0CMD[ST] only after the following conditions become true:

- SATA_P 0CMD[CR] is verified to be cleared to '0' and SATA_P 0CMD[FRE] has been set to 1;
- A functional device is present on the Port (as determined by SATA_P 0TFD[STS] BSY bit=0, SATA_P 0TFD[STS] DRQ bit=0, and SATA_P 0SSTS[DET]=3h) and the registers SATA_P 0CLB are programmed to valid values.

63.5.2 FIS Receive Enable (SATA_P 0CMD[FRE])

When SATA_P 0CMD[FRE] is set (causing SATA_P 0CMD[FR] to be set to 1), the Port receives FISes from the devices and copies them into system memory. When SATA_P 0CMD[FRE] is cleared (causing SATA_P 0CMD[FR] to be cleared to 0), received FISes are held in the RxFIFO, and when it is full, further FIS reception is blocked.

Software is allowed to manipulate SATA_P 0CMD[FRE] so that it may move the FIS receive area to a new location. When this bit is cleared to 0, software must first wait for SATA_P 0CMD[FR] to clear to 0, indicating that the Port DMA engine for FIS reception is in an idle condition. When SATA_P 0CMD[FR] and SATA_P 0CMD[FRE] are both cleared to 0, software may update the values of SATA_P 0FB. Prior to setting SATA_P 0CMD[FRE] to 1, software should ensure that SATA_P 0FB are set to valid values. Software should not write SATA_P 0FB while SATA_P 0CMD[FRE] is set to 1.

Software should set SATA_P 0CMD[FRE] to 1 prior to setting SATA_P 0CMD[ST] to 1. Software should not clear SATA_P 0CMD[FRE] while SATA_P 0CMD[ST] or SATA_P 0CMD[CR] is set to 1.

Upon global or Port reset, the SATA_P 0CMD[FRE] bit is cleared. The D2H Register FIS containing the device signature is accepted by the Port, and the signature field is updated.

NOTE

When the SATA block Port stops running due to an error (e.g., SATA_P 0IS[IFS] is set to 1), FISes may not be posted until the SATA_P 0CMD[ST] bit is cleared to 0 to recover from the error.

63.6 Register Descriptions

This section contains register memory maps, register groups, and bit-field descriptions.

63.6.1 Register Overview

The SATA block registers occupy 328 bytes of the 8-KB address space assigned to the SATA block and are divided into two parts: Generic control, and Port control.

Register space above the offset of 0x14F is reserved.

63.6.1.1 Register Basics:

All registers are 32-bits wide and can be accessed using 8, 16, or 32-bit wide transfers, except when noted otherwise in the register descriptions.

All registers that start below the offset 0x100 are global and apply to the entire SATA block.

63.6.1.2 Reserved Locations

When a register, field, or bit is reserved:

- Reads return zero.
- Writes have no effect.

63.7 Programmable Registers

This section provides high-level summaries of the Generic and Port control regsier maps.

The register names in SATA Memory Map are cross-referenced to the detailed register descriptions in the following section (double-click on the register name to link to the detailed description).

SATA memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1000_0000	HBA Capabilites Register (SATA_CAP)	32	R	0000_0000h	63.7.1/ 3843
1000_0004	Global HBA Control Register (SATA_GHC)	32	R/W	8000_0000h	63.7.2/ 3845
1000_0008	Interrupt Status Register (SATA_IS)	32	R/W	0000_0000h	63.7.3/ 3846
1000_000C	Ports Implemented Register (SATA_PI)	32	R	See section	63.7.4/ 3846
1000_0010	AHCI Version Register (SATA_VS)	32	R	0001_0100h	63.7.5/ 3847
1000_0014	Command Completion Coalescing Control (SATA_CCC_CTL)	32	R/W	See section	63.7.6/ 3848
1000_0018	Command Completion Coalescing Ports (SATA_CCC_PORTS)	32	R/W	0000_0000h	63.7.7/ 3849
1000_0024	HBA Capabilities Extended Register (SATA_CAP2)	32	R	0000_0004h	63.7.8/ 3850
1000_00A0	BIST Activate FIS Register (SATA_BISTAFR)	32	R	0000_0000h	63.7.9/ 3850
1000_00A4	BIST Control Register (SATA_BISTCR)	32	R/W	0000_0700h	63.7.10/ 3852
1000_00A8	BIST FIS Count Register (SATA_BISTFCTR)	32	R	0000_0000h	63.7.11/ 3855
1000_00AC	BIST Status Register (SATA_BISTSR)	32	R	0000_0000h	63.7.12/ 3855
1000_00BC	OOB Register (SATA_OOBR)	32	R/W	See section	63.7.13/ 3856
1000_00D0	General Purpose Control Register (SATA_GPCR)	32	R/W	0000_0000h	63.7.14/ 3857
1000_00D4	General Purpose Status Register (SATA_GPSR)	32	R/W	0000_0000h	63.7.15/ 3857

Table continues on the next page...

SATA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1000_00E0	Timer 1-ms Register (SATA_TIMER1MS)	32	R/W	0001_86A0h	63.7.16/ 3858
1000_00E8	Global Parameter 1 Register (SATA_GPARAM1R)	32	R	D800_0000h	63.7.17/ 3859
1000_00EC	Global Parameter 1 Register (SATA_GPARAM2R)	32	R	0000_0E1Eh	63.7.18/ 3861
1000_00F0	Port Parameter Register (SATA_PPARAMR)	32	R	0000_0022h	63.7.19/ 3862
1000_00F4	Test Register (SATA_TESTR)	32	R/W	0000_0000h	63.7.20/ 3863
1000_00F8	Version Register (SATA_VERSIONR)	32	R	3134_302Ah	63.7.21/ 3865
1000_0100	Port0 Command List Base Address Register (SATA_P0CLB)	32	R/W	0000_0000h	63.7.22/ 3865
1000_0108	Port0 FIS Base Address Register (SATA_P0FB)	32	R/W	0000_0000h	63.7.23/ 3866
1000_0110	Port0 Interrupt Status Register (SATA_P0IS)	32	R/W	0000_0000h	63.7.24/ 3867
1000_0114	Port0 Interrupt Enable Register (SATA_P0IE)	32	R/W	0000_0000h	63.7.25/ 3870
1000_0118	Port0 Command Register (SATA_P0CMD)	32	R/W	0000_0000h	63.7.26/ 3873
1000_0120	Port0 Task File Data Register (SATA_P0TFD)	32	R	0000_007Fh	63.7.27/ 3877
1000_0124	Port0 Signature Register (SATA_P0SIG)	32	R	FFFF_ FFFFh	63.7.28/ 3877
1000_0128	Port0 Serial ATA Status Register (SATA_P0SSTS)	32	R	0000_0000h	63.7.29/ 3878
1000_012C	Port0 Serial ATA Control {SControl} Register (SATA_P0SCTL)	32	R/W	0000_0000h	63.7.30/ 3879
1000_0130	Port0 Serial ATA Error Register (SATA_P0SERR)	32	R/W	0000_0000h	63.7.31/ 3880
1000_0134	Port0 Serial ATA Active Register (SATA_P0SACT)	32	R/W	0000_0000h	63.7.32/ 3882
1000_0138	Port0 Command Issue Register (SATA_P0CI)	32	R/W	0000_0000h	63.7.33/ 3883
1000_013C	Port0 Serial ATA Notification Register (SATA_P0SNTF)	32	w1c	0000_0000h	63.7.34/ 3884
1000_0170	Port0 DMA Control Register (SATA_P0DMACR)	32	R/W	0000_0044h	63.7.35/ 3884

Table continues on the next page...

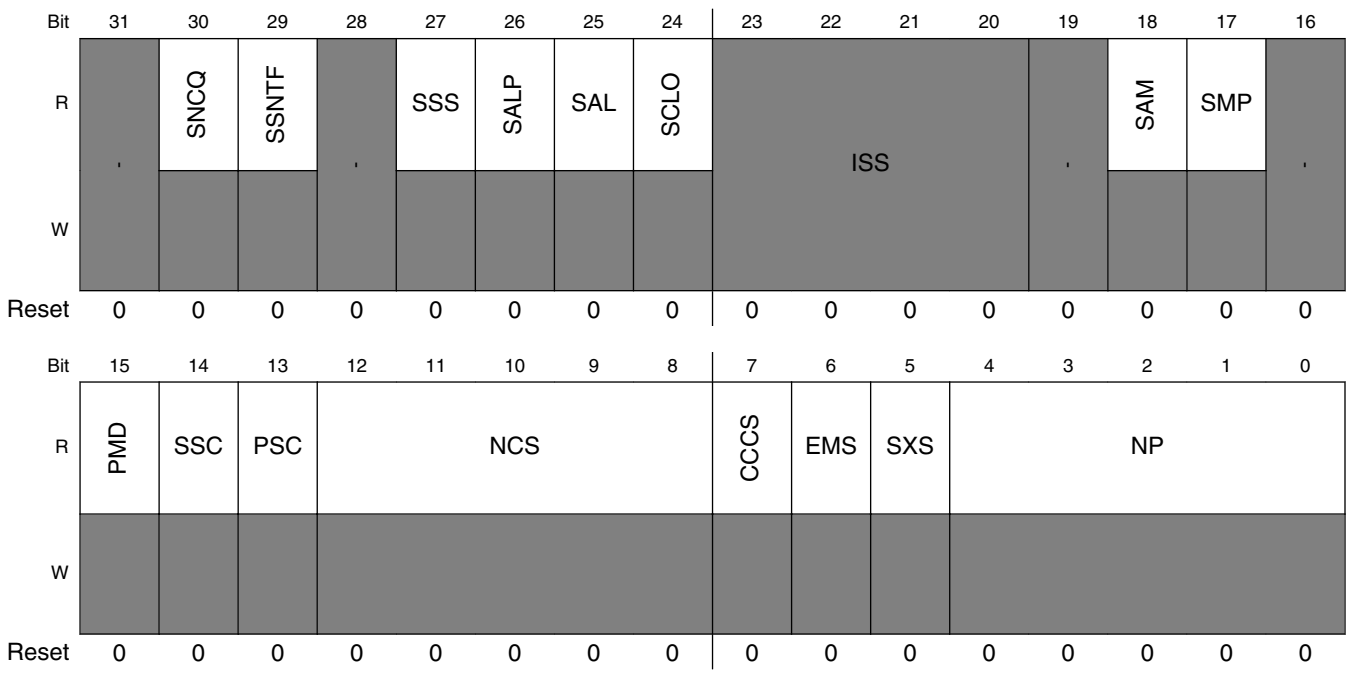
SATA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
1000_0178	Port0 PHY Control Register (SATA_P0PHYCR)	32	R/W	0000_0000h	63.7.36/ 3886
1000_017C	Port0 PHY Status Register (SATA_P0PHYSR)	32	R	0000_0000h	63.7.37/ 3887

63.7.1 HBA Capabilites Register (SATA_CAP)

This register indicates basic capabilities of the SATA block to the software.

Address: SATA_CAP is 1000_0000h base + 0h offset = 1000_0000h



SATA_CAP field descriptions

Field	Description
31 -	Reserved. Returns 0 on read.
30 SNCQ	Supports Native Command Queuing. SATA block supports SATA native command queueing by handling DMA Setup FIS natively.
29 SSNTF	Supports SNotification Register. SATA block supports SATA_P 0 SNTF (SNotification) register and its associated functionality.
28 -	Reserved. Returns 1 on read.

Table continues on the next page...

SATA_CAP field descriptions (continued)

Field	Description
27 SSS	Supports Staggered Spin-up. This bit is set by the system firmware/BIOS to indicate platform support for staggered devices' spin-up. SATA block supports this feature through the SATA_P 0 CMD[SUD] bit functionality.
26 SALP	Supports Aggressive Link Power Management. SATA block supports auto-generating (Port-initiated) Link Layer requests to the PARTIAL or SLUMBER power management states when there are no commands to process.
25 SAL	Supports Activity LED. SATA block supports activity indication using signal p 0 _act_led.
24 SCLO	Supports Command List Override. SATA block supports the SATA_P 0 CMD[CLO] bit functionality for Port Multiplier devices' enumeration.
23–20 ISS	Interface Speed Support. Reserved. Returns 0x2 on read.
19 -	Reserved. Returns 0 on read.
18 SAM	Supports AHCI Mode Only. SATA block supports AHCI mode only and does not support legacy, task-file based register interface.
17 SMP	Supports Port Multiplier. SATA block supports command-based switching Port Multiplier on any of its Ports.
16 -	Reserved.
15 PMD	PIO Multiple DRQ Block. SATA block supports multiple DRQ block data transfers for the PIO command protocol.
14 SSC	Slumber State Capable. SATA block supports transitions to the interface SLUMBER power management state.
13 PSC	Partial State Capable. SATA block supports transitions to the interface PARTIAL power management state.
12–8 NCS	Number of Command Slots. SATA block supports 32 command slots per Port.
7 CCCS	Command Completion Coalescing Support. SATA block supports command completion coalescing.
6 EMS	Enclosure Management Support. SATA block does not support enclosure management.
5 SXS	Supports External SATA. The options for this field are:

Table continues on the next page...

SATA_CAP field descriptions (continued)

Field	Description
	1 Indicates that the SATA block has one or more Ports that has a signal only connector (power is not part of that connector) that is externally accessible. When this bit is set to 1, the software can refer to the SATA_P 0 CMD[ESP] bit to determine whether a specific Port has its signal connector externally accessible. 0 Indicates that the SATA block has no Ports that have a signal only connector externally accessible. Reset Value: Configurable 1 when any of the SATA_P 0 CMD[ESP]=1 0 when all of the SATA_P 0 CMD[ESP]=0
4–0 NP	Number of Ports. 0's based value indicating the number of Ports supported by the SATA block: The options for this field are: • 0x00: 1 Port • 0x01: 2 Ports • 0x02: 3 Ports Reset Value: 0x00

63.7.2 Global HBA Control Register (SATA_GHC)

This register controls various global actions of the SATA block.

Address: SATA_GHC is 1000_0000h base + 4h offset = 1000_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	AE	-																													IE	HR
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SATA_GHC field descriptions

Field	Description
31 AE	AHCI Enable. This bit is always set since SATA block supports only AHCI mode as indicated by the SATA_CAP[SAM]=1.
30–2 -	Reserved
1 IE	Interrupt Enable. This global bit enables interrupts from the SATA block. When cleared, all interrupt sources from all the Ports are disabled (masked). When set, interrupts are enabled and any SATA block interrupt event causes intrq output assertion. This field is reset on Global reset (SATA_GHC[HR]=1).

Table continues on the next page...

SATA_GHC field descriptions (continued)

Field	Description
0 HR	HBA Reset. When set by the software, this bit causes an internal Global reset of the SATA block. All state machines that relate to data transfers and queueing return to an idle state, and all the Ports are re-initialized by sending COMRESET When staggered spin-up is not supported. When staggered spin-up is supported, then the software must spin-up each Port after this reset has completed. See Global Reset for details. The SATA block clears this bit when the reset action is done. A software write of 0 has no effect.

63.7.3 Interrupt Status Register (SATA_IS)

This register indicates which of the Ports within the SATA block have an interrupt pending and require service. This register is reset on Global reset (SATA_GHC[HR]=1).

- Size: 32 bits
- Address offset: 0x08
- Read/write access: Read/Write One to Clear
- Reset: 0x0000_00000

Address: SATA_IS is 1000_0000h base + 8h offset = 1000_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																														IPS	
W																															w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SATA_IS field descriptions

Field	Description
31–2 -	Reserved.
1–0 IPS	Interrupt Pending Status. When bit 1is set, this indicates that Port 0 has an interrupt pending. This bit is set when the Port has an interrupt event pending and the interrupt source is enabled (see the definition of the SATA_P 0 IE register). Bit 0 of the IPS field is not used.

63.7.4 Ports Implemented Register (SATA_PI)

This register indicates which Ports are exposed by the SATA block and are available for the software to use. It is loaded by the BIOS. For example, when the SATA block supports 8 Ports as indicated in the SATA_CAP[NP], only Ports 1, 3, 5, and 7 could be available, while Ports 0, 2, 4, and 6 being unavailable.

NOTE

The contents of this register are relevant to the SATA_CCC_PORTS (Command Completion Coalescing Ports) register.

Address: SATA_PI is 1000_0000h base + Ch offset = 1000_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																															PI
W																																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- See descriptions for reset values.

SATA_PI field descriptions

Field	Description
31–1 -	Reserved.
0 PI	Ports Implemented. BIOS must set this bit to 1

63.7.5 AHCI Version Register (SATA_VS)

This register indicates the major and minor version of the AHCI specification that the SATA block implementation supports. The SATA block supports version 1.20.

NOTE

The SATA block core currently complies fully with AHCI version 1.10 , and complies with AHCI version 1.2, except with respect to FIS-based switching. FIS-based switching is not currently supported.

BIST register locations (SATA_BISTAFR, SATA_BISTCR, SATA_BISTFCTR, and SATA_BISTSR) are shared between the SATA block Ports. Before accessing any of the BIST registers, the software must first select the required Port by writing the Port number to the SATA_TESTR[PSEL] field.

Address: SATA_VS is 1000_0000h base + 10h offset = 1000_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MJR																MNR															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

SATA_VS field descriptions

Field	Description
31–16 MJR	Major Version Number. Indicates that the major AHCI version is 1.
15–0 MNR	Minor Version Number. Indicates that the minor AHCI version is 20.

63.7.6 Command Completion Coalescing Control (SATA_CCC_CTL)

This register is used to configure the command completion coalescing (CCC) feature for the SATA block core. It is reset on Global reset.

Address: SATA_CCC_CTL is 1000_0000h base + 14h offset = 1000_0014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TV															
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CC								INT					-		EN
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- See descriptions for reset values.

SATA_CCC_CTL field descriptions

Field	Description
31–16 TV	Time-out Value. This field specifies the CCC time-out value in 1ms intervals. The software loads this value prior to enabling CCC. The options for this field are: • RW when SATA_CCC_CTL[EN]==0. • RO when SATA_CCC_CTL[EN]==1. A time-out value of 0x0000 is reserved and should not be used.
15–8 CC	Command Completions. This field specifies the number of command completions that are necessary to cause a CCC interrupt. The value 0x00 for this field disables CCC interrupts being generated based on the number of commands completed. In this case, CCC interrupts are only generated based on the timer. Software loads this value prior to enabling CCC: Field access is:

Table continues on the next page...

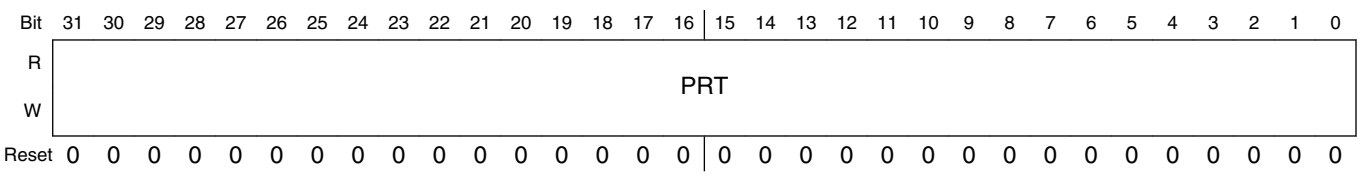
SATA_CCC_CTL field descriptions (continued)

Field	Description
	• RW when SATA_CCC_CTL[EN]==0 • RO when SATA_CCC_CTL[EN]==1
7–3 INT	Interrupt. Set this field to 0x01.
2–1 -	Reserved.
0 EN	Enable. **NOTE:** When field SATA_CCC_CTL[EN]==1, the software can not change the fields SATA_CCC_CTL[TV] and SATA_CCC_CTL[CC]. The options for this field are: 0 CCC feature is disabled and no CCC interrupts are generated. 1 CCC feature is enabled and CCC interrupts may be generated based on the time-out or command completion conditions.

63.7.7 Command Completion Coalescing Ports (SATA_CCC_PORTS)

This register specifies the Ports that are coalesced as part of the command completion coalescing (CCC) feature when SATA_CCC_CTL[EN]==1. It is reset on Global reset.

Address: SATA_CCC_PORTS is 1000_0000h base + 18h offset = 1000_0018h



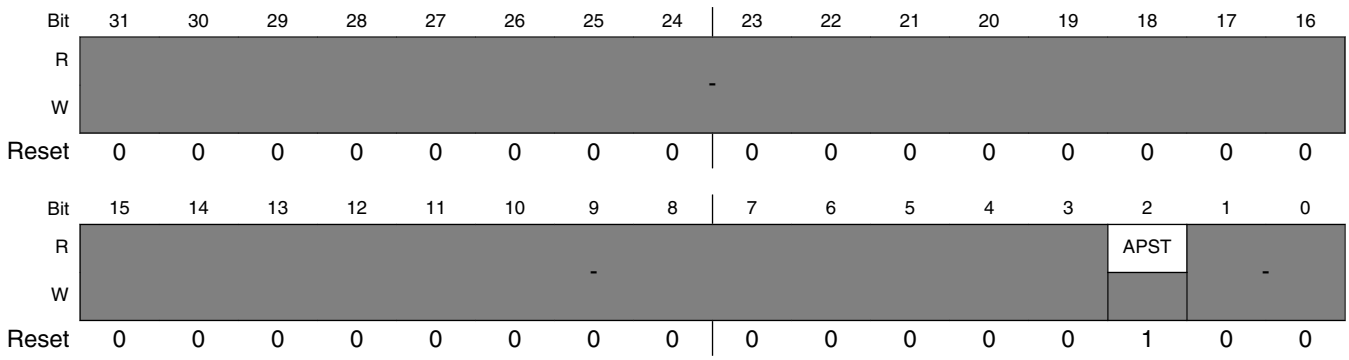
SATA_CCC_PORTS field descriptions

Field	Description
31–0 PRT	Ports. This field is bit significant. Each bit corresponds to a particular Port, where bit 0 corresponds to Port0. Bits set in this register must have the corresponding bit set in the SATA_PI (Ports Implemented Register). The options for this field are: 1 the corresponding Port is part of the CCC feature. 0 the corresponding Port is not part of the CCC feature.

63.7.8 HBA Capabilities Extended Register (SATA_CAP2)

This register indicates capabilites of the SATA block core to the software.

Address: SATA_CAP2 is 1000_0000h base + 24h offset = 1000_0024h



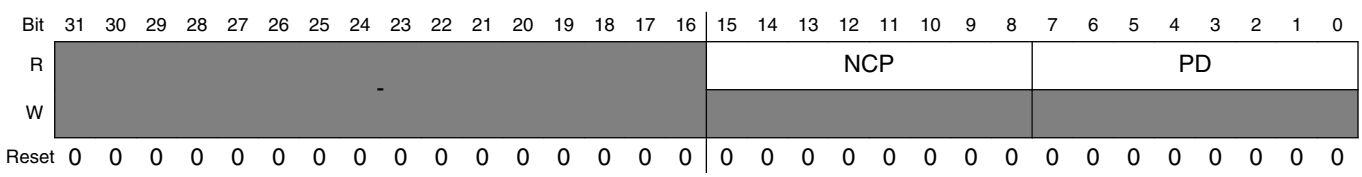
SATA_CAP2 field descriptions

Field	Description
31–3 -	Reserved
2 APST	Automatic Partial to Slumber Transitions. SATA block supports automatic Partial to Slumber transitions.
1–0 -	Reserved.

63.7.9 BIST Activate FIS Register (SATA_BISTAFR)

This register contains the pattern definition (bits [23:16] of the first DWORD) and data pattern (bits [7:0] of the second DWORD) fields of the received BIST Activate FIS. These fields define the SATA block loopback responder mode requested by the device. It is updated every time a new BIST Activate FIS is received from the device. Reset on Global or Port reset.

Address: SATA_BISTAFR is 1000_0000h base + A0h offset = 1000_00A0h



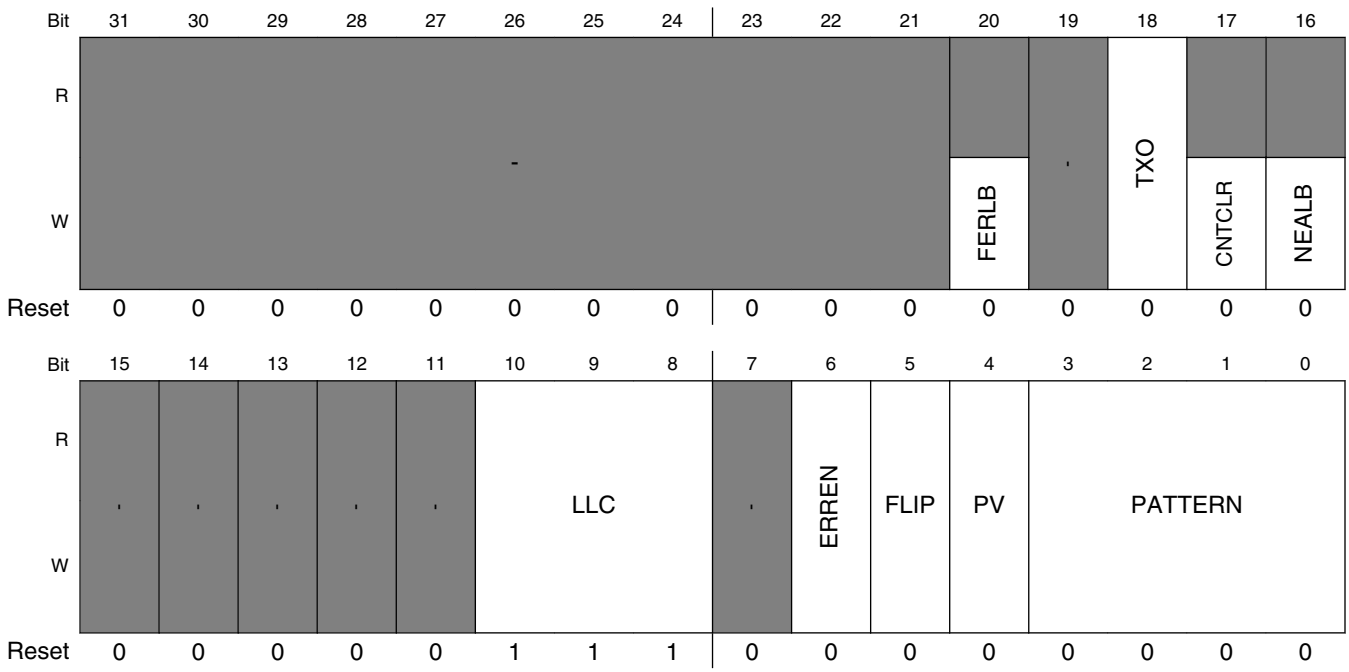
SATA_BISTAFR field descriptions

Field	Description
31–16 -	Reserved.
15–8 NCP	Least significant byte of the received BIST Activate FIS second DWORD (bits [7:0]). This value defines the required pattern for far-end transmit only mode (SATA_BISTAFR[PD]=0x80 or 0xA0): When none of these values is decoded, the simultaneous switching pattern is transmitted by default. 0xF1 Low transition density pattern (LTDP) 0xB5 High transition density pattern (HTDP) 0xAB Low frequency spectral component pattern (LFSCP) 0x7F Simultaneous switching outputs pattern (SSOP) 0x8B Lone Bit pattern (LBP) 0x78 Mid frequency test pattern (MFTP) 0x4A High frequency test pattern (HFTP) 0x7E Low frequency test pattern (LFTP)
7–0 PD	Pattern Definition Indicates the pattern definition field of the received BIST Activate FIS - bits [23:16] of the first DWORD. It is used to put the SATA block in one of the following BIST modes: For far-end transmit only modes SATA_BISTAFR[NCP] field contains the required data pattern. 0x10 Far-end retimed 0x08 Far-end analog (when PHY supports this mode) 0x80 Far-end transmit only 0xA0 Far-end transmit only with scrambler bypassed All other values should not be used by the device, otherwise, the FIS is negatively acknowledged with R_ERRp.

63.7.10 BIST Control Register (SATA_BISTCR)

This register is used in BIST initiator modes. It is loaded by the host software prior to sending BIST Activate FIS to the device (via TXBISTPD write). It is reset on a Global or Port reset.

Address: SATA_BISTCR is 1000_0000h base + A4h offset = 1000_00A4h



SATA_BISTCR field descriptions

Field	Description
31–21 -	Reserved.
20 FERLB	Far-end Retimed Loopback. When set, this bit is used to put the SATA block Link into Far-end Retimed mode, without the BIST Activate FIS, regardless whether the device is connected or disconnected (Link in NOCOMM state). This field is one-shot type and reads returns 0.
19 -	Reserved.
18 TXO	Transmit Only. This bit is used to initiate transmission of one of the non-compliant patterns defined by the SATA_BISTCR[PATTERN] value when the device is disconnected.
17 CNTCLR	Counter Clear This bit clears BIST error count registers. This field is one-shot type and reads returns 0. 1 Clear SATA_BISTFCTR, and SATA_BISTSR registers.

Table continues on the next page...

SATA_BISTCR field descriptions (continued)

Field	Description
16 NEALB	Near-End Analog Loopback This mode should be initiated either in the PARTIAL or SLUMBER power mode, or with the device disconnected from the Port PHY (Link NOCOMM state). BIST Activate FIS is not sent to the device in this mode. This bit places the Port PHY into near-end analog loopback mode. This field is one-shot type and reads returns 0: 1 Near-end analog loopback request. SATA_BISTCR[PATTERN] field contains the appropriate pattern.
15 -	Reserved.
14 -	Reserved.
13 -	Reserved.
12 -	Reserved.
11 -	Reserved.
10–8 LLC	Link Layer Control This field controls the Port Link Layer functions: scrambler, descrambler, and repeat primitive drop. Note the different meanings for normal and BIST modes of operation: • Bit8-SCRAM The options for this field are: 0 Scrambler disabled in normal mode, enabled in BIST mode 1 Scrambler enabled in normal mode, disabled in BIST mode • Bit9-DESCRAM The options for this field are: 0 Descrambler disabled in normal mode, enabled in BIST mode 1 Descrambler enabled in normal mode, disabled in BIST mode • Bit10-RPD The options for this field are: 0 Repeat primitive drop function disabled in normal mode, NA in BIST mode. 1 Repeat primitive drop function enabled in normal mode, NA in BIST mode. The SCRAM bit is cleared (enabled) by the Port when the Port enters a responder far-end transmit BIST mode with scrambling enabled (SATA_BISTAFR[PD]=0x80). In normal mode, the functions scrambler, descrambler, or RPD can be changed only during Port reset (SATA_P 0 SCTL[DET]=0x1)
7 -	Reserved.

Table continues on the next page...

SATA_BISTCR field descriptions (continued)

Field	Description
6 ERREN	Error Enable. This bit is used to allow or filter (disable) [internal errors outside the FIS boundary to set corresponding SATA_P 0 SERR bits. The options for this field are: 0 Filter errors outside the FIS, allow errors inside the FIS; 1 Allow errors outside or inside the FIS.
5 FLIP	Flip Disparity This bit is used to change disparity of the current test pattern to the opposite every time its state is changed by the software.
4 PV	Pattern Version This bit is used to select either short or long version of the SSOP, HTDP, LTDP, LFSCP, COMP patterns. The options for this field are: 0 Short pattern version 1 Long pattern version
3–0 PATTERN	This field defines one of the following SATA compliant patterns for far-end retimed/ far-end analog/ near-end analog initiator modes, or non-compliant patterns for transmit-only responder mode when initiated by the software writing to the SATA_BISTCR[TXO] bit. If the value is none of the listed below, Composite pattern (COMP) is transmitted by default. 0000b Simultaneous switching outputs pattern (SSOP) 0001b High transition density pattern (HTDP) 0010b Low transition density pattern (LTDP) 0011b Low frequency spectral component pattern (LFSCP) 0100b Composite pattern (COMP) 0101b Lone bit pattern (LBP) 0110b Mid frequency test pattern (MFTP) 0111b High frequency test pattern (HFTP) 1000b Low frequency test pattern (LFTP) All other values Reserved and should not be used.

63.7.11 BIST FIS Count Register (SATA_BISTFCTR)

This register contains the received BIST FIS count in the loopback initiator far-end retimed, far-end analog and near-end analog modes. It is updated each time a new BIST FIS is received. It is reset by Global reset, Port reset (COMRESET) or by setting the SATA_BISTCR[CNTCLR] bit. This register does not roll over and freezes when the FFFF_FFFFh value is reached. It takes approximately 65 hours of continuous BIST operation to reach this value.

Address: SATA_BISTFCTR is 1000_0000h base + A8h offset = 1000_00A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SATA_BISTFCTR field descriptions

Field	Description
31–0 -	Received BIST FIS Count

63.7.12 BIST Status Register (SATA_BISTSR)

This register contains errors detected in the received BIST FIS in the loopback initiator far-end retimed, far- end analog and near-end analog modes. It is updated each time a new BIST FIS is received. It is reset by Global reset, Port reset (COMRESET) or by setting the SATA_BISTCR[CNTCLR] bit.

Address: SATA_BISTSR is 1000_0000h base + ACh offset = 1000_00ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-								-								FRAMERR															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SATA_BISTSR field descriptions

Field	Description
31–24 -	Reserved.
23–16 -	Reserved.

Table continues on the next page...

SATA_BISTSR field descriptions (continued)

Field	Description
15–0 FRAMERR	Frame Error. This field contains the frame error count. It is accumulated (new value is added to the old value) each time a new BIST frame with a CRC error is received. The FRAMERR value does not roll over and freezes at FFFFh.

63.7.13 OOB Register (SATA_OOBR)

This register controls the Link layer OOB detection counters. The default values, MIN_COMWAKE, MAX_COMWAKE, MIN_COMINIT and MAX_COMINIT are calculated based on the RXOOB_CLK parameter and loaded on power-up or asynchronous SATA block reset.

Address: SATA_OOBR is 1000_0000h base + BCh offset = 1000_00BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WE	cwMin							cwMax								ciMin								ciMax							
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- See descriptions for reset values.

SATA_OOBR field descriptions

Field	Description
31 WE	Write Enable This bit is cleared when COMRESET is detected. The options for this field are: 1 SATA_OOBR bits [30:0] can be written 0 SATA_OOBR bits [30:0] are read-only
30–24 cwMin	COMWAKE Minimum Value This field is RW when WE=1 and RO when WE=0.
23–16 cwMax	COMWAKE Maximum Value This field is RW when WE=1 and RO when WE=0.
15–8 ciMin	COMINIT Minimum Value This field is RW when WE=1 and RO when WE=0.
7–0 ciMax	COMINIT Maximum Value This field is RW when WE=1 and RO when WE=0.

63.7.14 General Purpose Control Register (SATA_GPCR)

This 32-bit register is used for general purpose control. This register only exists when GP_CTRL parameter is set to “Include” otherwise this location is reserved.

The bits of this register are connected to the corresponding bits of the gp_ctrl output. Resets on power-up (system reset) only to the GP_CTRL_DEF value.

Address: SATA_GPCR is 1000_0000h base + D0h offset = 1000_00D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GP_CONTROL																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SATA_GPCR field descriptions

Field	Description
31–0 GP_CONTROL	General Purpose Control. Present only when GP_CTRL=Include(1). Reset Value: Configurable parameter GP_CTRL_DEF

63.7.15 General Purpose Status Register (SATA_GPSR)

This 32-bit register is used to monitor the general purpose status. This register only exists when GP_STAT parameter is set to “Include”, otherwise, this location is reserved.

The bits of this register reflect the state of the corresponding bits of the gp_status input. Signals connected to the gp_status input can be asynchronous to any of the DWC_ahsata clocks, however they must not change faster than five hclk/aclk periods, otherwise the GPSR register may never be updated with the intermediate changing values.

Address: SATA_GPSR is 1000_0000h base + D4h offset = 1000_00D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GP_STATUS																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

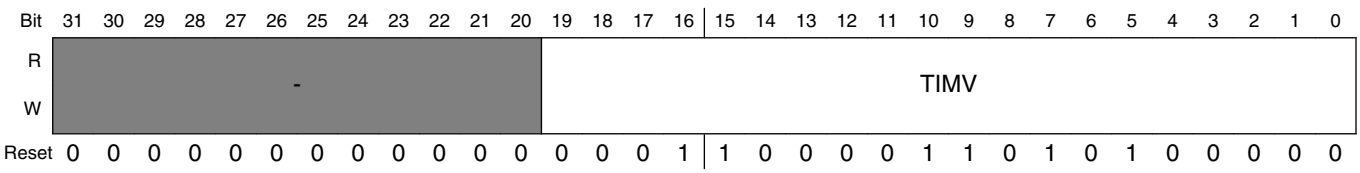
SATA_GPSR field descriptions

Field	Description
31–0 GP_STATUS	General Purpose Status. Present only when GP_STAT=Include(1)

63.7.16 Timer 1-ms Register (SATA_TIMER1MS)

This register is used to generate a 1-ms tick for the command completion coalescing (CCC) logic, based on the AHB bus clock frequency. The Software must initialize this register with the required value after power up before using the CCC feature. This register is reset to 100,000 (TIMV value for 100-MHz hclk) on power up and is not affected by Global reset.

Address: SATA_TIMER1MS is 1000_0000h base + E0h offset = 1000_00E0h



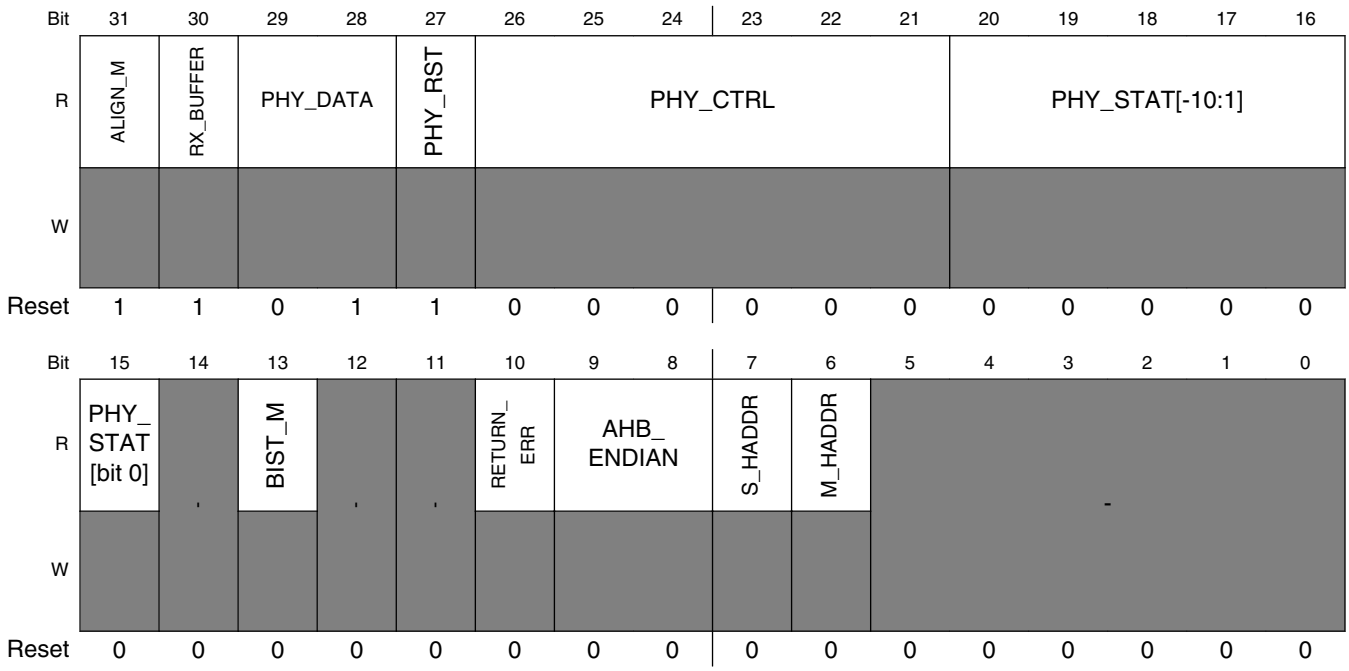
SATA_TIMER1MS field descriptions

Field	Description
31–20 -	Reserved.
19–0 TIMV	1ms Timer Value This field contains the following value for the internal timer to generate 1-ms tick: Fhclk*1000 where Fhclk = AHB clock frequency in MHz The options for this field are: • RW when SATA_CCC_CTL[EN]==0 • RO when SATA_CCC_CTL[EN]==1.

63.7.17 Global Parameter 1 Register (SATA_GPARAM1R)

This read-only register contains encoded information about the SATA block global configuration parameters' settings.

Address: SATA_GPARAM1R is 1000_0000h base + E8h offset = 1000_00E8h



SATA_GPARAM1R field descriptions

Field	Description
31 ALIGN_M	Rx Data Alignment 0 Misaligned —
30 RX_BUFFER	Rx Data Buffer — 1 Include
29–28 PHY_DATA	PHY Data Width 0x1: PHY DATA WIDTH = 2
27 PHY_RST	PHY Reset Mode 0 Low —
26–21 PHY_CTRL	PHY Control Width 0x20 (32) PHY Control Width is 32 bits.

Table continues on the next page...

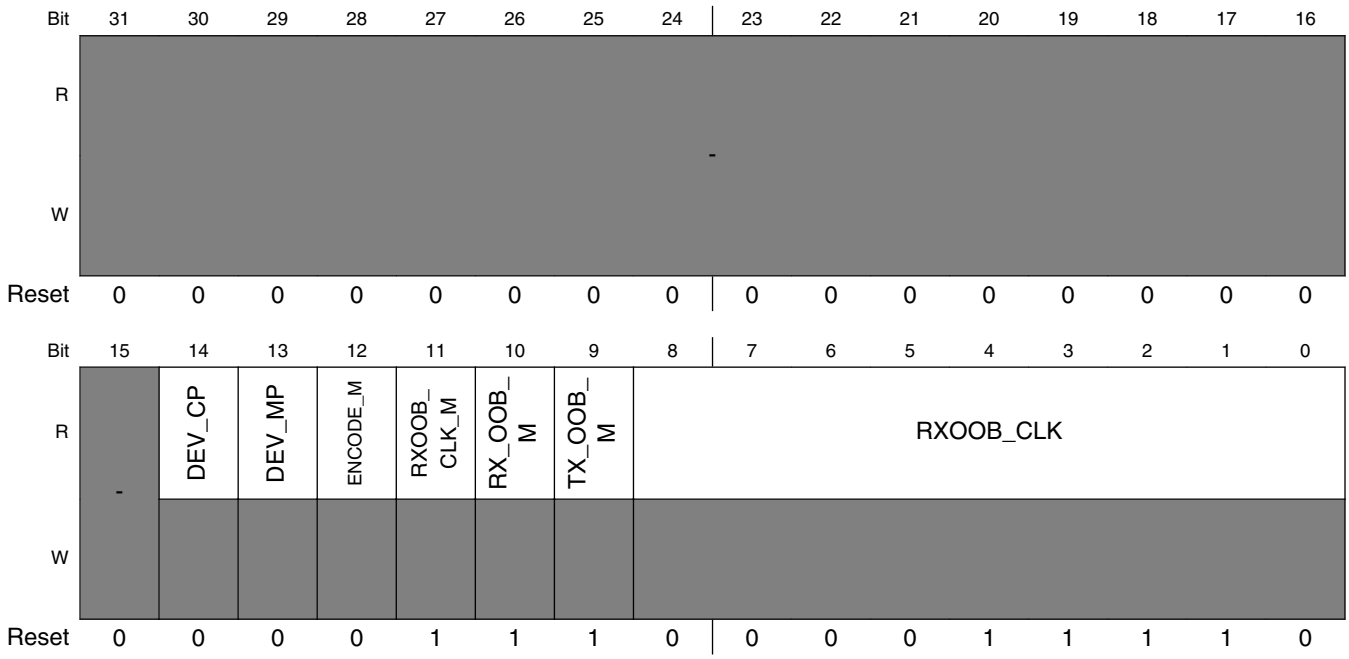
SATA_GPARAM1R field descriptions (continued)

Field	Description
20–15 PHY_STAT	PHY Status Width 0x20 (32) PHY Status Width is 32 bits.
14 -	Reserved. Returns 0.
13 BIST_M	BIST Loopback Checking Depth 0 FIS —
12 -	Reserved. Returns 1.
11 -	Reserved
10 RETURN_ERR	AHB Error Response — 1 True
9–8 AHB_ENDIAN	AHB Bus Endianness 0 Little Endian — —
7 S_HADDR	AHB Slave Address Bus Width 0 32 bits —
6 M_HADDR	AHB Master Address Bus Width 0 32 bits —
5–0 -	Reserved.

63.7.18 Global Parameter 1 Register (SATA_GPARAM2R)

This read-only register contains encoded information about the SATA block global configuration parameters' settings.

Address: SATA_GPARAM2R is 1000_0000h base + ECh offset = 1000_00ECh



SATA_GPARAM2R field descriptions

Field	Description
31–15 -	Reserved
14 DEV_CP	Cold Presence Detect — — 0 cold presence detect is not supported
13 DEV_MP	Mechanical Presence Switch — — 0 Mechanical presence switch is not supported
12 ENCODE_M	8b/10b Encoding/Decoding — — 1 8b/10b encoding/decoding is supported
11 RXOOB_CLK_M	Rx OOB Clock Mode

Table continues on the next page...

SATA_GPARAM2R field descriptions (continued)

Field	Description
	— — 1 separate clock
10 RX_OOB_M	Rx OOB Mode — — 1 Rx OOB signalling is supported
9 TX_OOB_M	Tx OOB Mode — — 1 Tx OOB signalling is supported
8–0 RXOOB_CLK	Rx OOB Clock Frequency This field returns 0x32 (50 decimal) The Rx OOB clock frequency is 50 MHz. .

63.7.19 Port Parameter Register (SATA_PPARAMR)

This read-only register contains encoded information about the selected Port configuration parameters' settings. The Port is selected by the SATA_TESTR[PSEL] field.

Address: SATA_PPARAMR is 1000_0000h base + F0h offset = 1000_00F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-				'	'	TX_MEM_M	TX_MEM_S	RX_MEM_M	RX_MEM_S	TXFIFO_DEPTH			RXFIFO_DEPTH		
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0

SATA_PPARAMR field descriptions

Field	Description
31–12 -	Reserved
11 -	Reserved
10 -	Reserved
9 TX_MEM_M	Tx FIFO Memory Read Port Type — 1 Sync
8 TX_MEM_S	Tx FIFO Memory Type 0 External —
7 RX_MEM_M	Rx FIFO Memory Read Port Type — 1 Sync
6 RX_MEM_S	Rx FIFO Memory Type 0 External —
5–3 TXFIFO_DEPTH	Tx FIFO Depth 0x4 Tx FIFO Depth set to 512 DWORDs
2–0 RXFIFO_DEPTH	Rx FIFO Depth 0x4 Rx FIFO Depth set to 512 DWORDs

63.7.20 Test Register (SATA_TESTR)

This register is used to put the SATA block slave interface into a test mode and to select a Port for BIST operation.

Address: SATA_TESTR is 1000_0000h base + F4h offset = 1000_00F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-													PSEL			-															TEST_IF
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SATA_TESTR field descriptions

Field	Description
31–19 -	Reserved
18–16 PSEL	Port Select This field is used to select a Port for BIST operation: The options for this field are: 0x0 Port0 is selected 0x0 Port1 is selected 0x0 Port2 is selected 0x0 Port3 is selected 0x0 Port4 is selected 0x0 Port5 is selected 0x0 Port6 is selected 0x0 Port7 is selected
15–1 -	Reserved
0 TEST_IF	TEST_IF: Test Interface Normal operation is disabled. The following registers can be accessed in this mode: - SATA_GHC register IE bit - SATA_BISTAFR register NCP and PD bits become read-write - SATA_BISTCR register LLC, ERREN, FLIP, PV, PATTERN - SATA_BISTFCTR, SATA_BISTSR become read-write - SATA_P 0 CLB , SATA_P 0 FB registers - SATA_P 0 IS register RW1C and UFS bits become read-write - SATA_P 0 IE register - SATA_P 0 CMD register ASP, ALPE, DLAE, ATAPI, PMA bits - SATA_P 0 TFD, SATA_P 0 SIG registers become read-write - SATA_P 0 SCTL register - SATA_P 0 SERR register RW1C bits become read-write bits - SATA_P 0 SACT, SATA_P 0 CI, SATA_P 0 SNTF registers become read-write - SATA_P 0 DMACR register - SATA_P 0 PHYCR register - SATA_P 0 PHYSR register becomes read-write Notes: • Interrupt is asserted when any of the SATA_IS register bits is set after setting the corresponding SATA_P 0 IS and SATA_P 0 IE registers and SATA_GHC[IE]=1. • SATA_CAP[SMPS], SATA_CAP[SSS], SATA_PI, SATA_P 0 CMD[ESP], SATA_P 0 CMD[CPD], SATA_P 0 CMD[MPSP], and SATA_P 0 CMD[HPCP] register bits are HwInit type and can not be used in Test mode. They are written once after power-on reset and become read-only. • Global SATA block reset must be issued (SATA_GHC[HR]=1) after TEST_WHEN bit is cleared following the Test mode operation.

Table continues on the next page...

SATA_TESTR field descriptions (continued)

Field	Description
	This bit is used to put the SATA block slave interface into the test mode: The options for this field are: 0 Normal mode: the read back value of some registers is a function of the SATA block state and does not match the value written. 1 Test mode: the read back value of the registers matches the value written.

63.7.21 Version Register (SATA_VERSIONR)

This 32-bit read-only register contains a hard-coded ASCII string that represents the version level of the SATA block.For the i.MX53, this register contains the ASCII string "140*" (hexadecimal 0x3134302A).

Address: SATA_VERSIONR is 1000_0000h base + F8h offset = 1000_00F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																															
W																																
Reset	0	0	1	1	0	0	0	1	0	0	1	1	0	1	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	0	1	0

SATA_VERSIONR field descriptions

Field	Description
31–0 -	SATA block hard-coded hexadecimal version value encoded in ASCII.

63.7.22 Port0 Command List Base Address Register (SATA_P0CLB)

Address: SATA_P0CLB is 1000_0000h base + 100h offset = 1000_0100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CLB																						-									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SATA_P0CLB field descriptions

Field	Description
31–10 CLB	Command List Base Address Indicates the 32-bit base physical address for the command list for this Port. This base is used when fetching commands to execute. The structure pointed to by this address range is 1 KB in length. This address must be 1-KB-aligned as indicated by bits [9:0] being read only.
9–0 -	Reserved.

63.7.23 Port0 FIS Base Address Register (SATA_P0FB)

Address: SATA_P0FB is 1000_0000h base + 108h offset = 1000_0108h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FB																								-							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

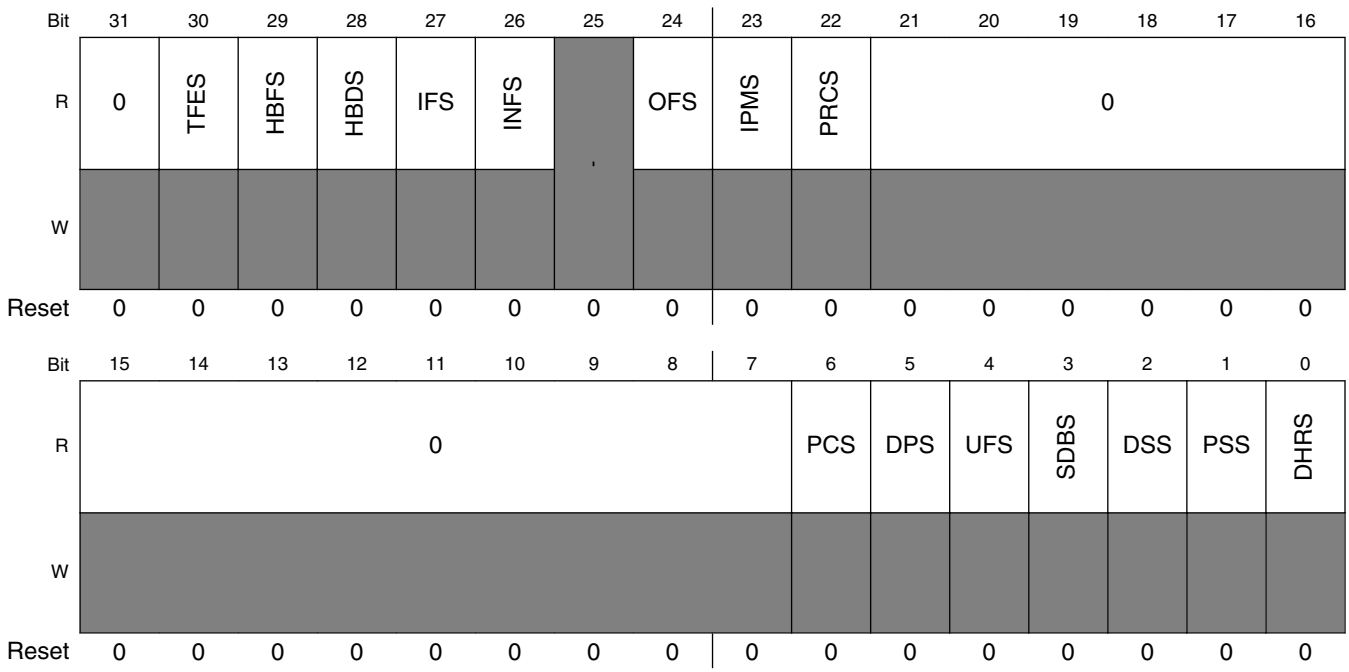
SATA_P0FB field descriptions

Field	Description
31–8 FB	FIS Base Address. Indicates the 32-bit base physical address for received FISes. The structure pointed to by this address range is 256 bytes in length. This address must be 256byte-aligned as indicated by bits [7:0] being read only. Reset: 0x000000
7–0 -	Reserved.

63.7.24 Port0 Interrupt Status Register (SATA_P0IS)

This register is used to generate SATA block interrupt when any of the bits are set. Bits in this register are set by some internal conditions, and cleared by the software writing ones in the positions it wants to clear. This register is reset on Global SATA block reset.

Address: SATA_P0IS is 1000_0000h base + 110h offset = 1000_0110h



SATA_P0IS field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved. Returns 0 on read.
30 TFES	Task File Error Status. This bit is set whenever the SATA_P 0 TFD[STS] register is updated by the device and the error bit (bit 0) is set.
29 HBFS	Host Bus Fatal Error Status. This bit is set when SATA block AHB Master detects an ERROR response from the slave.
28 HBDS	Host Bus Data Error Status. This bit is always cleared to 0.
27 IFS	Interface Fatal Error Status This bit is set when any of the following conditions is detected: • SYNC escape is received from the device during H2D Register or Data FIS transmission; • One or more of the following errors are detected during Data FIS transfer:

Table continues on the next page...

SATA_P0IS field descriptions (continued)

Field	Description
	- 10B to 8B Decode Error (SATA_P 0 SERR[DIAG_B]) - Protocol (SATA_P 0 SERR[ERR_P]) - CRC (SATA_P 0 SERR[DIAG_C]) - Handshake (SATA_P 0 SERR[DIAG_H]) - PHY Not Ready (SATA_P 0 SERR[ERR_C]) • Unknown FIS is received with good CRC, but the length exceeds 64 bytes; • PRD table byte count is zero. Port DMA transitions to a fatal state until the software clears SATA_P 0 CMD[ST] bit or resets the interface by way of Port or Global reset.
26 INFS	Interface Non-fatal Error Status This bit is set when any of the following conditions is detected: • One or more of the following errors are detected during non-data FIS transfer - 10B to 8B Decode Error (SATA_P 0 SERR[DIAG_B]) - Protocol (SATA_P 0 SERR[ERR_P]) - CRC (SATA_P 0 SERR[DIAG_C]), - Handshake (SATA_P 0 SERR[DIAG_H]) - PHY Not Ready (SATA_P 0 SERR[ERR_C]); • Command list underflow during read operation (i.e. DMA read) when the software builds command table that has more total bytes than the transaction given to the device.
25 -	Reserved
24 OFS	Overflow Status This bit is set when command list overflow is detected during read or write operation when the software builds command table that has fever total bytes than the transaction given to the device. Port DMA transitions to a fatal state until the software clears SATA_P 0 CMD[ST] bit or resets the interface by way of Port or Global reset.
23 IPMS	Incorrect Port Multiplier Status. Indicates that the HBA received a FIS from a device whose Port Multiplier field did not match what was expected. This bit may be set during enumeration of devices on a Port Multiplier due to the normal Port Multiplier enumeration process. The software must use the IPMS bit only after enumeration is complete on the Port Multiplier.
22 PRCS	PHY Ready Change Status This bit reflects the state of the SATA_P 0 SERR[DIAG_N] bit. When set to 1, indicates the internal p 0 _phy_ready signal changed state. To clear this bit, the software must clear the SATA_P 0 SERR[DIAG_N] bit to 0.

Table continues on the next page...

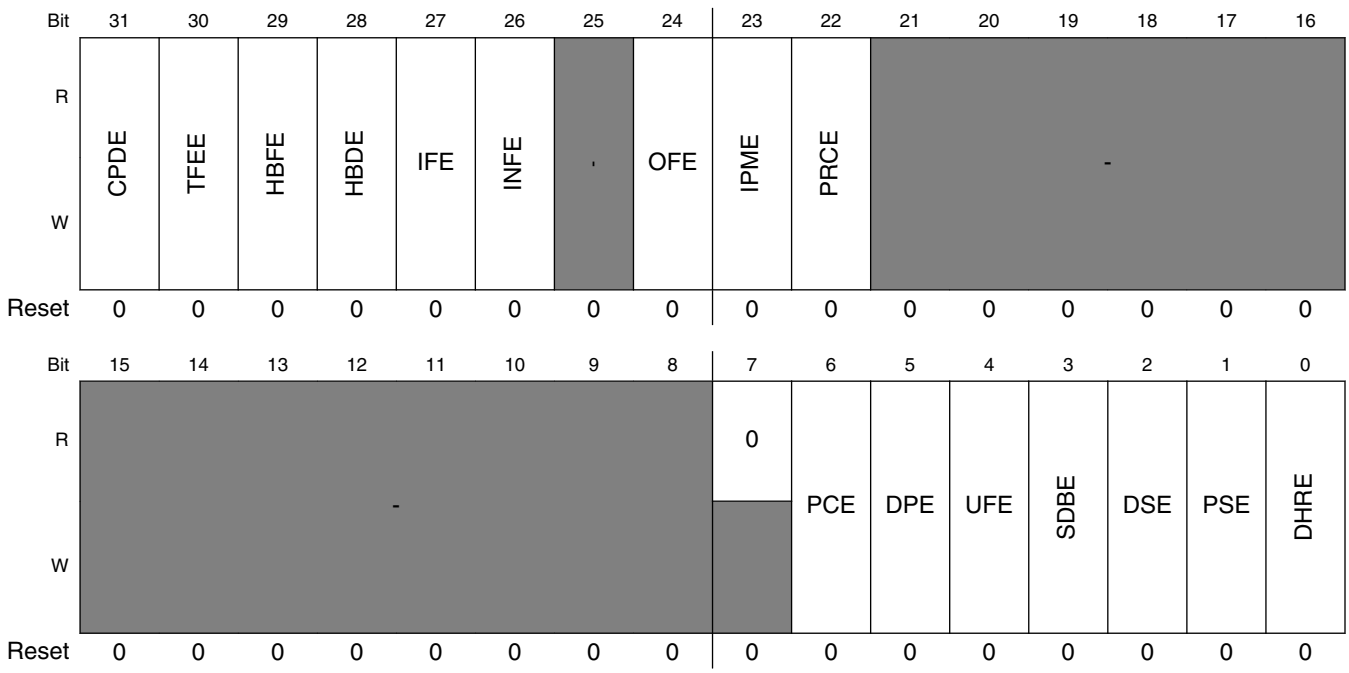
SATA_P0IS field descriptions (continued)

Field	Description
21–7 Reserved	This read-only field is reserved and always has the value zero. Reserved. Returns 0 on read.
6 PCS	Port Connect Change Status This bit is cleared only when SATA_P 0 SERR[DIAG_X] is cleared. This bit reflects the state of the SATA_P 0 SERR[DIAG_X] bit: 1 Change in Current Connect Status; 0 No change in Current Connect Status.
5 DPS	Descriptor Processed A PRD with the I bit set has transferred all of its data. **NOTE:** This is an opportunistic interrupt and must not be used to definitively indicate the end of a transfer. Two PRD interrupts could happen close in time together such that the second interrupt is missed when the first PRD interrupt is being cleared.
4 UFS	Unknown FIS Interrupt. When set to 1, indicates that an unknown FIS was received and has been copied into system memory. This bit is cleared to 0 by the software clearing the SATA_P 0 SERR[DIAG_F] bit to 0. **NOTE:** The UFS bit does not directly reflect the SATA_P 0 SERR[DIAG_F] bit. SATA_P 0 SERR[DIAG_F] bit is set immediately when an unknown FIS is detected, whereas the UFS bit is set when that FIS is posted to memory. The software should wait to act on an unknown FIS until the UFS bit is set to 1 or the two bits may become out of sync.
3 SDBS	Set Device Bits Interrupt. A Set Device Bits FIS has been received with the 'I' bit set and has been copied into system memory.
2 DSS	DMA Setup FIS Interrupt A DMA Setup FIS has been received with the 'I' bit set and has been copied into system memory.
1 PSS	PIO Setup FIS Interrupt. A PIO Setup FIS has been received with the 'I' bit set, it has been copied into system memory, and the data related to that FIS has been transferred. **NOTE:** This bit is set even when the data transfer resulted in an error.
0 DHRS	Device to Host Register FIS Interrupt A D2H Register FIS has been received with the 'I' bit set, and has been copied into system memory.

63.7.25 Port0 Interrupt Enable Register (SATA_P0IE)

This register enables and disables the reporting of the corresponding interrupt to the software. When a bit is set (1), and the corresponding interrupt condition is active, then the SATA block intrq output is asserted. Interrupt sources that are disabled (0) are still reflected in the status registers. This register is symmetrical with the SATA_P 0 IS register. This register is reset on Global SATA block reset.

Address: SATA_P0IE is 1000_0000h base + 114h offset = 1000_0114h



SATA_P0IE field descriptions

Field	Description
31 CPDE	Cold Port Detect Enable Read-only. Returns 0.
30 TFEE	Task File Error Enable Dependencies: when the following conditions are true, the intrq output signal is asserted: • This bit=1 • SATA_GHC[IE]=1 • SATA_P 0 IS[TFES]=1
29 HBFE	Host Bus Fatal Error Enable Dependencies: when the following conditions are true, the intrq output signal is asserted: • This bit=1 • SATA_GHC[IE]=1

Table continues on the next page...

SATA_P0IE field descriptions (continued)

Field	Description
	• SATA_P 0 IS[HBFS]=1
28 HBDE	Host Bus Data Error Enable Dependencies: when the following conditions are true, the intrq output signal is asserted: • This bit=1 • SATA_GHC[IE]=1 • SATA_P 0 IS[HBDS]=1
27 IFE	Dependencies: when the following conditions are true, the intrq output signal is asserted: • This bit=1 • SATA_GHC[IE]=1 • SATA_P 0 IS[IFS]=1
26 INFE	Interface Non-Fatal Error Enable Dependencies: when the following conditions are true, the intrq output signal is asserted: • This bit=1 • SATA_GHC[IE]=1 • SATA_P 0 IS[INFS]=1
25 -	Reserved
24 OFE	Overflow Enable Dependencies: when the following conditions are true, the intrq output signal is asserted: • This bit=1 • SATA_GHC[IE]=1 • SATA_P 0 IS[OFS]=1
23 IPME	Incorrect Port Multiplier Enable Dependencies: when the following conditions are true, the intrq output signal is asserted: • This bit=1 • SATA_GHC[IE]=1 • SATA_P 0 IS[IPMS]=1
22 PRCE	PHY Ready Change Enable Dependencies: when the following conditions are true, the intrq output signal is asserted: • This bit=1 • SATA_GHC[IE]=1 • SATA_P 0 IS[PRCS]=1
21–8 -	Reserved.
7 Reserved	This read-only field is reserved and always has the value zero. Reserved Returns 0 on .read.
6 PCE	Port Change Interrupt Enable Dependencies: when the following conditions are true, the intrq output signal is asserted:

Table continues on the next page...

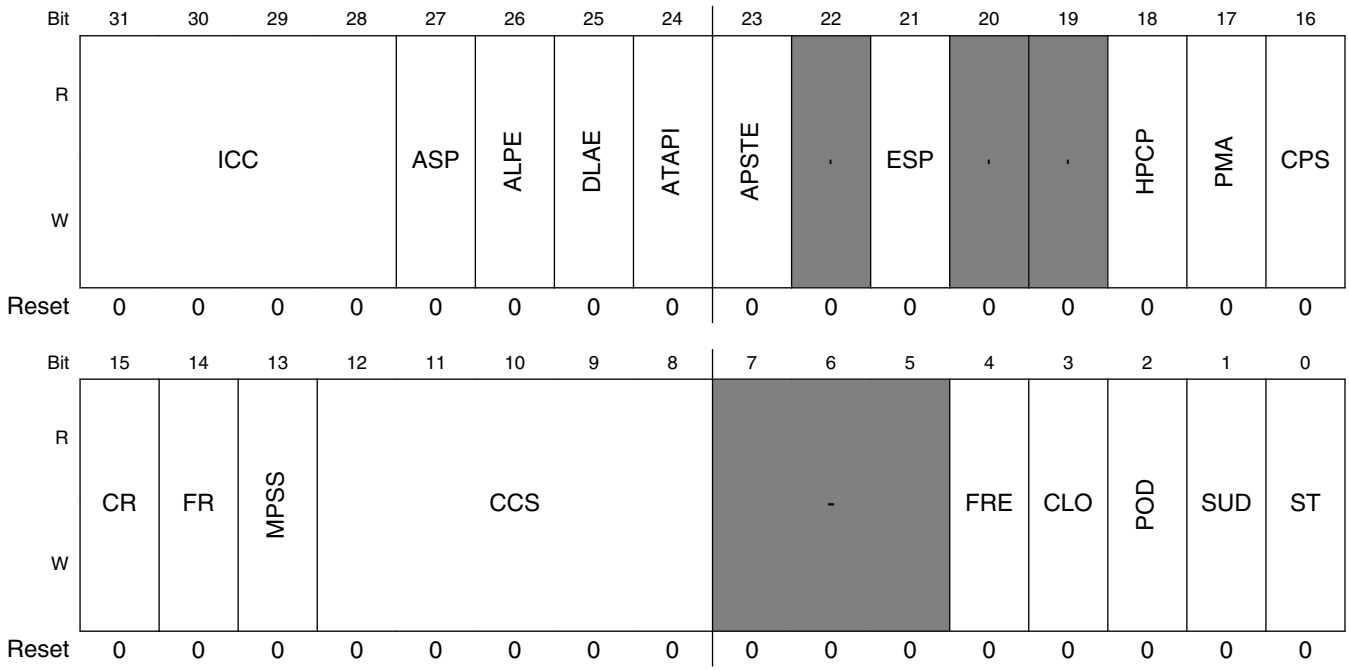
SATA_P0IE field descriptions (continued)

Field	Description
	• This bit=1 • SATA_GHC[IE]=1 • SATA_P 0 IS[PCS]=1
5 DPE	Descriptor Processed Interrupt Enable Dependencies: when the following conditions are true, the intrq output signal is asserted: • This bit=1 • SATA_GHC[IE]=1 • SATA_P 0 IS[DPS]=1
4 UFE	Unknown FIS Interrupt Enable Dependencies: when the following conditions are true, the intrq output signal is asserted: • This bit=1 • SATA_GHC[IE]=1 • SATA_P 0 IS[UFS]=1
3 SDBE	Set Device Bits FIS Interrupt Enable Dependencies: when the following conditions are true, the intrq output signal is asserted: • This bit=1 • SATA_GHC[IE]=1 • SATA_P 0 IS[SDBS]=1
2 DSE	DMA Setup FIS Interrupt Enable Dependencies: when the following conditions are true, the intrq output signal is asserted: • This bit=1 • SATA_GHC[IE]=1 • SATA_P 0 IS[DSS]=1
1 PSE	PIO Setup FIS Interrupt Enable Dependencies: when the following conditions are true, the intrq output signal is asserted: • This bit=1 • SATA_GHC[IE]=1 • SATA_P 0 IS[PSS]=1
0 DHRE	Device to Host Register FIS Interrupt Dependencies: when the following conditions are true, the intrq output signal is asserted: • This bit=1 • SATA_GHC[IE]=1 • SATA_P 0 IS[DHRS]=1

63.7.26 Port0 Command Register (SATA_P0CMD)

This register contains bits controlling various Port functions. All RW bits are reset on Global reset.

Address: SATA_P0CMD is 1000_0000h base + 118h offset = 1000_0118h



SATA_P0CMD field descriptions

Field	Description
31–28 ICC	Interface Communication Control This field is used to control power management states of the interface. When the Link layer is currently in the L_IDLE state, writes to this field cause the Port to initiate a transition to the interface power management state requested. When the Link layer is not currently in the L_IDLE state, writes to this field have no effect. • 0xF-0x7: Reserved • 0x6: Slumber. This causes the Port to request a transition of the interface to the Slumber state. The SATA device can reject the request and the interface remains in its current state. • 0x5-0x3: Reserved • 0x2: Partial. This causes the Port to request a transition of the interface to the Partial state. The SATA device can reject the request and the interface remains in its current state. • 0x1: Active. This causes the Port to request a transition of the interface into the active state. • 0x0: No-Op/ Idle. This value indicates to the software that the Port 0 is ready to accept a new interface control command, although the transition to the previously selected state may not yet have occurred. When the software writes a non-reserved value other than No-Op (0x0), the Port performs the action and update this field back to Idle (0x0).

Table continues on the next page...

SATA_P0CMD field descriptions (continued)

Field	Description
	When the software writes to this field to change the state to a state the link is already in (i.e., interface is in the active state and a request is made to go to the active state), the Port takes no action and returns this field to Idle. When the interface is in a low power state and the software wants to transition to a different low power state, the software must first bring the link to active and then initiate the transition to the desired low power state.
27 ASP	Aggressive Slumber/ Partial The options for this field are: • When set to 1, and SATA_P 0 CMD[ALPE]=1, the Port aggressively enters the SLUMBER state when one of the following conditions is true: - The Port clears the SATA_P 0 CI and the SATA_P 0 SACT register is cleared. - The Port clears the SATA_P 0 SACT register and SATA_P 0 CI is cleared. • When cleared to 0, and SATA_P 0 CMD[ALPE]=1, the Port aggressively enters the PARTIAL state when one of the following conditions is true: - The Port clears the SATA_P 0 CI register and the SATA_P 0 SACT register is cleared. - The Port clears the SATA_P 0 SACT register and SATA_P 0 CI is cleared.
26 ALPE	Aggressive Link Power Management Enable When set to 1, the Port aggressively enters a lower link power state (PARTIAL or SLUMBER) based on the setting of the SATA_P 0 CMD[ASP] bit. When cleared to 0, aggressive power management state transition is disabled.
25 DLAE	Drive LED on ATAPI Enable When set to 1, SATA_P 0 CMD[ATAPI]=1, and commands are active, the Port asserts p 0 _act_led output.
24 ATAPI	ATAPI Device is ATAPI This bit is used by the Port to control whether to assert p 0 _act_led output when commands are active. The options for this field are: 0 non-ATAPI device 1 ATAPI device
23 APSTE	Device is ATAPI This bit is used by the Port to control whether to assert p 0 _act_led output when commands are active. The options for this field are: 0 non-ATAPI device 1 ATAPI device
22 -	Reserved
21 ESP	External SATA Port When set to 1, indicates that this Port's signal only connector is externally accessible. When set to 1, SATA_CAP[SXS] is also set to 1. When cleared to 0, indicates that this Port's signal only connector is not externally accessible. Note: The ESP bit is mutually exclusive with SATA_P 0 CMD[HPCP]

Table continues on the next page...

SATA_P0CMD field descriptions (continued)

Field	Description
20 -	Reserved. Returns 0 on read.
19 -	Reserved. Returns 0 on read.
18 HPCP	Hot Plug Capable Port **NOTE:** The HPCP bit is mutually exclusive with SATA_P 0 CMD[ESP]. The options for this field are: 1 Indicates that this Port's signal and power connectors are externally accessible via a joint signal-power connector for blindmate device hot plug. 0 Indicates that this Port's signal and power connectors are not externally accessible.
17 PMA	Port Multiplier Attached The software is responsible for detecting whether a Port Multiplier is present; the SATA block Port does not auto-detect the presence of a Port Multiplier. The options for this field are: 1 A Port Multiplier is attached to this Port. 0 A Port Multiplier is not attached to this Port.
16 CPS	Cold Presence State This bit reports whether a device is currently detected on this Port as indicated by the p 0 _cp_det input state (assuming SATA_P 0 CMD[CPD]=1). The options for this field are: 1 device is attached to this Port 0 no device attached to this Port
15 CR	Command List Running When this bit is set to '1', the command list DMA engine for this Port is running. See AHCI state machine in AHCI specification section 5.3.2 for details on when this bit is set and cleared by the Port.
14 FR	FIS Receive Running When set to '1', the FIS Receive DMA engine for the Port is running. See AHCI specification section 10.3.2 for details on when this bit is set and cleared by the Port.
13 MPSS	Mechanical Presence Switch State The software must use this bit only when both SATA_CAP[SMPS] and SATA_P 0 CMD[MPSP] are set. This bit reports the state of a mechanical presence switch attached to this Port as indicated by the p 0 _mp_switch input state (assuming SATA_CAP[SMPS]=1 and SATA_P 0 CMD[MPSP]=1). The options for this field are: When SATA_CAP[SMPS]=0 then this bit is cleared to 0. 0 Switch is closed 1 Switch is open
12–8 CCS	Current Command Slot This field is set to the command slot value value of the command that is currently being issued by the Port.

Table continues on the next page...

SATA_P0CMD field descriptions (continued)

Field	Description
	• When SATA_P 0 CMD[ST] transitions from 1 to 0, this field is recleared to 0x00. • After SATA_P 0 CMD[ST] transitions from 0 to 1, the highest priority slot to issue from next is command slot 0. After the first command has been issued, the highest priority slot to issue from next is SATA_P 0 CMD[CCS]+1.For example, after the Port has issued its first command, when CCS=0x00 and SATA_P 0 CI is cleared to 0x3, the next command issued is from command slot 1. This field is valid only when SATA_P 0 CMD[ST] is set to 1.
7–5 -	Reserved.
4 FRE	FIS Receive Enable When set to 1, the Port may post received FISes into the FIS receive area pointed to by SATA_P 0 FB . When cleared, received FISes are not accepted by the Port, except for the first D2H register FIS after the initialization sequence, and no FISes are posted to the FIS receive area. The software must not set this bit until SATA_P 0 FB has been programmed with a valid pointer to the FIS receive area When the software wishes to move the base, this bit must first be cleared, and the software must wait for the SATA_P 0 CMD[FR] bit to be cleared.
3 CLO	Command List Override Setting this bit to 1 causes the SATA_P 0 TFD[STS] field BSY bit and the SATA_P 0 TFD[STS] field DRQ bit to be cleared. This allows a software reset to be transmitted to the device regardless of whether the BSY and DRQ bits are still set in the SATA_P 0 TFD[STS] field. This bit is cleared to 0 when SATA_P 0 TFD[STS] BSY bit and SATA_P 0 TFD[STS] DRQ bit have been cleared. A write to this register with a value of '0' has no effect. This bit should only be set to 1 immediately prior to setting SATA_P 0 CMD[ST] bit to 1 from a previous value of 0. Setting this bit to 1 at any other time is not supported and results in indeterminate behavior.
2 POD	Power On Device This bit is read/write when cold presence detection is supported on this Port as indicated by SATA_P 0 CMD[CPD]=1. This bit is read-only 1 when cold presence detection is not supported and SATA_P 0 CMD[CPD]=0. When set, the Port asserts the p 0 _cp_pod output pin so that it may be used to provide power to a cold-presence detectable Port.
1 SUD	Spin-Up Device This bit is read/write when staggered spin-up is supported as indicated by the SATA_CAP[SSS]=1. This bit is read-only 1 when staggered spin-up is not supported and SATA_CAP[SSS]=0. On an edge detect from 0 to 1, the Port starts a COMRESET initialization sequence to the device. Clearing this bit causes no action on the interface. Note: The SUD bit is read-only 0 on power-up until SATA_CAP[SSS] bit is written with the required value.
0 ST	Start When set to 1, the Port processes the command list. When cleared, the Port does not process the command list. Whenever this bit is changed from a 0 to a 1, the Port starts processing the command list at entry'0. Whenever this bit is changed from a 1 to a 0, the SATA_P 0 CI register is cleared by the Port upon transition into an idle state. Refer to AHCI specification, section 10.3.1, for important restrictions on when this bit can be set to 1. Note: SATA_P 0 SERR register must be cleared prior to setting ST bit to 1.

63.7.27 Port0 Task File Data Register (SATA_P0TFD)

This register contains Error and Status registers updated every time a new Register FIS, PIO Setup FIS, or Set Device Bits FIS is received from the device. Reset on Global or Port reset (COMRESET).

Address: SATA_P0TFD is 1000_0000h base + 120h offset = 1000_0120h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																ERR								STS							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

SATA_P0TFD field descriptions

Field	Description
31–16 -	Reserved
15–8 ERR	Error This field contains the latest copy of the task file error register.
7–0 STS	Status This field contains the latest copy of the task file status register. The bits that affect SATA block operation are: • Bit [7] BSY - Indicates the interface is busy • Bits [6:4] cs - Command specific • Bit [3] DRQ - Indicates a data transfer is requested • Bits [2:1] cs - Command specific • Bit [0] ERR - Indicates an error during the transfer **NOTE:** The Port updates the entire 8-bit field, not just the bits noted above.

63.7.28 Port0 Signature Register (SATA_P0SIG)

Address: SATA_P0SIG is 1000_0000h base + 124h offset = 1000_0124h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SIG																															
W																																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

SATA_P0SIG field descriptions

Field	Description
31–0 SIG	Signature This field contains the signature received from a device on the first D2H Register FIS. The bit order as follows: • Bits [31:24] - LBA High (Cylinder High) Register • Bits [23:16] - LBA Mid (Cylinder Low) Register • Bits [15:8] - LBA Low (Sector Number) Register • Bits [7:0] - Sector Count Register This field is updated once after a reset sequence. Reset on Global or Port reset.

63.7.29 Port0 Serial ATA Status Register (SATA_P0SSTS)

This 32-bit register conveys the current state of the interface and host. The Port updates it continuously and asynchronously. When the Port transmits a COMRESET to the device, this register is updated to its reset values (i.e., Global reset, Port reset, or COMINIT from the device

Address: SATA_P0SSTS is 1000_0000h base + 128h offset = 1000_0128h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																				IPM				SPD				-			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SATA_P0SSTS field descriptions

Field	Description
31–12 -	Reserved
11–8 IPM	Interface Power Management Indicates the current interface state. The options for this field are: 0x0 Device not present or communication not established 0x1 Interface in active state 0x2 Interface in Partial power management state 0x6 Interface in Slumber power management state All other values Reserved
7–4 SPD	Current Interface Speed Indicates the negotiated interface communication speed. The options for this field are: 0x0 Device not present or communication not established 0x1 1.5 Gb/s communication rate negotiated All other values Reserved and should not be used

Table continues on the next page...

SATA_P0SSTS field descriptions (continued)

Field	Description
3–0 -	Reserved

63.7.30 Port0 Serial ATA Control {SControl} Register (SATA_P0SCTL)

This 32-bit read-write register is used by the software to control SATA interface capabilities. Writes to this register result in an action being taken by the Port PHY interface. Reads from the register return the last value written to it. Reset on Global reset.

These bits are static and should not be changed frequently due to the clock crossing between the Transport and Link Layers. The software must wait for at least seven periods of the slower clock (clk_asic 0 or hclk) before changing this register

Address: SATA_P0SCTL is 1000_0000h base + 12Ch offset = 1000_012Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-																				IPM				SPD				DET			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SATA_P0SCTL field descriptions

Field	Description
31–12 -	Reserved
11–8 IPM	Interface Power Management Transitions Allowed This field indicates which power states the Port PHY interface is allowed to transition to. When an interface power management state is disabled, the Port does not initiate that state and any request from the device to enter that state is rejected via PMNAKp The options for this field are: 0x0 No interface power management state restrictions 0x1 Transitions to the Partial state disabled 0x2 Transitions to the Slumber state disabled 0x3 Transitions to both Partial and Slumber states disabled All other values. Reserved and should not be used
7–4 SPD	Speed Allowed This field indicates the highest allowable speed of the Port PHY interface. The options for this field are: **NOTE:** When the host software must change this field value, the host must also reset the Port (SATA_P 0 SCTL[DET] = 0x1) at the same time to ensure proper speed negotiation. 0x0 No speed negotiation restrictions

Table continues on the next page...

SATA_P0SCTL field descriptions (continued)

Field	Description
	0x1 Limit speed negotiation to SATA 1.5 Gb/s communication rate All other values Reserved and should not be used.
3–0 DET	Device Detection Initialization Controls the Port's device detection and interface initialization. The options for this field are: **NOTE:** This field may only be modified when SATA_P 0 CMD[ST] is 0. Changing this field while the SATA_P 0 CMD[ST]=1 results in undefined behavior. When SATA_P 0 CMD[ST] is set to 1, this field should have a value of 0x0. 0x0 No device detection or initialization action requested 0x1 Perform interface initialization sequence to establish communication. This results in the interface being reset and communication re initialized. 0x4 Disable the Serial ATA interface and put the Port PHY in offline mode. All other values reserved.

63.7.31 Port0 Serial ATA Error Register (SATA_P0SERR)

This 32-bit register represents all the detected interface errors accumulated since the last time it was cleared. The set bits in the SError register indicate that the corresponding error condition became true one or more times since the last time the bit was cleared. The set bits in this register are explicitly cleared by a write operation to the register, Global reset, or Port reset (COMRESET). The value written to clear the set error bits should have ones encoded in the bit positions corresponding to the bits that are to be cleared. All bits in the following table have a reset value of 0.

Address: SATA_P0SERR is 1000_0000h base + 130h offset = 1000_0130h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-					DIAG_X	DIAG_F	DIAG_T	DIAG_S	DIAG_H	DIAG_C	DIAG_D	DIAG_B	DIAG_W	DIAG_I	DIAG_N
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-				ERR_E	ERR_P	ERR_C	ERR_T	-						ERR_M	ERR_I
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SATA_P0SERR field descriptions

Field	Description
31–27 -	Reserved
26 DIAG_X	Exchanged This bit is set to 1 when PHY COMINIT signal is detected. This bit is reflected in the SATA_P 0 IS[PCS] bit.
25 DIAG_F	Unknown FIS Type This bit indicates that one or more FISes were received by the Transport layer with good CRC, but had a type field that was not recognized/known and the length was less than or equal to 64bytes. **NOTE:** When the Unknown FIS length exceeds 64 bytes, the DIAG_F bit is not set and the DIAG_T bit is set instead.
24 DIAG_T	Transport State Transition Error This bit indicates that a Transport Layer protocol violation was detected since the last time this bit was cleared. See Transport Check (TCHK) for details.
23 DIAG_S	Link Sequence Error This bit indicates that one or more Link state machine error conditions was encountered. One of the conditions that cause this bit to be set is device doing SYNC escape during FIS transmission.
22 DIAG_H	Handshake Error This bit indicates that one or more R_ERRp was received in response to frame transmission. Such errors may be the result of a CRC error detected by the device, a disparity or 8b/10b decoding error, or other error condition leading to a negative handshake on a transmitted frame.
21 DIAG_C	CRC Error
20 DIAG_D	Disparity Error This bit is always cleared to 0 since it is not used by the AHCI specification.
19 DIAG_B	10B to 8B Decode Error This bit indicates errors were detected by 10b8b decoder. This bit indicates that one or more CRC errors were detected by the Link layer during FIS reception. **NOTE:** This bit is set only when an error is detected on the received FIS data dword. This bit is not set when an error is detected on the primitive, regardless whether it is inside or outside the FIS.
18 DIAG_W	Comm Wake This bit is set when PHY COMWAKE signal is detected.
17 DIAG_I	PHY Internal Error This bit is set when the PHY detects some internal error as indicated by the assertion of the p 0 _phy_rx_err input. **NOTE:** The setting of this bit is controlled by the SATA_BISTCR[ERREN] bit: when ERREN==0 (default), only errors occurring inside the received FIS cause DIAG_I bit to be set; when ERREN==1, any error inside or outside the FIS causes the DIAG_I bit to be set.
16 DIAG_N	PHY Ready Change This bit indicates that the PHY Ready signal changed state. This bit is reflected in the SATA_P 0 IS[PRCS] bit.

Table continues on the next page...

SATA_P0SERR field descriptions (continued)

Field	Description
15–12 -	Reserved
11 ERR_E	Internal Error This bit is set to 1 when one or more AHB bus ERROR responses are detected on the master interface.
10 ERR_P	Protocol Error This bit is set to 1 when any of the following conditions are detected. • Transport state transition error (DIAG_T) • Link sequence error (DIAG_S) • RxFIFO overflow • Link bad end error (WTRM instead of EOF is received).
9 ERR_C	Non-Recovered Persistent Communication Error This bit is set to 1 when PHY Ready signal is negated due to the loss of communication with the device or problems with interface, but not after transition from active to Partial or Slumber power management state.
8 ERR_T	Non-Recovered Transient Data Integrity Error This bit is set when any of the following SATA_P 0 SERR register bits is set during Data FIS transfer: ERR_P (Protocol) • DIAG_C (CRC) • DIAG_H (Handshake) • ERR_C ("PHY Ready" negation)
7–2 -	Reserved
1 ERR_M	Recovered Communication Error This bit is set to 1 when PHY Ready condition is detected after interface initialization, but not after transition from Partial or Slumber power management state to active state.
0 ERR_I	This bit is set when any of the following SATA_P 0 SERR register bits is set during non- Data FIS transfer: • DIAG_C (CRC) • DIAG_H (Handshake) • ERR_C ("PHY Ready" negation)

63.7.32 Port0 Serial ATA Active Register (SATA_P0SACT)

Address: SATA_P0SACT is 1000_0000h base + 134h offset = 1000_0134h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SATA_P0SACT field descriptions

Field	Description
31–0 DS	Device Status This field is bit significant. Each bit corresponds to the TAG and command slot of a native queued command, where bit 0 corresponds to TAG 0 and command slot 0. Software sets this field prior to issuing a native queued command for a particular command slot. Prior to writing SATA_P 0 CI[TAG] to 1, the software sets DS[TAG] to 1 to indicate that a command with that TAG is outstanding. This field is cleared to 0 when: • The software writes SATA_P 0 CMD[ST from a 1 to a 0 . • The device sends a Set Device Bits FIS to the Port. The Port clears bits in this field that are set in the SActive field of the Set Device Bits FIS. The Port clears only bits that correspond to native queued commands that have completed successfully. This field is not cleared by the following: • Port reset (COMRESET). • Software reset. **NOTE:** Software must write this field only when SATA_P 0 CMD[ST] bit is set to 1.

63.7.33 Port0 Command Issue Register (SATA_P0CI)

Address: SATA_P0CI is 1000_0000h base + 138h offset = 1000_0138h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CI																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SATA_P0CI field descriptions

Field	Description
31–0 CI	Command Issued This field is bit significant. Each bit corresponds to a command slot, where bit 0 corresponds to command slot 0. This field is set by the software to indicate to the Port that a command has been built in system memory for a command slot and may be sent to the device. When the Port receives a FIS which clears the BSY, DRQ, and ERR bits for the command, it clears the corresponding bit in this register for that command slot. Bits in this field can only be set to 1 by the software when SATA_P 0 CMD[ST] is set to 1. **NOTE:** This field is reset when SATA_P 0 CMD[ST] is written from a 1 to a 0 by the software.

63.7.34 Port0 Serial ATA Notification Register (SATA_P0SNTF)

This register is used to determine when asynchronous notification events have occurred for directly connected devices and devices connected to a Port Multiplier.

Address: SATA_P0SNTF is 1000_0000h base + 13Ch offset = 1000_013Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																PMN															
W																	w1c															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SATA_P0SNTF field descriptions

Field	Description
31–16 -	Reserved
15–0 PMN	PM Notify This field indicates whether a particular device with the corresponding PM Port number issued a Set Device Bits FIS to the SATA block Port with the Notification bit set: • PM Port 0h sets bit 0, • PM Port 1h sets bit 1, ... • PM Port Fh sets bit 15. Individual bits are cleared by the software writing 1s to the corresponding bit positions. This field is reset on Global reset, but it is not reset by Port reset (COMRESET) or software reset.

63.7.35 Port0 DMA Control Register (SATA_P0DMACR)

This register contains bits for controlling the Port DMA engine. The software can change the fields of this register only when SATA_P 0 CMD[ST]=0. Power-up (system reset), Global reset, or Port reset (COMRESET) reset this register to the default value.

Address: SATA_P0DMACR is 1000_0000h base + 170h offset = 1000_0170h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																RXABL				TXABL				RXTS				TXTS			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0

SATA_P0DMACR field descriptions

Field	Description
31–16 -	Reserved
15–12 RXABL	Receive AHB Burst Limit This field allows the software to limit the AHB master write burst size. The options for this field are: **NOTE:** SATA block AHB master breaks the burst at 1-KB address boundary regardless of the RXABL value. **NOTE:** This field is read-write when SATA_P 0 CMD[ST]=0 and read-only when SATA_P 0 CMD[ST]=1. 0x0, 0x9-0xF Limit AHB burst size to 256 DWORDs 0x1 Limit AHB burst size to 1 DWORD 0x2 Limit AHB burst size to 2 DWORDs 0x3 Limit AHB burst size to 4 DWORDs 0x4 Limit AHB burst size to 8 DWORDs 0x5 Limit AHB burst size to 16 DWORDs 0x6 Limit AHB burst size to 32 DWORDs 0x7 Limit AHB burst size to 64 DWORDs 0x8 Limit AHB burst size to 128 DWORDs
11–8 TXABL	Transmit AHB Burst Limit This field allows the software to limit the AHB master read burst size. The options for this field are: **NOTE:** SATA block AHB master breaks the burst at 1-KB address boundary regardless of the TXABL value. **NOTE:** This field is read-write when SATA_P 0 CMD[ST]=0 and read-only when SATA_P 0 CMD[ST]=1. 0x0, 0x9-0xF Limit AHB burst size to 256 DWORDs 0x1 Limit AHB burst size to 1 DWORD 0x2 Limit AHB burst size to 2 DWORDs 0x3 Limit AHB burst size to 4 DWORDs 0x4 Limit AHB burst size to 8 DWORDs 0x5 Limit AHB burst size to 16 DWORDs 0x6 Limit AHB burst size to 32 DWORDs 0x7 Limit AHB burst size to 64 DWORDs 0x8 Limit AHB burst size to 128 DWORDs
7–4 RXTS	Receive Transaction Size This field defines the Port DMA transaction size in DWORDs for receive (system bus write, device read) operation. This field is read-write when SATA_P 0 CMD[ST]=0 and read-only when SATA_P 0 CMD[ST]=1. The maximum value of this field is determined by the RxFIFO depth parameter. When the software attempts to write a value exceeding this value, the maximum value would be set instead. 0x0 1 DWORD 0x1 2 DWORD 0x2 4 DWORD 0x3 8 DWORD 0x4 16 DWORDs (maximum value when RXFIFO_DEPTH=64)

Table continues on the next page...

SATA_P0DMACR field descriptions (continued)

Field	Description
	0x5 32 DWORD 0x6 64 DWORDs (maximum value when RXFIFO_DEPTH=128) 0x7 128 DWORDs (maximum value when RXFIFO_DEPTH=256) 0x8 256 DWORDs (maximum value when RXFIFO_DEPTH=512) 0x9 12 DWORDs (maximum value when RXFIFO_DEPTH=1024) 0xA 1024 DWORDs (maximum value when RXFIFO_DEPTH=2048) All other values are reserved and should not be used.
3–0 TXTS	Transmit Transaction Size This field defines the DMA transaction size in DWORDs for transmit (system bus read, device write) operation. The options for this field are: This field is read-write when SATA_P 0 CMD[ST]=0 and read-only when SATA_P 0 CMD[ST]=1. The maximum value of this field is determined by the TxFIFO depth parameter. When the software attempts to write a value exceeding this value, the maximum value would be set instead. 0x0 1 DWORD 0x1 2 DWORD 0x2 4 DWORD 0x3 8 DWORD 0x4 16 DWORDs (maximum value when TXFIFO_DEPTH=32) 0x5 32 DWORDs (maximum value when TXFIFO_DEPTH=64) 0x6 64 DWORDs (maximum value when TXFIFO_DEPTH=128) 0x7 128 DWORDs (maximum value when TXFIFO_DEPTH=256) 0x8 256 DWORDs (maximum value when TXFIFO_DEPTH=512) 0x9 512 DWORDs (maximum value when TXFIFO_DEPTH=1024) 0xA 1024 DWORDs (maximum value when TXFIFO_DEPTH=2048) All other values are reserved and should not be used.

63.7.36 Port0 PHY Control Register (SATA_P0PHYCR)

This register is used for Port PHY control.

Bits of this register are connected to the corresponding bits of the p 0 _phy_ctrl output Port.

NOTE

The SATA_P 0 PHYCR register supports only 32-bit write access

Address: SATA_P0PHYCR is 1000_0000h base + 178h offset = 1000_0178h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-											TEST_PDDQ	CR_READ	CR_WRITE	CR_CAP_DATA	CR_CAP_ADDR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CR_DATA_IN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SATA_P0PHYCR field descriptions

Field	Description
31–21 -	Reserved
20 TEST_PDDQ	Test IDDQ
19 CR_READ	CR Read. Reads from the referenced Address register.
18 CR_WRITE	CR Write. Writes the Write Data register to the referenced Address register.
17 CR_CAP_DATA	CR Capture Data. Captures phy_cr_data_in[15:0] into the Write Data register.
16 CR_CAP_ADDR	CR Capture Address. Captures phy_cr_data_in[15:0] into the Address register.
15–0 CR_DATA_IN	CR Address and Write Data Input Bus. Supplies and captures address and write data.

63.7.37 Port0 PHY Status Register (SATA_P0PHYSR)

This register is used to monitor PHY status.

The bits of this register reflect the state of the corresponding bits of the p 0 _phy_status input.

Signals connected to the p 0 _phy_status input can be asynchronous to any of the SATA block clocks, however they must not change faster than five hclk periods, otherwise the SATA_P 0 PHYSR register may never be updated with the intermediate changing values.

Address: SATA_P0PHYSR is 1000_0000h base + 17Ch offset = 1000_017Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													CR_ ACK	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CR_DATA_OUT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SATA_P0PHYSR field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero.
18 CR_ACK	CR Acknowledgement. Acknowledgement for the phy_cr_cap_addr, phy_cr_cap_data, phy_cr_write, and phy_cr_read control signals.
17–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 CR_DATA_OUT	CR Data Output Bus. Always presents last read data.

Chapter 64
Serial Advanced Technology Attachment PHY (SATA PHY)

64.1 Overview

The Serial-ATA PHY is an ultra low-power SATA physical layer that complies with *Serial ATA*, Revision 2.5.

64.1.1 General Product Description

The SATA2 PHY is a complete mixed-signal IP solution designed to implement SATA connectivity in a System-on-Chip (SoC) design targeted to a specific fabrication process.

The SATA2 PHY provides a unique combination of:

- Small area
- Low power
- High performance
- Testability and characterization (diagnostic) features

64.1.1.1 System Overview

Each SATA2 PHY lane takes a 10- or 20-bit input and produces a serial output at 10 or 20 times the input word rate, respectively.

The transmitted data is synchronous to the local refclk. The receiver is clock- forwarded (source-synchronous) and the received data and its accompanying clock are synchronous to refclk at the far end of the link. Synchronicity with refclk depends on the system application.

64.1.2 Features

The SATA2 PHY provides the following features:

- Data rates of 1.5/3.0 Gbps, selectable per transceiver
- Clock module:
- Pin-programmable MPLL bandwidth from 2.5-10 MHz in 1.25-MHz increments
- Selectable use of an alternative reference clock delivered from the ASIC
- Flexible, glitchless clock outputs to ASIC: word rate, 1/2 word rate, and keep-alive clock options buffered from RefClk
- Power on or off and suspension of RefClk
- Flexible baud rate: RefClk ratio. Includes all integers greater than 3 (excluding 6, 7, and 11) and less than 67, and all even integers up to and including 130.
- RefClk frequencies of 25 MHz or 50-156.25 MHz. Common RefClk frequencies of 100 MHz, 125 MHz, and 156.25 MHz are supported.
- Generation of low jitter spread-spectrum clock (0.5% downspread at 31.5 KHz)
- Transmit path:
- Multiple power-down modes
- Differential Tx amplitude with less than ± 10% variation across temperature, voltage, and process
- Pin-programmable transmit level
- Pin-programmable attenuation of {8, 9, 10, 12, 14, 16} / 16 of full scale
- Pin-programmable Tx boost of 0-5.75 dB in increments of ~ .37 dB
- 10- or 20-bit wide ASIC interface
- Per-lane word clock (in addition to word clock from MPLL) for maximum flexibility
- Out of Band (OOB) signaling
- Latency of 22-24 bit times from when the 10-bit data is sampled to when the first bit of the word is transmitted serially
- Receive path:
- Pin-programmable equalization of .5-4 dB in .5-dB increments
- Programmable Rx clock data recovery (CDR) based on a digital PLL (DPLL) for robust operation despite independently spread-spectrum references
- Rx CDR that can tolerate run lengths of thousands of bits
- Pin-programmable loss of signal (LOS) threshold
- Pin-programmable ACJTAG hysteresis level
- Multiple power-down modes
- Rx bandwidth of 2.5 GHz
- 10- or 20-bit wide ASIC interface
- Clock forwarding
- OOB signaling

- Latency of 22-31 bit times from when the last bit of a 10-bit word is received to when the 10-bit word is available on the ASIC interface
- Tx and Rx termination impedances that are automatically calibrated to match the external reference resistance

Test Features

The SATA2 PHY provides the following test features.

- Integrated test features:
- Combinatorial loopback for full testing of the ASIC/IP interface with the ASIC's native testing methodology
- Burn-in mode to toggle most internal modes without using clocks or controls
- IDDQ test mode
- Loopback:
- Tx-to-Rx serial analog loopback for wafer probe only
- Tx-to-Rx serial digital loopback
- Byte Error Rate Testing (BERT) independent per lane:
- 7th- and 15th-order polynomial pattern generation and recognition
- Generation of simple test patterns
- Byte error counting from polynomial patterns or simple test patterns
- Voltage margining with 10-bit resolution, synchronous or asynchronous operation
- Phase margining with UI/512 resolution, synchronous or asynchronous (when number of lanes is greater than one) operation
- Combined 2D margining
- High resolution scope per Rx signal pair:
- Acquisition of eye from patterns of known periodicity
- Acquisition and analysis of signals of modest periodicities (< 1,024 bytes)
- Analog DC testing:
- 10-bit A/D converter
- Selection and measurement of any individual Rx and Tx termination resistor
- Limit testing that enables vector-only tests, which pass results in a non-trivial limit range:
- High/low limits
- Whether register read is within limits
- Whether difference between register readback values is within limits

64.2 Architecture

64.2.1 Block Diagram

The figure below shows a high-level SATA2 PHY block diagram.

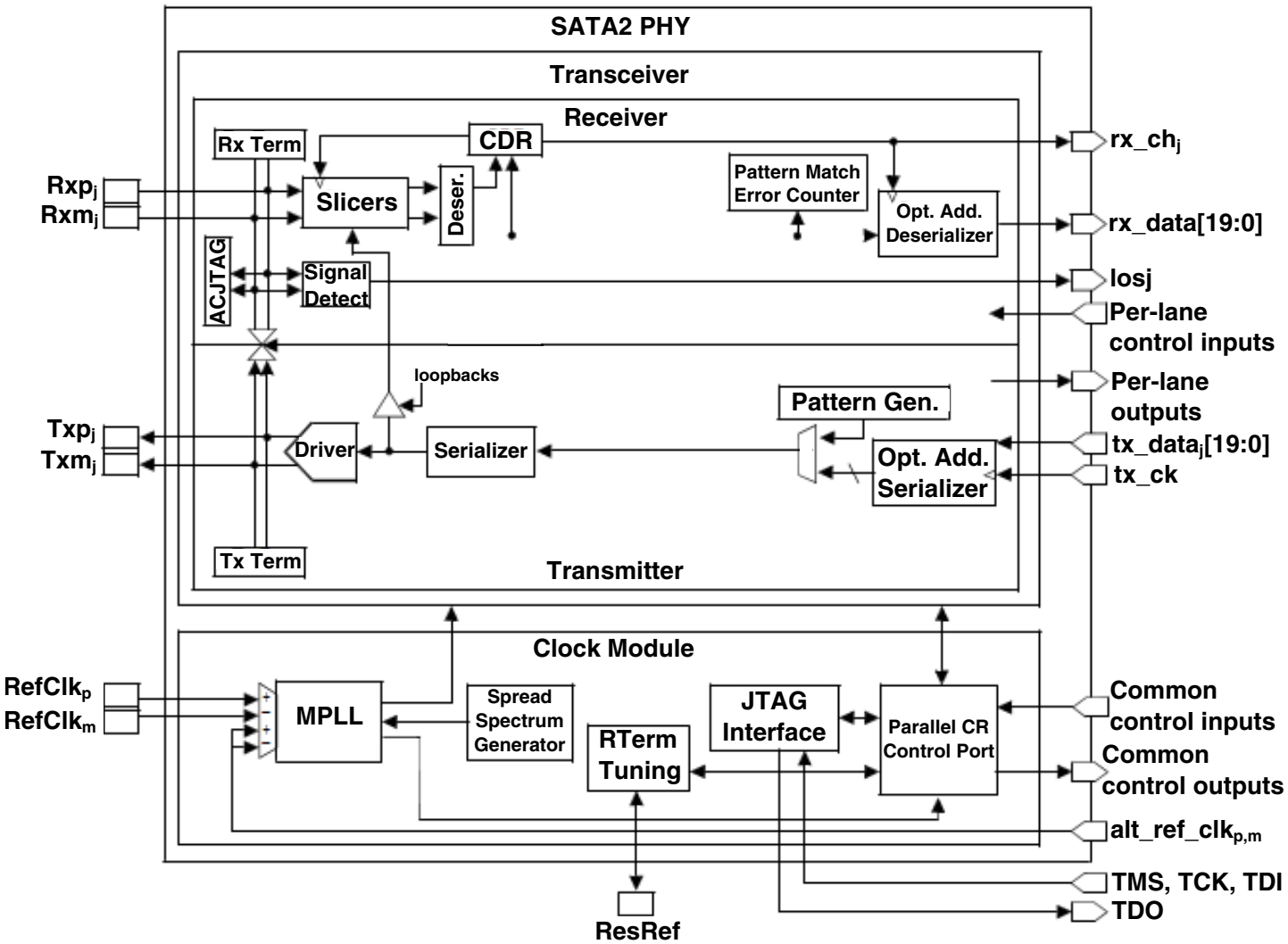


Figure 64-1. SATA2 PHY Block Diagram

64.2.2 Block Descriptions

This section describes blocks in the SATA2 PHY.

Transceiver Block - The Transceiver block is replicated per lane. This section describes blocks in the transceiver.

Receiver Block - This block converts the incoming, high-speed differential serial link to 10- or 20-bit parallel data and recovers the clock.

- **Rx Termination Block** - This block produces single-ended termination impedances of ~50 ohms to ground, differential ~ 100 ohms.

- **Rx Signal Detect** - This block detects the presence of a signal on the Rx signal pair. In a SATA implementation, this block detects OOB signals.

- **ACJTAG (Boundary Scan) Interface** - This block contains the logic for doing industry-standard (1149.1-2001 and 1149.6-2003) connectivity testing.

- **Rx Slicers** - This block converts the receive signal from a high-bandwidth analog signal to a sequence of 1s and 0s sampled when directed by the Clock and Data Recovery (CDR) block.

- **Deserializer** - The deserializer converts the output of the slicers to a 10-bit wide (1 coded byte) format. The deserializer uses COMMA characters to select the correct word alignment.

- **Clock and Data Recovery** - The CDR block comprises both a digital PLL (DPLL) and an analog PLL (APLL). The DPLL comprises phase detectors and a digital loop filter, which combine to produce a selected output phase. The APLL generates and filters the clock phase selected by the DPLL. The generated output clock feeds the slicers with multiple phases, while another output is divided down to become the word rate (baud rate / 10) recovered clock.

- **Rx Pattern Match and Error Count** - To identify errors during characterization of the channel quality, this block determines whether each byte of the received data sequence matches one of a few pseudo-random sequences. Detected byte errors are counted and available in the Pattern Matcher Control register (see lanej Memory Map/Register Definition).

- **Optional Additional Deserializer** - When enabled by the assertion of the wide_xface control, this block doubles the Rx datapath width (from 10 bits to 20 bits) and reduces the Rx_Ck rate to baud_rate/ 20. When wide_xface is deasserted, only bits [9:0] of the RxData bus are used; Rx_Ck runs at baud_rate / 10. When the half_rate signal is asserted, clock frequencies are halved, but still based on the true baud rate as previously described.

Transmitter Block - This block converts outgoing, parallel 10- or 20-bit data to a high-speed, differential serial link.

- **Tx Termination Block** - This block produces single-ended termination impedances of ~ 50 ohms to the I/O supply or ground.

- **Tx-to-Rx Analog Serial Loopback Switch** - This switch establishes a serial loopback connection between Tx and Rx differential signals for wafer test only. This analog loopback encompasses the complete analog signal path.

- **Tx-to-Rx Digital Serial Loopback Switch** - This switch establishes a serial loopback connection between the Tx and Rx circuits. This digital loopback encompasses most of the signal path, skipping the actual transmit driver and the first stage of the receiver.

- **Optional Additional Tx Serializer** - When enabled in Wide (20-bit) mode by assertion of the wide_xface control, this block reduces the width of the Tx datapath from two words to one word. In Wide mode, the Tx_Ck rate is expected to be 1/20th the baud rate. When the wide_xface control is deasserted (10-bit mode), only bits [9:0] of the TxData[] bus are used. In either 10- or 20-bit mode, this block converts Tx data from the Tx_Ck domain to the internal "baud / 10" domain. Tx_Ck is derived by buffering the cko_word or tx_cko_word clock. When half_rate is asserted, the clock frequencies are halved, but still based on the true baud rate as previously described.

- **Tx Pattern Generator** - This block produces one of a few pseudo-random sequences that can be easily matched in a receiver as part of the built-in Byte Error Rate Testing (BERT) function.

- **Tx Serializer** - This block serializes data, converting 10 bits at the word rate to 1 bit at the baud rate.

- **Tx Driver** -When fully enabled, this block drives a serialized signal with the programmed signal level and de- emphasis.

Clock Module - This section describes blocks in the Clock module.

- **Multiplying PLL** - The Multiplying PLL (MPLL) block converts the provided reference clock into the "baud / 2" rate required. This high-speed differential clock is buffered internally through an array of transceivers. The cko_word clock is produced at 1/10th or 1/20th the MPLL baud rate.

- **Spread Spectrum Generator** - This block generates the appropriate clock offsets for SATA spread spectrum support.

- **RTerm Tuning** -This block uses an external resistor (ResRef) as reference to determine the proper calibration setting for centering the Rx and Tx termination resistances at 50 ohms.

- **JTAG Interface** - This block enables test mode programming of internal test features.

NOTE

To use the SATA2 PHY in product applications, JTAG transactions are not required. To use the provided automatic test equipment (ATE) vectors and diagnostic capabilities, the JTAG interface is required.

64.3 Functional Description

This section describes SATA2 PHY functions.

64.3.1 Power Controls

This section describes power-down controls available to the ASIC.

64.3.1.1 Tx Power Controls

The table below provides the recommended Tx power state mappings.

Table 64-1. Recommended SATA Tx Power State Mappings

tx_en[2:0]	SATA
000 (OFF)	Disabled
001 (CM)	Slumber
010 (CM_CLK)	Partial
011 (ON)	Enabled/OOB

For tx_en[2:0] settings, see Per-Transceiver Control and Status Signals.

The table below lists all possible state changes. The SATA2 PHY supports only changes that are indicated as valid.

Most changes are immediate (only logic and signal delay times). Changes that are not immediate are complete when tx_done toggles, as noted in the table below. The tx_done signal also toggles due to the assertion of tx_clk_align.

If the tx_ck input changes in phase or frequency, the internal transmit clock must be resynchronized to the tx_ck input before driving data. This resynchronization occurs automatically during state transitions. If this resynchronization is required outside these transitions, resynchronization can be requested by asserting the tx_clk_align input.

Table 64-2. Transmitter State Transitions

From State	To State	Valid Change	tx_done Toggle	Clock Resync
OFF	CM	Yes	No	No
OFF	CM_CLK	Yes	Yes	Yes
OFF	ON	Yes	Yes	Yes
CM	OFF	Yes	No	No
CM	CM_CLK	Yes	Yes	Yes
CM	ON	Yes	Yes	Yes
CM_CLK	OFF	Yes	No	No
CM_CLK	CM	Yes	No	No
CM_CLK	ON	Yes	No	No
ON	OFF	Yes	No	No
ON	CM	Yes	No	No
ON	CM_CLK	Yes	No	No

64.3.1.2 Rx Power Controls

The receiver function is controlled through the rx_en, rx_pll_pwr_on, and rx_term_en pins. The successful change of the PLL power state (due to change in rx_pll_pwr_on) is signaled back to the ASIC by transitioning the rx_pll_state signal.

The LOS function continues to operate unless explicitly disabled. The table below provides the recommended Rx power state mappings.

Table 64-3. Recommended SATA Rx Power State Mappings

SATA	rx_en	rx_pll_pwr_on	rx_term_en
Disabled	0	0	0
Slumber	0	0	1

Table continues on the next page...

Table 64-3. Recommended SATA Rx Power State Mappings (continued)

SATA	rx_en	rx_pll_pwr_on	rx_term_en
Partial	0	0/1	1
Enabled/OOB	1	1	1

or fastest exit from partial, set rx_pll_pwr_on to 1'b1. For lower power, set rx_pll_pwr_on to 1'b0.

When rx_en is set to 1'b0, rx_ck is disabled; therefore, either asynchronous logic or another clock source must be used to move between states during this period.

64.3.1.3 Clock Module Power Controls

The table below provides the MPLL power state settings.

Table 64-4. Recommended SATA Power State Mappings

SATA	mpll_pwron (PCLK State)
Disabled	0
Slumber	1
Partial	1
Enabled/OOB	1

The Clock module contains the MPLL and is the entry point for the SATA2 PHY reference clock. The MPLL can be powered down using this module, and the reference clock coming into the PHY can be suspended.

NOTE

Suspending the reference clock requires the ASIC to be able to unsuspend the reference clock without a clock from the SATA2 PHY.

Before disabling the reference clock externally or suspending the clock via the mpll_ck_off pin, always power down the MPLL. The mpll_ck_off pin gates the reference clock on and off at the entry point to the SATA2 PHY. Before disabling the external reference clock, always set mpll_ck_off to prevent needless power consumption in the block that converts the reference clock to full logic levels. Setting mpll_ck_off disables all clocks (even cko_alive). When powering down the MPLL, consider whether you want cko_word to be suspended or switched to the reference clock and if you want the cko_alive clock to be available.

64.3.1.4 Power-Up Sequences

This section describes various power-up sequences.

64.3.1.4.1 Powering Up the Chip (Initial Power-Up)

To perform initial chip power-up:

1. Power up the power supplies.

NOTE

If you use the power_good indicator to determine when the power supplies have all reached 80% of their respective nominal values, ensure that you power up the lowest supply first.

2. Without using a clock from the SATA2 PHY, enable the external reference clock and wait for it to become stable.

3. To propagate refclk to the MPLL, set mpll_ck_off to 1'b0.

NOTE

Before setting mpll_ck_off to 1'b0, set the mpll_ncy, mpll_ncy5, and mpll_prescale signals to the appropriate values.

4. Perform a PHY reset by either toggling reset_n or writing a 1 to the Reset register (see Reset Register Reset Register) through the JTAG interface or Parallel CR Control port.

Upon writing the PHY reset bit in the reset register, the internal PHY reset is active immediately. Since the reset also affects the control register state machine, there will not be an acknowledgement of the write; that is, cr_ack will not be asserted.

NOTE

Diagnostic code should treat the *lack* of an acknowledgment of the write as a *successful* write; alternatively, it should treat the PHY *acknowledging* a write of the reset as a write *failure*. This is the opposite expectation of all other registers, where the lack is a failure and the *acknowledge* is successful. It is sufficient to wait 20 ref_clock cycles in order to determine that the acknowledgement has *not* occurred.

For information about the JTAG interface and the Parallel CR Control port, see Block Descriptions.

64.3.1.4.2 Powering Up the Clock Module

The following procedure assumes the chip is powered up and all blocks are in their power-down state. To power up the Clock module:

1. To start the MPLL, set the mpll_pwron signal to 1'b1.
2. Wait for op_done to transition.

The Tx and Rx can now be powered up as appropriate.

64.3.1.4.3 Powering Up the Tx

Assuming that the Clock module is powered up, tx_en[2:0] can be changed to any mode. Some transitions might require waiting for tx_done to transition before the transmitter is in the intended state.

64.3.1.4.4 Powering Up the Rx

The following procedure assumes that the Clock module is powered up.

To power up the Rx:

1. Turn on receiver terminations by setting rx_term_en to 1'b1.
2. Power up the Rx PLL by setting rx_pll_pwron to 1'b1.
3. Wait for rx_pll_state to go high.
4. Enable rx_ck and drive data out on the rx_data pins by setting rx_en to 1'b1.
5. Wait for rx_valid to indicate that the data is valid.

64.3.1.5 Power-Down Sequences

This section describes various power-down sequences.

64.3.1.5.1 Powering Down the Rx

To power down the Rx:

1. To disable the rx_ck and rx_data pins, set rx_en to 1'b0.
2. To power down the Rx PLL, set rx_pll_pwron to 1'b0.
3. Wait for rx_pll_state to go low.

64.3.1.5.2 Powering Down the Tx

To disable the transmitter, set tx_en[2:0] to the OFF or CM setting.

64.3.1.5.3 Powering Down the Clock Module

This procedure assumes the Tx and Rx are powered down.

To power down the Clock module:

1. If you want refclk to be output on cko_word while the MPLL is powered down, change cko_word_con to the refclk setting (3'b000).
2. To power down the MPLL, set mpll_pwron to 1'b0.
3. Wait for op_done to transition.
4. If refclk is to be suspended, or if mpll_ncy, mpll_ncy5, or mpll_prescale must be changed, first set mpll_ck_off to 1'b1.

64.3.2 Clock Module Operations

Figure 64-2 below shows the input and output clocks for one receiver/transmitter pair and the Clock module.

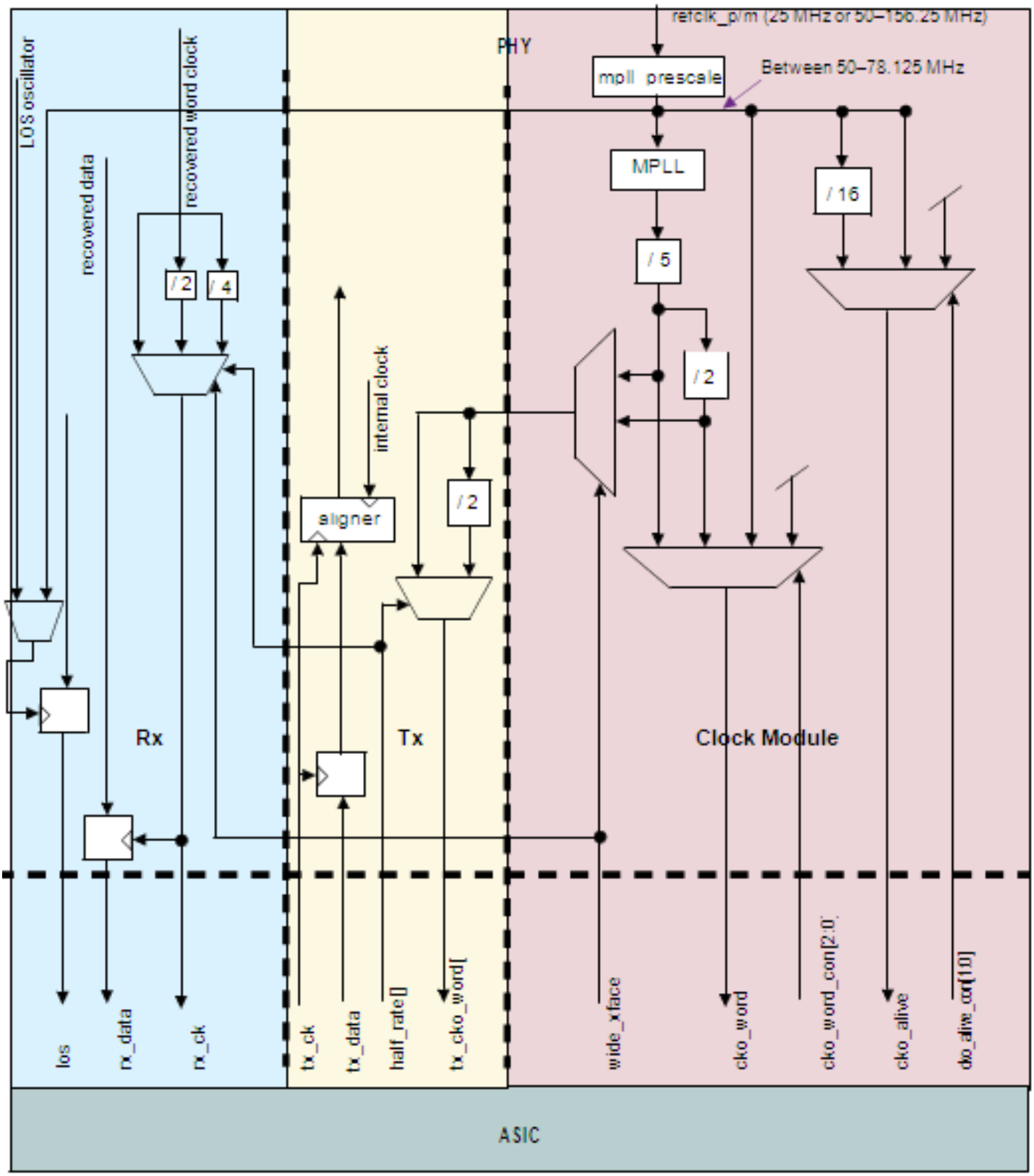


Figure 64-2. Clock interface

Note: tx_ck must be derived from tx_cko_word or cko_word.

64.3.3 Clock Inputs to the SATA2 PHY

The SATA2 PHY reference clock can be sourced from the ref_clk_p/m pads or the alt_ref_clk_p/m pins. The SATA2 PHY reference clock input has a frequency range of 25 MHz or 50-156.25 MHz.

Although the MPLL has a wide frequency range, there is a subset of specific input frequences that achieves the desired data rate. The reference clock is not used by the MPLL directly; instead, the clock passes through a prescaler that can pass the ref_clk through or divide/multiply the ref_clk frequency by 2. To guarantee that the MPLL input is in the range of 50-78.125 MHz, the prescaler must be set properly.

64.3.3.1 mpll_prescale[1:0]

The prescaler is a block that inputs the reference clock and outputs the clock used as input to the MPLL and the cko_word and cko_alive multiplexers.

The MPLL expects input clock (output from the prescaler) frequencies of 50-78.125 MHz. The table below provides some common settings. Set up the MPLL to operate at half the baud rate (except for SATA I where the MPLL operates at baud rate).

Valid refclk Range and Formulaic MPLL Settings provides a general formulaic approach for determining settings for less common scenarios. In the table below, the MPLL Divider column can be decoded into the appropriate mpll_ncy[4:0] and mpll_ncy5[1:0] settings using Table 64-7.

Table 64-5. mpll_prescale[1:0]: Sample Reference Clock Configurations

Reference Clock Frequency (MHz)	pll_prescale[1:0]	mpll_prescale Effect on ref clk	Input to MPLL (MHz)	MPLL Divider	Resultant Baud Rate (Gbps)
25	01	x 2	50	30	3
50	00	As is	50	30	3
60	00	As is	60	25	3
62.5	00	As is	62.5	24	3
75	00	As is	75	20	3
100	10	/ 2	50	30	3
120	10	/ 2	60	25	3
125	10	/ 2	62.5	24	3
150	10	/ 2	75	20	3

64.3.3.2 Valid refclk Range and Formulaic MPLL Settings

This section presents a formulaic approach for setting the MPLL values.

As mentioned in mpll_prescale[1:0], the MPLL input must be in the range of 50-78.125 MHz. Frequency doubling and halving is available at the refclk input. The following table lists all valid refclk ranges and the appropriate mpll_prescale[1:0] settings to place refclk in the proper range. This list includes valid refclk frequencies, but there is no guarantee that a particular refclk frequency can be used to create the intended baud rate.

Table 64-6. Valid refclk Frequencies

refclk Frequency Range (MHz)	mpll_prescale[1:0]	Effect on refclk Frequency	Clock Input to MPLL (MHz)
25	01	Doubled	50
50-78.125	00	No effect	50-78.125
100-156.25	10	Halved	50-78.125

To create the intended baud rate, the clock input to the MPLL (as determined from Table 2-5 on page 31) must evenly divide half the intended baud rate (or the baud rate in Half-Rate mode). The formula for Full- Rate mode is:

MPLL Divider = 0.5 x (baud rate) / (clock input to MPLL)

The formula for Half-Rate mode is:

MPLL Divider = (baud rate) / (clock input to MPLL)

The MPLL Divider value can then be used to determine the values for the mpll_ncy5 and mpll_ncy buses, as listed in the following table.

NOTE

The preceding formulae must yield a valid integer. If the result is not an integer, a refclk frequency that produces a valid integer must be used. The table below lists valid divider values.

Table 64-7. MPLL Divider Settings

MPLL Divider	mpll_ncy[4:0]	mpll_ncy5[1:0]
4	00000	00
5	00000	01
8	00001	00
9	00001	01
10	00001	01

Table continues on the next page...

Table 64-7. MPLL Divider Settings (continued)

MPLL Divider	mpll_ncy[4:0]	mpll_ncy5[1:0]
12	00010	00
13	00010	01
14	00010	10
15	00010	11
16	00011	00
17	00011	01
18	00011	10
19	00011	11
20	00100	00
21	00100	01
22	00100	10
23	00100	11
24	00101	00
25	00101	01
26	00101	10
27	00101	11
28	00110	00
29	00110	01
30	00110	10
31	00110	11
32	00111	00
33	00111	01
34	00111	10
35	00111	11
36	01000	00
37	01000	01
38	01000	10
39	01000	11
40	01001	00
41	01001	01
42	01001	10
43	01001	11
44	01010	00
45	01010	01
46	01010	10

Table continues on the next page...

Table 64-7. MPLL Divider Settings (continued)

MPLL Divider	mpll_ncy[4:0]	mpll_ncy5[1:0]
47	01010	11
48	01011	00
49	01011	01
50	01011	10
51	01011	11
52	01100	00
53	01100	01
54	01100	10
55	01100	11
56	01101	00
57	01101	01
58	01101	10
59	01101	11
60	01110	00
61	01110	01
62	01110	10
63	01110	11

64.3.3.3 Clock Module Power Control

For information about the Clock module power controls, see Clock Module Power Control.

64.3.3.4 Presence of refclk Signal

The reference clock must be present during all operational modes, except when the MPLL is powered down and mpll_ck_off has been set to 1'b1.

For more information, see Power-Up Sequences and Power-Down Sequences.

64.3.3.5 Power-On Reset

The SATA2 PHY asserts its internal power-on reset (POR) whenever the voltage level of the high supply has not yet reached the initial "power valid" trip point at ~70-75% of the rated supply voltage.

After a POR is asserted, to prevent accidental reactivation of the POR, the trip point is reduced to ~65-70% of the rated supply voltage. Reactivation of a POR causes the SATA2 PHY to go into a reset state again.

Note that deassertion of a POR causes a normal transition out of reset. During power-up, constraints on the state of the control inputs must be maintained-meaning that the POR mechanism in the SoC must ensure that control inputs are set to the intended states.

64.3.3.6 Resistor Calibration

During a reset or at the request of the ASIC via the rtune_do_tune pin, a resistor calibration occurs. When the operation is complete, the op_done pin is transitioned.

Resistor calibration consists of learning which state of the internal Resistor Calibration register causes an internal, digitally trimmed calibration resistor to best match the impedance applied to the ResRef pin. The calibration register value is then supplied to all Tx and Rx termination resistors.

During the calibration process (for a few tens of microseconds), up to 0.3 mW can be dissipated in the external ResRef resistor. At other times, no power is dissipated by the ResRef resistor.

64.3.4 Tx Operations

This section describes transmitter settings and operations.

64.3.4.1 Recommended Tx Settings

The following table outlines the recommended transmitter settings, and the sections that follow describe each setting in detail.

Table 64-8. Recommended Tx Settings

Application	tx_atten[2:0] (Short/ Medium/Long)[1, 1]	tx_boost[3:0] (Short/ Medium/Long)[1]	tx_lvl[4:0] (Short/Medium/ Long)[1]	x_edgerate[1:0]
SATA 1i	011	N/A	00110	01
SATA 1m	011	N/A	00110	01
SATA 1x	N/A	1001	00110	01
SATA 2i	100	N/A	10001	00
SATA 2m	100	N/A	10001	00
SATA 2x	000	1001	10001	00

1. Short/medium/long are typical values based on 5-cm, 23-cm, and 48-cm trace lengths on FR4 material. These values are provided as guidelines. It is recommended that the final setting be based on the characteristics of the actual serial channel.

64.3.4.2 Tx Amplitude Control

The transmitter has a programmable boost, level, and attenuation. The transmitter's full-scale amplitude is a function of the selected level (tx_lvl), attenuation (tx_atten), and boost (tx_boost).

The actual peak-to-peak differential amplitude is the Tx amplitude multiplied by the attenuation multiplier:

$$\text{Tx amp}_{\text{(p-p differential near-end)}} = \text{attenuation multiplier} \bullet \frac{1.24}{63.5} \left[(48 + 0.5 (\text{tx\_lvl[4:0]}) - V \right]_{\text{, p-p differential near-end}}$$

Using a larger transmit amplitude provides a larger eye to a far-end receiver (at the expense of a slight increase in power).

For tx_lvl[4:0] settings, see Common Signals.

64.3.4.3 Tx Boost Control

The transmitter can be programmed to provide boost (pre-emphasis/de-emphasis). Boost is achieved by reducing the drive level of a non-transition bit with respect to a transition bit.

The amount of boost the transmitter supplies is programmed through the tx_boost[3:0] pins as follows:

$$\text{boost} = - 20\log\left(1 - \frac{\text{txboost[3:0]} + 0.5}{32}\right) \text{ db}$$

except that setting tx_boost[3:0] to 4'b0000 actually produces 0 db of boost. This boost control produces results up to 5.75 dB in increments of ~0.37 dB. These pins can be transitioned asynchronously. The best boost setting is channel dependent. For lossy channels, the maximum boost setting is appropriate; for low- loss channels, lower settings provide a better eye. Table 64-8 provides the recommended boost settings for short, medium, and long trace lengths. If lossy channels are expected in the application, setting

the boost to its maximum value is appropriate. If more information is known about the channel, a good rule is to take the loss at Nyquist, subtract 3 dB, and find the tx_boost setting that most closely achieves this value. For example, if the worst-case loss at Nyquist is 6.5 dB, a good setting for tx_boost is one that achieves 3.5 dB of boost. From the preceding equation, the tx_boost setting is 4'b1010.

64.3.4.4 Tx Far-End Amplitude

The transmitter far-end amplitude is a function of boost, level, and attenuation.

The actual far-end differential amplitude can be defined as the most commonly transmitted level:

$$\left(-\frac{\text{boost}_{db}}{\text{attenuation factor}}\right) = \text{atten}_{multiplier} \cdot 10^{20}$$

$$\text{Tx amp}_{\text{(differential far-end)}} = \text{attenuation factor} \cdot \frac{1.24}{63.5}\left[\left(48 + 0.5\left(\text{tx\_lvl[4:0]}\right) - V\right]_{\text{, differential far-end}}\right.$$

With proper boost settings, the actual far-end differential amplitude is approximately equal to the far-end peak-to-peak differential amplitude.

64.3.4.5 Tx Edge Rate Control

The following table describes the edge rate control settings, which enable the SATA2 PHY to meet the edge rate requirements of all SATA variants.

Table 64-9. tx_edgerate Settings

tx_edgerate[1:0]	Value
00	Fast edge rate
01	Medium edge rate
10	Slow edge rate
11	Reserved

64.3.5 Rx Operations

This section describes receiver operations.

64.3.5.1 Recommended Rx Settings

The table below describes the recommended Rx settings.

Table 64-10. Recommended Rx Settings

Application	rx_eq_val[2:0]	los_lvl[4:0]	rx_dpll_mode[2:0]
SATA1i	101	10000	011 with SS 001 without SS
SATA1m	101	10000	011 with SS 001 without SS
SATA1x	101	11010	011 with SS 001 without SS
SATA2i	101	10010	011 with SS 001 without SS
SATA2m	101	10010	011 with SS 001 without SS
SATA2x	101	11010	011 with SS 001 without SS

64.3.5.2 Loss of Signal Detection

Flexible LOS detection is provided on a per-lane basis. The per-lane output of the LOS detection is provided through the LOS pins. LOS detection is controlled by filtering the raw LOS signal using the los_ctl pins and setting the LOS threshold.

Table 64-11 describes the los_ctl settings.

Table 64-11. los_ctl Settings

los_ctl[1:0]	Description
00	Disabled
01	Reserved
10	Filtering for SATA to enable OOB processing. The LOS signal is synchronous to the output of the prescaler (available on cko_alive or cko_word).
11	Heavy filtering. The LOS signal is synchronous to the prescaler's output (available on cko_alive or cko_word). Heavy filtering needs longer periods of inactivity or activity before indicating an LOS event because of slower response time to chatter on the receiver lines.

As defined in the SATA specification, OOB signaling is accomplished by measuring the gaps between bursts of energy. An energy burst is defined to be 160 UI long (107 ns long) at SATA I rates. The burst consists of repeating ALIGN primitives (k28.5, d10.2, d10.2, d27.3).

Table 64-12. Words Used in ALIGN Primitive

ALIGN words	+ Disparity	- Disparity
k28.5 (COMMA)	1 1 0 0 0 0 0 1 0 1	0 0 1 1 1 1 1 0 1 0
d10.2 (NYQUIST)	0 1 0 1 0 1 0 1 0 1	0 1 0 1 0 1 0 1 0 1
d27.3	1 1 0 1 1 0 0 0 1 1	0 0 1 0 0 1 1 1 0 0

The ALIGNs during OOB signaling are always sent at SATA I rates. The communication is realized by the length of the gap between bursts. The following three tables describe the OOB gap-sensing limits for COMWAKE, COMSET/COMINIT, and COMSAS, respectively.

Table 64-13. OOB Gap-Sensing Limits for COMWAKE

Gap Length	Decision
< 55 ns	Must not indicate a gap
55-101 ns	Can indicate COMWAKE
101-112ns	Must indicate COMWAKE
112-175ns	Can indicate COMWAKE
> 175 ns	Must not indicate there was a gap

Table 64-14. OOB Gap-Sensing Limits for COMRESET/COMINIT

Gap Length	Decision
< 175 ns	Must not indicate a gap
175-304 ns	Can indicate COMRESET/COMINIT
304-336 ns	Must indicate COMRESET/COMINIT
336-525 ns	Can indicate COMRESET/COMINIT
> 525 ns	Must not indicate there was a gap

Table 64-15. OOB Gap-Sensing Limits for COMSAS

Gap Length	Decision
< 525 ns	Must not indicate a gap
525-911.7 ns	Can indicate COMSAS
911.7-1,008 ns	Must indicate COMSAS
1,008-1,575 ns	Can indicate COMSAS
> 1,575 ns	Must not indicate there was a gap

For OOB detection/transmission, to synchronously transfer the LOS signal to the ASIC, the SATA2 PHY relies on a clock that is proportional to the reference clock. If the reference clock frequency is 25 MHz, it is multiplied by 2-50 MHz; if the reference clock frequency is 100 MHz, it is divided by 2-50 MHz. In addition, the LOS signal is preprocessed. Therefore, based on the results, it is recommended that the decision points indicated in Table 2-16 be used for detecting OOB signals.

Table 64-16. Recommended Decision Points (Assumes 25-MHz, 50-MHz or 100-MHz Reference Clock)

LOS Length (20-ns Periods)	Decision	True Limits (ns)
0-2	Invalid gap length	< 80
3-6	COMWAKE	70-150
7-10	Invalid gap length	140-230
11-22	COMINIT/COMRESET	220-470
22-28	Invalid gap length	460-590
29-75	COMSAS	580-1,530
>=76	Invalid gap length	>=1,520

NOTE

Recommended decision points assume a reference clock of 25 MHz, 50-MHZ, or 100 MHz.

To measure/count the LOS length, it is recommended that you use cko_alive with cko_alive_con[1:0] set to 2'b01. With this setting, the cko_alive frequency will be 50 MHz .

64.3.5.3 rx_dpll_mode[2:0]

The table below describes the rx_dpll_mode[2:0] settings. At each point in time, the actual gain setting is the larger of the gains indicated by the fast startup (if in use) and rx_dpll_mode[1:0]. The values of Phase

Update Gain (PHUG) and Frequency Update Gain (FRUG) correspond to the proportional and integral gains of the DPLL. Higher gains in the PHUG and FRUG settings cause an increase in jitter.

Table 64-17. rx_dpll_mode Settings

rx_dpll_mode[2:0]	PHUG	FRUG	fast_startup	Frequency Tolerance (ppm)
000	1	1	None	780

Table continues on the next page...

Table 64-17. rx_dpll_mode Settings (continued)

rx_dpll_mode[2:0]	PHUG	FRUG	fast_startup	Frequency Tolerance (ppm)
001[1, 1]	2	2	None	780
101	1	4	None	6,250
011[2, 2]	2	4	None	6,250
100	Reserved			
101				
110				
111				

1. Recommended setting (without Spread Sectrum (SS))
2. Recommended setting (with Spread Sectrum (SS))

64.3.5.4 Rx Equalizer Settings

The SATA2 PHY provides an Rx equalizer that can be programmed using the rx_eq_val[2:0] bits. The table below lists the approximate boost that each setting provides.

Table 64-18. Rx Equalizer Settings

rx_eq_val[2:0]	Boost (dB)
000	0.5
001	1.0
010	1.5
011	2.0
100	2.5
101	3.0
110	3.5
111	4.0

These pins can be transitioned asynchronously, but a dynamic change to these pins during data transmission might result in bit errors. Typically, these settings do not change during operation. Although the specification does not define receive equalization, a more robust link can be achieved by providing some equalization. A good general setting is to set the equalizer to approximately 3 dB of boost-suitable for many applications.

64.4 Control Registers

SATA2 PHY test features are controlled by registers that are accessible through the JTAG interface. Each register is a maximum of 16 bits wide.

Because some registers contain different fields, when changing a single field in a register, use a read-modify-write approach.

64.4.1 Register Fields

Some fields are single bits while other fields encompass the entire contents of the containment register. Some registers have Xs as reset values.

These registers are categorized as follows.

Registers That Reflect Inputs From the ASIC Interface

The values of these inputs are unknown on reset; therefore, the register's reset value is indeterminate.

These registers include:

- lane0.tx_stat
- lane0.rx_stat
- clock.freq_stat
- clock.ctl_stat
- clock.lvl_stat
- clock.creg_stat

Registers That Reflect Results of Operations Not Initialized on Reset

These registers might have actual Xs on part reset. However, those bits have no meaning until the associated function is enabled or initialized; until then, discard the read results of these register bits.

These registers include:

- lane0.out_stat
- lane0.pm_err
- clock.crcmp_stat
- clock.scope_count
- clock.adc_out

When reading a register, mask the result to view only the bits with which you are concerned. Correct masking disables the Xs from propagating through your logic.

64.4.1.1 Field Properties

The property string associated with each field is a concatenation of the applicable attributes listed in the table below. For example, the most common field property is RWCr, which denotes readable, writable, and subject to a reset.

Table 64-19. Register Attributes

Property	Description
R	Readable. A register read returns the current value of the register itself (if it is also writable) or the state to which the register is attached.
R2	Read operation on this register is pipelined. Two reads are required to get "current" value.
W	Writable
V	Volatile (that is, the value can change at any time)
I	Value is in signed integer format (2's complement).
Cr	Register is subject to being returned to its reset value on a reset.

In addition, each field requires two numeric parameters to specify the field's location in the register: a width (in bits) and an offset (in bits) from the register's least-significant bit (LSB).

64.4.1.2 Field Names

Fully specified field names consist of the concatenation of a register name and a field name, as in *register.field*.

Fields in registers comprising a single field (for example, clock.crcmp_lo_range.crcmp_lo_range) are typically named without the concatenation (for example, clock.crcmp_lo_range). In these cases, the single field name always matches the last portion of the register name.

64.4.1.3 Read/Modify/Write Operations

To modify one field in a register containing multiple fields, it is generally necessary to read the register, modify the target field, then write the entire result.

One approach to writing control registers is to monitor the contents of each field; this approach does not require read-modify-write operations. The read is not necessary if the current register value is known (reset value at the time of the reset updated with any subsequent modifications to values of other fields in the register). Therefore, volatile control fields such as DPLL PHASE are never combined in the same register with other control fields.

Methods that do not maintain this information must work by reading a register, modifying only the field being changed, then writing the register.

64.5 Signal Descriptions

64.5.1 Signal Descriptions Overview

The SATA2 PHY top-level signals are grouped as follows:

- Power supply signals
- Package I/O signals
- Per-transceiver control and status signals
- Per-transceiver datapath signals
- Common signals
- JTAG interface signals
- Parallel CR Control port signals

Per-transceiver pins are replicated for each lane in the SATA2 PHY. In the signal descriptions, this replication is denoted by "lane" in the signal name.

For example, a x2 PHY half_rate[lane] denotes two signals: half_rate[0] and half_rate[1]. tx_data_{lane}[19:0] denotes two buses: tx_data_0[19:0] and tx_data_1[19:0].

For information about actual pin layers and pin locations, refer to the LEF file provided in the database deliverables. All ASIC-side pins are at CMOS logic.

The figure below shows the SATA2 PHY top-level signals.

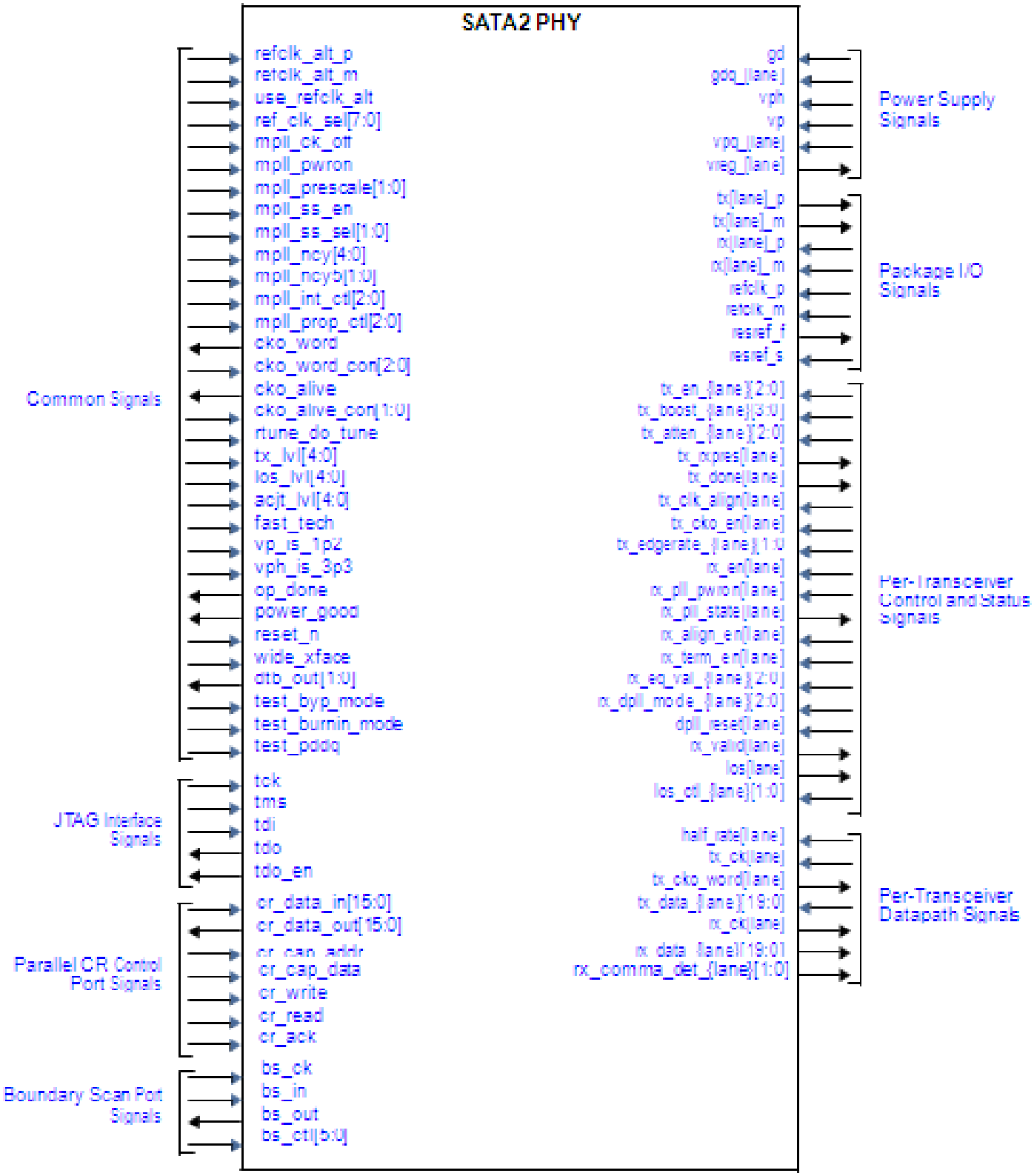


Figure 64-3. Top-Level IO Diagram

64.5.2 Signal Descriptions Information

In addition to describing the function of each signal, the signal descriptions include the following information:

- **Voltage Range:** Describes the voltage range expected on the pin.
- **Synchronous:** Indicates that the signal is asserted or deasserted with respect to a clock edge. (For synchronous-signal timing requirements.
- **Asynchronous:** Indicates that the signal is not asserted or deasserted with respect to a clock edge.

64.5.2.1 Power Supply Signals

The SATA2 PHY core requires two power supply voltages and a ground. Power connections abut directly between the Active and Pad sections; no routing of these power connections is required nor permitted.

Table 64-20. Power Supply Signals

Signal	IO	Description	Pad Section Connection
gd	I	Ground **Function:** Ground **Voltage Range:** 0 V **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** none - power/ground signal	gd
gdq_[lane]	I	Filtered Ground **Function:** Isolated version of gd for transmit jitter-sensitive devices. **Voltage Range:** 0 V **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** none - power/ground signal	gd
vph	I	High-Voltage Supply **Function:** Core Power **Voltage Range:** Varies (internal use only) **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** none - power/ground signal	vp25

Table continues on the next page...

Table 64-20. Power Supply Signals (continued)

Signal	IO	Description	Pad Section Connection
vp	I	Low-Voltage Supply **Function:** Core Power **Voltage Range:** Varies (internal use only) **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** none - power/ground signal	vp
vpq_[lane]	I	Filtered Low-Voltage Supply **Function:** Core Power **Voltage Range:**Varies (internal use only) **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** none - power/ground signal	vpq[lane] **Note:** For an x1 configuration, this pin is vpq.
vreg_[lane]	O	Regulated Voltage Supply for Transmitter **Function:** Regulated voltage supply for transmitter that is output from the PHY for attachment to an ESD clamp in the Pad section. **Voltage Range:** Varies (internal use only) **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** don't care	vreg[lane] **Note:** For an x1 configuration, this pin is vreg.

64.5.2.2 Package I/O Signals

The SATA2 PHY includes pins that must connect to package pins.

The table below describes these pins, which include the Tx, Rx, and reference clock differential pairs, and connections to an external resistor. These package I/O signals abut directly between the Active and Pad sections and connect to the Pad ports as described in the "Pad Section Connection" column. No additional routing of these signal connections is required nor permitted.

Table 64-21. Package I/O Signals

Signal	IO	Description	Pad Section Connection
• tx[0]_p • tx[0]_m	O	High-Speed Differential Transmit Pair **Function:** High-speed differential transmit outputs **Voltage Range:** 0 V-I/O voltage **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** don't care	• txp[0] • txm[0] Attach the same wire to the pad, macro, and chip interfaces.
• rx[0]_p • rx[0]_m	I	High-Speed Differential Receive Pair **Function:** High-speed differential receiver inputs **Voltage Range:** 0 V-I/O voltage **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** • rx[0]_p: none - loopback from tx0_p • rx[0]_p: none - loopback from tx0_m	• rxp[0] • rxm[0] Attach the same wire to the pad, macro, and chip interfaces.
• refclk_p • refclk_m	I	Differential Reference Clock Input **Function:** Input reference clock. If the sample pads are not used, but the refclk_alt_{p,m} signals are used instead, refclk_{p,m} can remain disconnected (not bonded or bumped). **Voltage Range:** 0 V-I/O voltage **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** • refclk_p: none - application-specific • refclk_m: none - application-specific	• refclkp_int • refclkm_int These wires are internal-only. Connect the chip interface to the refclkp and refclkm pins in the Pad section.

Table continues on the next page...

Table 64-21. Package I/O Signals (continued)

Signal	IO	Description	Pad Section Connection
• resref_f • resref_s	• O • I	Reference Resistor Connection **Function:** 191 ohms 1% precision resistor to ground. During resistor tuning, current is forced through the external resistor using resref_f while the induced voltage is sensed on resref_s. In the Pad section, the resref_f and resref_s pins are combined into a single external pad. **Voltage Range:** N/A **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** • resref_f: don't care • resref_s: none - connect to resistor on silicon	• rext • rext_s Connect one package pin to both resref_f on the IP and rext in the Pad section. Using a second wire, connect resref_s on the IP and rext_s in the Pad section.

64.5.2.3 Per-Transceiver Control and Status Signals

The table below describes the per-transceiver control/status and miscellaneous signals.

Table 64-22. Per-Transceiver Control and Status Signals

Signal	I/O	Description
Related to Transmitter		

Table continues on the next page...

Table 64-22. Per-Transceiver Control and Status Signals (continued)

Signal	I/O	Description
tx_en_{lane}[2:0]	I	Transmitter Enable **Function:** Enables transmitter. Transitions between 010 and 011 are staged to match the tx_data latency, so that all data on the parallel interface when tx_en is 011 is guaranteed to be output from the serial transmitter. • 000 (OFF): Tx off, no internal clocks running • 001 (CM): Hold common mode, no internal clocks running • 010 (CM_CLK): Hold common mode, transmitter clock divider running • 011 (ON): Transmitter fully enabled, all internal clocks running • 100: Reserved • 101: Reserved • 110: Reserved • 111: Reserved For information about the recommended Tx power state mappings, see Tx Power Controls. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** tx_ck **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
tx_boost_{lane}[3:0]	I	Transmit Boost Control **Function:** Programmed boost value (ratio of drive level of transition bit to non- transition bit) is: boost-20log[(tx_boost[3:0] + 0.5)/32] = db except that setting tx_boost[3:0] to 0000 actually produces 0 db of boost. This control produces results up to 5.75 dB in increments of ~ 0.37 dB. For more information about tx_boost_{lane}[3:0] and the recommended settings, see Tx Operations. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous; changes while the transmitter is on might briefly cause unwanted transmit waveforms **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction

Table continues on the next page...

Table 64-22. Per-Transceiver Control and Status Signals (continued)

Signal	I/O	Description
tx_atten_{lane}[2:0]	I	Transmit Attenuation Control **Function:** Provides discrete driver attenuation factors of either 16-, 14-, 12-, 10-, 9-, or 8-sixteenths of full drive level. • 000: 16/16 • 001: 14/16 • 010: 12/16 • 011: 10/16 • 100: 9/16 • 101: 8/16 • 11x: Reserved Attenuating the drive level in this way does not use less power than using the non-attenuated (16/16) drive level. Attenuation factor is dependent on boost level and $atten_{multiplier}$ (for more information about these calculations, see Tx Boost Control and Tx Far-End Amplitude, respectively). **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous; changes while the transmitter is on might briefly cause unwanted transmit waveforms **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
tx_rxpres[lane]	O	Receiver Detection **Function:** Reserved Leave these pins floating. **Voltage Range:** 0 V-core voltageOverride Ablility: 18-bit CREG_OVRD register
tx_done[lane]	O	Transmitter Operation Complete **Function:** Transitions (in either direction) when the transmitter has completed a requested power state transition or operation requested via the tx_en_{lane}[2:0] pins when a tx_clk_align operation is complete. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to**: tx_ck, but can be sampled asynchronously if necessary **Override Ablility:** 18-bit CREG_OVRD register

Table continues on the next page...

Table 64-22. Per-Transceiver Control and Status Signals (continued)

Signal	I/O	Description
tx_clk_align[lane]	I	Transmit Clock Align **Function:** Rising edge is a request to realign the internal Tx word clock to the applied tx_ck signal. Required only if the tx_ck path is interrupted or changed after the automatic alignments. Automatic alignments are triggered by tx_en_{lane}[2:0] transitions, which enable the transmit clocks. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous, but should be kept high for at least two tx_ck cycles **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
tx_cko_en[lane]	I	tx_cko_word Clock Enable **Function:** Enables tx_cko_word clock on a per-lane basis. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous, but transitions might take several cycles to be reflected in the clock output **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
tx_edgerate_{lane}[1:0]	I	Tx Edge Rate Control **Function:** Controls the edge rate of the serial output. • 00: Fastest edge rate • 01: Medium edge rate • 10: Slowest edge rate • 11: Reserved For more information about tx_edgerate_{lane}[1:0] and the recommended settings, see Tx Operations. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
Related to Receiver		
rx_en[lane]	I	Receive Clock and Data Outputs Enable **Function:** Enables receive clock and data outputs. The rx_en signal must be set to 1 only when both the rx_pll_pwron and rx_pll_state signals are set to 1'b1. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction

Table continues on the next page...

Table 64-22. Per-Transceiver Control and Status Signals (continued)

Signal	I/O	Description
rx_pll_pwron[lane]	I	Receive PLL Power-Up and Reset **Function:** Powers up and resets the receive PLL. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous, but must not be changed until the rx_pll_state signal matches rx_pll_pwron. **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
rx_pll_state[lane]	O	Receive PLL State **Function:** Indicates the current state of the receive PLL. This signal will match rx_pll_pwron when the PLL reaches the requested state. The PLL changes states as long as the MPLL is powered up, regardless of the state of incoming data lines. This signal does not indicate that the link is ready. After rx_en is asserted (set to 1'b1), rx_valid must be used to indicate that the link is ready. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register
rx_align_en[lane]	I	Word Alignment Enable **Function:** Assertion of this level-sensitive signal enables word alignment. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
rx_term_en[lane]	I	Rx Termination Enable **Function:** Assertion of this level-sensitive signal enables Rx termination. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
rx_eq_val_{lane}[2:0]	I	Receiver Equalization Control **Function:** Internal linear equalizer boost is ~ (rx_eq_val + 1) x 0.5 dB. For the recommended setting, see Recommended Rx Settings. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous, but transitions might cause bit errors while the boost level changes **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction

Table continues on the next page...

Table 64-22. Per-Transceiver Control and Status Signals (continued)

Signal	I/O	Description
rx_dpll_mode_{lane} [2:0]	I	DPLL Mode **Function:** Sets phase and frequency gain of receiver DPLL. rx_dpll_mode[1:0] define the steady state gains. For more information about this bus, see rx_dpll_mode[2:0]. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous while rx_en is deasserted; otherwise, must reset DPLL after change via the dpll_reset signal **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
dpll_reset[lane]	I	DPLL Frequency Register Reset **Function:** Rising edge of this signal triggers the reset of the DPLL's frequency and phase register. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous, but must be kept high for at least two rx_ck cycles **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
rx_valid[lane]	O	Receive Data Valid **Function:** Indicates that the receive DPLL is bit-locked and that the rx_data and rx_comma_det signals are valid for the corresponding lane. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** rx_ck **Override Ablility:** 18-bit CREG_OVRD register
los[lane]	O	Loss of Signal Output **Function:** Determines if a signal is present at the receiver based on the filtering selected by the los_ctl values. This signal enables higher layers to receive OOB signaling. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Output of the refclk prescaler. This same clock can be routed to the cko_alive clock output for synchronous sampling of the LOS signal. **Override Ablility:** 18-bit CREG_OVRD register

Table continues on the next page...

Table 64-22. Per-Transceiver Control and Status Signals (continued)

Signal	I/O	Description
los_ctl_{lane}[1:0]	I	LOS Detector and/or Filter Raw LOS Output Enable **Function:** Enables LOS detector and/or filtering of raw LOS output. • 00: Off • 01: Reserved • 10: Optimized for OOB signaling • 11: Heavy filtering For information about using this signal, see Loss of Signal Detection. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous; however, LOS output will be briefly unreliable during transition of the los_ctl signal **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction

64.5.2.4 Per-Transceiver Datapath Signals

Table 64-23. Per-Transceiver Datapath Signals

Signal	I/O	Description
Common to Tx and Rx		
half_rate[lane]	I	Half-Rate Mode Control **Function:** Halves the baud rate with respect to normal operation. This signal is asserted in SATA I operation and deasserted in SATA II operation. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** The transition of half_rate between the two modes of operation can be asynchronous to all clocks. **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
Related to Transmitter		
tx_ck[lane]	I	Transmit Clock **Function:** This clock must pulse at the word rate when the wide_xface signal is set to 1'b0 and at half the word rate when wide_xface is set to 1'b1. This clock must be a buffered version of the cko_word clock or tx_cko_word clock. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** none - must be buffered copy of cko_word

Table continues on the next page...

Table 64-23. Per-Transceiver Datapath Signals (continued)

Signal	I/O	Description
tx_cko_word[lane]	O	Per-Lane Word Clock **Function:** This clock runs whenever the MPLL is powered up. Clock frequency is the MPLL frequency / 10 or 5 if half_rate is set to 1'b1 or 1'b0, respectively. This clock can be used instead of cko_word if a per-lane word clock is required (that is, the Clock module can use this clock instead of cko_word.) **Note:** tx_cko_word is derived from either a divide-by-5 or divide-by-10 version of the MPLL clock based on wide_xface. If the lane is in Half-Rate mode, the tx_cko_word clock frequency is further divided by 2. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** don't care - must be connected to tx_ck for ATE testing
O tx_data_{lane} [19:0]	I	Transmit Data **Function:** Serial transmit data. Most-significant bit (MSB) is transmitted first. If wide_xface is set to 1'b0, only bits [9:0] are used. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** tx_ck **Override Ablility:** none - overriden by internal pattern generator
Related to Receiver		
rx_ck[lane]	O	Received Clock **Function:** This clock is the recovered per-lane clock. The rx_ck output runs when rx_en is set to 1'b1, but turns off in other modes. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** rx_data[19:0] (or rx_data[9:0] if wide_xface is set to 1'b0) **Override Ablility:** don't care
rx_data_{lane}[19:0]	O	Received Data **Function:** Serial received data. MSB is received first. If wide_xface is set to 1'b0, only bits [9:0] are used. This bus must be qualified with rx_valid. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** rx_ck **Override Ablility:** don't care

Table continues on the next page...

Table 64-23. Per-Transceiver Datapath Signals (continued)

Signal	I/O	Description
rx_comma_det_{lane}[1:0]	O	Comma Detect **Function:** Asserted when the associated word on rx_data is detected to be a comma (K28.5). This bus must be qualified with rx_valid. If wide_xface is set to 1'b1 (20-bit mode), bit 1 corresponds to the first word, and bit 0 corresponds to the second word. If wide_xface is set to 1'b0 (10-bit mode), only bit 0 is used. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** rx_ck **Override Ablility:** don't care

64.5.2.5 Common Signals

This section describes signals that are not replicated per transceiver. Common signals not included in this section are grouped into one of the following sections:

- JTAG Interface Signals
- Parallel CR Control Port Signals

Table 64-24. Common Signal Pins

Signal	I/O	Description
Related to MPLL or clocks		
• refclk_alt_p • refclk_alt_m	I	Alternate Reference Clock **Function:** Alternate reference clock inputs. If sample pads nor refclk_alt are not used, these pins can be left floating (they are pulled down internally). Delivery of a low-jitter reference clock (via refclk_alt_{p,m}) from the ASIC core is critical for meeting transmitter jitter specifications. If you are using the sample pads, connect these pins to refclk_alt_p and refclk_alt_m, respectively, in the Pad section. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** N/A **Override Ability:** none - application-specific; float or ground if unused

Table continues on the next page...

Table 64-24. Common Signal Pins (continued)

Signal	I/O	Description
use_refclk_alt	I	Alternate Reference Clock Control **Function:** Selects the MPLL inputs as an alternate reference clock. This signal must be static while mpll_ck_off is set to 1'b0. **Voltage Range:** 0 V-core voltage **Active State:** HIgh **Synchronous to:** N/A **Override Ablility:** 17-bit ATEOVRD JTAG register, and 18-bit CREG_OVRD register; JTAG override takes precedence
ref_clk_sel[7:0]	I	Reference Clock Frequency Select **Function:** Sets a fixed ppm offset to the MPLL output clock. • 8'd0 : No offset (Default) • 8'd1 to 8'd255 : Reserved. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
mpll_ck_off	I	Reference Clock Stop Enable **Function:** Disables the reference clock to the IP at the PHY's entry point. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous **Override Ablility:** 17-bit ATEOVRD JTAG register, and 18-bit CREG_OVRD register; JTAG override takes precedence
mpll_pwron	I	MPLL Power-Up **Function:** Enables power to the MPLL. When this signal is set to 0, the cko_word clock is glitchlessly stopped low, unless refclk production at cko_word has been selected. When this signal is set to 1, the internal MPLL reset and locking is timed based on refclk, then the cko_word clock is restarted. Before turning off the MPLL, tx_en must be in the OFF or CM state, and rx_en and rx_pll_pwron must be set to 1'b0. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous; must not be transitioned again before the op_done signal indicates that the previous operation is complete **Override Ablility:** 17-bit ATEOVRD JTAG register, and 18-bit CREG_OVRD register; JTAG override takes precedence

Table continues on the next page...

Table 64-24. Common Signal Pins (continued)

Signal	I/O	Description
mpll_prescale[1:0]	I	MPLL refclk Prescaler Control **Function:** Controls the MPLL's refclk prescaler. • 00: ref_clk is used as is. • 01: ref_clk doubler is enabled. • 10: ref_clk is divided by 2. • 11: Reserved Transitions must occur only while mpll_ck_off is set to 1'b1. For information about using this signal, see mpll_prescale[1:0] and Valid refclk Range and Formulaic MPLL Settings. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** 17-bit ATEOVRD JTAG register, and 18-bit CREG_OVRD register; JTAG override takes precedence
mpll_ss_en	I	Spread Spectrum Enable **Function:** Enables spread spectrum clock production (0.5% downspread at ~ 31.5 kHz) in the SATA2 PHY. If the applied RefClk is already spread spectrum, mpll_ss_en must be deasserted. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
mpll_ss_sel[1:0]	I	Spread Spectrum Select **Function:** Selects frequency spread for the spread spectrum clock. • 00: 5,000 ppm • 01: 4,500 ppm • 10: 4,000 ppm • 11: 3,000 ppm **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction

Table continues on the next page...

Table 64-24. Common Signal Pins (continued)

Signal	I/O	Description
mpll_ncy[4:0]	I	MPLL x4 Multiplier **Function:** MPLL x4 multiplier. Valid refclk Range and Formulaic MPLL Settings provides a formulaic approach for setting the MPLL values. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous; must not be changed while MPLL is running **Override Ablility:** 17-bit ATEOVRD JTAG register, and 18-bit CREG_OVRD register; JTAG override takes precedence
mpll_ncy5[1:0]	I	MPLL x1 Multiplier **Function:** MPLL x1 multiplier. "Valid refclk Range and Formulaic MPLL Settings" on page 31 provides a formulaic approach for setting the MPLL values. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous; must not be changed while MPLL is running **Override Ablility:** 17-bit ATEOVRD JTAG register, and 18-bit CREG_OVRD register; JTAG override takes precedence
mpll_int_ctl[2:0]	I	MPLL Integral Bandwidth Control **Function:** Integral charge pump control. Must be set to 3'b000. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
mpll_prop_ctl[2:0]	I	MPLL Proportional Bandwidth Control **Function:** Proportional charge pump control. Must be set to 3'b111. Must be static except during a reset or when the MPLL is disabled. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
Related to generation of output clocks		

Table continues on the next page...

Table 64-24. Common Signal Pins (continued)

Signal	I/O	Description
cko_word	O	Output Clock Based on Reference Clock or MPLL Output **Function:** Output clock based on the reference clock or the MPLL output. The cko_word_con bus selects the clock output. This clock is held low when cko_word_con is set to 3'b1xx. The tx_ck signal that is provided to the SATA2 PHY must be a buffered version of this clock. While switching between the reference clock and the MPLL, word clock is performed glitchlessly. • 000: Buffered version of mpll_prescaler output • 001: Word-rate clock is produced (baud rate / 10 in Full-Rate mode or baud rate / 5 in Half-Rate mode) • 010: Two-word-rate clock is produced (baud rate / 20 in Full-Rate mode or baud rate / 10 in Half-Rate mode) • 1xx: Disabled. Zero output produced. **Voltage Range**: 0 V-core voltage **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** don't care
cko_word_con[2:0]	I	cko_word Output Select **Function:** Selects the clock to be driven on the cko_word output. Changing settings causes cko_word to transition from one clock to another glitchlessly. • 000: Buffered version of prescaler output clock • 001: MPLL frequency / 5 clock • 010: MPLL frequency / 10 clock • 1xx: Disabled **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
cko_alive	O	Keep-Alive Clock **Function:** Keep-alive clock based on the ref_clk prescaler output. This clock can be half or twice the reference clock's frequency based on the mpll_prescaler control. This low-rate clock can be set to always be available for circuitry that the ASIC needs to run continuously provided ref_clk is enabled. This signal is controlled by cko_alive_con[1:0]. **Note:** The cko_alive clock is not suitable as a reference clock for other cores. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** don't care

Table continues on the next page...

Table 64-24. Common Signal Pins (continued)

Signal	I/O	Description
cko_alive_con[1:0]	I	cko_alive Output Select **Function:** Selects the clock to be driven on the cko_alive output. Changing settings causes cko_alive to transition from one clock to another glitchlessly. • 00: Output is disabled • 01: Prescaler output • 10: Prescaler output / 16 (very-low-speed clock) • 11: Reserved **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
Related to resistor tuning		
rtune_do_tune	I	Do Resistor Tune **Function:** Rising edge triggers a resistor calibration. When the operation is complete, the op_done pin is transitioned. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous; must be held high for at least two cko_word cycles. **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
Analog controls		

Table continues on the next page...

Table 64-24. Common Signal Pins (continued)

Signal	I/O	Description
tx_lvl[4:0]	I	Transmit Level **Function:** Fine resolution setting of transmit signal level, common to all lanes connected to one Clock module. • 00000: 0.937 V • 00001: 0.947 V • 00010: 0.957 V • 00011: 0.966 V • 00100: 0.976 V • 00101: 0.986 V • 00110: 0.996 V • 00111: 1.005 V • 01000: 1.015 V • 01001: 1.025 V • 01010: 1.035 V • 01011: 1.045 V • 01100: 1.054 V • 01101: 1.064 V • 01110: 1.074 V • 01111: 1.084 V • 10000: 1.094 V • 10001: 1.104 V • 10010: 1.113 V • 10011: 1.123 V • 10100: 1.133 V • 10101: 1.143 V • 10110: 1.152 V • 10111: 1.162 V • 11000: 1.172 V • 11001: 1.182 V • 11010: 1.191 V • 11011: 1.201 V • 11100: 1.211 V • 11101: 1.221 V • 11110: 1.230 V • 11111: 1.240 V Peak-peak output level (without attenuation) is: 1.24 x (48 + tx_lvl / 2) / 63.5 V. Setting the maximum Tx amplitude to greater than 1 V peak-to-peak differential results in overdrive (applying 1.2 V) to the thin-gate output transistors. For the recommended setting, see Recommended Tx Settings.

Table continues on the next page...

Table 64-24. Common Signal Pins (continued)

Signal	I/O	Description
		Voltage Range: 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
os_lvl[4:0]	I	LOS Detection Threshold Control **Function:** Controls LOS detection threshold. For the recommended los_lvl setting, see Recommended Rx Settings. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
acjt_lvl[4:0]	I	ACJTAG Receiver Comparator Level Control **Function:** Controls the ACJTAG receiver comparator level. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
fast_tech	I	Fast Technology Flag **Function:** IP is processed in one of the faster process variants. This signal must be set to 1'b0. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** N/A **Override Ablility:** 17-bit ATEOVRD JTAG register, and 18-bit CREG_OVRD register; JTAG override takes precedence
vp_is_1p2	I	Low-Voltage Supply Setting **Function:** Specifies the voltage level of the low-voltage supply. This signal must be set to 1'b0. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** N/A **Override Ablility:** 17-bit ATEOVRD JTAG register, and 18-bit CREG_OVRD register; JTAG override takes precedence

Table continues on the next page...

Table 64-24. Common Signal Pins (continued)

Signal	I/O	Description
vph_is_3p3	I	High-Voltage Supply Setting **Function:** Specifies the voltage level of the high-voltage supply. This signal must be set to 1'b0. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** N/A **Override Ablility:** 17-bit ATEOVRD JTAG register, and 18-bit CREG_OVRD register; JTAG override takes precedence
Digital controls		
op_done	O	Operation Complete Indicator **Function:** Indicates that the requested operation is complete by transitioning in either direction. Operations that generate a transition of op_done include: • mpll_pwron assertion: Transitions after the MPLL is operating and is at the correct frequency. If enabled and based on the MPLL, cko_word starts toggling before op_done transitions. • mpll_pwron deassertion: Asynchronously transitions after cko_word is suspended. • rtune_do_tune assertion: Transitions when resistor tuning is complete. • reset deassertion: Transitions when resistor tuning and MPLL power-up is complete. If one of these operations is in progress, the next operation must not be triggered until op_done transitions. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register; JTAG override takes precedence
power_good	O	Power Supply Valid Indicator **Function:** Indicates that the analog, digital, and I/O power supplies have all reached 80% of their respective nominal values, provided the low supply is powered up first. If the high supply is powered up first, do not use this pin as an indicator. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register; JTAG override takes precedence
reset_n	I	Reset **Function:** Reset signal. Resets all registers in the SATA2 PHY to their default value. Holding reset_n low holds the SATA2 PHY in a reset state. **Voltage Range:** 0 V-core voltage **Active State:** Low **Synchronous to:** Asynchronous; must be held low for a minimum of 5 ns **Override Ablility:** 17-bit ATEOVRD JTAG register

Table continues on the next page...

Table 64-24. Common Signal Pins (continued)

Signal	I/O	Description
wide_xface	I	Wide Interface Enable **Function:** When this signal is asserted, the Tx and Rx data interfaces are 20 bits wide and rx_ck is (and tx_ck must be) periodic in 20-bit intervals. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
dtb_out[1:0]	O	Debug Test Bus Output **Function:** Routes internal signals to external pins. This bus can be ignored or to aid characterization-multiplexed to high-speed output drivers. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** don't care
Test controls		
test_byp_mode	I	Test Bypass Mode **Function:** This signal sets up the ASIC interface so that all inputs are connected to outputs through a purely combinatorial path. Use this signal to enhance test coverage of the SATA2 PHY/ASIC interface. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous **Override Ability:** none - hold at 0 for test
test_burnin_mode	I	Test Burn-In Mode **Function:** Enables the SATA2 PHY to toggle as much circuitry as possible and to bias unused circuitry so that devices do not degrade asymmetrically. Use this signal in burn-in testing in conjunction with the test_byp_mode signal. When this pin is asserted, no other clocks are necessary, the external signal pins may be floated, and the ASIC-side signal pins do not matter. However, it is recommended to externally loop the transmitter back to the receiver through the appropriate DC blocking capacitors. This ensures that the receiver circuitry also toggles. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous **Override Ability:** none - hold at 0 for test

Table continues on the next page...

Table 64-24. Common Signal Pins (continued)

Signal	I/O	Description
test_pddq	I	Test IDDQ **Function:** Powers down all circuitry for IDDQ testing. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous **Override Ability:** none - hold at 0 for test

64.5.2.6 JTAG Interface Signals

The SATA2 PHY provides a JTAG interface that is compliant with *IEEE Std 1149.1*.

The table below describes the interface signals.

ATE and bench characterization features are accessed through internal registers using either the JTAG interface or Parallel CR Control port. Normal PHY startup and operation do not require using the JTAG interface.

Table 64-25. JTAG Signals

Signal	I/O	Description
tck	I	JTAG Clock **Function:** JTAG interface clock. Maximum frequency is 35 MHz. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** N/A **Override Ablility:** none; used for ATE testing
tms	I	JTAG State Machine Control **Function:** JTAG test mode select pin **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** tck **Override Ablility:** none; used for ATE testing
tdi	I	JTAG Data Input Port **Function:** JTAG test data input pin **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** tck **Override Ablility:** none; used for ATE testing

Table continues on the next page...

Table 64-25. JTAG Signals (continued)

Signal	I/O	Description
tdo	O	JTAG Output Port Data Value **Function:** JTAG test data output pin **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** tck **Override Ablility:** don't care
tdo_en	O	JTAG Output Enable **Function:** JTAG test data output enable pin. For compliance with the JTAG specification, the ASIC's test data out (TDO) output must be tri-stated when tdo_en is low. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** tck **Override Ablility:** don't care

64.5.2.7 Parallel CR Control Port Signals

ATE and bench characterization features are accessed through internal registers using either the JTAG interface or Parallel CR Control port. Normal PHY startup and operation do not require using the Parallel CR Control port.

Table 64-26. Parallel CR Control Port

Signal	I/O	Description
cr_data_in[15:0]	I	CR Address and Write Data Input **Function:** Supplies and captures address and write data. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous **Override Ablility:** none - don't care when not performing CR operation
cr_data_out[15:0]	O	CR Data Output **Function:** Always presents last read data. **Voltage Range:** 0 V-core voltage **Active State:** N/A **Synchronous to:** Asynchronous **Override Ablility:** don't care

Table continues on the next page...

Table 64-26. Parallel CR Control Port (continued)

Signal	I/O	Description
cr_cap_addr	I	CR Capture Address **Function:** Captures cr_data_in into the Address register. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
cr_cap_data	I	CR Capture Data **Function:** Captures cr_data_in into the wrdata register. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
cr_write	I	CR Write **Function:** Writes the wrdata register to the referenced Address register. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
cr_read	I	CR Read **Function:** Reads from the referenced Address register. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous **Override Ablility:** 18-bit CREG_OVRD register accessed using crsel instruction
cr_ack	O	CR Acknowledgement **Function:** Acknowledgement for the cr_cap_addr, cr_cap_data, cr_write, and cr_read control signals. **Voltage Range:** 0 V-core voltage **Active State:** High **Synchronous to:** Asynchronous **Override Ablility:** don't care

64.6 Timing and Specifications

64.6.1 SATA2 PHY Implementation-Specific Timing

The table below describes the timings for a standalone SATA2 PHY.

Table 64-27. SATA2 PHY Implementation-Specific Timing

Name	Description	Value
Transmit latency	Measured from when the 10-bit data is sampled to when the first word bit is serially transmitted.	22-24 bit times
Transmit latency	Measured from when the 10-bit data is sampled to when the last word bit is serially transmitted.	• 10-bit wide interface: 31-33 bit times • 20-bit wide interface: 41-43 bit-times
Receive latency	Measured from when the first bit of a 10-bit word is received to when the 10-bit word is available to be sampled at the ASIC interface. Alignment can be turned off.	• 10-bit wide interface: 31-40 bit times • 20-bit wide interface: 41-50 bit times
Receive latency	Measured from when the last bit of a 10-bit word is received to when the 10-bit word is available to be sampled at the ASIC interface. Alignment can be turned off.	22-31 bit times
refclk spin-up latency	Time from deassertion of a reset (assuming mpll_ck_off is deasserted) or deassertion of mpll_ck_off to when the reference clock is available for the MPLL and cko_alive.	< 1.65 µs (or 31 µs if refclk doubler is used)
MPLL lock latency	Time from assertion of mpll_pwron (assuming refclk is on and stable) to when the MPLL is ready and cko_word begins toggling.	< 41 µs
Resistor tuning	Time to perform a resistor tune	2 µs
Receive PLL lock latency	Time from assertion of rx_pll_pwron (assuming the MPLL is ready) to when rx_pll_state is asserted. Does not require incoming data transmission.	9 µs
Receive CDR lock time	Time from end of loss of signal or assertion of rx_en to assertion of rx_valid. These are worst- case values; nominal lock times might be shorter. Assuming 30 kHz minimum frequency spread spectrum clocking. Non-spread spectrum receiver lock will be significantly shorter.	37 µs
Transmit enable latency	Time from tx_en change to serial lines leaving common mode.	• CM to ON: 300 ns • CM_CLK to ON: Same as transmit latency
LOS detection latency	For information about loss of signal detection, see Loss of Signal Detection.	20 ns

64.6.1.1 Synchronous Tx Inputs

The Tx input, tx_data, is clocked into the Tx block at the rate of either baud / 10 or baud / 20 (based on the wide_xface setting) with a source-synchronous clock, tx_ck (that must be derived from either tx_cko_word or cko_word).

The figure below shows timing for the Tx inputs.

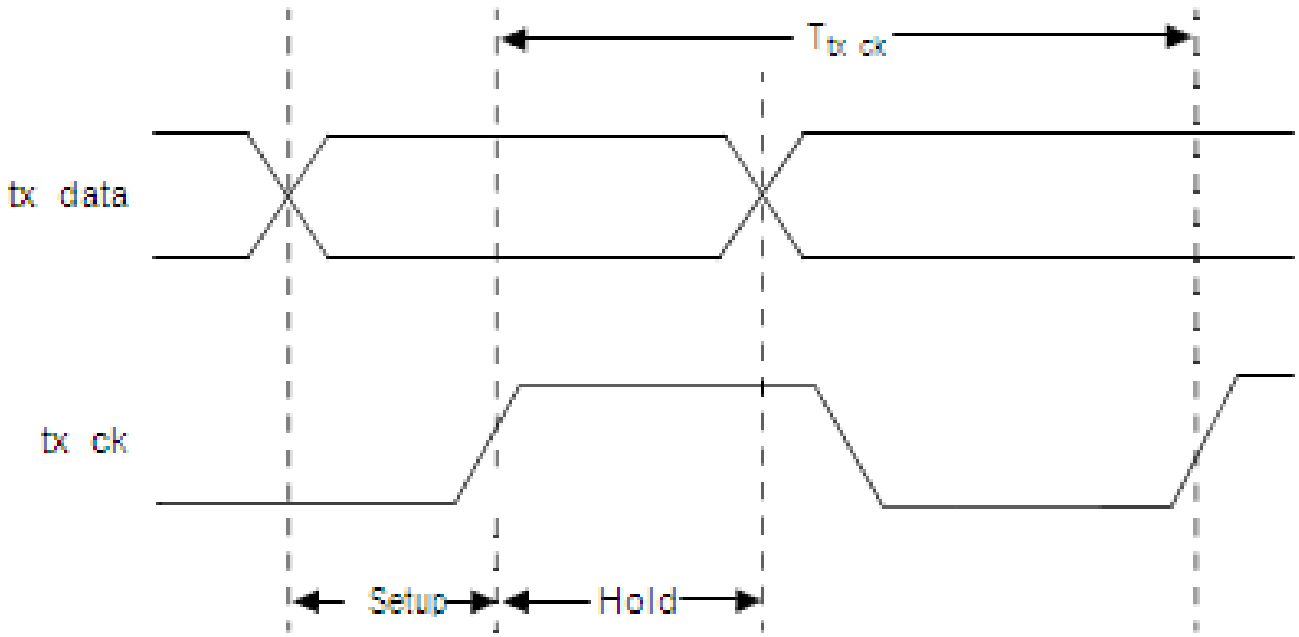


Figure 64-4. Parallel Input Tx Timing

64.6.1.2 Synchronous Rx Outputs

The Rx output data, rx_data, is generated with the rising edge of rx_ck (that has a Trx_ck of either 10 or 20 baud based on wide_xface). The data is stable based on the rising edge of this clock.

The figure below shows timing for the Rx outputs.

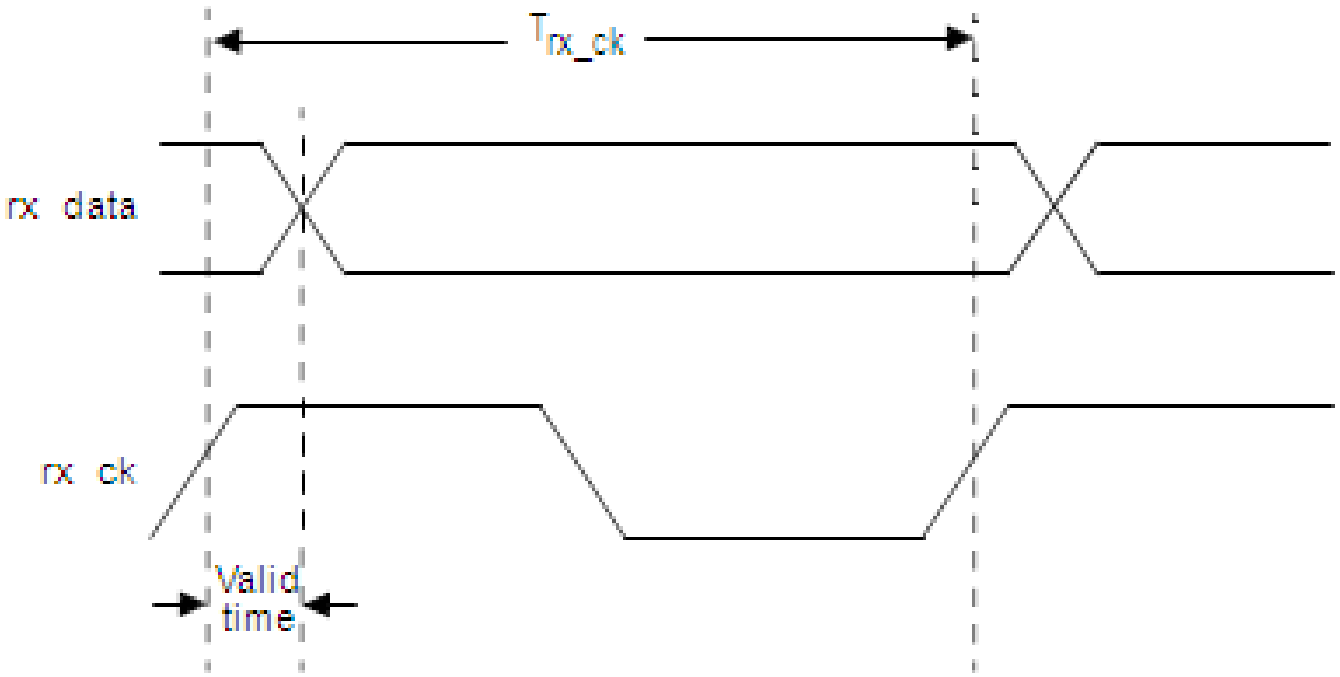


Figure 64-5. Parallel Output Rx Timing

64.6.1.3 Asynchronous Tx and Rx I/O

There are no static timing requirements for inputs or outputs noted as asynchronous in Table 64-23. The inputs take a short time to propagate to the associated circuitry that they control. Signals that do not have restrictions specified in Table 64-23 have no high or low pulse-width requirements. All outputs must be either synchronized before using or sampled appropriately for the pin's function.

64.6.1.4 Control Register Bus Interface

Timing of the Parallel Control Register bus is accomplished with a standard four-phase asynchronous handshake using the four CR request lines and the cr_ack signal.

For information about using this interface, see Parallel CR Control Port Testing.

Because the interface is asynchronous, it does not need to be timed with a static timing tool. There is zero setup time from cr_data_in[15:0] to the rising edge of cr_cap_{addr, data}. There is zero hold time from the rising edge of cr_ack to cr_data_in[15:0].

64.6.1.5 JTAG Interface Timing

The JTAG interface complies with interface pin timings as defined in the JTAG specification; therefore, you should not have timing issues interfacing this port with other internal or external JTAG implementations.

The tck clock signal has a maximum frequency of 35 MHz, but no minimum frequency. This clock can be stopped at any point without loss of state. The figure below shows timing for the JTAG interface.

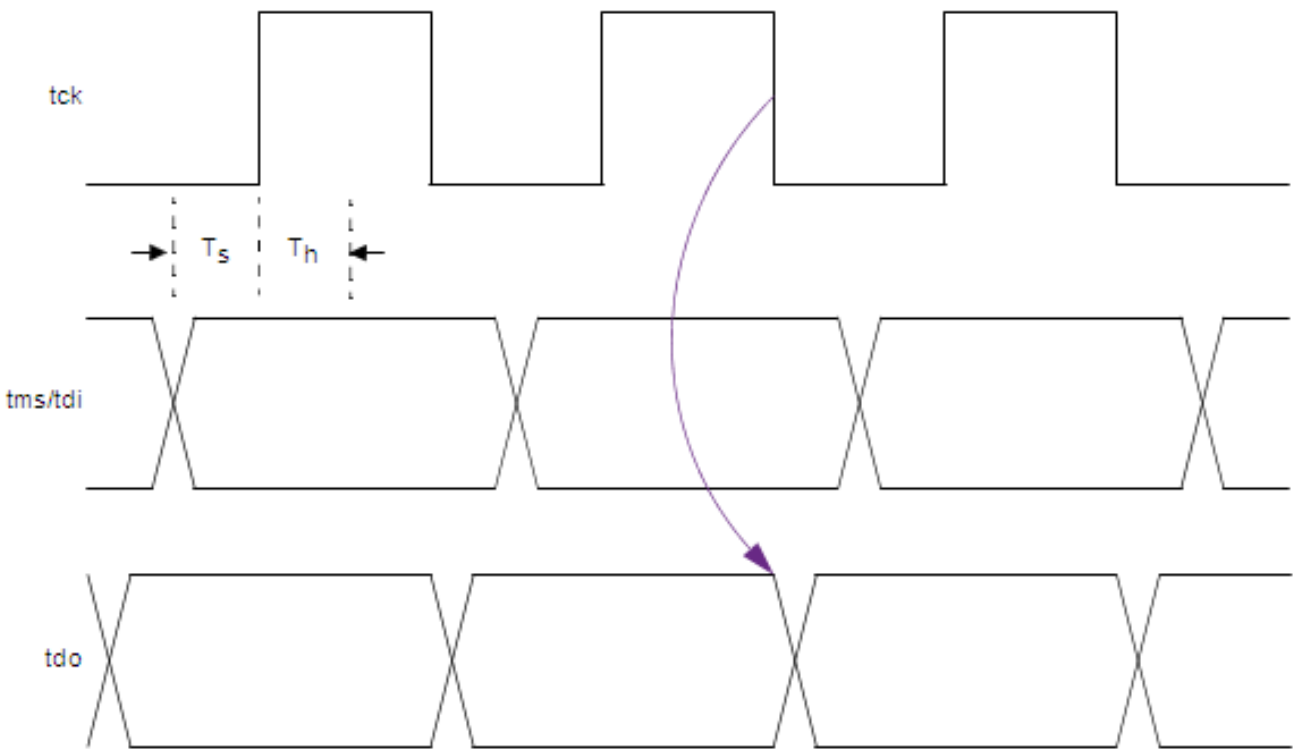


Figure 64-6. JTAG Interface Timing

NOTE

Setup (Ts) and hold (Th) times for tms and tdi are defined in the .lib files. The tdo signal transitions on the falling edge of tck.

The maximum frequency of operation for the JTAG port is 35 MHz. For detailed timing information, refer to the .lib files in the database deliverables.

If you are using the JTAG interface for off-chip connections, you must instantiate appropriate Low Voltage Transistor-Transistor Logic (LVTTL) input buffers in the ASIC for the JTAG inputs. To drive the tdo pin in compliance with the JTAG specification, you must also instantiate a tri-state output buffer.

64.6.2 Silicon Testing

This section describes SATA2 PHY interfaces and features that can be used in silicon testing.

64.6.2.1 Boundary Scan Port

The Boundary Scan port on the SATA2 PHY enables any ASIC using the macro to be fully compliant with the 1149.1-2001 and 1149.6-2003 standards.

The SATA2 PHY contains scannable, sequential elements to support ACJTAG and boundary scan. These elements are connected in a scan chain and made available on the ASIC interface. Concatenate this scan chain with any other scan chains on the ASIC.

64.6.2.1.1 Per-Lane Block Diagram

The figure below shows the Boundary Scan circuitry included in each lane. The signals to control the Boundary Scan circuitry are derived from the Scan Port interface pins. There are no boundary scan cells in the Clock module.

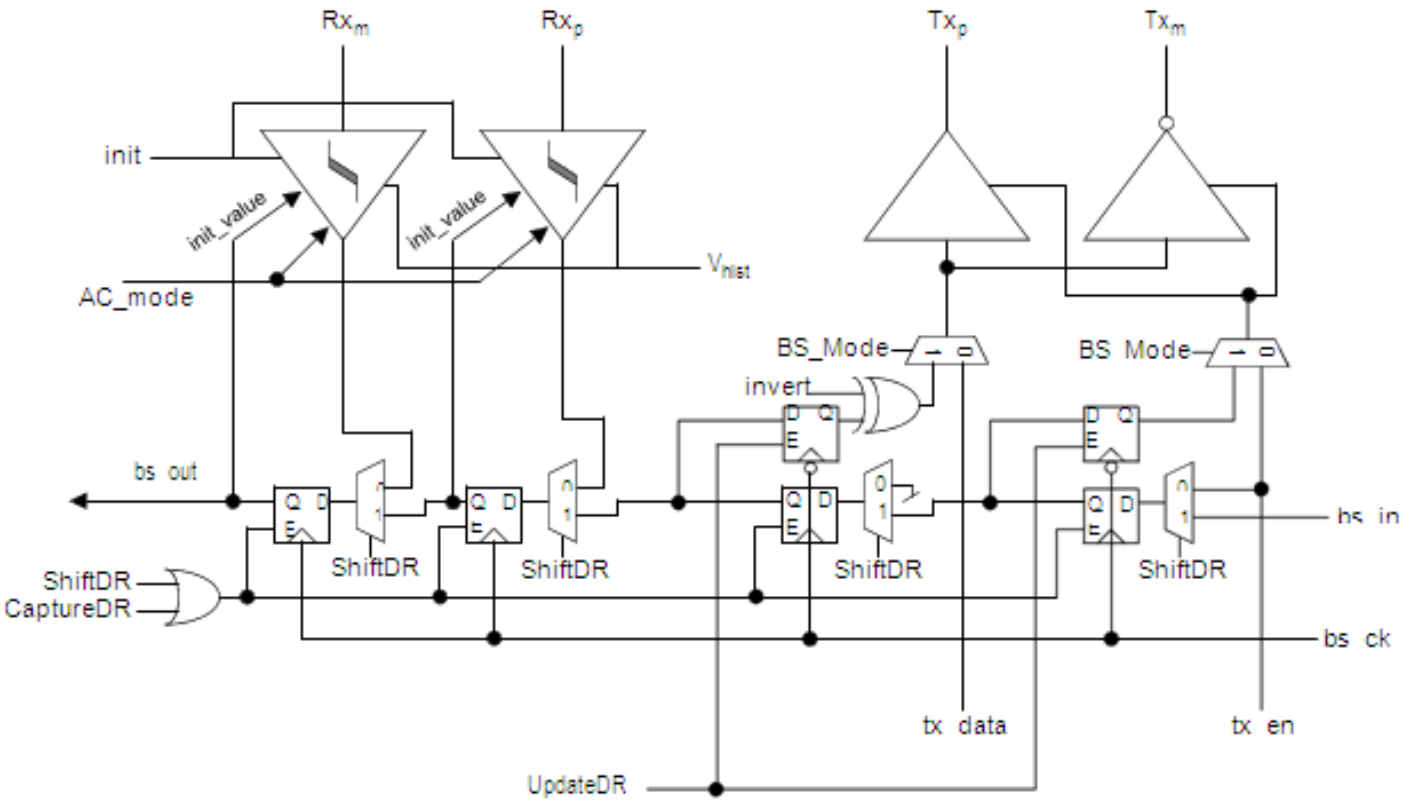


Figure 64-7. Per-Lane ACJTAG Details

The following figures show the function of the comparator associated with the receive pins shown in the previous figure.

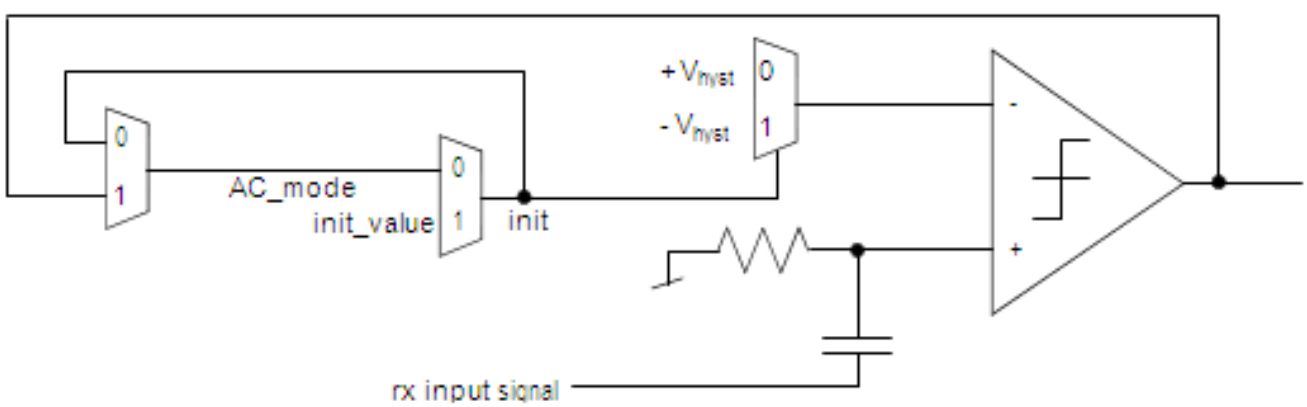


Figure 64-8. Comparator in AC mode

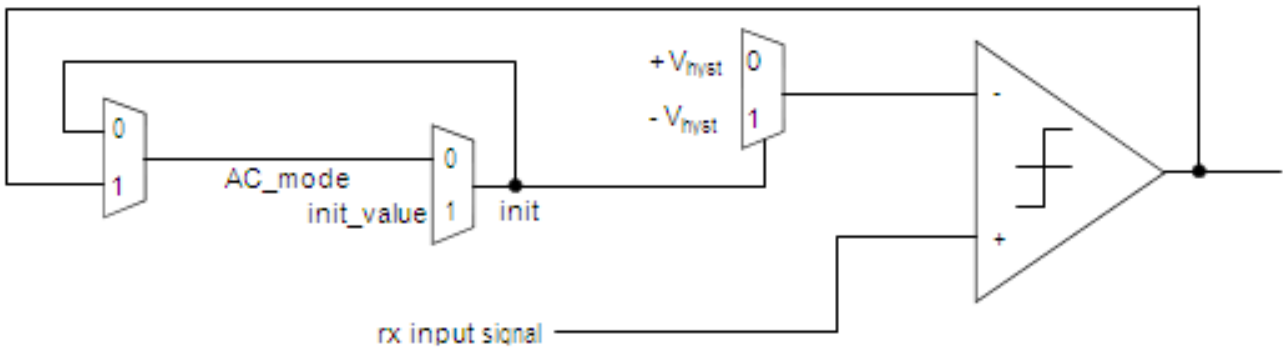


Figure 64-9. Comparator in DC mode

64.6.2.2 JTAG Interface Silicon Testing

The SATA2 PHY's JTAG interface provides access to an internal TAP controller implementing 1149.6-2003- compliant IDCODE and USERCODE instructions.

In addition, the CRSEL instruction is available for writing and reading registers in the SATA2 PHY. This instruction provides access to features available for bench characterization and ATE (manufacture) testing. Read and/or write register access for normal operations is neither necessary nor recommended.

64.6.2.2.1 Interface Options

The SATA2 PHY's JTAG interface excludes the optional TRSTb reset pin as defined in the 1149.6-2003 specification.

64.6.2.2.2 Resets

Entry into the TEST-LOGIC-RESET state clears the IR register and DR-shift register; no other registers are affected.

Note that due to the JTAG state machine's topology, TEST-LOGIC-RESET is entered regardless of the JTAG machine's current state-by clocking in at least five consecutive 1s on TMS.

After a power-on reset, TEST-LOGIC-RESET is the default state for the JTAG state machine.

64.6.2.2.3 IR Codes

The table below lists the supported JTAG instructions. The least-significant bit (LSB) of the IR and DR registers is the first bit shifted into TDI.

Table 64-28. Supported JTAG Instructions

Instruction	IR	DR	Operation
BYPASS	All others	1 bit	TDI to TDO bypass (bit is preloaded to 1)
IDCODE	8'h01	32 bits	Shift out ID Code
USERCODE	8'h0d	32 bits	Shift out USER Code
CRSEL	8'h31	18 bits	Control register access (see Control Register Operations)
DSCAN	8'h51	TBD bits	MUX-D scan testing
ATEOVRD	8'h5d	17 bits	Override reset, MPLL settings, and technology-specific parameters

64.6.2.2.4 ID Code

The table below provides the 32-bit JTAG ID code for the SATA2 PHY.

Table 64-29. JTAG ID for the SATA2 PHY

Bit Range	Signal	Value
31:16	ID_VAL_HI	1017
15:0	ID_VAL_LO	74CD

64.6.2.2.5 USER Code

The 32-bit JTAG USER code reflects the hard-wired programming of the SATA2 PHY in an ASIC. Various bits reflect the status of ASIC interface pins defined in Per-Transceiver Control and Status Signals. The table below provides the register mapping for the USER code.

Table 64-30. JTAG USERCODE

Bit Range	Signal
31:0	000, use_refclk_alt, mpll_prescale[1:0], mpll_ncy[4:0], mpll_ncy5[1:0], wide_xface, fast_tech, vp_is_1p2, vph_is_3p3, tx_lvl[4:0], los_lvl[4:0], acjt_lvl[4:0]

64.6.2.2.6 Control Register Operations

Internal registers can be accessed through the JTAG interface when operating with the CRSEL instruction (for a list of supported JTAG instructions, see Table 64-28).

The JTAG controller implements two shadow registers that access the macro's register space: an address register (Addr_reg) and data register (Data_reg).

Addr_reg is loaded with the address of the register to be accessed. Data_reg holds data to be written to or data that has been read from the addressed register. Four operations involving Addr_reg and Data_reg and the reading and writing of registers are performed with the shifting of 18 bits (16 bits of data or address and 2 bits of opcode) through the DR scan path of the JTAG state machine. The first 16 bits that are shifted out when in Shift-DR are the current contents of the Data_reg register for any operation (before the register read occurs).

Data_reg is also updated with the 16 bits of address or data shifted in as part of one of the following operations.

- Address (op = 2'b00): Addr_reg and Data_reg are loaded with the 16 bits of address shifted in.
- Write (op = 2'b01): Data_reg is loaded with the 16 bits of data shifted in. The register addressed by the contents of Addr_reg is written with this value.
- Read (op = 2'b11): Addr_reg is loaded with the 16 bits of address shifted in. A read of the register addressed by the contents of Addr_reg is initiated. The read data is latched into Data_reg with the passing through of the Capture-DR state. Therefore, a subsequent operation is required to shift out the read data.
- Data (op = 2'b10): This No Operation (NOP) operation can be used to shift out the current contents of Data_reg (for example, after a read operation). Data_reg is consequently loaded with the 16 bits shifted in.

NOTE

A standalone register read requires two passes down the DR scan path to actually do the read and retrieve the data. However, the second pass can be overlapped or effectively pipelined with a subsequent operation.

64.6.2.2.7 JTAG Override Register (jtag_ovrd)

Reset, MPLL controls, and technology-specific parameters can be overridden using the JTAG interface.

The Instruction register (ATEOVRD)-at JTAG IR address 8'h5d-enables the JTAG Override register to be written to

Reset Value: 17'b0 0000 0000 0000 0000

Table 64-31. JTAG Override Register (jtag_ovrd)

Field Name	Bits	Description
jtag_ovrd	0	JTAG override. Enables control of this register. Active high.
reset	1	Reset override. Active high.
mpll_ck_off	2	Reference clock stop enable override. Active high.
mpll_pwron	3	MPLL power-on override. Active high.
use_refclk_alt	4	Alternate reference clock control override. Active high.
mpll_prescale	6:5	Reference clock prescaler control override. For more information, see the mpll_prescale[1:0] signal description in Table 64-24.
mpll_ncy	11:7	MPLL x4 multiplier override. For more information, see the mpll_ncy[4:0] signal description on Table 64-24.
mpll_ncy5	13:12	MPLL x1 multiplier override. For more information, see the mpll_ncy5[1:0] signal description on Table 64-24.
fast_tech	14	Fast technology flag override. For more information, see the fast_tech signal description on Table 64-24.
vp_is_1p2	15	Low-voltage supply is 1.2-V override. For more information, see the vp_is_1p2 signal description on Table 64-24.
vph_is_3p3	16	High-voltage supply is 3.3-V override. For more information, see the vph_is_3p3 signal description on Table 64-24.

64.6.2.3 Parallel CR Control Port Testing

ATE and bench characterization features are accessed through internal registers via the JTAG interface or Parallel CR Control port. Normal SATA2 PHY startup and operation do not require the use of neither the JTAG interface nor the Parallel CR Control port.

The Parallel CR Control port comprises two internal registers (Address and Data) as well as control signals to manipulate these registers and to use them to perform control register reads and writes. The cr_data_in signal can be used to write to either the Address or Data register (depending on the control signals). The cr_data_out signal is always the Data register's output.

Communicating with the control port itself is accomplished with a standard four-phase asynchronous handshake using the four CR request lines and the cr_ack signal.

This protocol can be summarized by the following sequence.

1. The ASIC waits until the cr_ack signal is deasserted from a previous operation. At this point, all request signals are deasserted.

2. The ASIC sets up data on cr_data_in (if necessary) and asserts the target request signal.

3. After capturing the data and completing the requested operation, the SATA2 PHY sets up data on cr_data_out (if necessary) and asserts the cr_ack signal.

4. After capturing data from cr_data_out (if necessary), the ASIC deasserts the request signal.

5. The SATA2 PHY deasserts the cr_ack signal to indicate that the SATA2 PHY is ready for the next request.

64.6.2.3.1 Addressing

Before reading or writing a register, an address must be supplied and latched into the port's Address register.

This transaction requires the following steps.

1. Supply the address on cr_data_in[].

2. Assert the cr_cap_addr signal.

3. Wait for the cr_ack signal to be asserted.

4. Deassert cr_cap_addr.

5. Wait for cr_ack to be deasserted.

The following figure shows timing for this transaction.

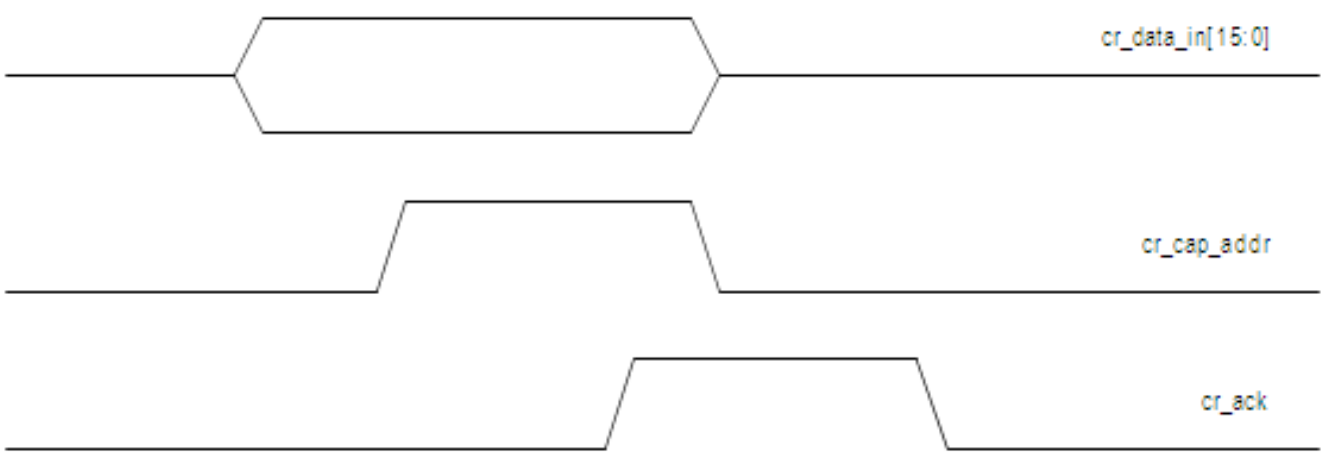


Figure 64-10. Capture Address Transaction (Precedes Read or Write)

64.6.2.3.2 Register Write

Writing to an internal register requires providing the write data and latching it into the port's Data register, then executing the write itself.

This transaction requires the following steps.

1. Supply the data on cr_data_in[].
2. Assert the cr_cap_data signal.
3. Wait for the cr_ack signal to be asserted.
4. Deassert cr_cap_data.
5. Wait for cr_ack to be deasserted.
6. Assert the cr_write signal.
7. Wait for cr_ack to be asserted.
8. Deassert cr_write.
9. Wait for cr_ack to be deasserted.

The following figure shows timing for this transaction.

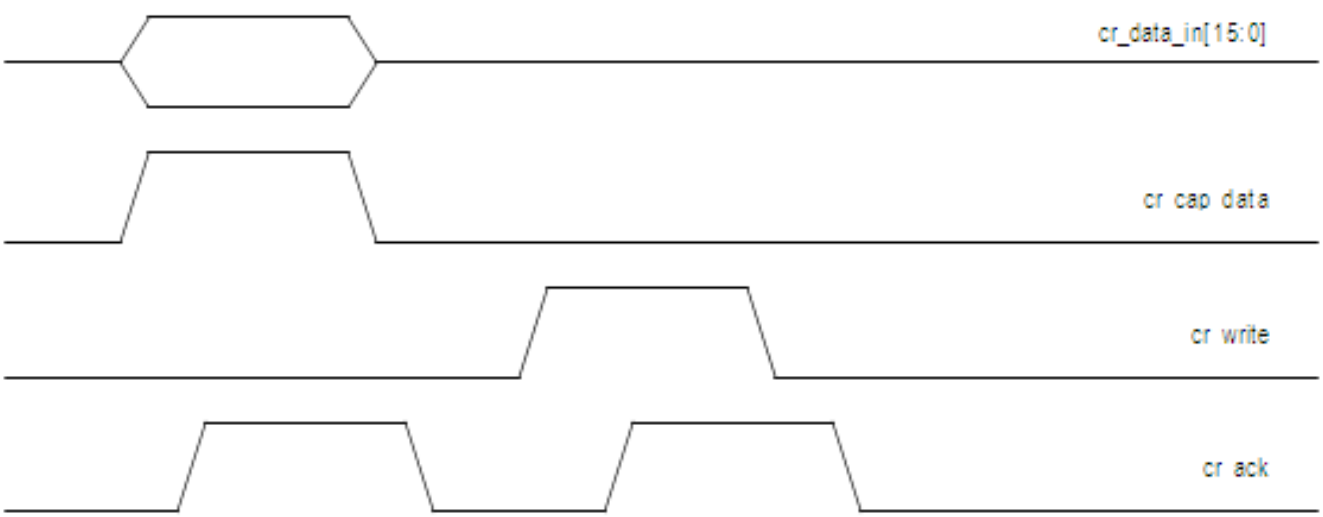


Figure 64-11. Write Transaction

64.6.2.3.3 Register Read

Reading from an internal register requires the following steps (to latch the read value into the port's Data register).

1. Assert the cr_read signal.
2. Wait for the cr_ack signal to be asserted.
3. Capture the data from cr_data_out[].
4. Deassert cr_read.
5. Wait for cr_ack to be deasserted.

The figure below shows timing for this transaction.

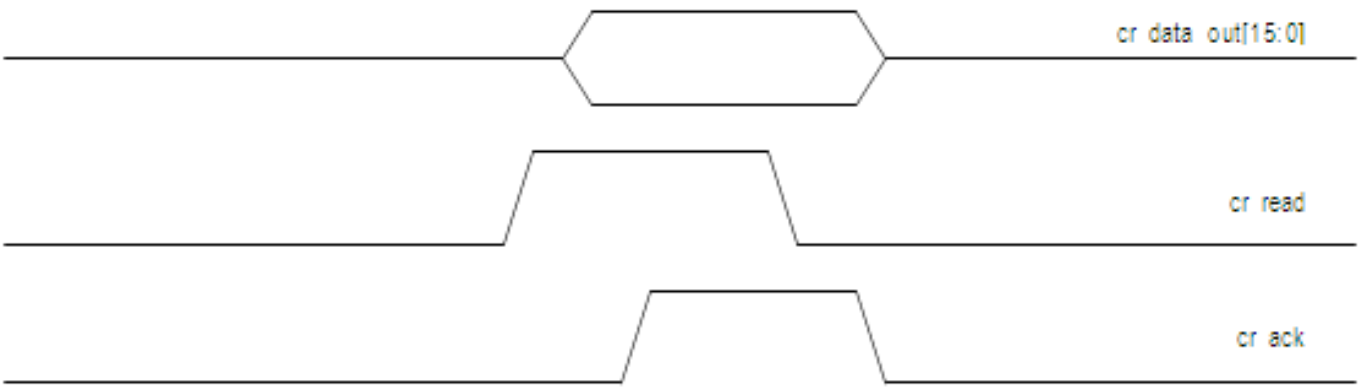


Figure 64-12. Read Transaction

64.6.2.4 Diagnostic Features

The SATA2 PHY provides the following diagnostic features to enhance bench characterization and ATE testing.

- Loopback functions:
- Digital serial loopback
- Serial loopback for wafer probe only
- For information about the loopback functions, see Loopback Functions.
- Asynchronous operation in a synchronous test environment for enhanced coverage of functional tests: For information about asynchronous operation, see Asynchronous Operation.
- BERT independent per lane:
- 7th- and 15th-order polynomial pattern generation and recognition
- Generation of simple test patterns
- Byte error counting from polynomial pattern
- For information about the BERT, see Byte Error Rate Tester.
- Margining:

- Voltage margining with 10-bit resolution, synchronous or asynchronous operation
- Phase margining with sub-ps resolution, synchronous or asynchronous (when number of lanes is greater than one)
- For information about margining, see Margining.
- High-resolution scope per Rx signal pair:
- Acquisition of eye or signal from 7th- or 15th-order polynomial patterns
- For information about scope function, see Scope Function.
- Analog DC test:
- Signal selection matrix embedded in the SATA2 PHY
- 10-bit A/D converter
- Selection and measurement of any individual Rx and Tx termination resistors
- For information about the SATA2 PHY's analog DC test capabilities, see Analog DC Test Capabilities.
- Limit testing:
- Specification of high/low limits
- Determination of whether register read was within limits
- Determination of whether difference between register read values was within limits
- For information about limit testing, see Limit Testing.
- Integrated test modes:
- Bypass test mode
- Burn-in test mode
- IDDQ test mode
- For information about the SATA2 PHY's integrated test modes, see Integrated Test Modes.
- Temperature monitor:
- Real-time die temperature
- No extra pins required, access is through JTAG interface or Parallel CR Control port.
- For information about the temperature sensor, see Temperature Sensor.

64.6.2.4.1 Loopback Functions

The figure below depicts the SATA2 PHY's loopback functions.

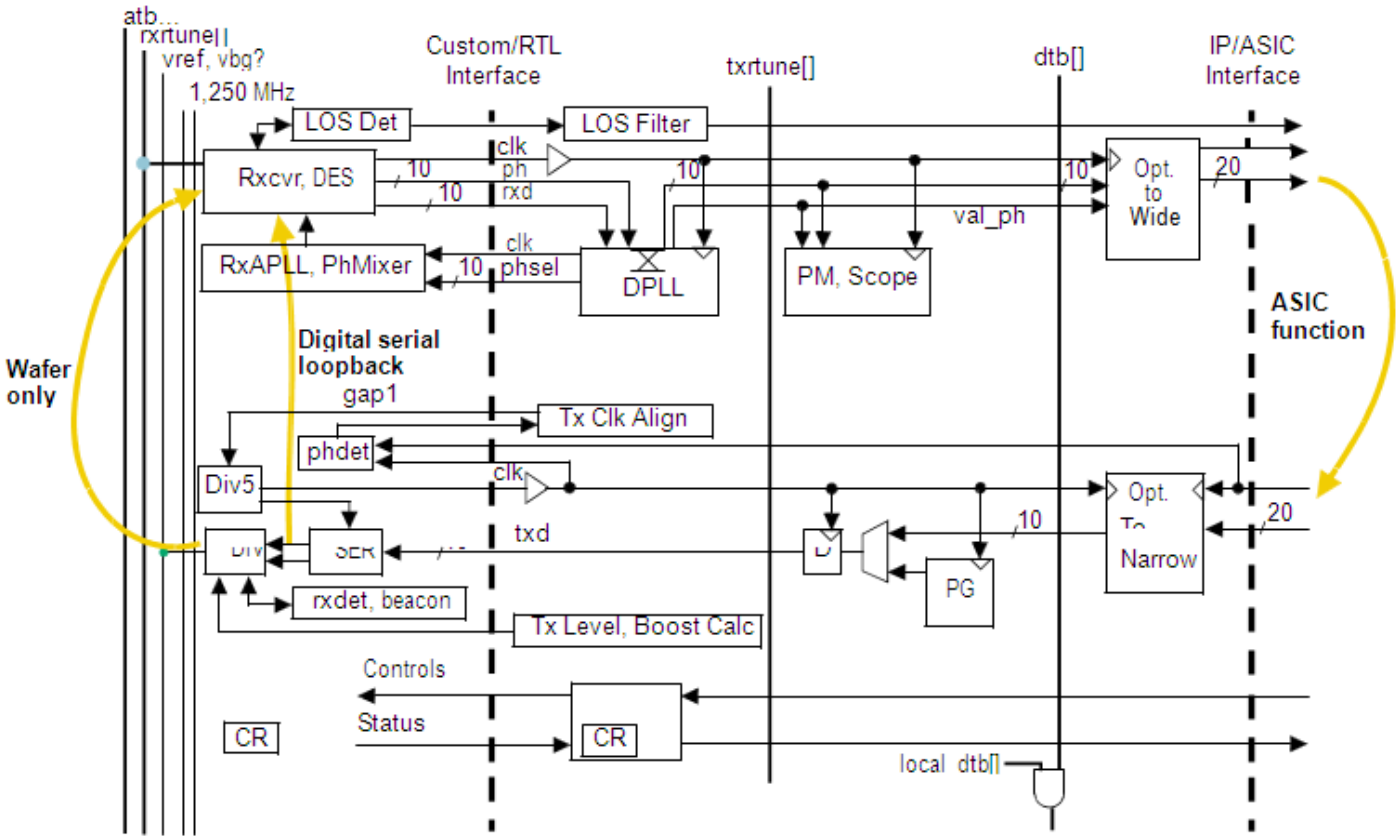


Figure 64-13. Loopback Functional Diagram

64.6.2.4.1.1 Rx-to-Tx Parallel Data Loopback

This loopback function must be provided by the ASIC.

64.6.2.4.1.2 Tx-to-Rx Digital Serial Data Loopback

This loopback function selects serial data located just before the Tx driver and inserts the data into the Rx just after the first stage of the receiver.

This function enables verification of the entire datapath, except for actual output drivers and the Rx's first stage. This loopback function is useful for testing the ASIC and is activated by asserting the rxlbi_en field in the Receive Analog Control register (lanej.rx_ana.ctrl), as indicated in Receive Analog Control Register (lane0_RX_ANA_CONTROL).

64.6.2.4.1.3 Tx-to-Rx Serial Analog Loopback

This loopback function is intended to be used only in manufacturing test at wafer probe. Without wafer probes landing on the Tx or Rx serial differential pads, this loopback function tests the entire transceiver at wafer test, where providing loopback traces with sufficient signal integrity through a probe card is typically impossible.

This loopback function is activated by asserting the rxlbe_en field in the Receive Analog Control register (lanej.rx_ana.ctrl.rxlbe_en), as indicated in Receive Analog Control Register (lane0_RX_ANA_CONTROL).

Using this loopback function with typical packaged parts (even without connections to the Tx and Rx serial differential pins) would provide too much parasitic capacitance loading to enable operation at speed.

64.6.2.4.1.4 Full Analog Loopback for In-Package ATE Test

This loopback function requires the use of loopback traces in the ATE load board.

64.6.2.4.2 Asynchronous Operation

ATE test environments are typically synchronous, where only a single reference clock is available.

In addition, when loopbacks (internal to the PHY or wired into a package load board) are used in production test, only a single Clock module is typically part of the test setup; therefore, only synchronous operation is possible. However, a variety of circuit defects can exist that might have little or no effect on CDR operation when operating synchronously.

The SATA2 PHY addresses this coverage gap for multi-lane PHYs. Each pair of lanes is interconnected with the ability to use the recovered clock of the other member of a pair as reference for receive CDR.

One member (master) of the pair of lanes is programmed to ignore the received signal and produce a specified frequency offset by a specified number of ppm from the actual reference clock. The other member (slave) of the pair uses the synthesized offset clock as reference.

While receiving data synchronous to the actual reference clock, the slave's CDR must constantly slew to maintain a fixed recovered clock phase despite the offset frequency of the slave's reference. This task exercises the slave's DPLL and phase mixer in the same way they would operate during normal, asynchronous operation. Margining tests performed under this condition reveal any defects in the slave's CDR.

64.6.2.4.3 Byte Error Rate Tester

The Byte Error Rate Tester (BERT) comprises two independently programmed modules: a pattern generator and a pattern matcher/error counter.

64.6.2.4.3.1 BERT Pattern Generator

The following table provides a list of patterns that can be generated with the built-in pattern generator.

Table 64-32. Pattern Generator mode[2:0] Control

mode[2:0]	Description
000	Pattern generator is disabled.
001	15th order polynomial: $X^{15} + X^{14} + 1$
010	7th order polynomial: $X^7 + X^6 + 1$
100	Fixed 8- or 10-bit pattern from bottom of PAT0 field
101	2-byte DC balanced pattern constructed as {PAT0, ~PAT0}
111	4-byte DC balanced pattern constructed as {0x000, PAT0, 0x3FF, ~PAT0}

64.6.2.4.3.2 BERT Pattern Matcher and Error Counter

The pattern matcher is capable of synchronizing to and detecting errored bytes in multiple types of patterns. Errored bytes are counted in the error counter.

Note that there is no dependence on the pattern generator in the same lane. That pattern generator does not need to be enabled nor programmed for the same pattern.

The following table describes the mode[2:0] field, which selects the expected pattern and operating mode.

Table 64-33. mode[2:0] Control of the Pattern Matcher

mode[1:0]	Description
000	Disabled
001	lfsr15
010	lfsr7
011	d[n] = d[n-10]
100	d[n] = !d[n-10]
101	Reserved
110	Reserved
111	Reserved

NOTE

When using either the LFSR7 or LFSR15 pattern, the rx_align_en needs to be de-asserted at the pin or through the register override.

For modes 1 and 2, the pattern matcher operates by generating the expected pattern and synchronizing the generated pattern to the incoming pattern. Synchronization and error counting are initiated by asserting and deasserting the sync bit. Therefore, during synchronization in high-error-rate conditions, synchronization can fail if an error occurs during the last 15 bits prior to clearing of the sync bit.

The error counter is 22 bits wide.

The contents are presented according to these rules:

- When the contents are less than 2^{15}, the bottom 15 bits are presented in the count field and the ov14 field is cleared.
- When the contents are greater than 2^{22} but less than or equal to 2^{15}, the top 15 bits are presented in the count field and the ov14 field is set. This result effectively scales the contents down by a factor of 2^{7}.
- When the 22-bit counter overflows, the count field is set to 0x7FFF, and the ov14 field is set.

The error counter is typically used in a "clear after read" mode. The error registers are reset on a read-only when the pattern matcher is enabled. If the pattern matcher is disabled, the registers return the error count that was indicated when the pattern matcher was disabled and never reset the error counter.

64.6.2.4.4 Margining

Margining is the process of learning about the quality of a communications system by studying how its BER is changed when known impairments are introduced to the system.

Typical BER targets for systems are extremely low (for example, 1e-18; at 2.5 Gbps; that is, one error every 12.7 years). Margining makes it possible to extrapolate from a short set of measurements to project bounds on the actual error rate, assuming there are no additional sources of low-rate errors (for example, metastability), which do not occur during the measurement, but would dominate the BER over large periods.

One common margining technique-phase margining-is referred to as a "bathtub curve." The figure below shows an example of this technique, where the extrapolated phase margin is a measurement of margin between the extrapolated system performance and the performance target.

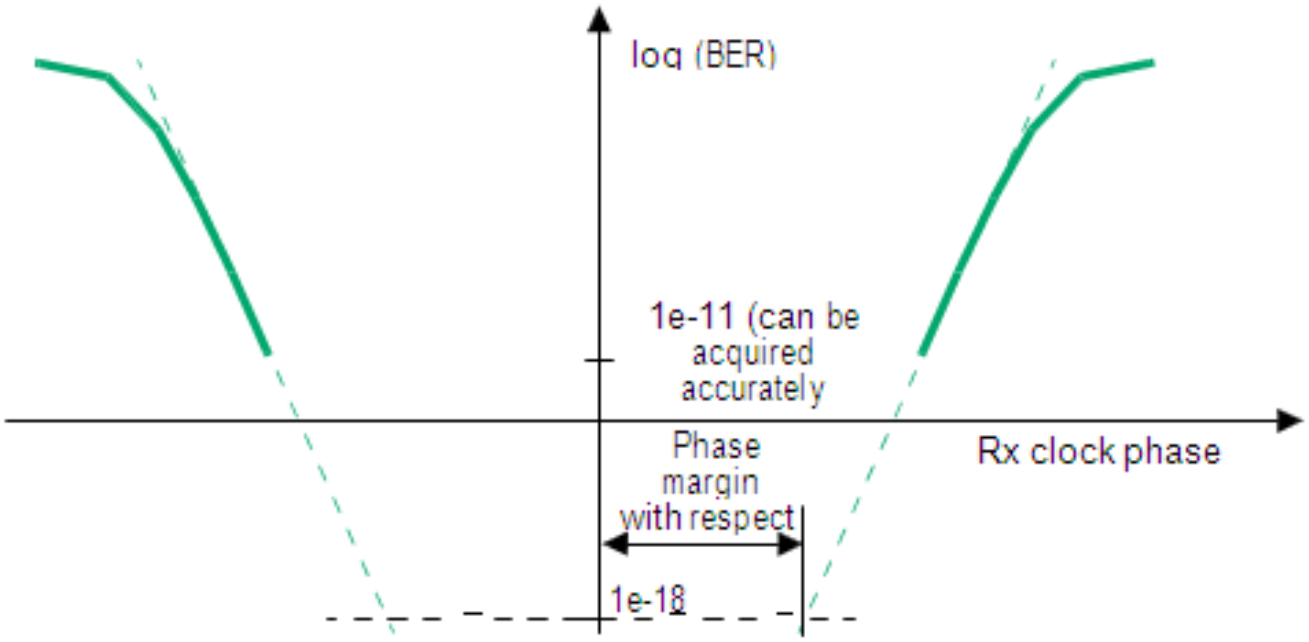


Figure 64-14. Example of Phase Margining Technique

Margining can be used on the testbench to study the effects of other changes (for example, crosstalk, trace lengths, power supply noise, temperature, power supply voltage) on system performance.

In ATE testing, during test time intervals in the order of 100 ms, simple error counting can pass defective parts that might produce BER results in the order of 1e-8. However, during that same period, margining can be used to provide a better bound on the performance of shipped parts.

64.6.2.4.4.1 Phase Margining

Phase margining is more common than voltage margining, but phase margining requires you to turn off (or freeze) the receiver's clock recovery-an important receiver function. Generally, to perform phase margining, you must ensure that the transmitter and receiver are operating from a common clock.

Synchronous phase margining requires the following steps.

1. 1. Provide a common clock to the transmitting and receiving devices.

   2. Determine the mean selected phase of the receiver's CDR by reading the DPLL Phase register (lanej.phase)-of the lane to be margined-a number of times and averaging the values.

3. Enable the pattern generator in the relevant transmitter. For information about the BERT pattern generator, see BERT Pattern Generator.

4. Enable the pattern matcher and error counter in the receiver under test. For information about the BERT pattern matcher and error counter, seeBERT Pattern Matcher and Error Counter.

5. Freeze the receiver's CDR.

In lanej.rx_ctl, set the phug_value field to 1'b1, the ovrd_dpll_gain field to 1'b1, and the frug_value field to 2'b00 (default).

Set lanej.freq to 13'b0_0000_0000_0000.

In lanej.rx_ctl, set the phug_value field to 2'b00.

6. Force a selected phase offset from the mean phase identified in step 2 by adding a value to the value measured in step 2 and writing the sum to lanej.phase.

7. Measure the BER at this phase offset.

8. Repeat steps 6 and 7 to get the BER versus programmed phase offset.

5.2.5.4.2 Voltage Margining

Voltage margining can be performed in an asynchronous environment without shutting down the CDR, because the receiver is clock-forwarded. Voltage margining is performed by adding an offset voltage to the received signal.

Note that when the applied offset voltage is large relative to the peak-to-peak received signal, the Rx CDR's phase detector develops a dead zone that adds to the sampling jitter, causing additional loss in margin.

Voltage margining requires the following steps.

1. Enable the pattern generator in the relevant transmitter. For information about the BERT pattern generator, see BERT Pattern Generator.

2. Enable the pattern matcher and error counter in the receiver under test. For information about the BERT pattern matcher and error counter, see BERT Pattern Matcher and Error Counter.

3. Enable the margining DAC in the Clock module.

4. Set the DAC to its midrange by setting clock.dac_ctl.dac_val to 10'b01_1111_1111 (511). The DAC is 10 bits wide and the DAC's midrange corresponds to a zero offset.

5. Enable the DAC by setting clock.dac_ctl.dac_mode to 2'b11.

6. Program an additive offset voltage by changing the value of clock.dac_ctl.dac_val from 511. The resolution (1 LSB) of the DAC is VP25 x 279e-6 volts.

7. Measure the BER at this voltage offset.

8. Repeat steps 4 and 5 to get the BER versus programmed voltage offset.

64.6.2.4.5 Scope Function

Scope function (capturing and viewing the signals received at each input) can be achieved by combining the SATA2 PHY's signal acquisition capabilities with software running on an external host and utilizing external graphics display capabilities.

Note that scope function differs from the function of expensive lab equipment in some important ways; for example:

- Scope: The actual signal delivered to the receiver's slicers is measured. When a lab hardware scope is used, different reflections are present. In some cases, an alternative board-not the board with the receiver-is plugged in, enabling use of an external connector. Consequently, the reflection environment is changed entirely. In other cases, an active probe is used to pick up the signal near the receiver. This technique picks up reflections from the receiver that are not actually present at the receiver (at least 11-20 mm from the closest possible pickup point).
- Scope: The signal is acquired with the actual bandwidth limitations of the slicers in the receivers. This situation might not suit your measurement goals.
- A limitation of the on-die scope is that only periodic signals of known periodicity can be acquired, though the periodicity can be as long as 2^16 bytes.
- Scope: Jitter of the receiving device's Rx Analog Phase-Locked Loop (APLL) and MPLL is present in the measurement. This jitter is generally larger than the sampling jitter of a quality piece of bench equipment. In a loopback situation (same MPLL for both Tx and Rx), much of this jitter is cancelled, yielding the appearance of Tx jitter that is lower than is actually present.
- Scope function is analogous to an undersampling scope, not a real-time, oversampling scope.
- Bench hardware typically has voltage and time accuracy that are carefully specified and directly traceable to external standards. Tracing voltage, offset, and phase accuracy to external standards is more indirect.

The scope's greatest strength is its ease of use in debugging and understanding a system in situ. The scope is not intended to replace traditional lab equipment for the purpose of compliance testing, but the scope provides a nondestructive view of incoming data.

64.6.2.4.6 Analog DC Test Capabilities

This section describes the SATA2 PHY's analog DC test capabilities.

64.6.2.4.6.1 Analog Test Bus

The analog test bus (ATB) comprises four signals routed throughout the Active section, through the Clock module and each lane.

Some of these signals can be brought to the internal ADC while others can be brought to any Tx or Rx pin. The table below describes the ATB signals.

Table 64-34. Analog Test Bus Signals

Signal	Description	Can be Brought to:
atb_s_p	Zero current "sense" connections	ADC, Txp, Rxp
atb_s_m	Zero current "sense" connections	ADC, Txm, Rxm
atb_f_p	Large current (up to about 3 mA) "force" connection	Txp, Txm, Rxp, Rxm

Many internal circuits have connections that can be enabled to these buses. Note that these connections are intended to be used in a very-low-frequency (near DC) situation. Series resistance between connections on the bus are typically > 30 ohms, while bus parasitic capacitance can be multiple picoFarads, yielding bandwidths below 2 MHz. When connections are made to nodes carrying high-frequency signals such as Tx or Rx serial pins, explicit 10-k ohm resistors are used so that connecting the ATB bus does not impair the lane's operation.

The figure below shows a high-level map of the ATB.

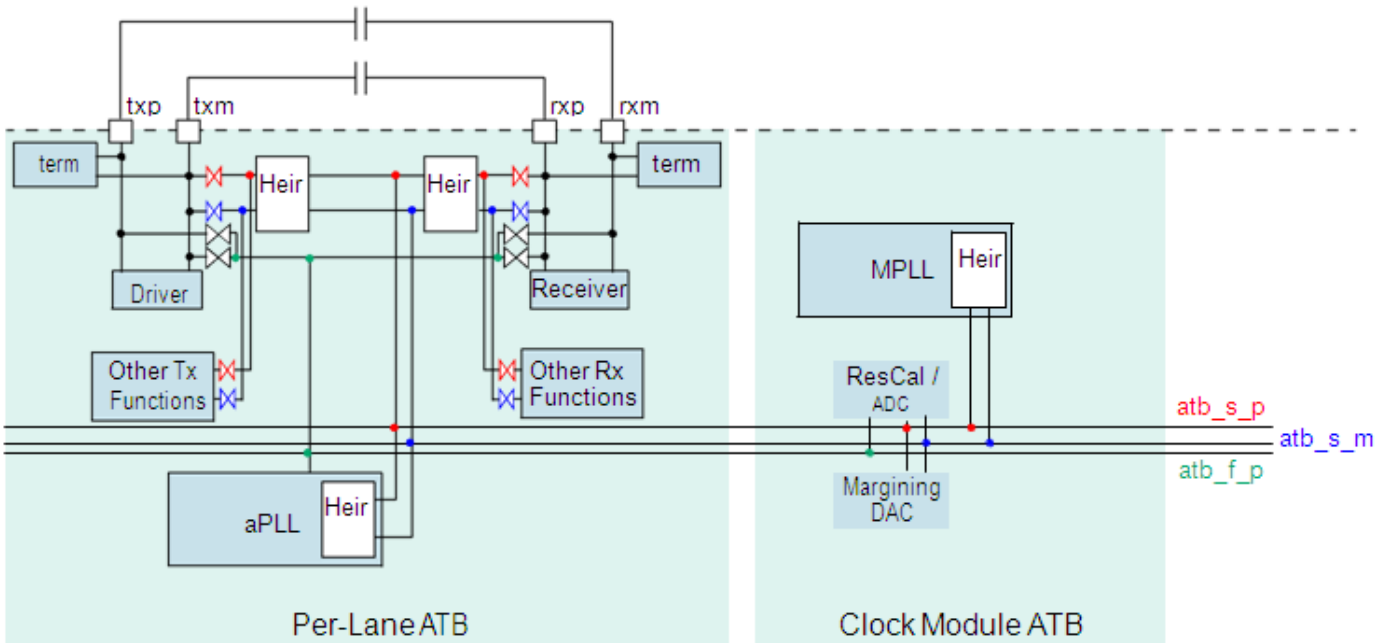


Figure 64-15. High-Level Map of ATB

Bringing a selected internal signal to either a package pin or the internal ADC is a matter of activating the correct combination of hierarchical switches and leaf-level switches. Control of these switches is distributed across several registers, as described in the table below.

Table 64-35. Location of ATB Switch Controls

Circuit Area	Control Register
Transmit	Hierarchical switch: "atb_en" in Receive Analog Control Register (lane0_RX_ANA_CONTROL) Leaf-level switches: Transmit ATB 1 Control Register (lane0_TX_ANA_ATBSEL1) and Transmit ATB 2 Control Register (lane0_TX_ANA_ATBSEL2)
Receive	Hierarchical switch: "atb_en" in Receive Analog Control Register (lane0_RX_ANA_CONTROL) Leaf-level switches: Receive ATB Register (lane0_RX_ANA_ATB)
CDR aPLL	Hierarchical switch: "atb_sense_sel" in Rx PLL Programming 2 Register (lane0_PLL_PRG2) Leaf-level switches: Rx PLL Measurement Register (lane0_PLL_PRG3)
MPLL	Hierarchical switch: "atb_sense_sel" in Rx PLL Programming 2 Register (lane0_PLL_PRG2) Leaf-level switches: MPLL Test Register (clock_MPLL_TEST)

Disabling Rx Termination

To bring ATB signals out on the Rx pins, it is recommended that you disable the resistive terminations that are otherwise present at these pins.

When using the Rx pins connected to an external measuring device, disable the receive termination resistors by deasserting the rx_term_en bit. In addition, disable the LOS by setting the los_ctl bits to 2'b00.

The Tx pins must not be used for this purpose, because the driver contains some impedances and leakage that cannot be completely eliminated.

64.6.2.4.6.2 10-Bit DAC

The 10-bit DAC is required for voltage margining and is used as part of the 10-bit AD converter (see 10-Bit ADC).

In addition, the DAC can be connected to the DC test bus for other purposes and is generally controlled by the DAC Control register (clock.dac_ctl). (For information about this register, see "DAC Control Register (clock.dac_ctl)" on page 47.)

To use the 10-bit DAC:

1. Place the DAC in one of the DAC operating modes by setting clock.dac_ctl.dac_mode to a value of 4-7 (see DAC Control Register (clock_DAC_CTL)).
2. Connect the DAC output to the global atb_s_p bus by setting clock.rtune_ctl.dac_chop (see Resistor Tuning Control Register (clock_RTUNE_CTL)).
3. To produce the intended output voltage, write a value to clock.dac_ctl.dac_val (see DAC Control Register (clock_DAC_CTL)).

64.6.2.4.6.3 10-Bit ADC

The 10-bit ADC uses circuit components that are present in the Clock module, because the 10-bit DAC is required for voltage margining, and the offset calibrated comparator is present for resistor calibration.

64.6.2.4.7 Limit Testing

In most ASIC production test environments, the tests simply run a vector (drive controls into the DUT and verify that the correct output is produced) due to either limitations of the tester itself or time constraints in the development of more sophisticated tests. Therefore, tasks that would otherwise seem simple (for example, reading a register and verifying that the result is < 20) become effectively impossible.

The limit testing feature can be used to place upper and lower bounds on a masked version of a register read or to place bounds on a masked version of the difference between two masked register reads.

The figure below shows the hardware functions involved in limit testing.

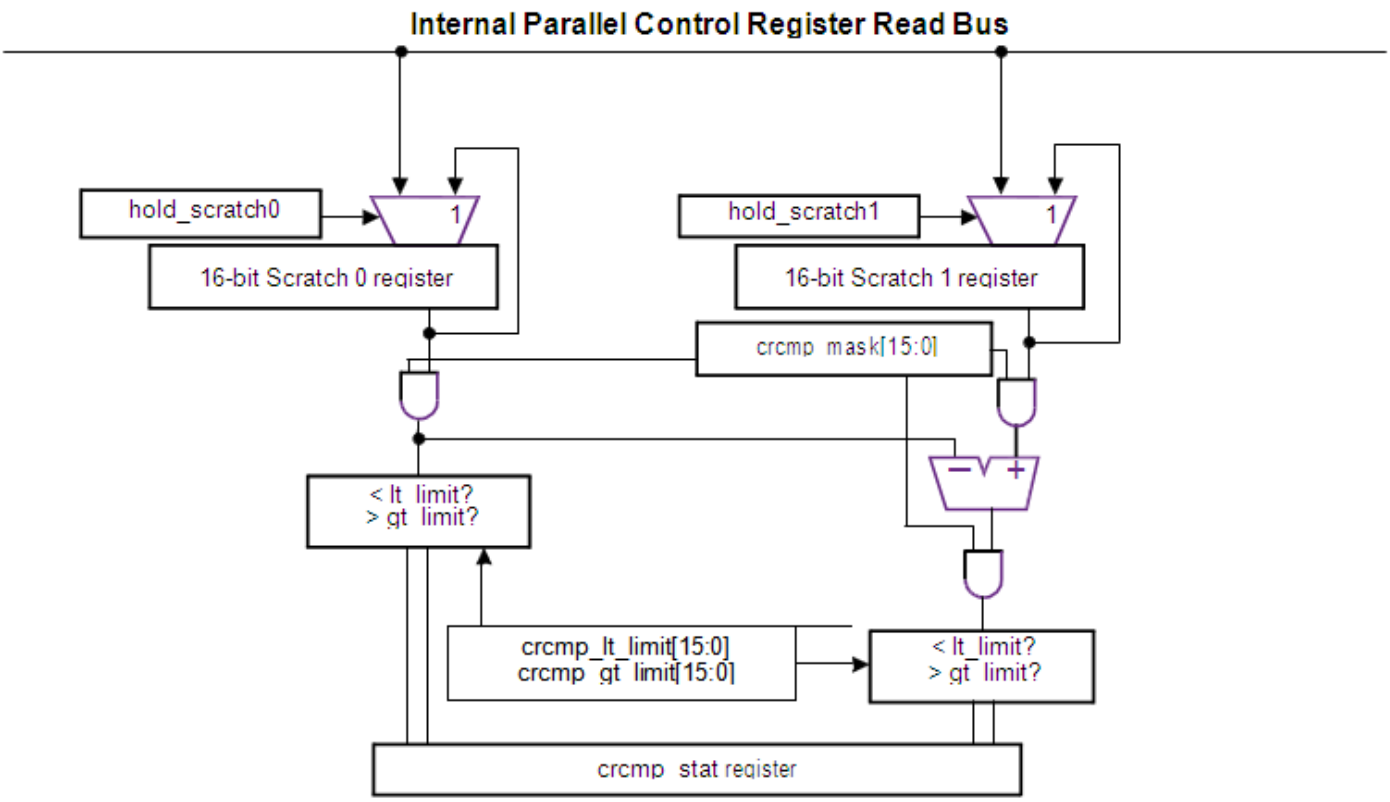


Figure 64-16. Hardware for Limit Testing

Each internal scratch register is updated each time a register read occurs through the JTAG interface, unless the associated hold_scratch1 or hold_scratch0 bit is set.

Note that the values written to the crcmp_lt_limit and crmp_gt_limit registers are also masked with the crcmp_mask field before being used.

64.6.2.4.8 Integrated Test Modes

The SATA2 PHY integrates the following test modes, which can be used to enhance characterization and ATE testing.

64.6.2.4.8.1 IDDQ Test Mode

To enable IDDQ test mode, set the pddq_h signal to 1'b1 (I/0 voltage level). This setting powers down circuitry that cannot be powered down as described in Power-Down Sequences.

All preconditioning of the SATA2 PHY must be done before asserting pddq_h, because the device becomes non-functional when pddq_h is set to 1'b1.

64.6.2.4.8.2 Bypass Test Mode

Setting the test_byp_mode signal to 1'b1 connects the parallel data inputs to the parallel data outputs through a purely combinatorial path. This mode enables testing of the ASIC/SATA2 PHY interface; the passthru.v Verilog model represents this mode.

This interface is designed to be orders of magnitude slower than the operational frequency, so no timing information is provided. However, if necessary, using 1ns for a hold time would be a very conservative number.

64.6.2.4.8.3 Burn-In Test Mode

Setting the test_burnin_mode signal to 1'b1 enables the SATA2 PHY for burn-in testing. An on-board oscillator is activated and multiplexed to the prescaler inputs to be used as the reference clock.

This oscillator removes the requirement for an external reference. As much circuitry as possible is toggled, and unused circuitry is biased so that devices do not degrade asymmetrically.

64.6.2.4.9 Burn-In Test Requirements

For burn-in testing, the IP must be placed in a mode so that the IP's circuitry is stressed in its intended mode of operation.

This requirement involves providing the IP a reference clock, applying power, powering up the device as described in Power-Up Sequences, and sending parallel data to the transmitter. Externally, the transmitter pins must be AC-coupled through a capacitor (typically 0.1 µF) to the receiver, and the rbias resistor must be connected. Burn-in test mode satisfies this requirement-power must be applied to the IP and Burn-in test mode must be activated. Externally, the rbias resistor must be connected.

64.6.2.4.10 Temperature Sensor

The die's temperature is derived using measured voltages from the internal analog circuitry. These voltages are first converted to a digital value through an ADC, and in turn these values are simply read from standard register reads.

To determine the die temperature, you must then apply the two measurements in an equation.

The figure below shows the flow for measuring and calculating the temperature.

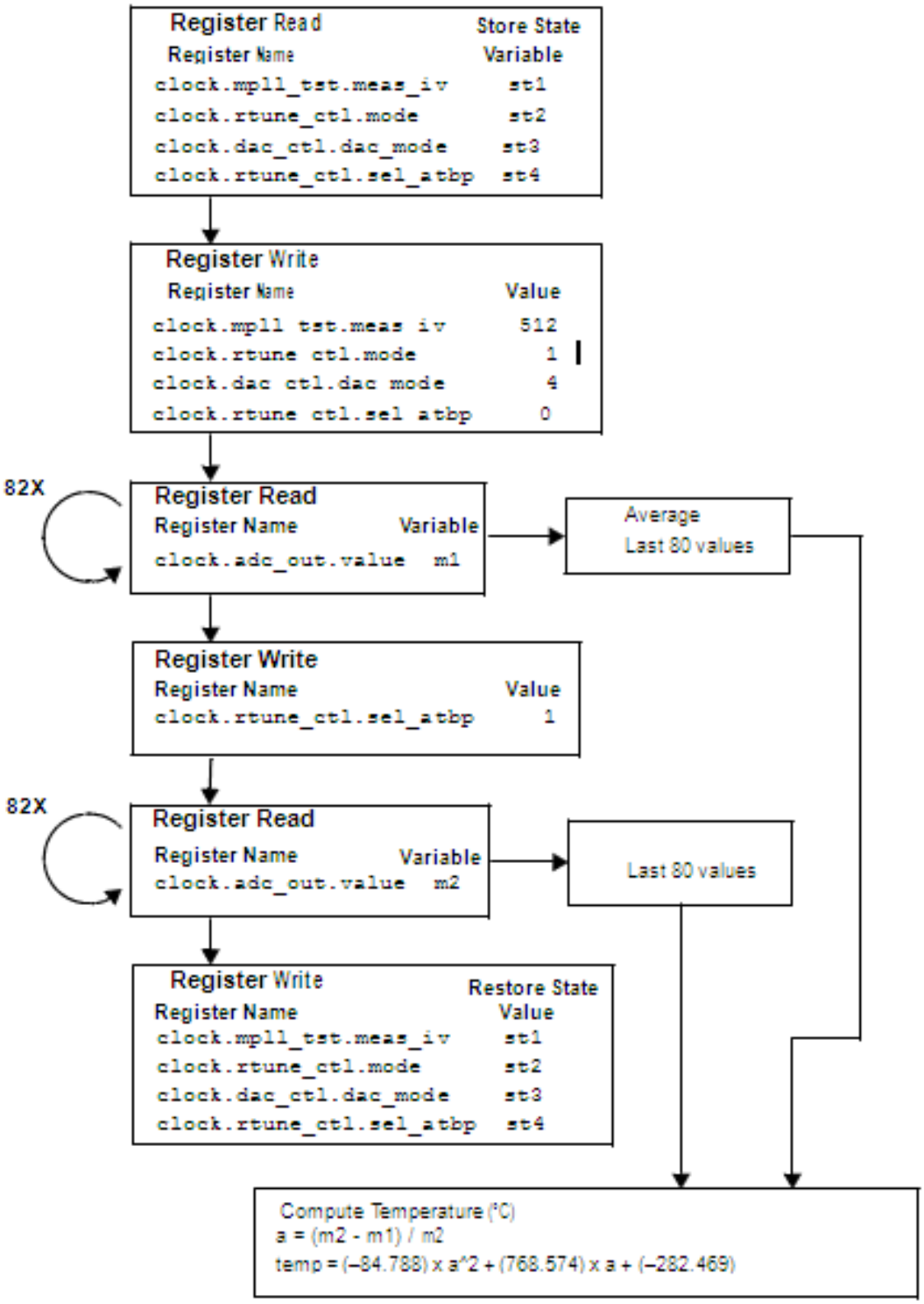


Figure 64-17. Temperature Measurement Flow

Synopsys has verified the temperature sensor's function in the lab environment using temperature control equipment and an additional high-precision temperature sensor fabricated with the IP on a test chip. The equation for the temperature calculation is a quadratic curve fit to simulation and experimental data. The temperature measurement has proven to be supply-independent, with an absolute error of less than 5° C across process. The temperature measurement can be made without disturbing data traffic.

NOTE

Refer to the SATA Temperature Sensor application note (AN4380) for more details regarding programming and its usage.

64.6.2.5 ATE Testing

The features described in Diagnostic Features were used to develop an analog, production-ready test program. This test suite tests the analog functions of the IP using simple pass/fail vectors that can be used on a low-cost digital tester.

Test vectors are entered into the core through the JTAG interface. By request, a program for generating digital test patterns as well as documentation that provides implementation guidelines will be provided .

64.7 clock Memory Map/Register Definition

This section describes registers in the Clock module. Only one instance of each register exists in the Clock module, regardless of the number of lanes.

Registers in the Clock module have 16-bit addresses. These addresses are noted in the register map and in the section for the applicable register.

NOTE

SATA PHY registers are only accessible by the corresponding controller (SATA_P0_PCSR_PHYCR and SATA_P0_PCSR_PHYSR) or in debug through the JTAG port. SATA PHY is not memory mapped to processor address space, so the absolute addresses shown is the relative address and is not valid. See SATA Memory Map for more information.

clock memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0001	Creg Compare Upper Limit Register (clock_CRCMP_LT_LIMIT)	16	R/W	0000h	64.7.1/ 3970
0002	Creg Compare Lower Limit Register (clock_CRCMP_GT_LIMIT)	16	R/W	FFFFh	64.7.2/ 3970
0003	Creg Compare Mask Register (clock_CRCMP_MASK)	16	R/W	FFFFh	64.7.3/ 3971
0004	Creg Compare Control Register (clock_CRCMP_CTL)	16	R/W	0000h	64.7.4/ 3971
0005	Creg Compare Status Register (clock_CRCMP_STAT)	16	R	0000h	64.7.5/ 3972
0006	Scope Sample Count Register (clock_SCOPE_SAMPLES)	16	R/W	0100h	64.7.6/ 3973
0007	Scope Count Result Register (clock_SCOPE_COUNT)	16	R	0000h	64.7.7/ 3973
0008	DAC Control Register (clock_DAC_CTL)	16	R/W	01FFh	64.7.8/ 3974
0009	Resistor Tuning Control Register (clock_RTUNE_CTL)	16	R/W	0020h	64.7.9/ 3975
000A	ADC Output Register (clock_ADC_OUT)	16	R	0000h	64.7.10/ 3976
000B	Spread Spectrum Phase Register (clock_SS_PHASE)	16	R/W	0000h	64.7.11/ 3976
000C	JTAG Chip ID (High Bits) Register (clock_CHIP_ID_HI)	16	R	0011h	64.7.12/ 3977
000D	JTAG Chip ID (Low Bits) Register (clock_CHIP_ID_LOW)	16	R	74CDh	64.7.13/ 3977
000E	Frequency Status Register (clock_FREQ_STAT)	16	R	0000h	64.7.14/ 3978
000F	Control Status Register (clock_CTL_STAT)	16	R	0000h	64.7.15/ 3979
0010	Level Status Register (clock_LVL-STAT)	16	R	0000h	64.7.16/ 3980
0011	Creg Status Register (clock_CREG_STAT)	16	R	0000h	64.7.17/ 3980
0012	Frequency Override Register (clock_FREW_OVRD)	16	R/W	4547h	64.7.18/ 3981
0013	Control Override Register (clock_CTL_OVRD)	16	R/W	0854h	64.7.19/ 3982
0014	Level Override Register (clock_LVL_OVRD)	16	R/W	4210h	64.7.20/ 3983

Table continues on the next page...

clock memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0015	Creg Override Register (clock_CREG_OVRD)	16	R/W	0040h	64.7.21/ 3984
0016	MPLL Control Register (clock_MPLL_CTL)	16	R/W	0000h	64.7.22/ 3985
0017	MPLL Test Register (clock_MPLL_TEST)	16	R/W	0000h	64.7.23/ 3986
0018	Spread Spectrum Frequency Register (clock_SS_FREQ)	16	R/W	332Fh	64.7.24/ 3987
0019	Clock Select Status Register (clock_SEL_STAT)	16	R	0000h	64.7.25/ 3988
001A	Clock Select Override Register (clock_SEL_OVRD)	16	R/W	0000h	64.7.26/ 3989
7F3F	Reset Register (clock_RESET)	16	W	0000h	64.7.27/ 3989

64.7.1 Creg Compare Upper Limit Register (clock_CRCMP_LT_LIMIT)

Address: 0x0001

Reset value: 16'b 0000 0000 0000 0000

This register contains the less-than-limit compare point.

Address: clock_CRCMP_LT_LIMIT is 0h base + 1h offset = 0001h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read / Write	crcmp_lt_limit															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

clock_CRCMP_LT_LIMIT field descriptions

Field	Description
15–0 crcmp_lt_limit	Less-than-limit compare point

64.7.2 Creg Compare Lower Limit Register (clock_CRCMP_GT_LIMIT)

Address: 0x0002

Reset value: 16'b 1111 1111 1111 1111

This register contains the greater-than-limit compare point.

Address: clock_CRCMP_GT_LIMIT is 0h base + 2h offset = 0002h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read / Write	crcmp_gt_limit															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

clock_CRCMP_GT_LIMIT field descriptions

Field	Description
15–0 crcmp_gt_limit	Greater-than-limit compare point

64.7.3 Creg Compare Mask Register (clock_CRCMP_MASK)

Address: 0x0003

Reset value: 16'b 1111 1111 1111 1111

This register contains the compare/scratch value mask.

Address: clock_CRCMP_MASK is 0h base + 3h offset = 0003h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read / Write	crcmp_mask															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

clock_CRCMP_MASK field descriptions

Field	Description
15–0 crcmp_mask	Mask for comparisons

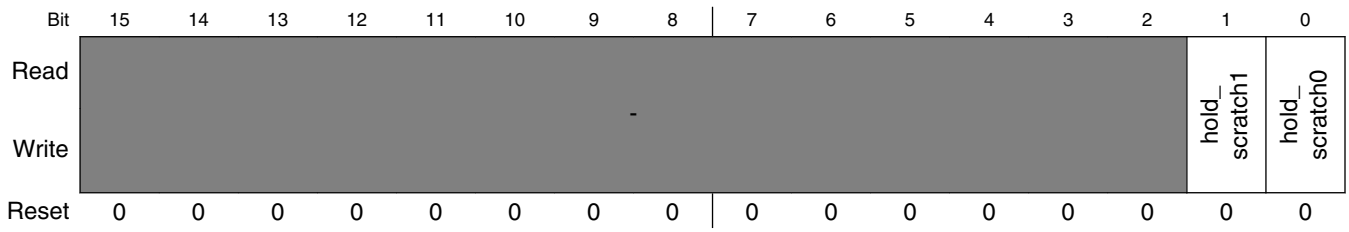
64.7.4 Creg Compare Control Register (clock_CRCMP_CTL)

Address: 0x0004

Reset value: 16'b 0000 0000 0000 0000

This register contains the scratch space control bits.

Address: clock_CRCMP_CTL is 0h base + 4h offset = 0004h



clock_CRCMP_CTL field descriptions

Field	Description
15–2 -	Reserved
1 hold_scratch1	Scratch1 is not updated on register reads.
0 hold_scratch0	Scratch0 is not updated on register reads.

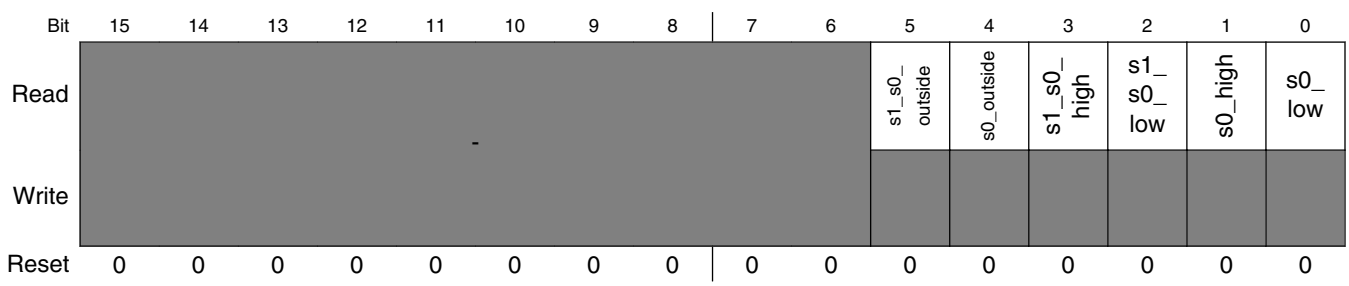
64.7.5 Creg Compare Status Register (clock_CRCMP_STAT)

Address: 0x0005

Reset value: 16'b xxxx xxxx xxxx xxxx

This register contains the results of scratch register comparisons to various limits.

Address: clock_CRCMP_STAT is 0h base + 5h offset = 0005h



clock_CRCMP_STAT field descriptions

Field	Description
15–6 -	Reserved
5 s1_s0_outside	Logical OR of S1_S0_LOW and S1_S0_HIGH Useful for determining if the difference is near signed zero.
4 s0_outside	Logical OR of S0_LOW and S0_HIGH Useful for determining if the value is near signed zero.

Table continues on the next page...

clock_CRCMP_STAT field descriptions (continued)

Field	Description
3 s1_s0_high	Masked (Scratch1 - Scratch0) is higher than CRCMP_HT_LIMIT.
2 s1_s0_low	Masked (Scratch1 - Scratch0) is lower than CRCMP_LT_LIMIT.
1 s0_high	Masked Scratch0 is higher than CRCMP_HT_LIMIT.
0 s0_low	Masked Scratch0 is lower than CRCMP_LT_LIMIT

64.7.6 Scope Sample Count Register (clock_SCOPE_SAMPLES)

Address: 0x0006

Reset value: 16'b 0000 0001 0000 0000

This register specifies the number of samples to count.

Address: clock_SCOPE_SAMPLES is 0h base + 6h offset = 0006h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	scope_samples															
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

clock_SCOPE_SAMPLES field descriptions

Field	Description
15–0 scope_samples	The number of samples to count

64.7.7 Scope Count Result Register (clock_SCOPE_COUNT)

Address: 0x0007

Reset value: 16'b xxxx xxxx xxxx xxxx

This register provides the results of scope counting. A write to this register starts the counting process. A value of FFFF indicates that the count is still in progress.

Address: clock_SCOPE_COUNT is 0h base + 7h offset = 0007h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	scope_count															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

clock_SCOPE_COUNT field descriptions

Field	Description
15–0 scope_count	Results of scope counting

64.7.8 DAC Control Register (clock_DAC_CTL)

Reset value: 16'b x000 0001 1111 1111

This register supports DAC values and controls.

Address: clock_DAC_CTL is 0h base + 8h offset = 0008h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	-	dac_mode			ovrd_rtune_rx	ovrd_rtune_tx	dac_val									
Write																
Reset	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1

clock_DAC_CTL field descriptions

Field	Description
15 -	Reserved
14–12 dac_mode	DAC output mode: 000 Powers down DAC 001 Reserved 010 High-range margining (VP25 x 418e-6 res) 011 Low-range margining (VP25 x 279e-6 res) 100 100% range DAC, 0% offset 101 36% range DAC, 0% offset 110 36% range DAC, 33% offset 111 36% range DAC, 66% offset
11 ovrd_rtune_rx	Writes DAC_VAL[5:0] to the Rx rtune bus

Table continues on the next page...

clock_DAC_CTL field descriptions (continued)

Field	Description
10 ovrd_rtune_tx	Writes DAC_VAL[5:0] to the Tx rtune bus
9–0 dac_val	Digital value to be used for DAC

64.7.9 Resistor Tuning Control Register (clock_RTUNE_CTL)

Reset value: 16'b xxxx x000 0010 0000

This register contains resistor tuning controls.

Address: clock_RTUNE_CTL is 0h base + 9h offset = 0009h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	-					adc_trig	rtune_trig	rtune_dis	cmp_invert	dac_chop	rsc_x4	sel_atbp	pwron_lcl	frc_pwron	mode	
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

clock_RTUNE_CTL field descriptions

Field	Description
15–11 -	Reserved
10 adc_trig	Triggers ADC conversion
9 rtune_trig	Triggers manual resistor calibration
8 rtune_dis	Disables automatic resistor recalibrations
7 cmp_invert	Inverts output of comparator (to reverse successive approximation register (SAR) feedback loop)
6 dac_chop	Polarity of chop control for DAC
5 rsc_x4	Sets x4 in rescal circuitry
4 sel_atbp	Selects atb_s_p for A/D measurement
3 pwron_lcl	Value of power-on to force
2 frc_pwron	Overrides internal power-on

Table continues on the next page...

clock_RTUNE_CTL field descriptions (continued)

Field	Description
1–0 mode	Resistor tune SAR mode: 00 Normal restune 01 ADC 10 Rx Resistor test 11 Tx Resistor test

64.7.10 ADC Output Register (clock_ADC_OUT)

Address: 0x000A

Reset value: 16'b xxxx xxxx xxxx xxxx

This register contains the results of the ADC process. A read from this register starts a new A/D conversion.

Address: clock_ADC_OUT is 0h base + Ah offset = 000Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	-					fresh	value									
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

clock_ADC_OUT field descriptions

Field	Description
15–11 -	Reserved
10 fresh	Flag indicates that a new A/D conversion result is present.
9–0 value	A/D conversion result Based on RTUNE_CTL.MODE, this value is the result of either the last conversion (MODES 0 or 1) or the current Tx/Rx cal value (MODES 3/2).

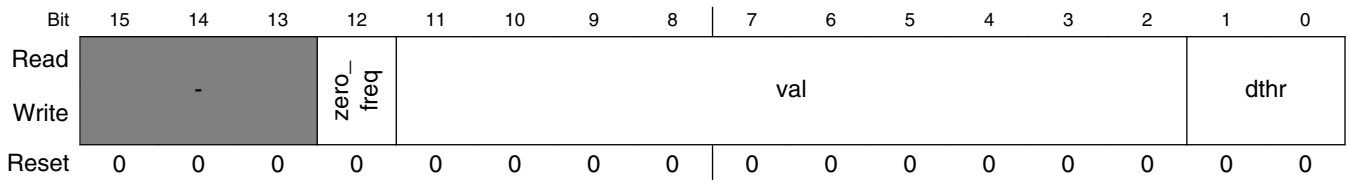
64.7.11 Spread Spectrum Phase Register (clock_SS_PHASE)

Address: 0x000B

Reset value: 16'b xxx0 0000 0000 0000

This register contains the current MPLL phase selector value.

Address: clock_SS_PHASE is 0h base + Bh offset = 000Bh



clock_SS_PHASE field descriptions

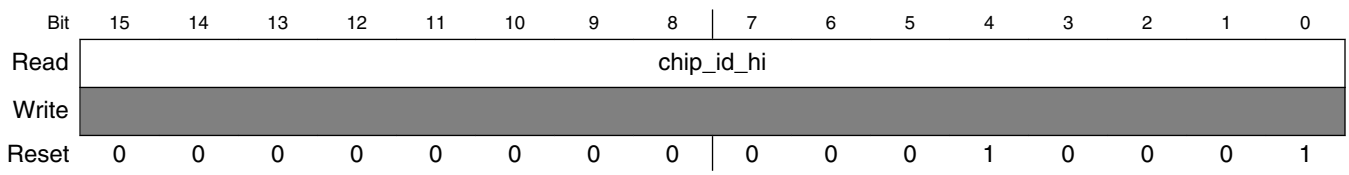
Field	Description
15–13 -	Reserved
12 zero_freq	Zero frequency register Must be set for PHASE writes to not be immediately overwritten.
11–2 val	Phase value from zero reference
1–0 dthr	Bits below the useful resolution

64.7.12 JTAG Chip ID (High Bits) Register (clock_CHIP_ID_HI)

Address: 0x000C

This register contains the internal chip ID (high 16 bits) of the JTAG interface.

Address: clock_CHIP_ID_HI is 0h base + Ch offset = 000Ch



clock_CHIP_ID_HI field descriptions

Field	Description
15–0 chip_id_hi	Internal chip ID (high 16 bits)

64.7.13 JTAG Chip ID (Low Bits) Register (clock_CHIP_ID_LOW)

Address: 0x000D

This register contains the internal chip ID (low 16 bits) of the JTAG interface.

Address: clock_CHIP_ID_LOW is 0h base + Dh offset = 000Dh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	chip_id_lo															
Write																
Reset	0	1	1	1	0	1	0	0	1	1	0	0	1	1	0	1

clock_CHIP_ID_LOW field descriptions

Field	Description
15–0 chip_id_lo	Internal chip ID (low 16 bits)

64.7.14 Frequency Status Register (clock_FREQ_STAT)

Address: 0x000E

Reset value: 16'b xxxx xxxx xxxx xxxx (depends on inputs)

This register indicates the status of frequency control inputs.

Address: clock_FREQ_STAT is 0h base + Eh offset = 000Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	reserved	prescale		ncy					ncy5		int_ctl			prop_ctl		
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

clock_FREQ_STAT field descriptions

Field	Description
15 Reserved	This field is reserved. Always reads as 1
14–13 prescale	Prescaler control
12–8 ncy	Divide-by-4 cycle control
7–6 ncy5	Divide-by-5 control
5–3 int_ctl	Integral charge pump control
2–0 prop_ctl	Proportional charge pump control

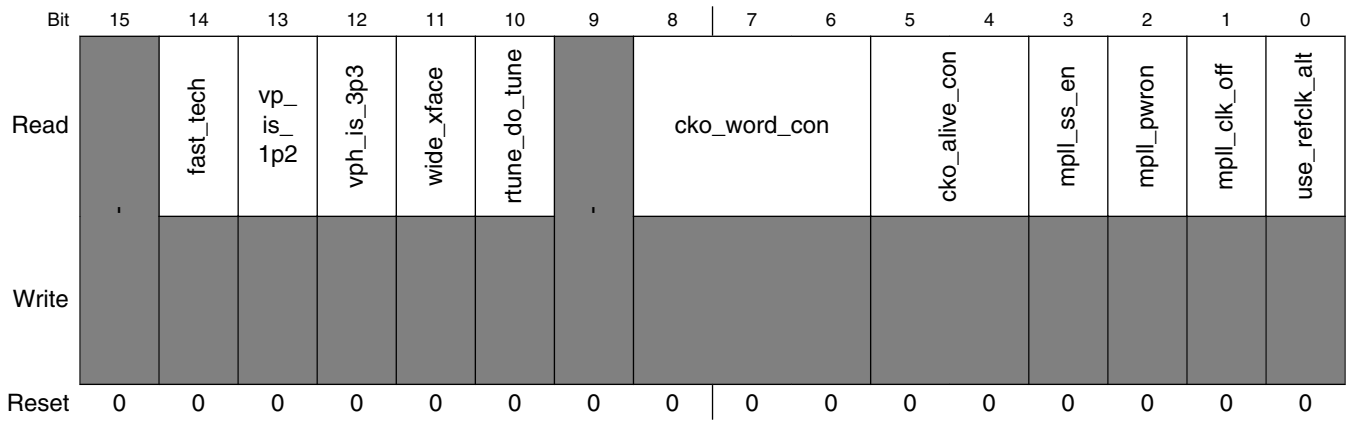
64.7.15 Control Status Register (clock_CTL_STAT)

Address: 0x000F

Reset value: 16'b xxxx xxxx xxxx xxxx (depends on inputs)

This register indicates the status of various control inputs.

Address: clock_CTL_STAT is 0h base + Fh offset = 000Fh



clock_CTL_STAT field descriptions

Field	Description
15 -	Reserved
14 fast_tech	Technology is fast
13 vp_is_1p2	Low voltage supply is 1.2 V
12 vph_is_3p3	High voltage supply is 3.3 V
11 wide_xface	Wide interface control
10 rtune_do_tune	Manual resistor tune control
9 -	Reserved
8–6 cko_word_con	cko_word MUX control
5–4 cko_alive_con	cko_alive MUX control
3 mpll_ss_en	Spread spectrum enable

Table continues on the next page...

clock_CTL_STAT field descriptions (continued)

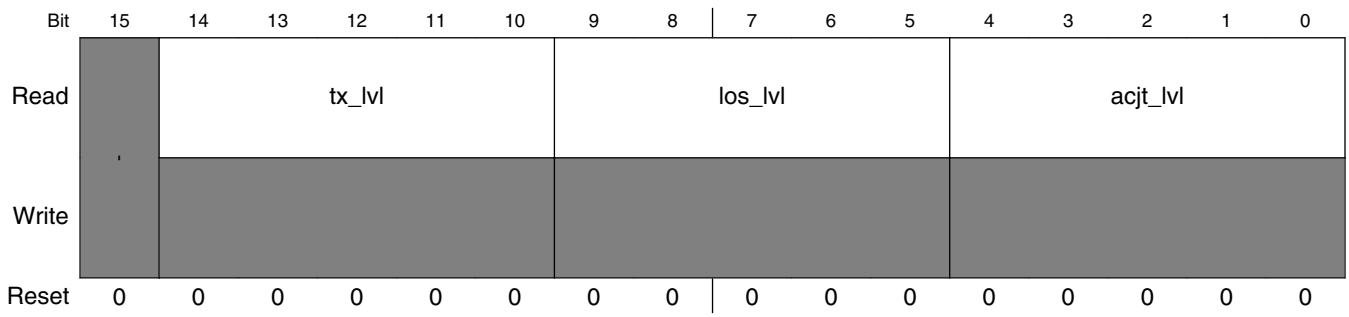
Field	Description
2 mpll_pwron	MPLL power-on control
1 mpll_clk_off	Reference clock is off
0 use_refclk_alt	Alternate refclk is used

64.7.16 Level Status Register (clock_LVL-STAT)

Reset value: 16'b xxxx xxxx xxxx xxxx (depends on inputs)

This register indicates the status of level control inputs.

Address: clock_LVL-STAT is 0h base + 10h offset = 0010h



clock_LVL-STAT field descriptions

Field	Description
15 -	Reserved
14–10 tx_lvl	Transmit level
9–5 los_lvl	Loss of Signal Detector level
4–0 acjt_lvl	ACJTAG comparator level

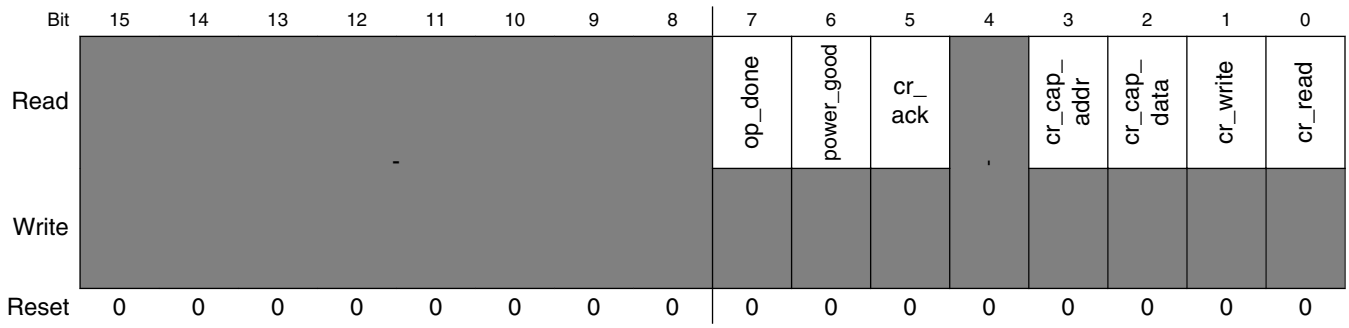
64.7.17 Creg Status Register (clock_CREG_STAT)

Address: 0x0011

Reset value: 16'b xxxx xxxx xxxx xxxx (depends on inputs)

This register indicates the status of creg control I/O.

Address: clock_CREG_STAT is 0h base + 11h offset = 0011h



clock_CREG_STAT field descriptions

Field	Description
15–8 -	Reserved
7 op_done	Operation is complete output
6 power_good	Power good output
5 cr_ack	Creg request acknowledgement
4 -	Reserved
3 cr_cap_addr	Captures address request
2 cr_cap_data	Captures data request
1 cr_write	Write request
0 cr_read	Read request

64.7.18 Frequency Override Register (clock_FREW_OVRD)

Reset value: 16'b 0100 0101 0100 0111

This register contains the override of frequency control inputs.

Address: clock_FREW_OVRD is 0h base + 12h offset = 0012h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	ovrd	prescale		ncy					ncy5		int_ctl			prop_ctl		
Write																
Reset	0	1	0	0	0	1	0	1	0	1	0	0	0	1	1	1

clock_FREW_OVRD field descriptions

Field	Description
15 ovrd	Enables override of all bits in this register
14–13 prescale	Prescaler control: 00 No scaling 01 Doubles refclk frequency 10 Halves refclk frequency 11 Reserved
12–8 ncy	Divide-by-4 cycle control MPLL Divider period = 4 x (NCY + 1) + NCY5. Valid only when NCY5 <=NCY.
7–6 ncy5	Divide-by-5 control MPLL Divider period = 4 x (NCY + 1) + NCY5. Valid only when NCY5 <=NCY
5–3 int_ctl	Integral charge pump control Integral current = (n + 1) / 8 x full_scale
2–0 prop_ctl	Proportional charge pump control Proportional current = (n + 1) / 8 x full_scale

64.7.19 Control Override Register (clock_CTL_OVRD)

Reset value: 16'b 0000 1000 0101 0100

This register contains the override of various control inputs.

Address: clock_CTL_OVRD is 0h base + 13h offset = 0013h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	ovrd_static	fast_tech	vp_is_1p2	vph_is_3p3	wide_xface	rtune_do_tune	ovrd_clk	cko_word_con			cko_alive_con		mpll_ss_en	mpll_pwron	mpll_clk_off	use_refclk_alt
Write																
Reset	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0	0

clock_CTL_OVRD field descriptions

Field	Description
15 ovrd_static	Overrides static controls (bits [14:10])
14 fast_tech	Technology is fast
13 vp_is_1p2	Low-voltage supply is 1.2 V
12 vph_is_3p3	High-voltage supply is 3.3 V
11 wide_xface	Wide interface control
10 rtune_do_tune	Manual resistor tune control
9 ovrd_clk	Overrides clock controls (bits [8:0])
8–6 cko_word_con	cko_word mux control
5–4 cko_alive_con	cko_alive mux control
3 mpll_ss_en	Spread spectrum enable
2 mpll_pwron	MPLL power-on control
1 mpll_clk_off	Reference clock is off
0 use_refclk_alt	Uses alternate refclk

64.7.20 Level Override Register (clock_LVL_OVRD)

Address: 0x0014

Reset value: 16'b 0100 0010 0001 0000

This register contains the override of level control inputs.

Address: clock_LVL_OVRD is 0h base + 14h offset = 0014h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	ovrd	level tx_lvl					los_lvl					acjt_lvl				
Reset	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0

clock_LVL_OVRD field descriptions

Field	Description
15 ovrd	Overrides all level controls
14–10 level tx_lvl	Transmit level
9–5 los_lvl	Loss of Signal Detector
4–0 acjt_lvl	ACJTAG comparator level

64.7.21 Creg Override Register (clock_CREG_OVRD)

Address: 0x0015 Reset value: 16'b xxxx xxx0 0100 0000 This register contains the override of creg control I/O.

Address: clock_CREG_OVRD is 0h base + 15h offset = 0015h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	-							ovrd_out	op_done	power_good	cr_ack	ovrd_in	cr_cap_addr	cr_cap_data	cr_write	cr_read
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

clock_CREG_OVRD field descriptions

Field	Description
15–9 -	Reserved
8 ovrd_out	Overrides outputs (bits [7:5])
7 op_done	Operation is complete output
6 power_good	Power good output
5 cr_ack	Creg request acknowledgement
4 ovrd_in	Overrides inputs (bits [3:0])
3 cr_cap_addr	Captures address request
2 cr_cap_data	Captures data request
1 cr_write	Writes request

Table continues on the next page...

clock_CREG_OVRD field descriptions (continued)

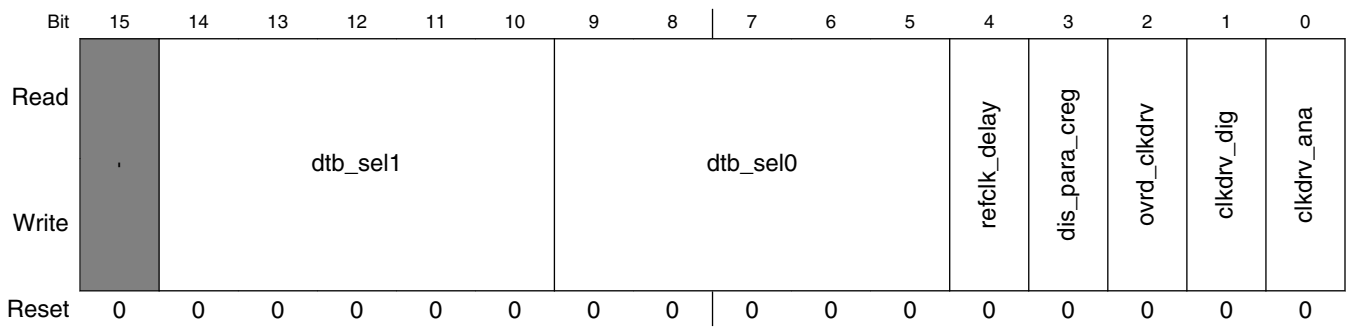
Field	Description
0 cr_read	Reads request

64.7.22 MPLL Control Register (clock_MPLL_CTL)

Reset value: 16'b xxxx xx00 0000 0000

This register contains MPLL controls.

Address: clock_MPLL_CTL is 0h base + 16h offset = 0016h



clock_MPLL_CTL field descriptions

Field	Description
15 -	Reserved
14–10 dtb_sel1	Selects wire to drive onto DTB bit 1: All other bits: Disabled 00000 Disabled 00001 mpll_gear_shift 00010 mpll_reset 00011 mpll_pwron (at analog boundary) 00100 reset_n 00101 cr_ack 00110 power_good 00111 op_done 01000 cr_read 01001 cr_write 01010 cr_cap_data 01011 cr_cap_addr 01100 rtune_do_tune 01101 cko_alive_con[0] 01110 cko_alive_con[1] 01111 cko_word_con[0]

Table continues on the next page...

clock_MPLL_CTL field descriptions (continued)

Field	Description
	10000 cko_word_con[1] 10001 cko_word_con[2] 10010 mpll_pwron (ASIC control) 10011 mpll_ck_off
9–5 dtb_sel0	Selects wire to drive onto DTB bit 0: All other bits: Disabled 00000 Disabled 00001 mpll_gear_shift 00010 mpll_reset 00011 mpll_pwron (at analog boundary) 00100 reset_n 00101 cr_ack 00110 power_good 00111 op_done 01000 cr_read 01001 cr_write 01010 cr_cap_data 01011 cr_cap_addr 01100 rtune_do_tune 01101 cko_alive_con[0] 01110 cko_alive_con[1] 01111 cko_word_con[0] 10000 cko_word_con[1] 10001 cko_word_con[2] 10010 mpll_pwron (ASIC control) 10011 mpll_ck_off
4 refclk_delay	Delays refclk output of prescaler
3 dis_para_creg	Disables parallel creg interface
2 ovrd_clkdrv	Overrides clock driver controls
1 clkdrv_dig	Value for digital clock drivers
0 clkdrv_ana	Value for analog clock drivers

64.7.23 MPLL Test Register (clock_MPLL_TEST)

Address: 0x0017

Reset value: 16'b 0000 0000 0000 0000

This register contains MPLL test controls.

Address: clock_MPLL_TEST is 0h base + 17h offset = 0017h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	ovrd_ctl	gearshift_val	reset_val	meas_iv											meas_gd	atb_sense
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

clock_MPLL_TEST field descriptions

Field	Description
15 ovrd_ctl	Overrides MPLL reset and gearshift controls
14 gearshift_val	Value to override for mpll_gearshift
13 reset_val	Value to override for mpll_reset
12–2 meas_iv	Measures various MPLL controls: • Bit 2: Measures dcc_vcntrl_p on atb_sense_p • Bit 3: Measures dcc_vcntrl_m on atb_sense_m • Bit 4: Measures 1-V supply voltage on atb_sense_m • Bit 5: Measures vp_cp voltage on atb_sense_p; gd on atb_sense_m • Bit 6: Measures VCO supply voltage on atb_sense_p; gd on atb_sense_m • Bit 7: Measures clock tree supply voltage on atb_sense_p; gd on atb_sense_m • Bit 8: Measures vp16 on atb_sense_p; gd on atb_sense_m • Bit 9: Measures vref on atb_sense_p; gd on atb_sense_m • Bit 10: Measures vcntrl on atb_sense_m • Bit 11: Measures copy of bias current in oscillator on atb_force_m • Bit 12: Enables phase linearity testing of phase interpolator and VCO
1 meas_gd	Measures Ground For correct measurements, this field must be set when various meas_iv bits are set.
0 atb_sense	Hooks up ATB sense lines

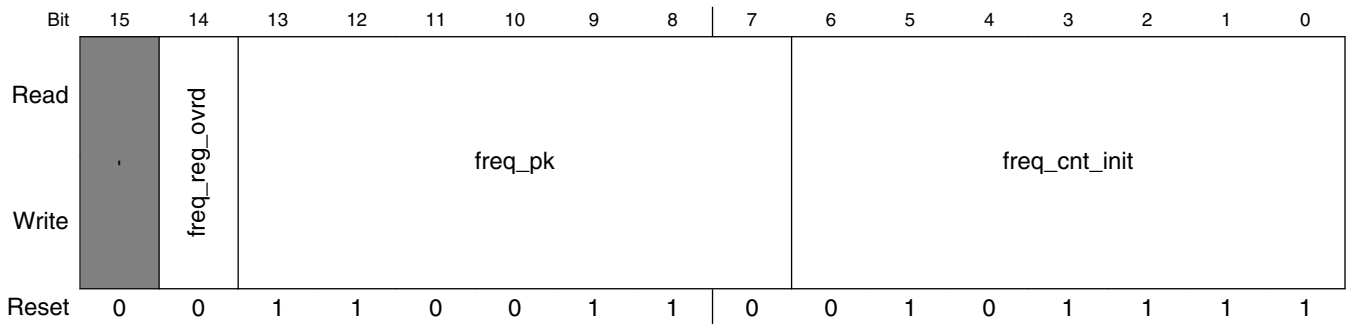
64.7.24 Spread Spectrum Frequency Register (clock_SS_FREQ)

Address: 0x0018

Reset value: 16'b x011 0011 0010 1111

This register contains the frequency register override, peak frequency value, and frequency counter step values.

Address: clock_SS_FREQ is 0h base + 18h offset = 0018h



clock_SS_FREQ field descriptions

Field	Description
15 -	Reserved
14 freq_reg_ovrd	Override control, indicating that overridden value is active
13–7 freq_pk	Peak frequency value
6–0 freq_cnt_init	Frequency counter step value. **Note:** This value is independent of the freq_pk value.

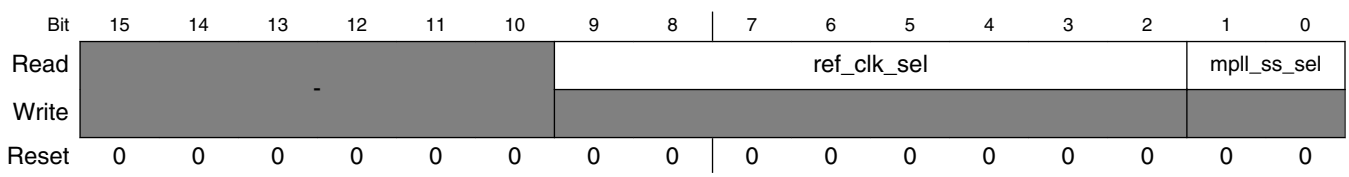
64.7.25 Clock Select Status Register (clock_SEL_STAT)

Address: 0x0019

Reset value: 16'bxxxx xxxx xxxx xxxx (depends on inputs)

This register indicates the status of the ref_clk_sel and mpll_ss_sel inputs.

Address: clock_SEL_STAT is 0h base + 19h offset = 0019h



clock_SEL_STAT field descriptions

Field	Description
15–10 -	Reserved
9–2 ref_clk_sel	Reference clock select input
1–0 mpll_ss_sel	MPLL spread spectrum select input

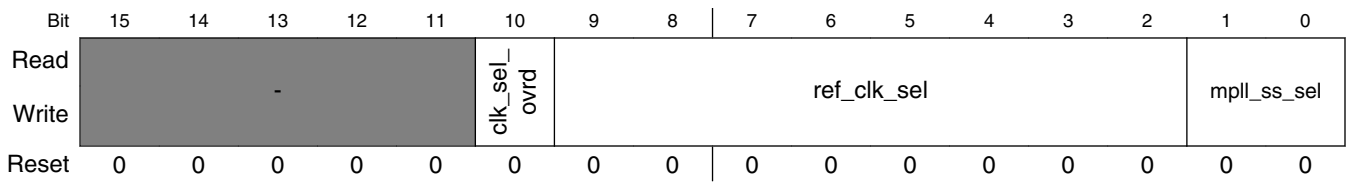
64.7.26 Clock Select Override Register (clock_SEL_OVRD)

Address: 0x001A

Reset value: 16'b0000 0000 0000 0000

This register contains the clock select override, the ref_clk_sel override value, and the mpll_ss_sel override value.

Address: clock_SEL_OVRD is 0h base + 1Ah offset = 001Ah



clock_SEL_OVRD field descriptions

Field	Description
15–11 -	Reserved
10 clk_sel_ovrd	Override control, indicating that the overridden value is active
9–2 ref_clk_sel	Reference clock select
1–0 mpll_ss_sel	MPLL spread spectrum select

64.7.27 Reset Register (clock_RESET)

Address: 0x7F3F

Reset value: 16'b xxxx xxxx xxxx xxx0

This register is a write-only register (not a real register) that resets the SATA2 PHY.

Upon writing the PHY reset bit in the reset register, the internal PHY reset is active immediately. Since the reset also affects the control register state machine, there will not be an acknowledgement of the write; that is, cr_ack will not be asserted.

NOTE

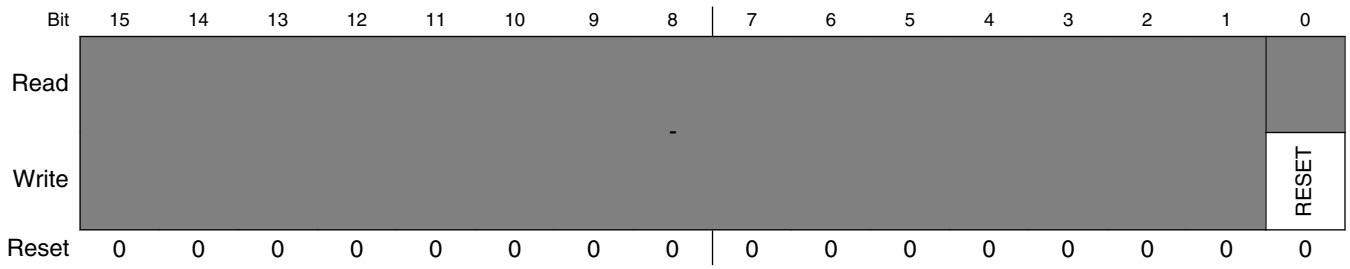
Diagnostic code should treat the *lack* of an acknowledgment of the write as a *successful* write; alternatively, it should treat the PHY *acknowledging* a write of the reset as a write *failure*. This

is the opposite expectation of all other registers, where the lack is a failure and the *acknowledge* is successful.

NOTE

It is sufficient to wait 20 ref_clock cycles in order to determine that the acknowledgement has not occurred.

Address: clock_RESET is 0h base + 7F3Fh offset = 7F3Fh



clock_RESET field descriptions

Field	Description
15–1 -	Reserved
0 RESET	Writing a 1 to this field resets the SATA2 PHY.

64.8 lanej Memory Map/Register Definition

Register Addresses

The SATA2 PHY comprises two parts, the Clock module and one or more lane modules. Only one Clock module exists in each assembly, but there can be multiple lanes.

NOTE

SATA PHY registers are only accessible by the corresponding controller (SATA_P0_PCSR_PHYCR and SATA_P0_PCSR_PHYSR) or in debug through the JTAG port. SATA PHY is not memory mapped to processor address space, so the absolute addresses shown is the relative address and is not valid. See SATA Memory Map for more information.

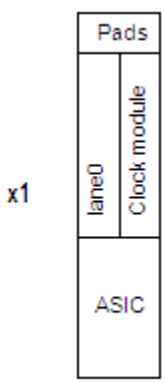
Broadcast Addressing

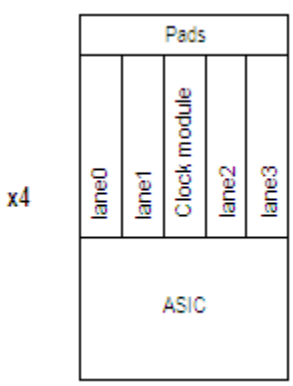
In addition to supporting writes to a single register, the register controller in the SATA2 PHY also supports broadcast writes. Broadcast writes make it easy (a single register write operation) to write the same value into N instantiations of the same register that exist in N lanes. To do a broadcast write to all lanes, simply replace the upper byte of the address with 0xA3. Note that broadcast reading of registers is not possible.

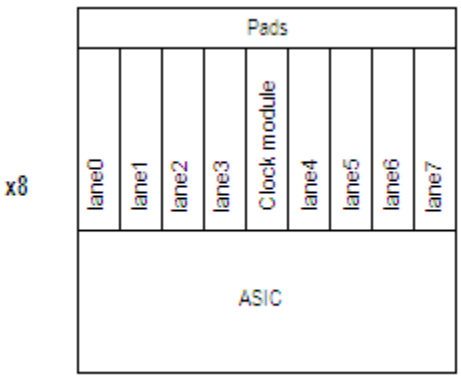
Per-Lane Register Numbering

NOTE

In an N lane SATA2 PHY, there are a total of N instantiations of each register. Each register name begins with "lane0." When referring to a particular lane, the "0" is replaced with the lane number, as in lane3.dpll.freq. The upper byte in the 16-bit address for all registers in the lane is 0x2{lane#}, where lane# is 0x0 for an x1 configuration, between 0x0 and 0x7 for an x8 configuration, and between 0x0 and 0x3 for an x4 configuration. Lane numbering always starts from 0 at the far-left side, as shown in the following three figures:







In the tables in this section, the lower byte in the 16-bit address is noted as the base address.

lane0 memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
2001	Transmit Input Status Register (lane0_TX_STAT)	16	R	0000h	64.8.1/ 3993
2002	Receiver Input Status Register (lane0_RX_STAT)	16	R	0000h	64.8.2/ 3994
2003	Output Status Register (lane0_OUT_STAT)	16	R	0000h	64.8.3/ 3995
2004	Transmit Input Override Register (lane0_TX_OVRD)	16	R/W	0007h	64.8.4/ 3996
2005	Receive Input Override Register (lane0_RX_OVRD)	16	R/W	1416h	64.8.5/ 3997
2006	Output Override Register (lane0_OUT_OVRD)	16	R/W	0011h	64.8.6/ 3998
2007	Debug Control Register (lane0_DBG_CTL)	16	R/W	0000h	64.8.7/ 3998
2010	Pattern Generator Control Register (lane0_PG_CTL)	16	R/W	0000h	64.8.8/ 4000
2018	Pattern Matcher Control Register (lane0_PM_CTL)	16	R/W	0000h	64.8.9/ 4001
2019	Pattern Matcher Error Register (lane0_PM_ERR)	16	R/W	0000h	64.8.10/ 4002
201A	DPLL Phase Register (lane0_DPLL_PHASE)	16	R/W	0000h	64.8.11/ 4002
201B	DPLL Frequency Register (lane0_DPLL_FREQ)	16	R/W	0000h	64.8.12/ 4003
201C	Scope Control Register (lane0_SCOPE_CTL)	16	R/W	0000h	64.8.13/ 4004

Table continues on the next page...

lane0 memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
201D	Receiver Control Register (lane0_RX_CTL)	16	R/W	000Fh	64.8.14/ 4004
201E	Receiver Debug Register (lane0_RX_DBG)	16	R/W	0000h	64.8.15/ 4005
2030	Receive Analog Control Register (lane0_RX_ANA_CONTROL)	16	R/W	0020h	64.8.16/ 4007
2031	Receive ATB Register (lane0_RX_ANA_ATB)	16	R/W	0000h	64.8.17/ 4007
2032	Rx PLL Programming 2 Register (lane0_PLL_PRG2)	16	R/W	0000h	64.8.18/ 4008
2033	Rx PLL Programming 1 Register (lane0_PLL_PRG1)	16	R/W	02A9h	64.8.19/ 4009
2034	Rx PLL Measurement Register (lane0_PLL_PRG3)	16	R/W	0000h	64.8.20/ 4010
2035	Transmit ATB 1 Control Register (lane0_TX_ANA_ATBSEL1)	16	R/W	0000h	64.8.21/ 4011
2036	Transmit ATB 2 Control Register (lane0_TX_ANA_ATBSEL2)	16	R/W	0000h	64.8.22/ 4012
2037	Transmit Analog Control Register (lane0_TX_ANA_CONTROL)	16	R/W	0000h	64.8.23/ 4013

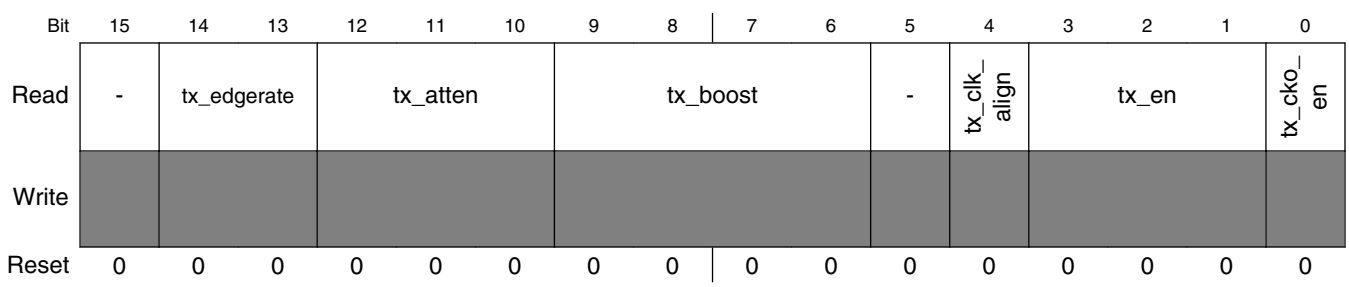
64.8.1 Transmit Input Status Register (lane0_TX_STAT)

Address: 0x2001

Reset value: 16'b xxxx xxxx xxxx xxxx (depends on inputs)

This register indicates the status of transmit control inputs.

Address: lane0_TX_STAT is 0h base + 2001h offset = 2001h



lane0_TX_STAT field descriptions

Field	Description
15 -	Always reads as 1
14–13 tx_edgerate	Edge rate control
12–10 tx_atten	Attenuation amount control
9–6 tx_boost	Boost amount control
5 -	Always reads as 0
4 tx_clk_align	Command to align clocks
3–1 tx_en	Transmit enable control
0 tx_cko_en	tx_cko clock enable

64.8.2 Receiver Input Status Register (lane0_RX_STAT)

Address: 0x2002

Reset value: 16'b xxxx xxxx xxxx xxxx (depends on inputs)

This register indicates the status of receiver control inputs.

Address: lane0_RX_STAT is 0h base + 2002h offset = 2002h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	-		los_ctl		dpll_reset	rx_dpll_mode			rx_eq_val			rx_term_en	rx_align_en	rx_en	rx_pll_pwron	half_rate
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

lane0_RX_STAT field descriptions

Field	Description
15–14 -	Always reads as 1
13–12 los_ctl	LOS filtering mode control

Table continues on the next page...

lane0_RX_STAT field descriptions (continued)

Field	Description
11 dpll_reset	DPLL reset control
10–8 rx_dpll_mode	DPLL mode control
7–5 rx_eq_val	Equalization amount control
4 rx_term_en	Receiver termination enable
3 rx_align_en	Receiver alignment enable
2 rx_en	Receiver enable control
1 rx_pll_pwron	PLL power state control
0 half_rate	Digital half-rate data control

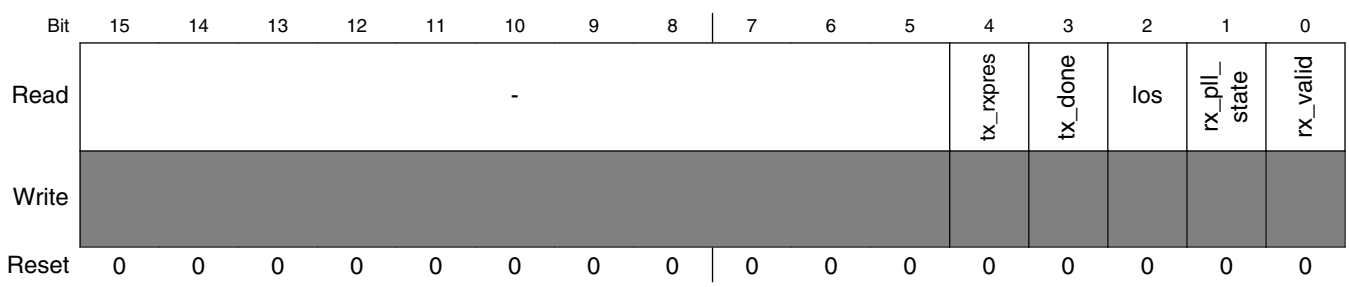
64.8.3 Output Status Register (lane0_OUT_STAT)

Address: 0x2003

Reset value: 16'b xxxx xxxx xxxx xxxx (depends on inputs)

This register indicates the status of output signals.

Address: lane0_OUT_STAT is 0h base + 2003h offset = 2003h



lane0_OUT_STAT field descriptions

Field	Description
15–5 -	Always reads as 1
4 tx_rxpres	Transmit receiver detection result

Table continues on the next page...

lane0_OUT_STAT field descriptions (continued)

Field	Description
3 tx_done	Transmit operation is complete output
2 los	Loss of signal output
1 rx_pll_state	Current state of Rx PLL
0 rx_valid	Receiver valid output

64.8.4 Transmit Input Override Register (lane0_TX_OVRD)

Address: 0x2004

Reset value: 16'b 0000 0000 0000 0111

This register contains the override transmitter control inputs.

Address: lane0_TX_OVRD is 0h base + 2004h offset = 2004h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	ovrd	tx_edgerate		tx_atten			tx_boost				tx_dis_align	tx_clk_align	tx_en			tx_cko_en
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

lane0_TX_OVRD field descriptions

Field	Description
15 ovrd	Enables override of all bits in this register
14–13 tx_edgerate	Edge rate control
12–10 tx_atten	Attenuation amount control
9–6 tx_boost	Boost amount control
5 tx_dis_align	Disables clock alignment FSM
4 tx_clk_align	Command to align clocks
3–1 tx_en	Transmit enable control
0 tx_cko_en	tx_cko clock enable

64.8.5 Receive Input Override Register (lane0_RX_OVRD)

Address: 0x2005

Reset value: 16'b x001 0100 0001 1110

This register contains the override of receiver control inputs.

Address: lane0_RX_OVRD is 0h base + 2005h offset = 2005h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	-	ovrd	los_ctl		dpll_reset	rx_dpll_mode			rx_eq_val			rx_term_en	rx_align_en	rx_en	rx_pll_pwron	half_rate
Reset	0	0	0	1	0	1	0	0	0	0	0	1	0	1	1	0

lane0_RX_OVRD field descriptions

Field	Description
15 -	Reserved
14 ovrd	Enables override of all bits in this register
13–12 los_ctl	LOS filtering mode control
11 dpll_reset	DPLL reset control
10–8 rx_dpll_mode	DPLL mode control
7–5 rx_eq_val	Equalization amount control
4 rx_term_en	Receiver termination enable
3 rx_align_en	Receiver alignment enable
2 rx_en	Receiver enable control
1 rx_pll_pwron	PLL power state control
0 half_rate	Digital half-rate data control

64.8.6 Output Override Register (lane0_OUT_OVRD)

Address: 0x2006

Reset value: 16'b xxxx xxxx xx01 0001

This register contains the override of output signals.

Address: lane0_OUT_OVRD is 0h base + 2006h offset = 2006h

lane0_OUT_OVRD field descriptions

Field	Description
15–6 -	Reserved
5 ovrd	Enables override of all bits in this register
4 tx_rxpres	Transmit receiver detection result
3 tx_done	Transmit operation is complete output
2 los	Loss of signal output
1 rx_pll_state	Current state of Rx PLL
0 rx_valid	Receiver valid output

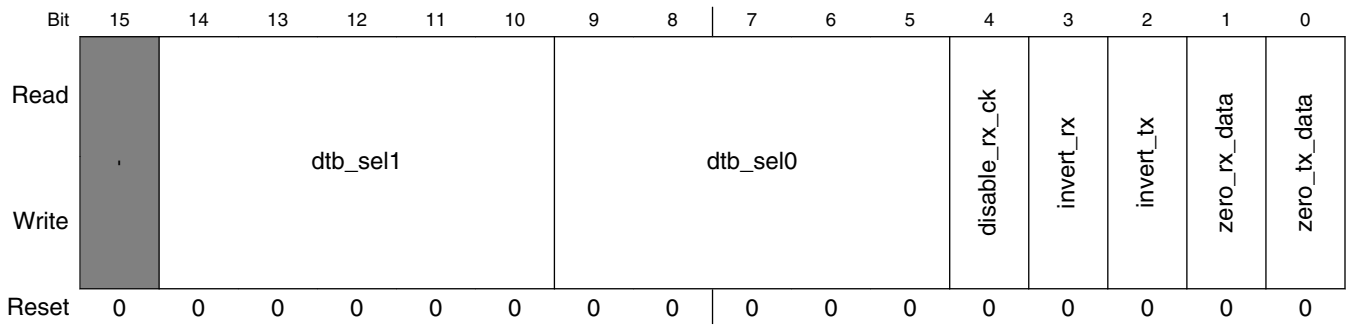
64.8.7 Debug Control Register (lane0_DBG_CTL)

Address: 0x2007

Reset value: 16'b x000 0000 0000 0000

This register contains debug controls.

Address: lane0_DBG_CTL is 0h base + 2007h offset = 2007h



lane0_DBG_CTL field descriptions

Field	Description
15 -	Reserved
14–10 dtb_sel1	All other bits: Disabled Selects wire to drive onto DTB bit 1: 00000 Disabled 00001 half_rate 00010 tx_en[0] 00011 tx_en[1] 00100 tx_en[2] 00101 tx_clk_align 00110 n/a 00111 rx_pll_pwron 01000 rx_en 01001 dpll_reset 01010 rx_valid 01011 rx_pll_state 01100 los 01101 tx_done 01110 rx_ck (output to ASIC) 01111 ck_rx (PLL output) 10000 ck_los 10001 tx_ck (ASIC input) 10010 ck_tx_out (serializer output) 10011 pll_pwron (analog input) 10100 pll_reset 10101 ser_clk_kill 10110 LBERT pg strobe 10111 rx_present_p (sampled) 11000 rx_present_m (sampled) 11001 acjt receiver o/p from rx_p 11010 acjt receiver o/p from rx_m
9–5 dtb_sel0	All other bits: Disabled Selects wire to drive onto DTB bit 0:

Table continues on the next page...

lane0_DBG_CTL field descriptions (continued)

Field	Description
	00000 Disabled 00001 half_rate 00010 tx_en[0] 00011 tx_en[1] 00100 tx_en[2] 00101 tx_clk_align 00110 n/a 00111 rx_pll_pwron 01000 rx_en 01001 dpll_reset 01010 rx_valid 01011 rx_pll_state 01100 los 01101 tx_done 01110 rx_ck (output to ASIC) 01111 ck_rx (PLL output) 10000 ck_los 10001 tx_ck (ASIC input) 10010 ck_tx_out (serializer output) 10011 pll_pwron (analog input) 10100 pll_reset 10101 ser_clk_kill 10110 LBERT pattern generator strobe 10111 rx_present_p (sampled) 11000 rx_present_m (sampled) 11001 acjt receiver o/p from rx_p 11010 acjt receiver o/p from rx_m
4 disable_rx_ck	Disables rx_ck output
3 invert_rx	Inverts receive data (pre-LBERT)
2 invert_tx	Inverts transmit data (post-LBERT)
1 zero_rx_data	Overrides all receive data to zeros
0 zero_tx_data	Overrides all transmit data to zeros

64.8.8 Pattern Generator Control Register (lane0_PG_CTL)

Address: 0x2010

Reset value: 16'b xx00 0000 0000 0000 0000

This register contains pattern generator controls.

Address: lane0_PG_CTL is 0h base + 2010h offset = 2010h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	-		pat0										trigger_err	mode		
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

lane0_PG_CTL field descriptions

Field	Description
15–14 -	Reserved
13–4 pat0	Pattern for modes 3-5
3 trigger_err	Inserts a single error into the LSB
2–0 mode	Selects a pattern to generate: 000 Disabled 001 lfsr15. X^15 + X^14 + 1 010 lfsr7. X^7 + X^6 + 1 011 Fixed word (PAT0)3'b 100 DC balanced word (PAT0, ~PAT0) 101 Fixed pattern: (000, PAT0, 3ff, ~PAT0) 110 Reserved 111 Reserved

64.8.9 Pattern Matcher Control Register (lane0_PM_CTL)

Address: 0x2018

Reset value: 16'b xxxx xxxx xxxx 0000

This register contains pattern matcher controls.

Address: lane0_PM_CTL is 0h base + 2018h offset = 2018h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	-												sync	mode		
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

lane0_PM_CTL field descriptions

Field	Description
15–4 -	Reserved
3 sync	To enable checking, "Synchronize pattern matcher LFSR with incoming data" must be turned on, then turned off.
2–0 mode	All other bits: Reserved Pattern to match: 000 Disabled 001 lfsr15 010 lfsr7 011 d[n] = d[n-10] 100 d[n] = !d[n-10]

64.8.10 Pattern Matcher Error Register (lane0_PM_ERR)

Address: 0x2019

Reset value: 16'b xxxx xxxx xxxx xxxx (a read resets the register)

This register is the pattern match error counter. When the clock to the error counter is turned off, reads and writes to the register are queued until the clock is turned back on.

Address: lane0_PM_ERR is 0h base + 2019h offset = 2019h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	ov14	count														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

lane0_PM_ERR field descriptions

Field	Description
15 ov14	If this field is active, the count is multiplied by 128. If ov14 is set to 1 and count = 2^15 - 1, indicates overflow of counter.
14–0 count	Current error count If the ov14 field is active, the count is multiplied by 128.

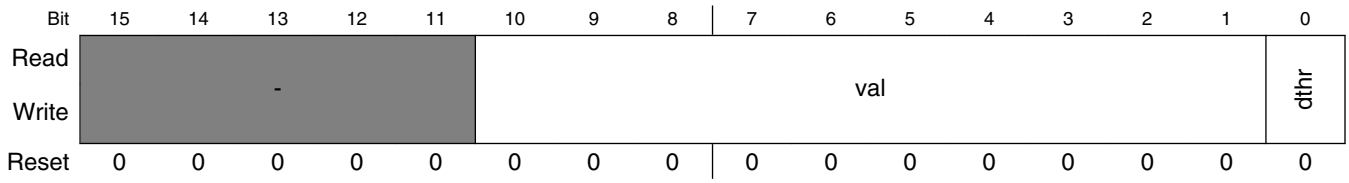
64.8.11 DPLL Phase Register (lane0_DPLL_PHASE)

Address: 0x201A

Reset value: 16'b xxxx x000 0000 0000

This register contains the current phase selector value.

Address: lane0_DPLL_PHASE is 0h base + 201Ah offset = 201Ah



lane0_DPLL_PHASE field descriptions

Field	Description
15–11 -	Reserved
10–1 val	Phase is .UI/512 x VAL ps from zero reference
0 dthr	Bits below the useful resolution

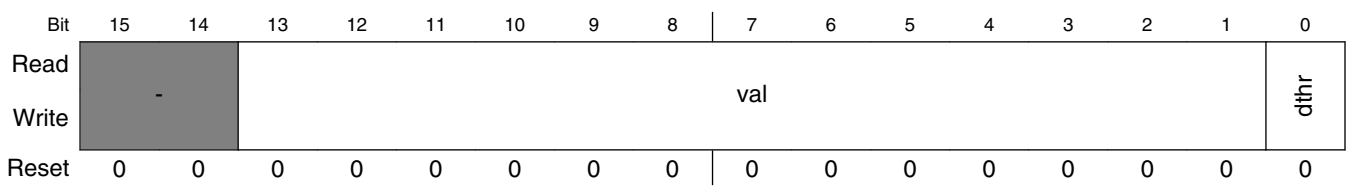
64.8.12 DPLL Frequency Register (lane0_DPLL_FREQ)

Address: 0x201B

Reset value: 16'b xx00 0000 0000 0000

This register contains the current frequency integrator value.

Address: lane0_DPLL_FREQ is 0h base + 201Bh offset = 201Bh



lane0_DPLL_FREQ field descriptions

Field	Description
15–14 -	Reserved
13–1 val	Frequency is 1.526 x VAL ppm from the reference
0 dthr	Bits below the useful resolution

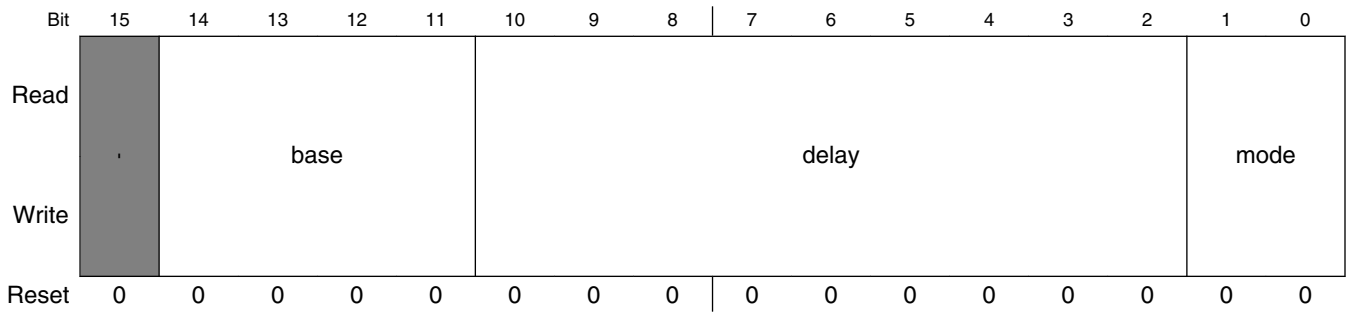
64.8.13 Scope Control Register (lane0_SCOPE_CTL)

Address: 0x201C

Reset value: 16'b x000 0000 0000 0000

This register contains control bits for the per-transceiver scope portion.

Address: lane0_SCOPE_CTL is 0h base + 201Ch offset = 201Ch



lane0_SCOPE_CTL field descriptions

Field	Description
15 -	Reserved
14–11 base	The bit to be sampled when mode = 2'b01
10–2 delay	Number of symbols to skip between samples
1–0 mode	Mode of counters: 00 Off 01 Sample every 10 bits (see base) 10 Sample every 11 + 10 x delay bits 11 Sample every 12 + 10 x delay bits

64.8.14 Receiver Control Register (lane0_RX_CTL)

Address: 0x201D

Reset value: 16'b x000 0000 0000 1111

This register contains control bits for the receiver in the recovered domain.

Address: lane0_RX_CTL is 0h base + 201Dh offset = 201Dh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	-	switch_val	ovrd_switch	mode_bp			frug_value		phug_value		ovrd_dpll_gain	phdet_pol	phdet_edge		phdet_en	
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

lane0_RX_CTL field descriptions

Field	Description
15 -	Reserved
14 switch_val	Value to override the data/phase MUX
13 ovrd_switch	Overrides the value of the data/phase MUX
12–10 mode_bp	Sets BP 2:0 to longer timescale (for FTS patterns): 10 Starts phase update gain (PHUG) profile at 4/3 cycles 11 Starts frequency update gain (FRUG) profile at 46/42 additional cycles 12 Ends frequency update gain (FRUG) profile at 142/110 additional cycles
9–8 frug_value	Overrides value for frequency update gain (FRUG)
7–6 phug_value	Overrides value for phase update gain (PHUG)
5 ovrd_dpll_gain	Overrides phase update gain (PHUG) and frequency update gain (FRUG) values
4 phdet_pol	Reverses polarity of phase error
3–2 phdet_edge	Edges to use for phase detection Top bit is rising edges, bottom is falling.
1–0 phdet_en	Enables phase detector Top bit is odd slicers, bottom is even.

64.8.15 Receiver Debug Register (lane0_RX_DBG)

Address: 0x201E

Reset value: 16'b xxxx xxxx 0000 0000

This register contains control bits for receiver debugging.

Address: lane0_RX_DBG is 0h base + 201Eh offset = 201Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	-								dtb_sel1				dtb_sel0			
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

lane0_RX_DBG field descriptions

Field	Description
15–8 -	Reserved
7–4 dtb_sel1	Selects wire to go on DTB bit 1 0000 Disabled 0001 ana_los 0010 los_ref_pos 0011 los_ref_neg_l 0100 sata_los 0101 los_rck 0110 eios_idle 0111 pcie_los 1000 coast_dpll 1001 misalign 1010 realign 1011 com_detect 1100 idl_detect 1101 fts_detect 1110 fts_err 1111 bp_state[1]
3–0 dtb_sel0	Selects wire to go on DTB bit 0 0000 Disabled 0001 ana_los 0010 los_ref_pos 0011 los_ref_neg_l 0100 sata_los 0101 los_rck 0110 eios_idle 0111 pcie_los 1000 coast_dpll 1001 misalign 1010 realign 1011 com_detect 1100 idl_detect 1101 fts_detect 1110 fts_err 1111 bp_state[1]

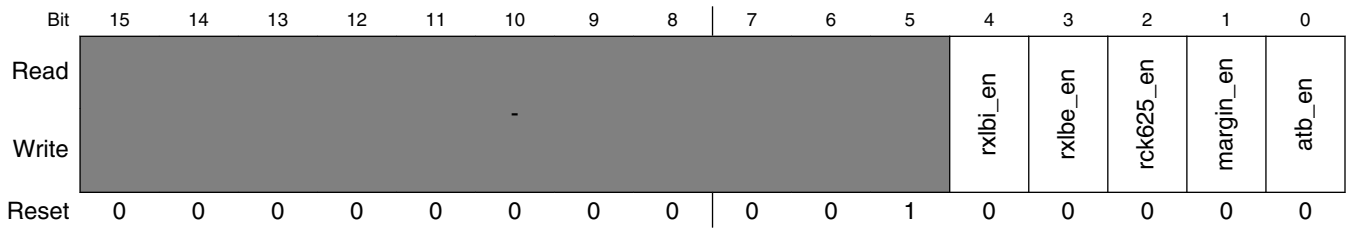
64.8.16 Receive Analog Control Register (lane0_RX_ANA_CONTROL)

Address: 0x203C

Reset value: 16'b xxxx xxxx xx10 0000

This register contains Rx control bits.

Address: lane0_RX_ANA_CONTROL is 0h base + 2030h offset = 2030h



lane0_RX_ANA_CONTROL field descriptions

Field	Description
15–5 -	Reserved
4 rxlbi_en	Digital serial (internal) loopback enable bit When this field is set to 1, an output from the serializer is connected to the first comparator stage.
3 rxlbe_en	Wafer level (external) loopback enable bit When this field is set to 1, the lane's output (Tx) is connected to the lane's input (Rx) through pass gates.
2 rck625_en	rck625 enable bit When this field is set to 1, pll_alt_ref is driven by ck_i_p / 2.
1 margin_en	1 RWCr Margin enable bit When this field is set to 1, margining is enabled. When doing margining, ensure that you set atb_en so that atb_s_p/m are connected.
0 atb_en	ATB enable bit When this field is set to 1, internal atb_s_p,m = external atb_s_p,m.

64.8.17 Receive ATB Register (lane0_RX_ANA_ATB)

Address: 0x2031

Reset value: 16'b xxxx xxxx xx00 0000

This register contains Rx ATB bits.

Address: lane0_RX_ANA_ATB is 0h base + 2031h offset = 2031h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	-										sensem_vref_los	sensem_vcm	sensem_rx_m	sensep_rx_p	forcep_rx_m	forcep_rx_p
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

lane0_RX_ANA_ATB field descriptions

Field	Description
15–6 -	Reserved
5 sensem_vref_los	Connects atb_s_m to vref_los (vref_rx / 14)
4 sensem_vcm	Connects atb_s_m to Rx vcm Use in margining.
3 sensem_rx_m	Connects atb_s_m to rx_m Use for measuring Rx termination.
2 sensep_rx_p	Connects atb_s_p to rx_p Use for measuring Rx termination.
1 forcep_rx_m	Connects atb_f_p to rx_m Use for measuring Rx termination.
0 forcep_rx_p	Connects atb_f_p to rx_p Use for measuring Rx termination.

64.8.18 Rx PLL Programming 2 Register (lane0_PLL_PRG2)

Address: 0x2032

Reset value: 16'b xxxx xxxx 0000 0000

This is an 8-bit programming register.

Address: lane0_PLL_PRG2 is 0h base + 2032h offset = 2032h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	-								atb_sense_sel	frc_hcpl	hcpl_lcl	frc_pwron	pwron_lcl	frc_reset	reset_lcl	enable_test_pd
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

lane0_PLL_PRG2 field descriptions

Field	Description
15–8 -	Reserved
7 atb_sense_sel	Controls proportional charge pump current: 0 Disconnects PLL from analog test bus. No PLL signals can be viewed on the ATB. 1 Enables signals internal to PLL to connect to the analog test bus.
6 frc_hcpl	Enables override of hcpl default value. Enables hcpl_lcl to control high-coupling mode.
5 hcpl_lcl	Forces coupling in VCO: Field is valid only when frc_hcpl is set to 1'b1. 0 Forces coupling in VCO to minimum. 1 Forces coupling in VCO to maximum.
4 frc_pwron	Enables override of default value of pll_pwron. Enables pwron_lcl to control PLL power-on.
3 pwron_lcl	Controls power to PLL: Field is valid only when frc_pwron is set to 1'b1. 0 Powers down PLL. 1 Supplies power to PLL.
2 frc_reset	Enables override of default value of pll_pwron. Enables pwron_lcl to control PLL power-on.
1 reset_lcl	Resets PLL: Field is valid only when frc_reset is set to 1'b1. 0 PLL is in normal mode. 1 PLL is held/placed in reset.
0 enable_test_pd	Controls phase interpolator test mode: 0 Disables phase interpolator test mode. 1 Tests phase linearity of phase interpolator and VCO.

64.8.19 Rx PLL Programming 1 Register (lane0_PLL_PRG1)

Address: 0x2033

Reset value: 16'b xxxx xx10 1010 1001

This is a 10-bit programming register.

Address: lane0_PLL_PRG1 is 0h base + 2033h offset = 2033h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	-							sel_rxck	prop_cntrl			int_cntrl			-	
Write																
Reset	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	1

lane0_PLL_PRG1 field descriptions

Field	Description
15–9 -	Reserved
8 sel_rxck	Uses recovered clock as reference to the PLL: 0 Uses MPLL output as reference to the PLL 1 Uses recovered clock as reference to PLL
7–5 prop_cntrl	Controls proportional charge pump current Proportional current = (n + 1) / 8 x full_scale Default value = 3'b101: 0.75 x full_scale
4–2 int_cntrl	Controls integral charge pump current Integral current = (n + 1) / 8 x full_scale Default value = 3'b010: 0.375 x full_scale
1–0 -	Reserved

64.8.20 Rx PLL Measurement Register (lane0_PLL_PRG3)

Address: 0x2034

Reset value: 16'b xxxx xx00 0000 0000

This is a 10-bit programming register.

Address: lane0_PLL_PRG3 is 0h base + 2034h offset = 2034h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	-						meas_bias	meas_vcntrl	meas_vref	meas_vp16	meas_startup	meas_vco	meas_vp_cp	meas_1v	meas_crowbar	-
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

lane0_PLL_PRG3 field descriptions

Field	Description
15–10 -	Reserved
9 meas_bias	Measures copy of bias current in oscillator on atb_force_m.
8 meas_vcntrl	Measures vcntrl on atb_sense_m. If meas_vref is also set, atb_sense_p,m measures vref - vcntrl.
7 meas_vref	Measures vref on atb_sense_p; gd on atb_sense_m. If meas_vcntrl is also set, atb_sense_p,m measures vref - vcntrl.
6 meas_vp16	Measures vp16 on atb_sense_p; gd on atb_sense_m.
5 meas_startup	Measures startup voltage on atb_sense_p; gd on atb_sense_m.
4 meas_vco	Measures VCO supply voltage on atb_sense_p; gd on atb_sense_m.
3 meas_vp_cp	Measures vp_cp voltage on atb_sense_p; gd on atb_sense_m. If meas_1v is also set, atb_sense_p,m measures vpcp - vp.
2 meas_1v	Measures 1-V supply voltage on atb_sense_m. If meas_vp_cp is also set, atb_sense_p,m measures vpcp - vp.
1 meas_crowbar	Measures crowbar bias voltage on atb_sense_p; gd on atb_sense_m.
0 -	Reserved

64.8.21 Transmit ATB 1 Control Register (lane0_TX_ANA_ATBSEL1)

Address: 0x2035

Reset value: 16"b xxxx xxxx 0000 0000

This register contains Tx ATB control bits.

Address: lane0_TX_ANA_ATBSEL1 is 0h base + 2035h offset = 2035h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	-								vbpf_s_p	txm_s_m	txm_f_p	txp_s_p	txp_f_p	vreg_s_m	vref_s_p	vgr_s_p
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

lane0_TX_ANA_ATBSEL1 field descriptions

Field	Description
15–8 -	Reserved
7 vbpf_s_p	vbpf in edge rate control circuit on ATB_S_P To validate this field, set lane0.tx_ana.atbsel2.atb_en.
6 txm_s_m	txm on ATB_S_M To validate this field, set lane0.tx_ana.atbsel2.atb_en.
5 txm_f_p	txm connected to ATB_S_P For termination resistance measurements.
4 txp_s_p	txp connected to ATB_S_P To validate this field, set lane0.tx_ana.atbsel2.atb_en.
3 txp_f_p	txp connected to ATB_F_P For termination resistance measurements.
2 vreg_s_m	Regulator output voltage on ATB_S_M To validate this field, set lane0.tx_ana.atbsel2.atb_en.
1 vref_s_p	tx_vref voltage on ATB_S_P To validate this field, set lane0.tx_ana.atbsel2.atb_en.
0 vgr_s_p	Regulator gate voltage on ATB_S_P To validate this field, set lane0.tx_ana.atbsel2.atb_en.

64.8.22 Transmit ATB 2 Control Register (lane0_TX_ANA_ATBSEL2)

Address: 0x2036

Reset value: 16'b xxxx xxxx 0000 0000

This register contains Tx ATB control bits.

Address: lane0_TX_ANA_ATBSEL2 is 0h base + 2036h offset = 2036h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	-								atb_en	vrefrxd_s_m	vcm_s_p	vbns_s_m	vbps_s_p	vbnf_s_m	enlpbk	en_txilpbk
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

lane0_TX_ANA_ATBSEL2 field descriptions

Field	Description
15–8 -	Reserved
7 atb_en	RWCr 7 RWCr Connects internal and external ATB buses Required for all ATB measurements.
6 vrefrxd_s_m	Reference voltage for RX_DETECT on ATB_S_M To validate this field, set the atb_en field.
5 vcm_s_p	Vcm replica on ATB_S_P To validate this field, set the atb_en field.
4 vbns_s_m	vbps in edge rate control circuit on ATB_S_M To validate this field, set the atb_en field.
3 vbps_s_p	vbps in edge rate control circuit on ATB_S_M To validate this field, set the atb_en field.
2 vbnf_s_m	vbnf in edge rate control circuit on ATB_S_M To validate this field, set the atb_en field.
1 enlpbk	Enables Tx external loopback Ensure that internal loopback is not on.
0 en_txilpbk	Enables Tx internal loopback

64.8.23 Transmit Analog Control Register (lane0_TX_ANA_CONTROL)

Address: 0x237

Reset value: 16"b xxxx xxxx 0000 0000

This register contains Tx power state control bits.

Address: lane0_TX_ANA_CONTROL is 0h base + 2037h offset = 2037h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	-								frc_pwrst	en_lcl		frc_do	dataovrd_lcl	frc_beacon	bcn_lcl	-
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

lane0_TX_ANA_CONTROL field descriptions

Field	Description
15–8 -	Reserved
7 frc_pwrst	Locally forces power state When this field is set to 1, the tx_en[1:0] input is overridden by en_lcl.
6–5 en_lcl	Locally forces tx_en[1:0]: 00 Power off 01 Tx idle (slow) 10 Transmit data 11 Tx idle (fast)
4 frc_do	Forces dataovrd locally When set to 1, this field overrides the input data_ovrd value.
3 dataovrd_lcl	RWCr Local dataovrd control value To validate this field, set lane0.tx_ana.control.frc_do.
2 frc_beacon	Forces beacon to local value (bcn_lcl) When this field is set to 1, BCN_LVL overrides input value.
1 bcn_lcl	Local beacon on/off control value To validate this field, set lane0.tx_ana.control.frc_beacon.
0 -	Reserved

Chapter 65
Security Controller (SCC)

65.1 Introduction

Information security within a data processing platform is dependent upon the platform's ability to provide mandatory and optional information protection services. A secure platform is intended to protect sensitive data such as encryption keys, software and premium data content from unauthorized access in the form of inspection (read), modification (write), or execution (use). Security assurance refers to the degree of confidence that security claims are actually met and is therefore associated with the correctness of the security design and the integrity of the resources upon which it depends. The Freescale Platform Independent Security Architecture (PISA) incorporates the hardware and software resources necessary to provide high assurance for the security claims of a given embedded platform. PISA provides a secure foundation upon which platform architects can construct application-specific security mechanisms. The Security Controller (SCC) is an integral part of that PISA security foundation. A block diagram of the SCC is shown in Figure 65-1.

65.1.1 Overview

The SCC-AES is the second generation of the Security Controller. It implements secure RAM that can be used either as general-purpose memory for storing data and software, or as special confidentiality-preserving memory that protects disclosure-sensitive data such as cryptographic keys, passwords, code, or PIN numbers. The RAM within the SCC-AES is divided into 4 partitions, each of which contains 4 Kbytes, and is byte accessible. A partition can be allocated to either the TrustZone or non-TrustZone domains of the ARM, which can then determine how this partition can be accessed by the other domain and other bus masters. The SCC-AES aborts accesses to a partition that are not in accordance with the access permissions set by the partition's current owner.

The SCC-AES also incorporates cryptographic logic and a DMA engine that can be used to safely export the data stored within a partition to external RAM or non-volatile memory. As the data is exported it is encrypted with AES using a secret non-volatile 256-bit key (different in each SCC-AES). This secret key cannot be read or changed by software, or other bus masters, or by JTAG/scan logic. Before this 256-bit key is used to encrypt or decrypt data for a partition, the key value is modified by the SCC-AES hardware based upon the ID of the partition owner and the access permissions that the owner has set. Software can optionally supply additional key modification data that is also mixed with the basic secret key value. These key modifications ensure that data owned and exported by one domain cannot be decrypted and placed within a partition owned by the other domain, or in a partition with less restricted access permissions.

The SCC-AES also maintains the security state of the SoC. Upon reset, the SCC-AES hardware performs a number of security checks, and the High Assurance Boot (HAB) software on the SoC performs additional checks. If all checks pass, the SCC-AES's security state is set to Secure, otherwise the state is set to Non-secure. In Secure state, the Secret Key is available for cryptographic operations. In the Non-secure state, a known default key is used in place of the Secret Key. This allows the SCC-AES to be tested without compromising the confidentiality of data that has been encrypted using keys derived from the Secret Key. If RTIC and/or SAHARA are present on the SoC, the SCC-AES monitors their security status outputs. The SCC-AES also monitors indications of failures or potential security intrusions that are output by other components on the SoC. If security failures are detected, the SCC-AES's security state transitions to Fail state. In Fail state the SCC-AES shuts off access to, and begins erasing, the SCC-AES partitions that contain confidential data. Cryptographic operations are disabled in Fail state.

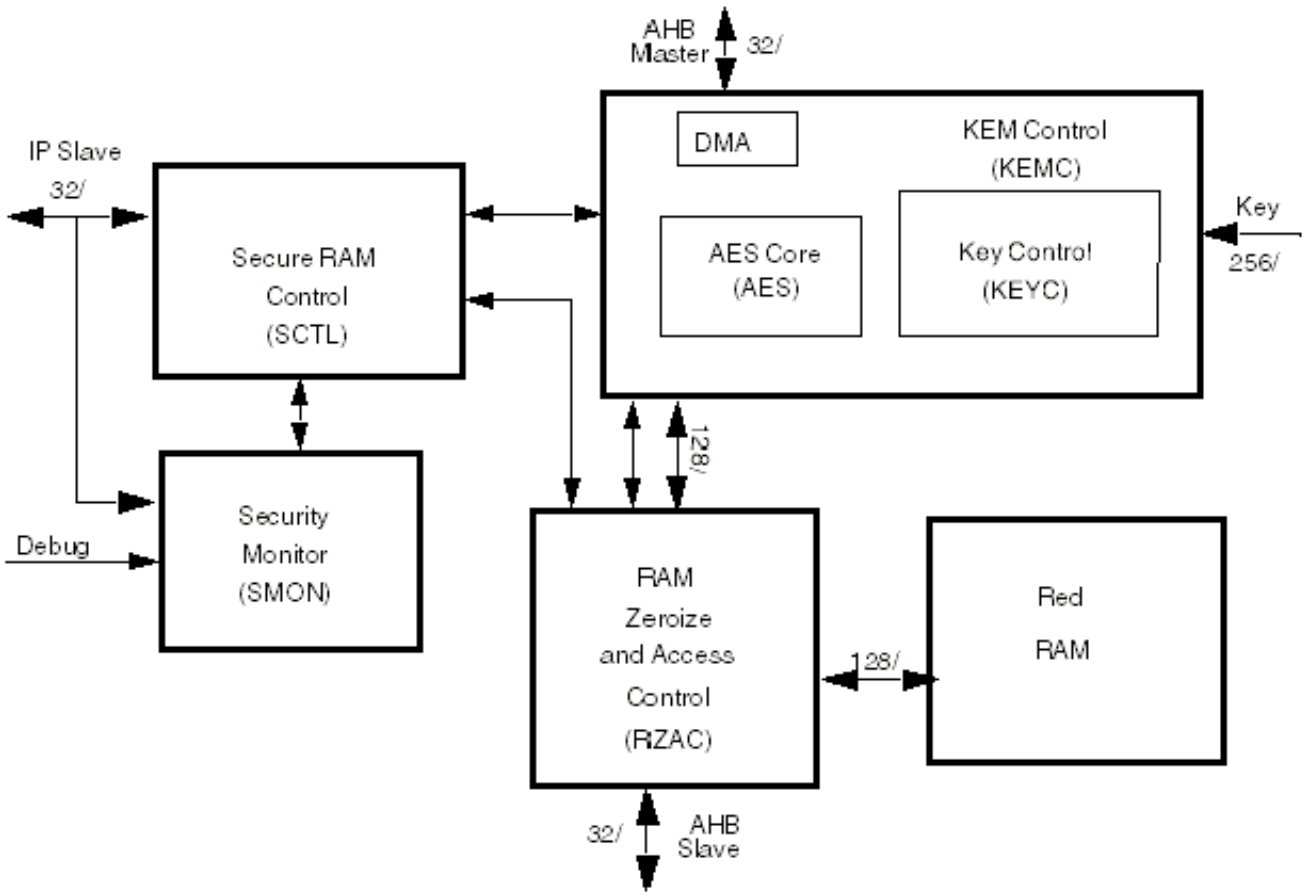


Figure 65-1. Block Diagram of SCC-AES

65.1.2 Features

The SCC-AES provides the following features:

- An autonomous hardware security state controller with security violation inputs, which are tied to all platform security violation signals to trigger a security shutdown when a violation is detected.
- Access Control to ensure supervisor-mode and user-mode access separation.
- Controls to ensure high assurance internal boot is the only procedure that can put the SCC-AES into the Secure state after Reset.
- A dedicated AES cryptographic engine to encrypt plaintext (RED) data or decrypt ciphertext (BLACK) data.
- A port to accept an OTP 256-bit statistically unique secret key, referred to as the Security Key, for use by the AES engine only when it is in the Secure state and which is provided/programmed at manufacture and unavailable for external test or inspection.

- Clearing of certain security sensitive registers upon scan entry.
- Special scan interface logic to detect whether scan test has been active since the last reset and consequently whether the current hardware security state can be trusted.
- A self-clearing (zeroizing) Red RAM, which clears itself upon reset, command, or failure and can therefore be used to store security-sensitive Red data (that is, disclosure-sensitive plaintext) such as cryptographic keys.
- A Security Timer, which is an independent security watchdog timer whose time-out triggers a security violation. Its use is mandatory in the Initialization state and optional in other SCC states.
- Bus Master access controls to prevent unauthorized access to Red Memory.
- An Algorithm Sequence Checker (ASC), which can be used by software to force software synchronization to the ASC's internal linear feedback shift register (LFSR) as a software assurance check. Its use is mandatory in the Initialization state and optional in other SCC states.
- A Bit Bank counter that can be used with the ASC to ensure that a cryptographic function uses the same number of algorithm bits as traffic bits to ensure that no traffic data is accidentally left in the clear.
- A Plaintext/Ciphertext comparator that may be used to ensure that a cryptographic algorithm has not been replaced with a function that XORs the data with a simple fixed pattern.
- Instruction storage and execution capability.

NOTE

Detailed information about the SCC is provided in the Security Manual. Contact your Freescale representative for information about obtaining this document.

Chapter 66
Smart Direct Memory Access Controller (SDMA)

66.1 Introduction

The Smart Direct Memory Access (SDMA) controller offers highly-competitive DMA features combined with software-based virtual-DMA flexibility. It enables data transfers between peripheral I/O devices and internal/external memories.

The SDMA controller helps maximize system performance by off-loading the ARM core in dynamic data routing.

66.1.1 Overview

The figure below shows a block diagram of the SDMA controller. It includes the custom RISC core along with its RAM, ROM, DMA units, the CRC unit, and the scheduler.

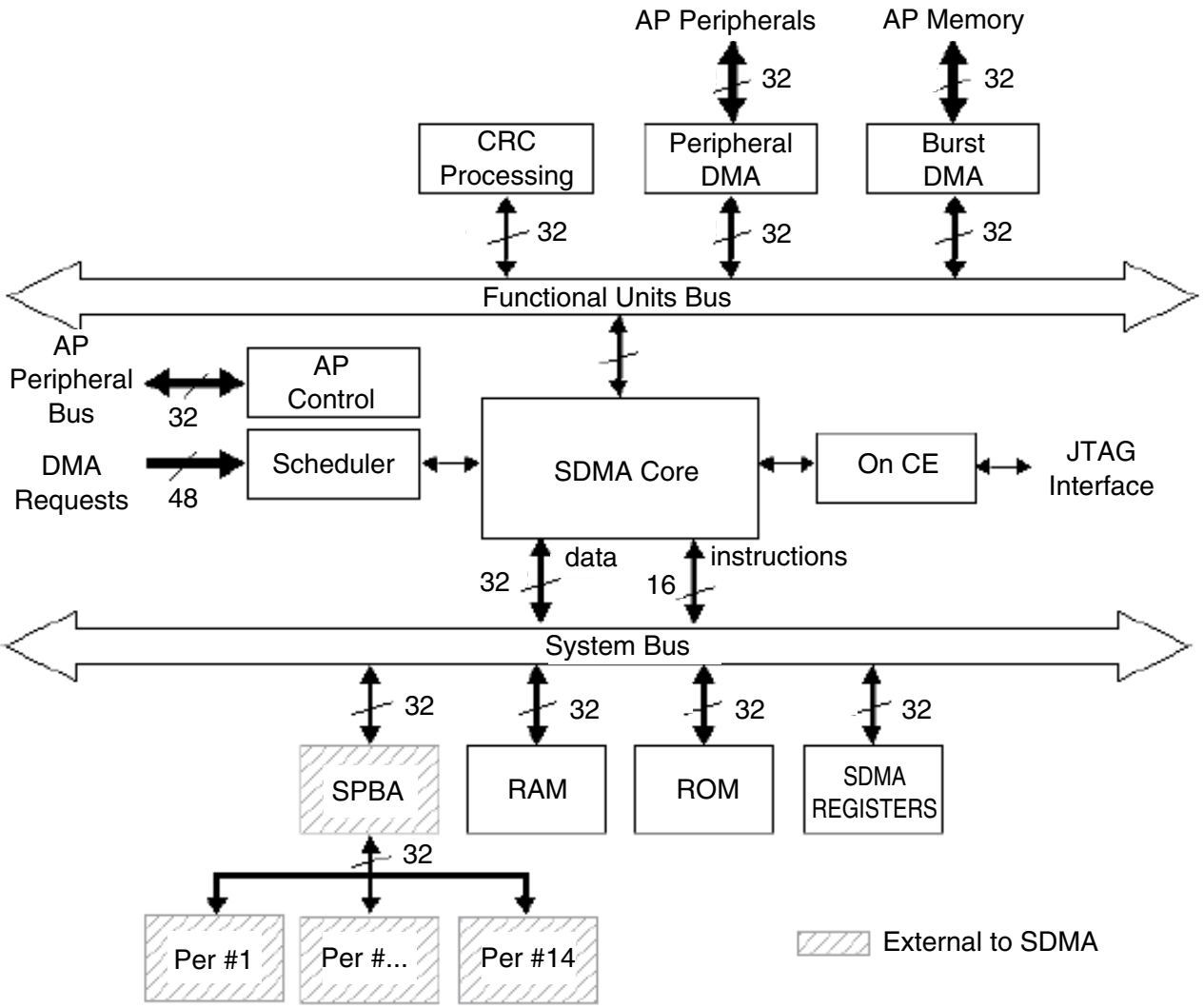


Figure 66-1. SDMA Block Diagram

The SDMA core executes short routines that perform DMA transfers; these routines are called *scripts*. The SDMA core interfaces to its own memory via the SDMA system bus. The SDMA system bus supports a 32-bit data path and a 16-bit address bus. The system bus datapath is used for both 16-bit instruction (program) memory access and 32-bit data access. DMA units interface to the core via the Functional Unit Bus and use dedicated registers to perform DMA transfers.

The SDMA memory contains a ROM and a RAM. The ROM contains startup scripts (for example, boot code) and other common utilities, which are referenced by the scripts that reside in the RAM. The internal RAM is divided into a context area and a script area (more details about this mapping are available in Instruction Memory Map and Data Memory Map).

Every transfer channel requires one context area to keep the contents of all the core and unit registers while inactive. Channel scripts are downloaded into the internal RAM by the SDMA using a dedicated channel that is started during the boot sequence. Downloads are invoked using commands and pointers provided by the ARM platform. Every channel contains a corresponding channel script located in RAM and/or ROM that can be reconfigured independently as-needed. Channel scripts can be stored in an external memory and downloaded when needed. The SDMA can be configured with any mixture of scripts to enable an endless combination of supported services.

The scheduler monitors and detects DMA requests, mapping them to channels, and mapping individual channels to a pre-configured priority. At any given point, the scheduler presents the highest priority channel that requires service to the SDMA core. A special SDMA core instruction is used to "conditionally yield" the current channel being executed to an eligible channel that requires service. If (and only if) there is an eligible channel pending, will the current channel execution be preempted.

There are two yield instructions that differently determine the eligible channels: In the first version, eligible channels are pending channels with a strictly higher priority than the current channel priority. In the second version (yieldge), eligible channels are pending channels with a priority that is greater or equal to the current channel priority. The scheduler detects devices that need service through its 48 DMA request inputs. After a request is detected, the scheduler determines the channel(s) that is (are) triggered by this request and marks it (them) as pending in the "Channel Pending (EP)" register. The priorities of all the pending channels are continuously evaluated in order to update the highest pending priority. The channel pending flag is cleared by the channel script when the transfer has completed.

The ARM platform control block contains the control registers used to configure the 32 individual channels. There are 48 Channel Enable registers, and every register maps one DMA request to any desired combination of channels. The 32 Priority registers are used to assign a programmable 1-of-7 level priority to every possible channel. This block also contains all other control registers that the ARM platform can access.

The 48 DMA requests that are connected to the scheduler come from a variety of sources. The "receive register full" and "transmit register empty" signals found in the UART and USB ports are typical examples of DMA requests that can be connected to the SDMA. These requests can be used to trigger a specific SDMA channel, or several channels.

There is an OnCE compatible debug port for product development. The OnCE includes support for setting breakpoints, single-step and trace, and register dump capability. In addition, all memory locations are accessible from the debug port.

66.1.2 Features

The following are the SDMA features:

- Multi-channel DMA supporting up to 32 time-division multiplexed DMA channels
- Hardware or software driven triggers for each channel
- 48 hardware driven triggers that can be mapped to any channel.
- Memory accesses including linear addressing, FIFO addressing and 2D addressing
- Fast context-switching with two-level, priority-based preemptive multi-tasking
- 16-bit instruction-set micro-RISC engine (the SDMA core)
- Two DMA units with some or all the following features:
  - Auto-flush and prefetch capability
  - Flexible address management (increment, decrement, and no address changes on source and destination address)
  - Misaligned data-transfer support
  - Uni-directional and bi-directional flows (copy mode)
  - Up to eight-word buffers for configurable burst transfers
- Support of byte-swapping and CRC calculations
- An available API and library of scripts
- Little-Endian and Big-Endian modes
- Hardware handshakes for low-power entry sequence
- Security support to lock contents of the SDMA script RAM.
- 4-Kbyte ROM containing startup scripts (for example, boot code) and other common utilities that can be referenced by RAM-located scripts
- 8-Kbyte RAM area is divided into a processor context area and a code space area used to store channel scripts that are downloaded from the system memory
- Debug support, including a OnCE port, real-time monitors, and embedded cross-trigger events
- Supported clock frequencies in process:
  - Configurable clock options for the SDMA core and the ARM platform DMA units
    - 1:2 ratio with maximum of SDMA core running at ARM platform Peripheral Bus speed and DMA running at max DMA frequency.
    - 1:1 ratio when both SDMA core and ARM platform DMA clocks are set to the ARM platform Peripheral Bus speed.
- Peripheral bus interface for configuration register programming by the ARM platform
- The SDMA RISC engine (arithmetic and logic operations, plus CRC acceleration), which is referred to as the "SDMA core."
- An internal peripheral bus connected to the Shared Peripherals Bus Interface (SPBA) that enables access to up to 14 shared peripherals. SDMA supports 32-bit accesses to word peripherals and 16-bit accesses to half-word peripherals.

- The peripheral DMA unit that is hooked-up to the ARM platform Crossbar Switch to service ARM peripherals
- The burst DMA unit is able to perform burst accesses to the external memory
- All the DMA units are 32-bit AHB masters. They are connected to different buses, thus allowing concurrent accesses.

66.2 Functional Description

Figure 66-2 shows the SDMA topology, and is composed of the following components:

- SDMA Core (SDMA Core)
- SDMA Scheduler (Scheduler)
- Functional Units:
  - CRC (CRC Calculation Unit)
  - Burst DMA (Burst DMA Unit)
  - Peripheral DMA (Peripheral DMA Unit)
- ARM platform Control for ARM control register access.
- Internal RAM and ROM Memory (SDMA Programming Model)
- OnCE debug Port (The OnCE Controller)

The functional unit bus provides access by the SDMA core to the CRC hardware accelerator and the DMA units. The system bus provides access to SDMA internal memory and also supports up to 14 peripherals.

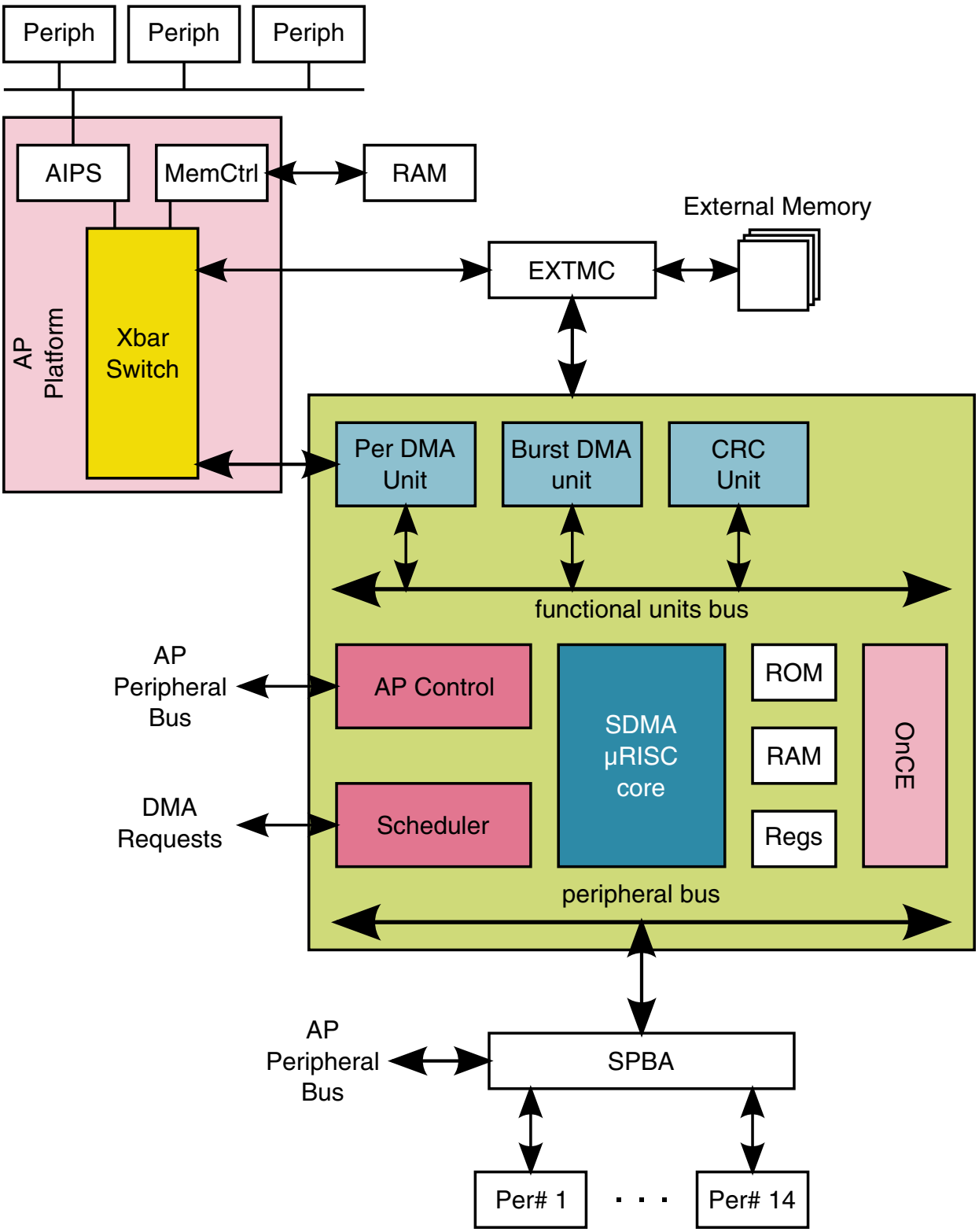


Figure 66-2. SDMA Connections

NOTE

EDITOR NOTE: This copy of the figure is for the case where the BP interfaces are NOT connected. Unconnected interfaces are removed from this diagram since not visible at all to the user.

66.3 SDMA Core

The SDMA core is a customized RISC-like processor that is specifically developed to control DMA units and perform L1 tasks like byte-stuffing or framing.

The SDMA core incorporates on-chip debug capability using the OnCE.

The SDMA core is based on a 32-bit register architecture with 16-bit instructions. There are eight general purpose 32-bit registers, four flags (T, LM, SF, and DF), and four PCU registers (PC, RPC, SPC, and EPC) that can address 16,384 16-bit instructions.

66.3.1 SDMA Core Structure

The following figure shows the structure of the SDMA core. It also shows the different registers, calculation resources, and possible data movements.

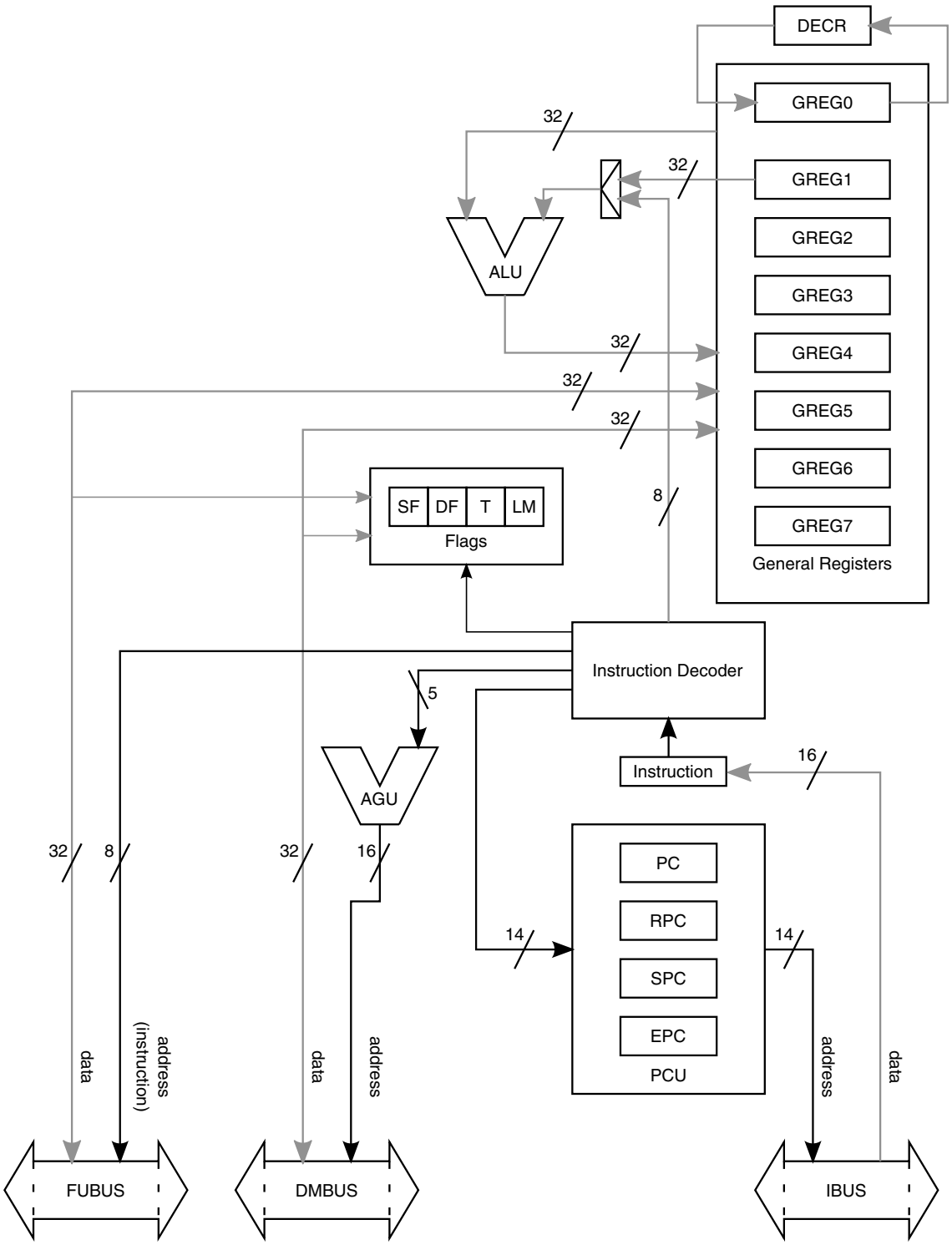


Figure 66-3. SDMA Core

- The Program Control Unit (PCU) is described in Program Control Unit (PCU). It handles the state of the core and generates the instruction fetch addresses. Instructions are retrieved from the Instruction Bus (IBUS) and stored in the SDMA

core instruction register prior to their decoding. The PCU contains the following registers:

- The Program Counter (PC) contains the address of the current instruction.
- The Return Program Counter (RPC) contains the address of the instruction that follows a jump to the subroutine.
- The Start Program Counter (SPC) contains the address of the first instruction of the current hardware loop.
- End Program Counter (EPC) contains the address of the last instruction of the current hardware loop.

- The other core registers are the general purpose registers (GREGn) and the flags.
  - The general purpose registers can be used to hold data and addresses. They can be loaded with immediate values (for example, 8-bit data that are encoded in the instruction), results of calculations that were performed with the ALU, 32-bit data that comes from the memory or peripherals via the Data Memory Bus (DMBUS), 32-bit data that comes from the DMAs or CRC via the Functional Units Bus (FUBUS) or another general purpose register. Their content can be the operands of the ALU, the data to send on either bus (DMBUS or FUBUS), or a pointer to memory (DMBUS address).
  - The general register 0 (GREG0) is also the hardware loop counter. In hardware loops, it cannot be used for any other purpose. This register uses a dedicated decrement unit (DECR) shown in Figure 66-3.
  - The flags reflect the status of operations:
    - SF and DF are set when the last load or store on either bus (FUBUS or DMBUS) received an error response.
    - LM is set when the core is executing instructions inside a hardware loop.
    - T is set when the ALU operation result was 0 or the loop counter reaches 0 (the latter is preponderant when an ALU operation is the last instruction of a hardware loop).
- The ALU has two operands: any general register and either a second general register or an immediate value. The result is always stored into the first general register. A NOP function can be utilized by moving a register's contents into itself (For example, the instruction: mov R0,R0).
- The 16-bit instructions are fetched via the instruction bus (IBUS) whose address is driven by the PC. The SDMA RAM and ROM are visible to the core as 16-bit devices through this interface.
- The memory (RAM and ROM), memory mapped registers, and external peripherals are accessed via the DMBUS. The address is always taken from a general register whose content is added to a 5-bit immediate value. This is the only available addressing mode. The DMBUS is a 32-bit data bus. Except for the peripherals that

are external to the SDMA, the address accuracy is the 32-bit word (for example, adding 1 to an address points to the next word, not the next byte).
- The functional units are accessed via the FUBUS connection. The data is exchanged with any general register, but the address (which in fact is the instruction and the selector of the functional unit) comes from an 8-bit field of the corresponding load or store.

66.3.2 Program Control Unit (PCU)

This part of the SDMA core is dedicated to the control of the RISC engine, as implied by the instructions that are executed. Its behavior is determined by the instruction type and the inputs of the SDMA.

It contains the PC, RPC, SPC, and EPC registers that are described in SDMA Core Structure.

66.3.2.1 Instruction Types

The state sequence and the delay of execution vary according to the type of the instruction. There are six possible categories of instructions, as follows:

1. Standard: Most of the instructions belong to this category, and always last 1 cycle.
2. ldf/stf: These are respectively the load and store instructions that access the functional units. They last 1+n cycles where n is the number of wait-states of the targeted functional unit.
3. ld/st: These are the load and store instructions that access the memory and peripherals. They last 1+n cycles where n is the number of wait-states of the targeted device (1 for the ROM, RAM, and memory mapped registers, 1 + the external peripheral wait-states). These instructions always last at least two cycles, but the core is able to handle them in one cycle. The first wait-state is inserted outside the core.
4. Branch: These are all the instructions that cause the Program Counter to point to another instruction other than the following one (for example, one that breaks the sequential flow). There are the absolute jumps, the conditional branches, the jump to the sub-routines, and the return from the sub-routine.
5. Loop,Modified Load or Store: The hardware loop instruction modifies the potential behavior of any load or store inside the loop (for example, when the LM flag is set). A jump may be implied after any such load or store if it received an error. The error causes an early exit of the loop, which means a jump to the instruction that follows the one that is pointed to by EPC. An additional cycle is required by the PCU to perform the jump (+1 to the ld/st/ldf/stf original execution delay). Although there is

usually an implicit jump after the last instruction of the loop when the PC goes back to SPC, this is performed at no cycle cost.

6. Done: The done, yield, or yieldge instructions are used to control channel switching. When no channel switching is performed, these instructions last a single cycle. When there is a change of channel or context switch, the delay is variable and depends on many factors (as detailed in Context Switching).

66.3.2.2 PCU States

The PCU state is visible through outputs of the SDMA (see Real-Time Debug Outputs) or the OnCE status register(see OnCE Status Register (OSTAT)).

The PCU state is a four-bit field that can take the values shown in the following table. Figure 66-4 shows the possible state transitions and the corresponding conditions.

Table 66-1. PCU States

Value	State	Description
0	Program	The is the usual instruction cycle.
1	Data	This state is inserted when there are wait-states during a load or a store on the data bus (ld/st type).
2	Change of Flow	This is the second cycle of any instruction that breaks the sequence of instructions (branch and done types). This state lasts only a single cycle; it is always followed by the Program state.
3	Error in Loop	This state is used when an error causes a hardware loop exit (loop-modified load or store type). This state only lasts a single cycle; it is always followed by the Program state.
4	Debug	he SDMA is stopped in debug mode.
5	Functional Unit	This state is inserted when there are wait-states during a load or a store on the functional units bus (ldf/stf type).
6	Sleep	No script is running: The core is idle after saving the last channel context.
7	Save	The context switch FSM is saving the current channel.
8	Program in Sleep	Same as Program except there is no associated channel, this state is used when instructions are executed after entering debug mode, whereas the core was in either Sleep mode.
9	Data in Sleep	This is the same as Data except there is no associated channel.
10	Change of Flow in Sleep	This is the same as Change of Flow except there is no associated channel. This state only lasts a single cycle, and is always followed by the Program in Sleep state.
11	Error in Loop in Sleep	This is the same as Error in Loop except there is no associated. channel. This state only lasts a single cycle, and is always followed by the Program in Sleep state.
12	Debug in Sleep	This is the same as Debug except the core was put in debug mode when no channel was active.

Table continues on the next page...

Table 66-1. PCU States (continued)

Value	State	Description
13	Functional Unit in Sleep	This is the same as Functional Unit except there is no associated channel.
14	Sleep after Reset	This shows that no script is running, and the core is idle after a reset. When a channel becomes active, no context is restored but the core starts its boot program located at address 0 (or the address available in register in Channel 0 Boot Address (SDMAARM_CHN0ADDR)).
15	Restore	The context switch FSM is restoring the next channel context.

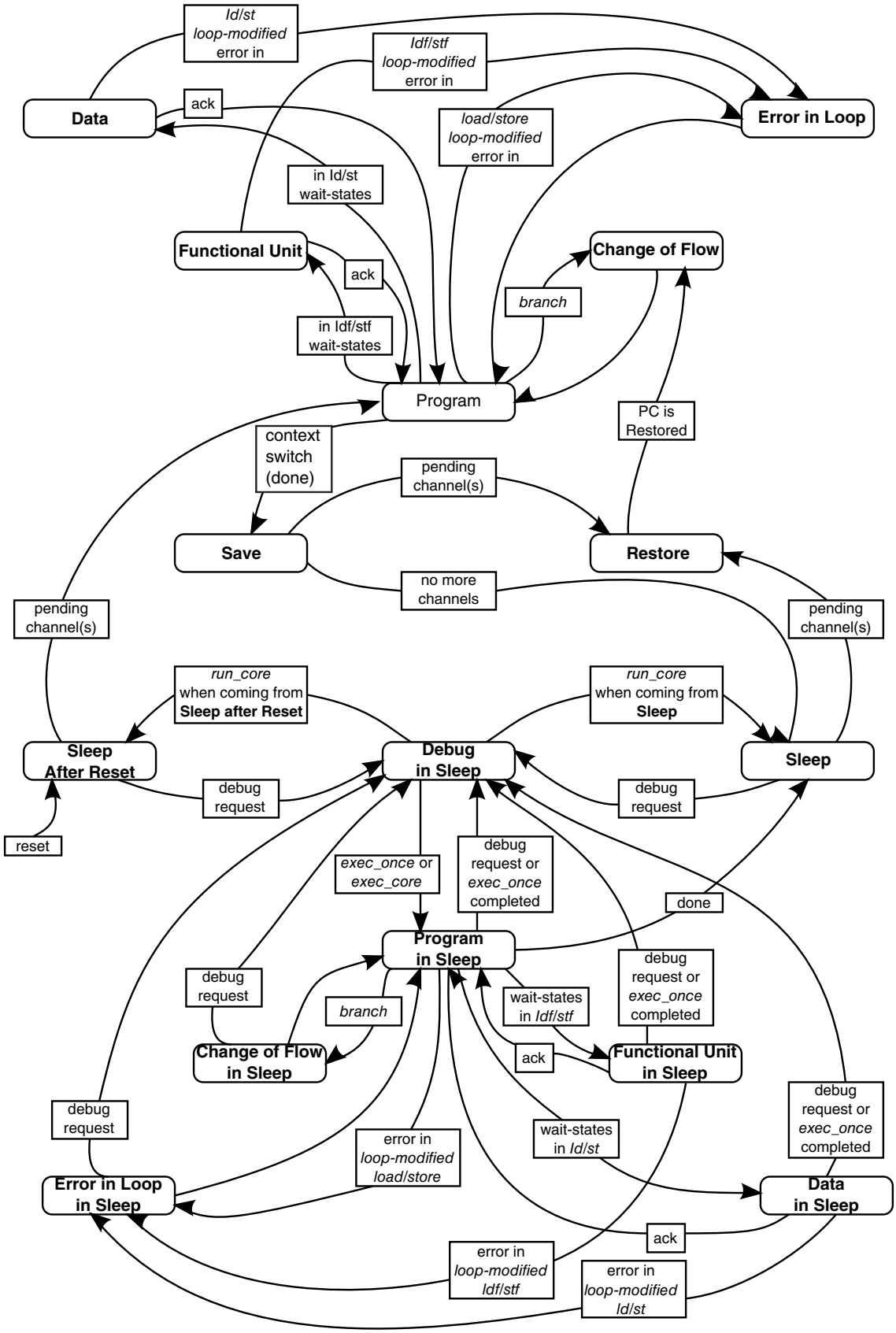


Figure 66-4. PCU State Diagram

66.3.3 SDMA Core Memory

The SDMA has two memory spaces: one for the instructions and one for the data. As both spaces share the same resources (ROM and RAM devices), the system bus manages possible conflicts when the core accesses the same resource for both an instruction read and a data read or write.

Program and data memory is further described in Address Space.

Instructions of 16-bit width are stored in 32-bit wide devices and can be accessed as data. The mapping is Big Endian: an even instruction address (terminated by 0) accesses the most significant part of the 32-bit data (bits [31:16]), and an odd instruction address (terminated by 1) accesses the least significant part of the 32-bit data (bits [15:0]). Instructions can be fetched out of internal ROM or RAM.

Data can be read from ROM, RAM, memory mapped registers, and external peripherals, and written to the same devices (except the ROM).

The ROM contains bootload scripts, channel scripts, and common subroutines which may be referenced by channel scripts elsewhere in the ROM or RAM.

The RAM is divided into a context area and a code space area which may be used to store channel scripts. The RAM contains undefined values after a hardware reset. Channel scripts and initial context values are downloaded into RAM using channel 0 which is reserved for bootload functions.

66.4 Scheduler

All channel scheduling hardware is included in the Scheduler.

66.4.1 Primary Functions

The scheduler is a hardware-based design used to coordinate the timely execution of 32 virtual DMA channels by the SDMA core on the basis of channel status and priority.

The scheduler performs the following functions:

- Monitors, detects, and registers the occurrence of any one of the 48 DMA requests
- Links a specific request to a channel or group of channels (channel mapping)
- Ignores requests that are not mapped to a previously configured channel
- Maintains a list of all the channels that are requesting service

- Assigns a pre-programmed priority level (1 of 7) to every channel requesting service
- Detects and flags overrun/underrun conditions

66.4.2 Channels and DMA Requests

66.4.2.1 Channels

A Virtual Channel (hereafter simply called a channel) manages a flow of data through the SDMA. Flows are typically unidirectional.

The SDMA can have up to 32 simultaneously operating channels, numbered from 0 to 31. Channel 0 is usually dedicated to control the SDMA script downloading. All the channels can be assigned by the ARM platform software.

66.4.2.2 DMA Requests

A DMA request is caused by externally (for example, external to the SDMA) controlled conditions (for example, UART receive FIFO reaches a threshold). The SDMA currently supports up to 48 DMA requests.

66.4.2.3 Mapping from DMA Requests to Channels and Priorities

A channel can stall waiting on a single DMA request. A single DMA request can awake more than one channel (in fact, any request can awake any combination of channels).

The mapping between DMA requests and channels is program-controlled. There is a storage element assigned for each of the 48 requests that contains a bitmap table of the channels that are awakened by the event.

Every channel also has a three-bit register that indicates its priority.

66.4.3 Scheduler Functional Description

Scheduler Overview describes the behavior of the SDMA scheduler-from the channel enabling conditions to the highest priority pending channel selection.

66.4.3.1 Scheduler Overview

The scheduler algorithm is built in hardware. It is provided with possibilities for the ARM platform to control its behavior.

The scheduler processes incoming DMA requests, maps detected requests to 0, one, or several channels, maintains a list of channels that are requesting service (pending channels), identifies the top priority and its associated channel, and selects the next active channel when the current channel yields.

The following figure shows a functional overview.

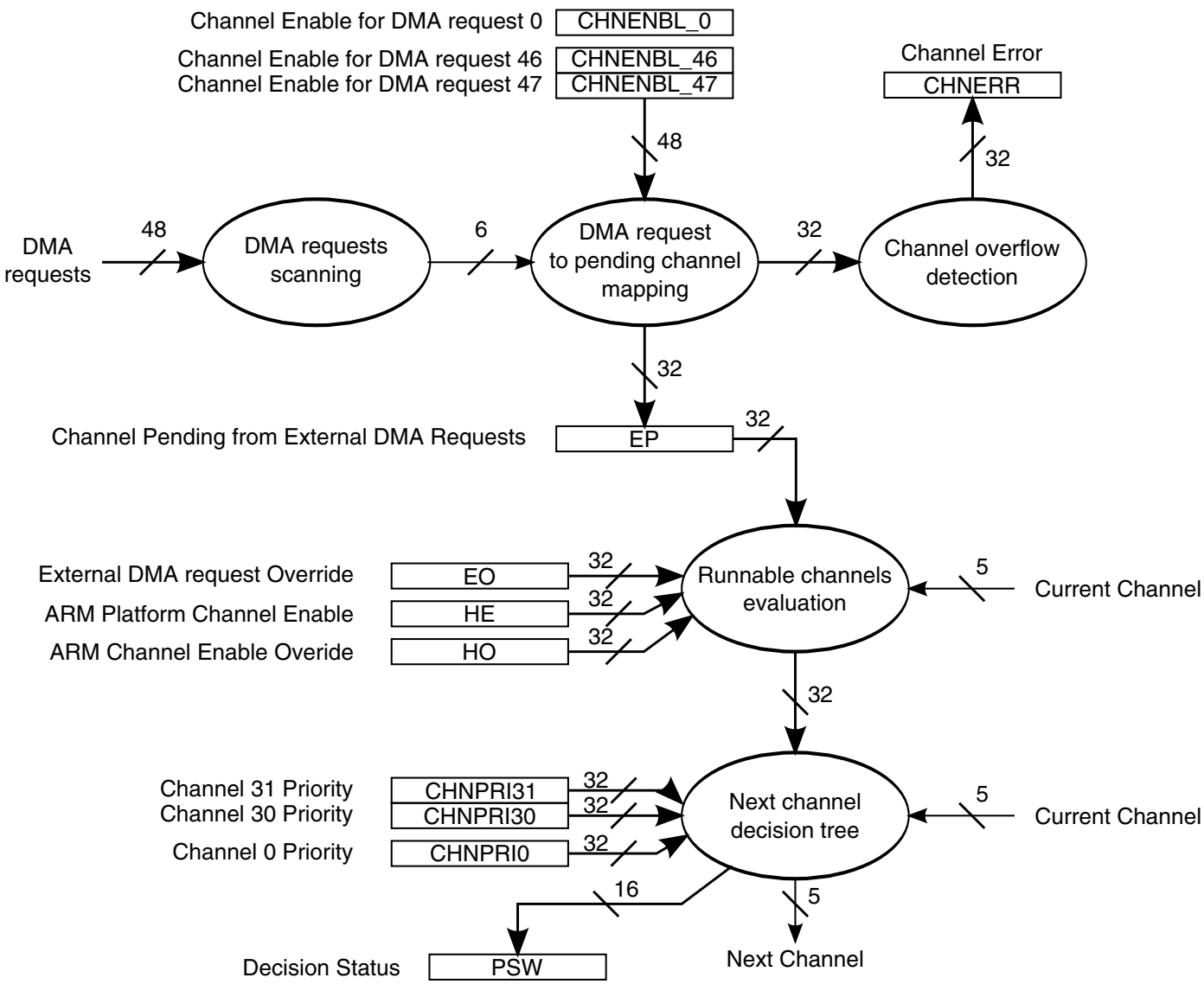


Figure 66-5. SDMA Hardware Scheduler

66.4.3.2 DMA Requests Scanning

The scheduler contains a 48-bit edge detection device that detects the rising edge of every DMA request and transmits the request number to the next stage.

The DMA requests are assumed to be generated on the same reference clock as the SDMA core clock; they are detected as soon as the signal goes from a 1-to-n-cycles low state to a 1-to-m-cycles high state.

This system is able to detect single-cycle pulses as well as level-based DMA requests such as a FIFO threshold crossing. In this case, the SDMA provides a memory mapped register that can be used by the channel script to monitor the DMA requests lines, and thus determines whether the data transfer is done or not done, and then continues with the transfer or closes the channel.

When several DMA requests are detected at the same time, they are forwarded to the next scheduler stage at the rate of one request per cycle. No request is lost.

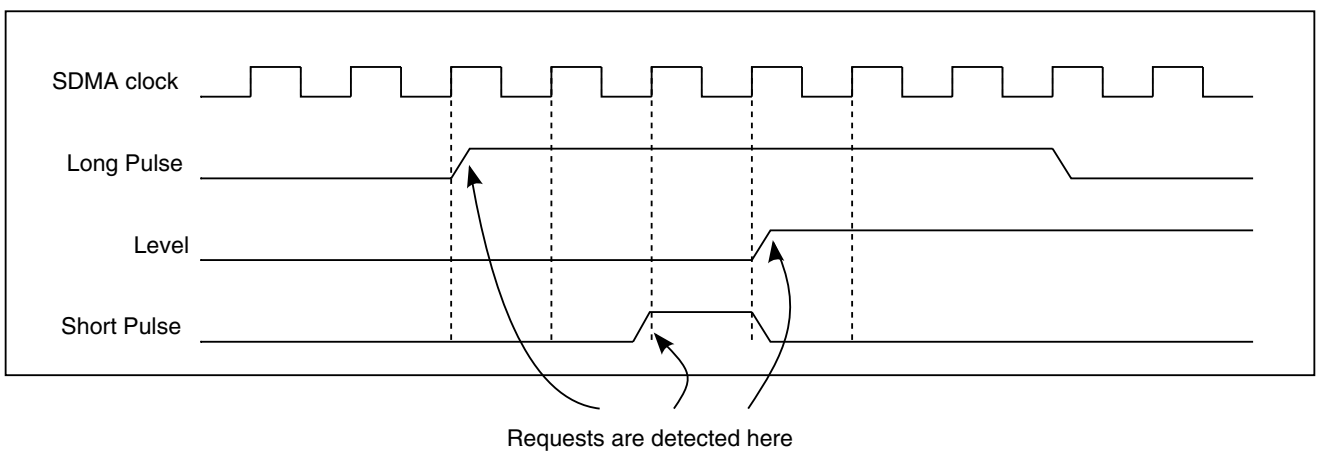


Figure 66-6. Examples of Valid DMA Requests

The DMA request inputs are connected to various sources that depend on the SoC. The exact list of DMA request inputs and their associated number is available in each respective project-specific chapter.

66.4.3.3 Mapping DMA Requests to Pending Channels

Whenever a DMA request is detected by the first stage, its number is used in the second stage to determine the channels that have to be activated.

This is performed with an array of 48 registers that are 32 bits wide: There are 48 Channel Enable Registers (CHNENBLn), one register per DMA request. The DMA request number selects the Channel Enable Registers, and every bit of this 32-bit register indicates that the corresponding channel must be activated when it is a 1.

This information is passed on the EP register. For every bit of the Channel Enable Register that is set, the corresponding bit of the EP register is also set, and the remaining bits of EP are left unchanged. The transformation of EP is summarized by the following equation:

EP = EP or CHNENBLn

The EP register is used to know which channels require service because they received a DMA request.

Typical contents of the CHNENBLn registers are all 0s, except for a single bit set. For example, a DMA request triggers one channel, but all 0s or several 1s are possible. One DMA request could activate several channels, and the channel execution sequence can be controlled by the channel priorities and numbers, as explained in the next sections. The following table illustrates an example configuration.

NOTE

From the table, the DMA request 0 is programmed to simultaneously trigger channels 0, 1, and 31. Also, DMA requests 30-47 are not used in this example. The remaining channels 2 to 30, are configured to be triggered by DMA requests 29 to 1, respectively.

Table 66-2. Channel Enable RAM Programming Example

DMA Request Number	Channel																														
	31																														0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table continues on the next page...

Table 66-2. Channel Enable RAM Programming Example (continued)

DMA Request Number	Channel 31																															0
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Table continues on the next page...

Table 66-2. Channel Enable RAM Programming Example (continued)

DMA Request Number	Channel 31																														0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table continues on the next page...

Table 66-2. Channel Enable RAM Programming Example (continued)

DMA Request Number	Channel																															
	31																															0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

66.4.3.4 Channel Overflow

A channel overflow occurs when a DMA request requires service from channel *n* by setting bit *n* of the register EP, but this bit is already set, meaning channel *n* is already pending. This can come from an overrun/underrun condition.

This detection is possible only when the DMA requests are pulses, because a level-based DMA request stays high until it is serviced, even though an underrun or overrun condition occurs, thus preventing another edge detection of the DMA request.

The channel overflow information is saved in the 32-bit CHNERR register (1 bit per channel). You can configure the SDMA to trigger an interrupt to the ARM platform when there are 1s in CHNERR. Every bit of CHNERR is masked with the corresponding bit of INTRMASK and if it gives a 1, the corresponding bit of INTR is set, triggering the interrupt.

66.4.3.5 Runnable Channels Evaluation

The EP register is used in conjunction with several other 32-bit registers to determine the channels that are runnable.

Registers EO, DO, HO and HE, are controlled by the ARM platform. EP is controlled by the DMA requests and their mapping to channels.

Several channels may be runnable at any given time. The i^{th} channel is runnable if (and only if) the condition below is true:

(HE[i] or HO[i]) and (DO[i]) and (EP[i] or EO[i])

After reset, the HE[i], HO[i], EP[i], and EO[i] bits are all cleared whereas the DO[i] bits are all set. The functions associated with DO are not available for this device. When DO[i] is set, the scheduler condition becomes:

(HE[i] or HO[i]) and (EP[i] or EO[i])

The registers in these equations are controlled as follows:

- ARM platform (host) channel enable flag HE[i] may be set or cleared by the ARM platform with the HSTART and STOP_STAT registers. It can also be cleared by the i^{th} channel script.

  Typical usage is for the ARM platform to set this flag to activate the channel. The flag is cleared by the SDMA core when the transfer is done.

- Externally triggered channel pending flag EP[i] is set by the scheduler when the channel was activated by a DMA request. It can be cleared by the i^{th} channel script.
- The ARM platform channel override flag HO[i] may be set or cleared by the ARM platform. When set, it enables the i^{th} channel to run without the involvement of the ARM platform.

  Typical usage is for the ARM platform to set this flag for channels that do not need ARM platform supervision such as channels that are controlled by DMA request events (EP).

- DO should always be set to 1 so that the runnable channel evaluation considers only HO, HE, EP, and EO.
- Externally triggered channel override flag EO[i] may be set or cleared by the ARM platform. When set, it prevents the i^{th} channel from stopping and stalling on incoming peripheral DMA requests. This is the case when the channel is not handling data transfers with peripherals (for example, a memory to memory transfer).

The SDMA can clear the HE[i], and EP[i] bits by means of adone or notify instruction. The done instruction causes a reschedule; thus, enabling another channel to preempt the current one, while the notify instruction does not. The done and notify instructions can clear either HE[i] or EP[i] (never more than one at a time).

Table 66-3. Runnable Channel Selection Control

Register	Set by	Cleared By
HO	Write to HOSTOVR register	Write to HOSTOVR register
HE	Write to HSTART register	Write to STOP_STAT register or by the channel script with the done or notify instructions.
DO	Write to DSPOVR register	Write to DSPOVR register
EO	Write to EVTOVER register	Write to EVTOVER register
EP	Set by external DMA request event input.	By the channel script with the done or notify instructions

66.4.3.6 Next Channel Decision Tree

The next channel number is computed from the runnable channels list, the current channel number, and their respective priorities.

It is re-evaluated every cycle, but is only used when the current channel yields or terminates by executing a yield, yieldge, or done instruction.

The decision tree is based on the selection of the runnable channel that has the highest priority.

The highest priority channel is selected according to the following rules:

- Runnable channels are sorted by priority.
- If one of the channels with the highest priority had been preempted by a channel with a higher priority, but did not want to yield to a channel of the same priority (for example, it executed a yield, not a yieldge), it is elected as the next channel.
- The channels that belong to the highest priority group are sorted by their number and the channel that has the highest number in this group becomes the next channel. For example, if priorities are the same, channel 31 will be selected before channel 30.

When the current channel requires a reschedule with a yield(ge) or a done instruction, the context switch decision is based on the instruction parameter, the current channel number and priority, and the next channel number and priority. The possible cases are all listed in the following table. The grayed cells correspond to unusual cases that should not occur with a typical usage of the SDMA.

Table 66-4. Channel Switching Decision with a yield, yield(ge), or done

Instruction	Current Channel	Next Channel	Priorities Comparison	New Running Channel/Comments
yield (done 0)	Runnable	Not runnable	none	Current
	Runnable	Runnable	Current > Next	Current
			Current = Next	Current
			Current < Next	Next[1]
	Not runnable	Not runnable	none	none[2] (occurs when the channel was disabled by the ARM platform)
	Not runnable	Runnable	none	Next[1] (occurs when the channel was disabled by the ARM platform)
yieldge (done 1)	Runnable	Not runnable	none	Current
	Runnable	Runnable	Current > Next	Current
			Current = Next	Next[1]
			Current < Next	Next[1]
	Not runnable	Not runnable	none	none[2] (occurs when the channel was disabled by the ARM platform)
	Not runnable	Runnable	none	Next[1] (occurs when the channel was disabled by the ARM platform)
done (done>1)	Not runnable	Not runnable	none	none[2]
	Runnable	Not runnable	none	Current[3] (occurs when the done instruction does not disable the channel runnable condition)
	Not runnable	Runnable	none	Next[1]
	Runnable	Runnable	none	Current[3] (occurs when the done instruction does not disable the channel runnable condition)

1. Current channel script execution is stopped, its context is saved; the next channel context is restored and its script execution resumes
2. Current channel context is saved and SDMA enters IDLE mode
3. Current channel context is saved, then restored, and the current channel script resumes execution

Finally, when the SDMA is in IDLE mode and a runnable channel is elected as the next channel, its context is immediately restored and the script execution resumes.

The *combinatorial-decision* tree supports dynamic modifications of the EP, EO, HE, HO, and DO flags as well as dynamic modifications of the channel priorities. The propagation times are detailed in Scheduler Pipeline Timing Diagram.

The decision tree status is available in the PSW register, which is continuously updated. It contains the next channel priority, the next channel number, the current channel priority, and the current channel number. When a priority is read as 0, it means the channel is not runnable.

A few examples of decisions are presented below:

- Channel 31 is running with priority 5, channels 13 and 24 are pending with the same priority 5; channel 24 is eligible as the next channel since 24 > 13.
- Channel 31 is running with priority 7, channels 13 and 24 are pending with priority 5; channel 31 is the next channel because its priority is greater than the other pending channels.
- Channels 7, 23, and 29 are pending with the same priority. Channel 7 is active and runs a yieldge; it is preempted by channel 29. After a period of time, channel 29 runs a yieldge, it is then preempted by channel 23 that is the selected channel since channel 29 is the current channel. Later, channel 23 runs a yieldge and is preempted by channel 29. Channels 23 and 29 will go on switching after every yieldge until one of them terminates. It is only at that point that channel 7 becomes eligible again.
- Channel 11 is running with priority 3, and channel 15 is pending with priority 4. When the channel 31 script executes a yield instruction, it gets preempted by channel 15; then channels 6 and 18 with priority 3 become pending. Because channel 11 was preempted after executing a yield and there is no pending channel with a strictly greater priority, it is eligible as the next channel (although its number 11 < 18).

66.4.3.7 Scheduler State Diagram

The Figure 66-7 summarizes the behavior of the SDMA scheduler with details about the exact mechanism of the priority decision tree. It is important to understand the scheduler is a hardwired pipeline, which means all the stages are performed simultaneously every cycle, but a change on any given stage is reflected on the next stage after the delays presented in Scheduler Pipeline Timing Diagram.

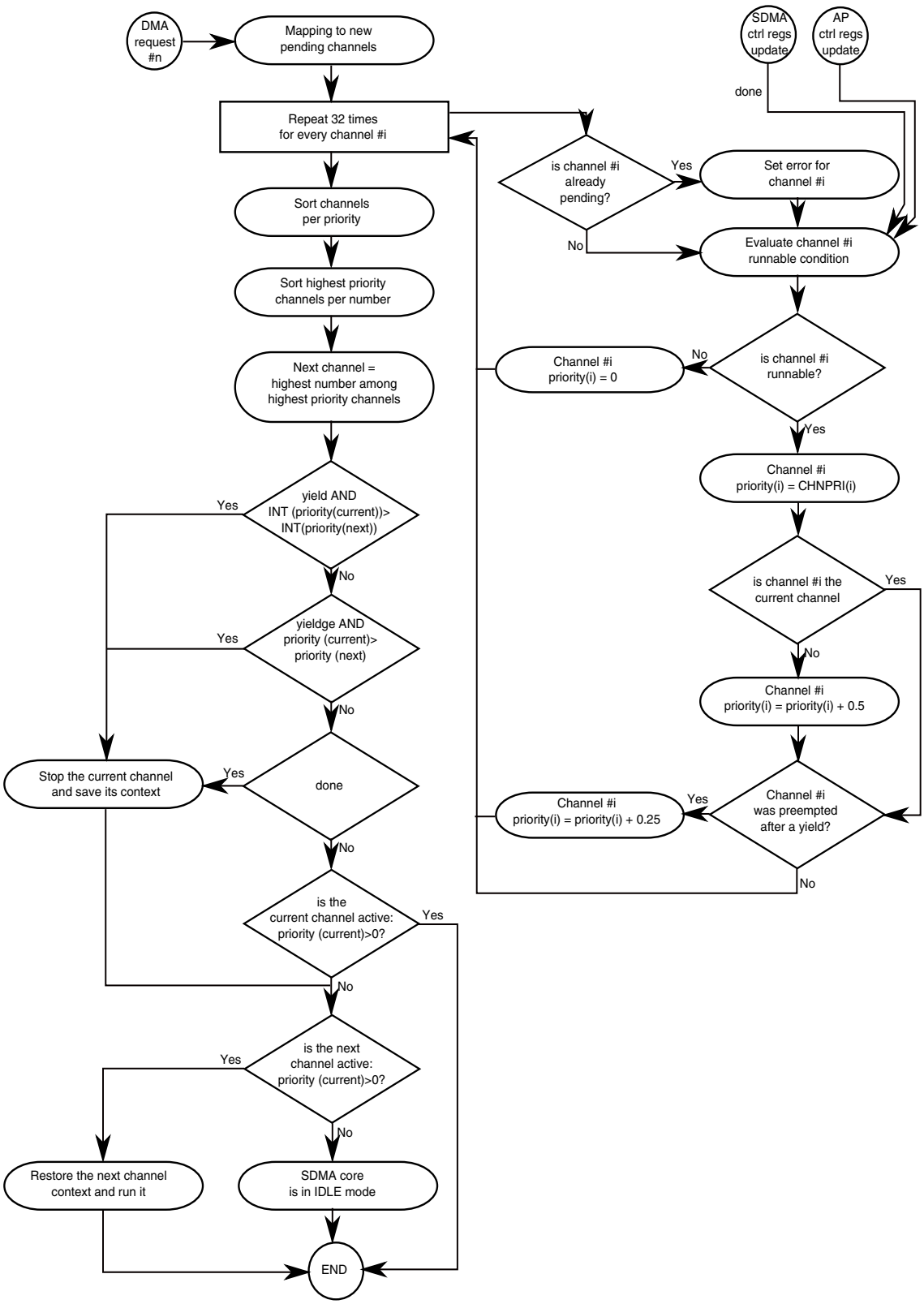


Figure 66-7. Scheduler State Diagram

66.4.3.8 Scheduler Pipeline Timing Diagram

The SDMA scheduler process of DMA-request and control-register modifications is not immediate.

The figure below shows the exact delays of all the tasks. The reference clock is the SDMA core clock.

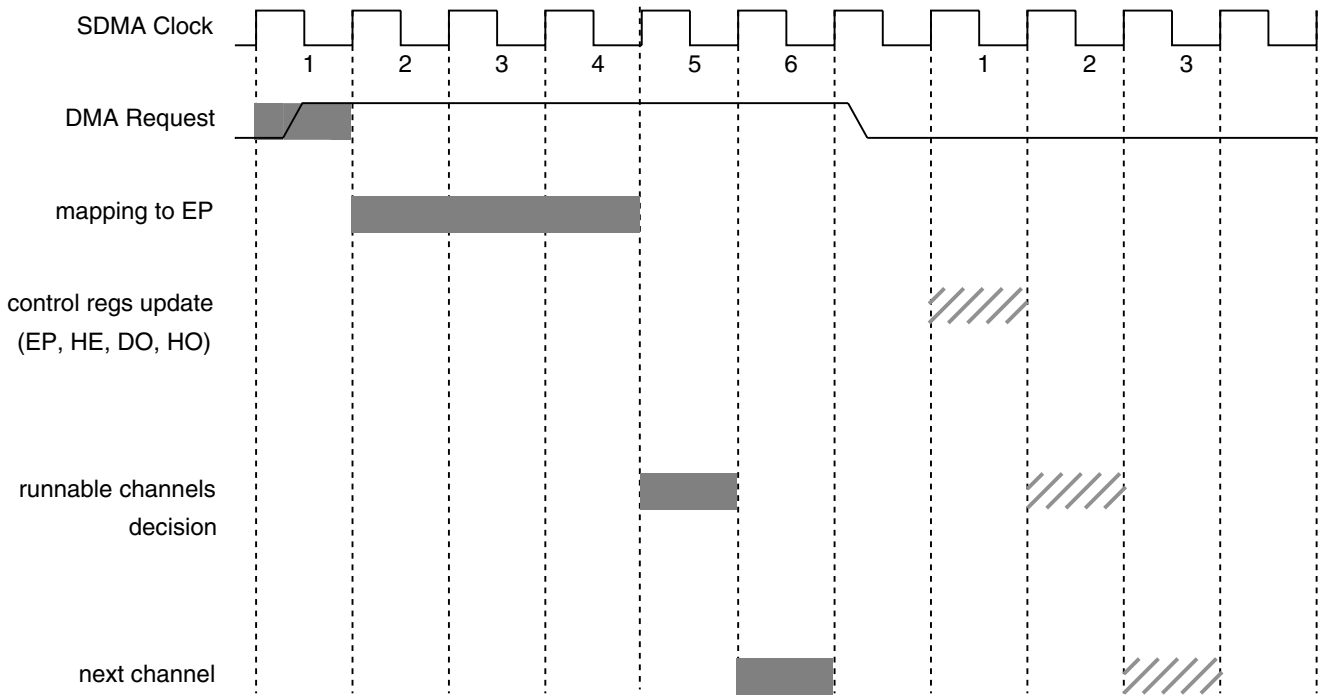


Figure 66-8. Scheduler Timing Diagram

Two numbers can be inferred from this timing diagram. First, it takes six SDMA core clock cycles to update the next channel from a DMA request. Second, it takes three SDMA core clock cycles to update the next channel from a direct modification of the condition registers (EP, DO, HE, or HO) by any processor. The processors that can modify these bits include SDMA with a done instruction or the ARM platform with a write access through the corresponding control port on their respective peripheral bus).

66.4.3.9 Channel-DMA Request Mapping

The 48 DMA request inputs to the SDMA scheduler are listed in project-specific chapters. Refer to the respective chapters for this information.

66.4.3.10 Examples: How to Start a Channel

A channel can be started when the following equation is true for channel *i*:

(HE[i] or HO[i]) and (DO[i]) and (EP[i] or EO[i])

Once this equation is true, the scheduler can start this channel according to the priority of all pending channels. Several examples of configuration are listed below:

1. To start a channel triggered by ARM platform software:
   - Initially, configure HO[i]=0, DO[i]=1, and EO[i]=1 using registers indicated in Table 66-3.
   - ARM platform software triggers the channel by writing to the HSTART register to set HE[i]=1, thereby setting the above equation true.
2. To start a channel triggered by DMA request event.
   - Initially, configure HO[i]=1, DO[i]=1, and EO[i]=0 using registers indicated in Table 66-3.
   - The DMA request is asserted to trigger the channel by setting EP[i]=1, which makes the above equation true.

66.4.4 Context Switching

On execution of a done or yield(ge) instruction, the current channel may be changed either because it has finished (which necessarily happens when the done instruction is executed), or it was preempted by a higher priority channel (which is possible but not systematic when the yield(ge) is executed).

Upon a channel change the SDMA goes through a context switch procedure. When the current channel yields or ends, the context for that channel is saved into the context RAM locations for that channel. When the next channel starts running, its context is first restored from RAM.

Since context RAM is not yet initialized by reset, there will be no context restore at the beginning of the first channel (bootload channel) run after reset. It is expected that the bootload channel will be used to initialize the context for all other channels. When the bootload channel finishes running or yields, SDMA will enter its SAVE state and save that channel's context into RAM. Then, if the bootload channel is called again later, the context will be restored from RAM when the channel starts again.

The context structure for each channel is defined in Context Switching-Programming andTable 66-10. There will be one context area reserved for each channel. When a channel ends or yields, the SDMA core registers are automatically saved into the context RAM and later restored from the context RAM when the channel is next run. The total

RAM space reserved for 32-channel contexts is either 3K or 4K depending on whether the SMSZ bit is set in the CHN0ADDR register, which enables an additional 8 words of scratch RAM for each context.

66.4.4.1 Context Switch Modes

The exact procedure to save the context of the old channel, and to restore the context of the new channel depends on the context switch mode selected by the ARM platform in the CONFIG control register.

The following are the context switch modes:

- By default, the "dynamic" context switch is set. This mode provides the most efficient context switch for an average of eight cycles to stop the current channel, save its context, restore the next channel context, and resume its execution. It consists of saving modified registers of the current channel in the background (for example, during the channel execution)-which leaves very few registers to save when the switch is decided-resuming execution of the next channel as soon as possible (for example, when the minimal set of registers is restored), and continuing the restore phase during this execution.
- In "dynamic with no loop" mode, the same principle is followed except the modified registers are only saved in the background when the loop flag is not set. This mode offers almost the same effectiveness as the previous one, but it prevents the system from accessing the RAM during loops to save power. This is the recommended mode for an efficient context-switch when the loop bodies are short.
- In "dynamic power" mode, no background saving is performed, which reduces power consumption to the minimum. The modified registers are only saved when the context switch starts. The restore phase is the same as before. This is the mode that achieves the optimal power consumption at the cost of a slower context-switch.
- In a "static" context switch, all the registers are saved when a context switch is decided, and all the registers are restored before starting the execution of the new channel. This mode enables a predictable behavior of the context switch since all the registers are restored prior to the channel start and all registers are saved after the channel termination.

NOTE

Static context mode should be used for the first channel called after reset to ensure that the all context RAM for that channel is initialized during the context SAVE phase when the channel is done or yields. Subsequent calls to the same channel or different channels may use any of the dynamic context modes. This will ensure that all context locations for the bootload

channel are initialized, and prevent undefined values in context RAM from being loaded during the context restore if the channel is re-started later.

66.4.4.2 Context Switch Procedure

The Program Control Unit goes into the *save* state, the current context is spilled into memory, and the next channel context is restored according to the context-switch mode that was selected by the ARM platform.

The context switch procedure is as follows:

1. Load the current context's spill base address.
2. Spill the modified registers of the current channel to memory according to the selected context switch mode while the channel is running.

   On a done or yield(ge) that causes the channel preemption, the PCU goes into the *save* state. In *static* mode, all the registers are saved; whereas, in either *dynamic* mode, the registers that were modified but not yet saved are then saved, and the PCU registers and flags are finally saved.

3. Put the SDMA core into *sleep* and wait for new channels to be serviced. This step is skipped if there are pending channels when the current channel is saved.

   As soon as there is at least one pending channel, the PCU goes into its *restore* state to restore the context of the channel that was elected by the scheduler.

   Once a channel is elected, it remains the current channel until its script requests a rescheduling operation with a done or yield(ge) instruction. That means the current channel cannot be modified by the ARM platform, even if it is no more runnable or if its priority is modified.

   The ARM platform can however force a reschedule by writing the corresponding bit in the CONFIG register, which has the same effect as if the script had executed a done instruction. That feature should only be used to stop the SDMA in emergency cases.

4. Load the context base-address of the new channel.

   In "static" mode, all the registers are restored. In either "dynamic" modes, only the PCU registers are restored.

   The new channel is running. In "static" mode, no more activity regarding context restoring or saving is performed. In either "dynamic" modes, the registers are restored in the background every time an access to the context RAM is possible, and

priority is given to restoring the registers that are required by the next instruction to be executed. When a register has not been restored and the next instruction needs it, this instruction gets stalled until the register was restored.

In "dynamic" and "dynamic with no loop" modes, background saving of dirty registers is performed every time an access to the context RAM is possible and allowed by the context switch mode.

NOTE

The contents of a channel context space in the context RAM depends on the selected context switch mode. In "dynamic" and "dynamic with no loop" modes, the contents of the context RAM tend to match the contents of the SDMA registers (except for the PCU registers and flags that are never saved in the background). In "dynamic power" and "static" modes, the contents of the context RAM remain unchanged until the channel terminates with a done or gets preempted.

66.4.4.3 Context Map in Memory

Refer to Context Switching-Programming.

66.5 Functional Units

The functional units are small systems that are used by the SDMA core to perform complex calculations like the CRC unit, or to handle data transfers between the core and a bus domain external to the SDMA.

The SDMA core is able to control and exchange data with these systems by sending instructions and reading or writing data from/to the functional units' registers via the FUBUS. This is done with the ldf and stf instructions.

The following sections provide introductions to the available functional units. Functional Units Programming Model provides descriptions the functional units' behaviors.

66.5.1 CRC Calculation Unit

The Cyclic Redundancy Check (CRC) unit can perform CRC calculation. A single byte of data can be processed every cycle, but up to four bytes can be simultaneously loaded

The CRC unit supports the following set of polynomials:

CRC32: $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$

CRC16: $X^{16}+X^{15}+X^2+1$

CCITT16: $X^{16}+X^{12}+X^5+1$

IS136: $X^{12}+X^{10}+X^8+X^5+X^4+X^3+1$

CRC10: $X^{10}+X^9+X^5+X^4+X+1$

CRC8: X^8+X^2+X+1

parity: X^8+1

66.5.1.1 CRC Structure

The following figure describes the overall structure of the CRC unit and introduces its registers that are accessible by the SDMA core via the FUBUS.

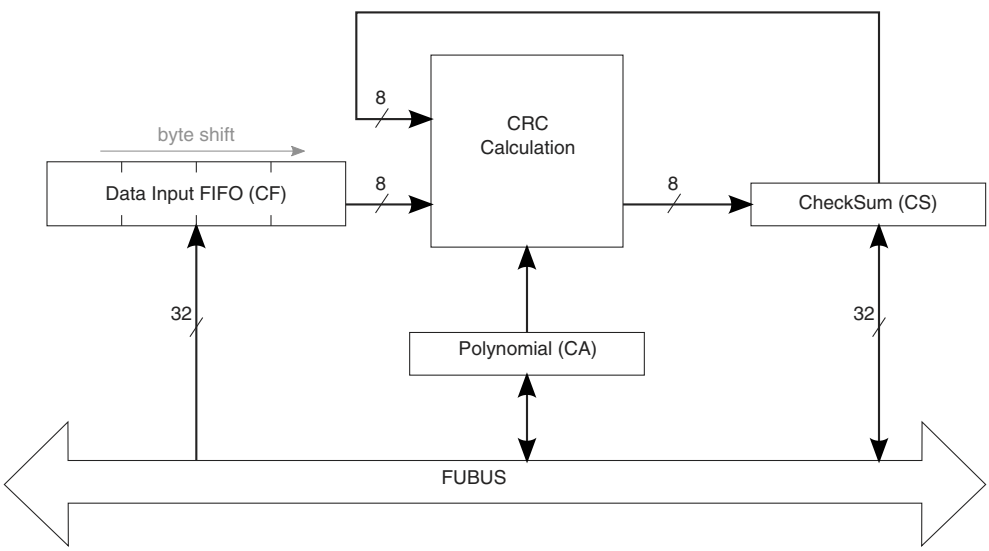


Figure 66-9. CRC Structure

66.5.1.2 CRC Data Processing

CRC processing requires the following three stages:

1. The preliminary initialization stage consists of selecting the desired polynomial, and storing the initialization pattern into the checksum register.
2. Data processing itself, which comes to feeding incoming bytes to the CRC input FIFO - bytes can be written one at a time (8-bit write access from the SDMA core), by pairs (16-bit write), or groups of four (32-bit write). One byte is processed every

cycle: Any subsequent access may stall the SDMA core until completion of the previous calculation.

3. The CRC result (or checksum) can be retrieved at any time, but all data must be processed before it is made available.

66.5.1.3 CRC Registers

The CRC enables reading and writing to three register addresses, which trigger the operations described in CRC Data Processing. The following are the three registers:

- *CA (CRC algorithm)*-The CA selects the desired polynomial. Reading and writing this register correspond to retrieving and changing the polynomial.
- *CS (CRC checksum)*-Writing to this register initializes the checksum accumulator, and reading this register yields the result from the CRC calculation (the result width depends on the polynomial size).
- *CF (CRC FIFO)*-Writing to this register loads the data into the input FIFO and triggers the CRC calculation process. It is not possible to read this register.

66.5.1.4 CRC Summary

Every operation mentioned in CRC Registers takes time to be executed by the CRC unit.

When the CRC receives a command, it immediately acknowledges the SDMA core provided it is not busy completing a previous operation. Therefore, wait-states are inserted by the CRC when a command succeeds another command that is not yet competed.

The following table lists the number of cycles that the CRC takes to execute every possible command. When an operation lasts one cycle, it means the CRC is able to process another command in the next clock cycle (for example, no wait-state is inserted by the CRC because of the former operation that is processing).

Table 66-5. CRC Processing Summary

Operation	Command	Delay	Comments
Set the polynomial	Write CA	1	
Retrieve the polynomial	Read CA	1	
Initialize the checksum	Write CS	1	
Retrieve the checksum	Read CS	1	
Store 1 byte to process	Write CF	1	
Store 2 bytes to process	Write CF	2	CRC is busy for 1 cycle

Table continues on the next page...

Table 66-5. CRC Processing Summary (continued)

Operation	Command	Delay	Comments
Store 4 bytes to process	Write CF	4	CRC is busy for 3 cycles

66.5.2 Burst DMA Unit

The burst DMA unit enables the SDMA core to perform data transfers to and from the ARM platform memory.

It is optimized for accessing SDRAM-like devices. It does not provide control to assign a privilege level to the DMA access. The burst DMA unit provides the SDMA with means to do the following:

- Perform up to 8-beat read and write bursts to the ARM platform memory, which optimizes throughput when accessing SDRAM-type devices because of an internal, 36-byte FIFO
- Access the ARM platform memory at once or twice the SDMA core frequency
- Copy data from one ARM platform memory location to another ARM platform memory location at the ARM platform bus speed, which provides a very high throughput
- Control the method for addressing the ARM platform memory (automatic increment of addresses or frozen addresses-the former aimed at accessing RAM-like memory and the latter aimed at accessing single-address FIFOs)
- Enable or disable automatic prefetch when reading data from the ARM platform memory. When the prefetch mode is selected, the burst DMA automatically triggers external bursts to fill its FIFO without waiting for the SDMA core to request the corresponding data, greatly improving throughput.
- Rely on the DMA to automatically flush its FIFO content when there is enough data to generate an 8-beat burst to the ARM platform memory. Or, it forces a flush when a data transfer must terminate.
- In the former case, the SDMA core may only be stalled when it tries writing data and there is not enough room left in the FIFO. In the latter case, the core is stalled until the data is effectively written to the ARM platform memory.

In automatic flush mode, the core receives an acknowledge that does not reflect the actual error status when the data is effectively written into the ARM platform memory. This error status is retrieved by a later access to the burst DMA. Terminating a write data transfer with a forced flush command guarantees that any bus error to the ARM platform memory is caught.

- Handle address alignment issues between the ARM platform memory map and the SDMA core data. This enables the core to read or write 32-bit data from the burst DMA, whereas the corresponding ARM platform address is not 32-bit aligned. This drastically improves the SDMA scripts' efficiency since the same loop that transfers 32 bits at a time can be used regardless of the start and end addresses in the ARM platform memory space.

This unit structure and registers are described in Burst DMA Structure and Burst DMA Registers.

66.5.2.1 Burst DMA Structure

The burst DMA is essentially made up of a 36-byte FIFO, address registers, and a controlling state-machine. The 36-byte FIFO enables eight-word buffering with address alignment, and the state-machine manages clock adaptation when required.

The burst DMA is depicted in the figure below.

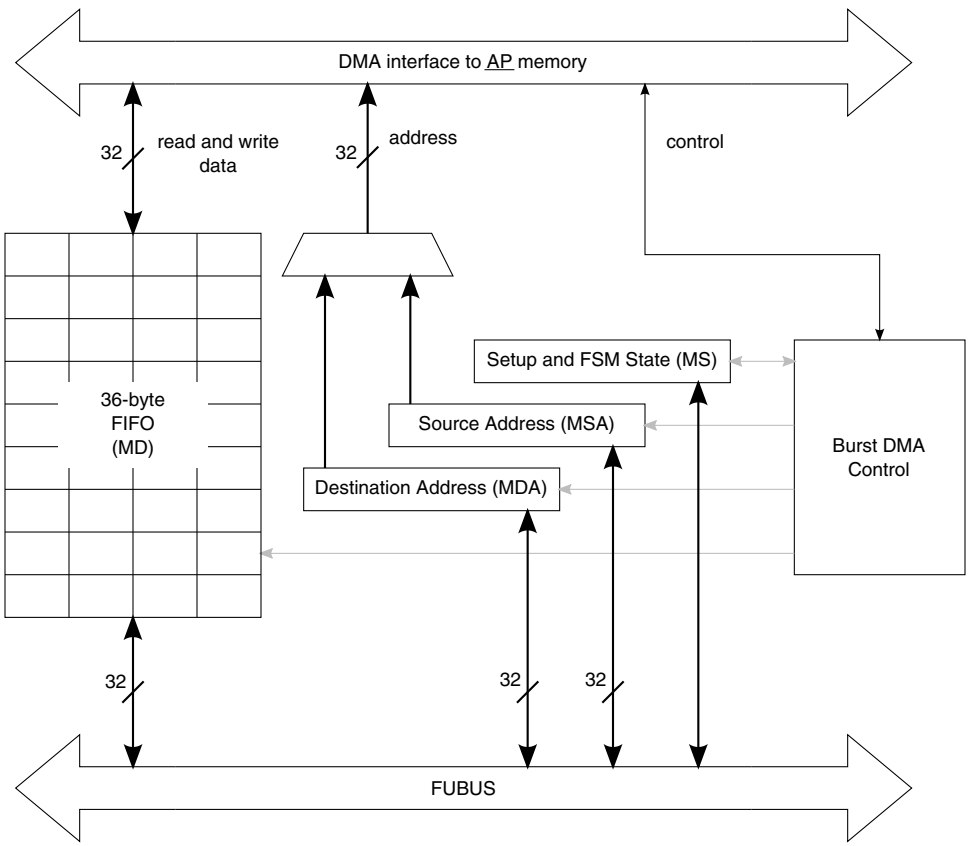


Figure 66-10. Burst DMA Structure

66.5.2.2 Burst DMA Registers

There are four registers, as follows, that may be accessed from the SDMA core:

- MSA (Memory Source Address) - Holds the source byte address in the ARM platform memory map for reading data from this location. This register is automatically modified every time the core reads new data from the FIFO.
- MDA (Memory Destination Address) - Holds the destination byte address in the ARM platform memory map for writing data to this location. This register is automatically modified every time the core writes new data into the FIFO.

- MD (Memory Data) - Labels the 36-byte FIFO access point: Reading a byte, halfword, or word from MD respectively retrieves the first 1, 2, or 4 bytes of the FIFO (for example, the bytes that were stored first by the DMA state-machine when transferring data from the ARM platform memory).
- When the FIFO does not hold as many bytes as required by the SDMA core, the core is stalled until the missing bytes are read from the ARM platform memory. In the case of prefetch mode, the DMA controller decides when it should start a burst to ARM platform memory in order to reduce the risk to not have the required data for the future accesses of the core. When there is no prefetching, a burst is triggered when the required data is not available in the FIFO.

  Writing a byte, halfword, or word to MD stores 1, 2, or 4 bytes, respectively, at the end of the FIFO (for example, these bytes are transmitted to the ARM platform memory after all the other bytes that were previously stored in the FIFO). When the FIFO does not have enough room left to hold the written data, the SDMA core is stalled until a sufficient amount of FIFO contents are flushed out to the ARM platform memory. Flushing is decided by the DMA controller when there are enough bytes in the FIFO to perform the largest allowed burst to ARM platform memory (the exact size depends on the burst start address and the AHB 1 Kbyte boundary rule). However, the SDMA core has the ability to force the flushing operation at any time, for example, when at the end of the data transfer, prior to channel closure.

- MS (Memory Setup) - Contains the state of the burst DMA control, the two flags that define whether each address register is incremented after every access to the external memory, and another flag that is set when a bus error occurred.

66.5.2.3 Burst DMA Data Transfers

Three typical usages have been identified that involve the burst DMA: the data transfer startpoint, the endpoint, or both.

Every case requires a different procedure, as listed in the following sections:

66.5.2.3.1 Data Retrieval from the ARM platform Memory

The following steps retrieve data from ARM platform memory using the burst DMA unit:

- Set up the MS flags to reflect the mode for the source address (incremented or frozen according to the type of accessed device: memory or peripheral FIFO), then initialize the source address register itself (MSA).
- Read data from the FIFO using the *ldf MD* instruction as many times as needed. If an error occurred during the fetch from ARM platform memory, the DMA control tags

the error status on the data and the SDMA core SF flag is set when reading this data from the FIFO.

66.5.2.3.2 Storing Data Into the ARM platform Memory

The following steps store data from ARM platform memory using the burst DMA unit:

- Set up the MS flags to reflect the mode for the destination address (incremented or frozen according to the type of accessed device: memory or peripheral FIFO), then initialize the destination address register itself (MDA).
- Store data into the FIFO using the *stf MD* instruction as many times as needed.
- When the transfer is finished and if the DMA worked in automatic flush mode, force the flush of the FIFO. This instruction is stalled until all the FIFO data is effectively sent to the ARM platform memory and the error status of the transfer is available in the DF flag.

66.5.2.3.3 Transferring Data Between Two ARM platform Memory Locations-Burst DMA Unit

The following steps copy data between two ARM platform memory locations using the burst DMA unit:

- Set up the MS flags to reflect the modes for the source and destination addresses (all the combinations are possible), then initialize the source address register (MSA) and the destination address register (MDA). Both addresses must be word-aligned.
- Use as many *stf MD* instructions with the *COPY* flag as needed. Every instruction triggers a burst read of a given number of words from the source address (this number is provided to the burst DMA via the SDMA core general purpose register, which is referenced in the *stf* instruction). Once all the data is loaded into the FIFO, the DMA empties it with a write burst of the same count to the destination address. The DMA acknowledges prior to instruction completion, which frees the SDMA core for other tasks at no delay cost.
- Once the transfer is done, there should be a final access to the burst DMA to check the error status.

66.5.3 Peripheral DMA Unit

The peripheral DMA unit is the second functional unit that connects the SDMA to the ARM platform memory.

Unlike the burst DMA, it does not support burst transfers and is optimized for accessing peripherals. It does not provide control to assign a privilege level to the DMA access. Its feature list comprises the following:

- Access to the ARM platform peripherals or memory at once or twice the SDMA core frequency
- Data copy from one ARM platform memory location to another ARM platform memory location at memory bus speed, improving throughput
- Control of the method for addressing the ARM platform memory (automatic increment or decrement of addresses or frozen addresses, the first ones aimed at accessing RAM-like memory and the last one aimed at accessing single-address FIFOs)
- Selectable automatic prefetch when reading data from the ARM platform memory. In prefetch mode, the peripheral DMA automatically fetches another data-without waiting for the SDMA core to request it-when its data register is empty, which improves the throughput
- Selectable automatic flush. In this mode, the SDMA core may only be stalled when it tries writing data and the previous write operation is not finished yet; whereas, in forced flush mode, the core is stalled until the data is effectively written to the ARM platform memory.
- In automatic flush mode, the core receives an acknowledge that does not reflect the actual error status when the data is effectively written into the ARM platform memory or the peripheral. This error status is retrieved by a later access to the peripheral DMA. Terminating a write data transfer with a forced flush command guarantees that any bus error to the ARM platform memory has been caught.

  This unit structure and registers are described in Peripheral DMA Structure and Peripheral DMA Registers.

66.5.3.1 Peripheral DMA Structure

The peripheral DMA is made up of a 32-bit data register, two address registers, and a controlling state-machine. The state-machine manages clock adaptation, when required.

It is shown in the following figure.

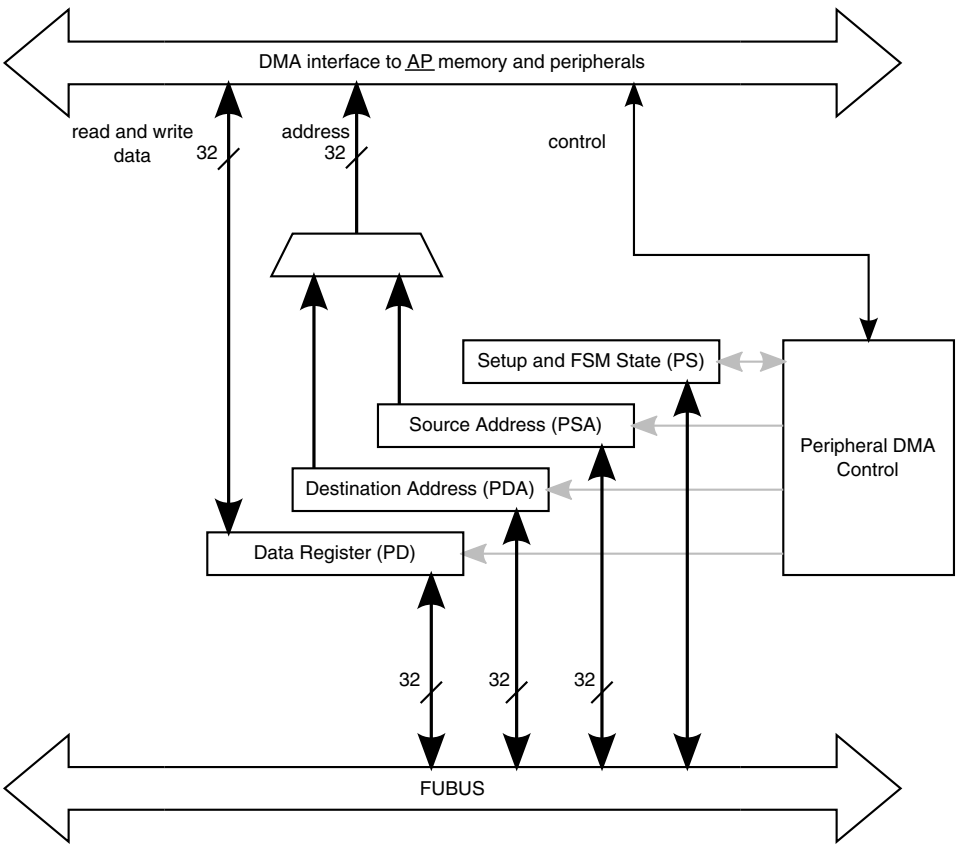


Figure 66-11. Peripheral DMA structure

66.5.3.2 Peripheral DMA Registers

According to Figure 66-11, the peripheral DMA has four registers that may be read or written by the SDMA core:

- *PD (Peripheral Data)* is the DMA 32-bit data register.
- *PSA (Peripheral Source Address)* holds the source byte address in the ARM platform memory map for reading data from this location. This register is automatically modified every time the core reads a new data from PD.

- *PDA (Peripheral Destination Address)* holds the destination byte address in the ARM platform memory map for writing data to this location. This register is automatically modified every time the core writes a new data into PD.
- *PS (Peripheral Setup)* contains the state of the peripheral DMA control, two configuration fields that define the way address registers are modified after every data access, two additional configuration fields that define the data size to access the source and destination devices, and another field that contains the latest transfer error status.

66.5.3.3 Peripheral DMA Data Transfers

There are three typical usages that involve the peripheral DMA, whether it is the data transfer start-point, endpoint, or both.

Every case requires a different procedure, as described in Data Retrieval from the ARM platform Memory or Peripheral, Storing Data into the ARM platform Memory or Peripheral, and Transferring Data Between Two ARM platform Memory Locations-Peripheral DMA Unit.

66.5.3.3.1 Data Retrieval from the ARM platform Memory or Peripheral

The following steps retrieve data from ARM platform memory using the peripheral DMA unit:

- Set up the PS fields to reflect the mode and data size for the source (incremented, decremented, or frozen address register; 8-bit, 16-bit, or 32-bit data transfers), then initialize the source address register itself (PSA) with an address that is aligned to the programmed data size.
- Read data from PD using the ldf PD instruction as many times as needed. If an error occurs during the fetch from the ARM platform memory or peripheral, the DMA control tags the error status on the data and the SDMA core SF flag is set when reading this data from PD.

66.5.3.3.2 Storing Data into the ARM platform Memory or Peripheral

The following steps store data to ARM platform memory using the peripheral DMA unit:

- Set up the PS fields to reflect the mode and data size for the destination (incremented, decremented, or frozen address register; 8-bit, 16-bit, or 32-bit data transfers), then initialize the destination address register itself (PDA) with an address that is aligned to the programmed data size.

- Store data into PD using the *stf PD* instruction as many times as needed.
- When the transfer is finished and if the peripheral DMA worked in automatic flush mode, force the flush of PD. This instruction is stalled until PD contents are effectively sent to the ARM platform memory or peripheral, and the error status of the transfer is available in the DF flag.

66.5.3.3.3 Transferring Data Between Two ARM platform Memory Locations-Peripheral DMA Unit

The following steps copy data between two ARM platform memory locations using the peripheral DMA unit:

- Set up the PS fields to reflect the modes and data size for the source and destination addresses (all the combinations of addressing modes are possible, but both data sizes must be identical), then initialize the source address register (PSA) and the destination address register (PDA). Both addresses must be aligned with the programmed data size.
- Use as many *stf PD* instructions with the *COPY* flag as needed. Every instruction triggers a single read from the source address; a single write of the received data immediately follows. The DMA acknowledges prior to instruction completion, which frees the SDMA core for other tasks at no delay cost.
- Once the transfer is done, there should be a final access to the peripheral DMA to check the error status.

66.6 SDMA Security Support

The SDMA provides support to SDMA software to block unauthorized updates to the scripts in RAM.

SDMA supports the following Security modes:

- Open Mode: has full control to load scripts and context into SDMA RAM. This is the default mode.
- Locked Mode: The ARM platform loads scripts and channel contexts at startup when it is still executing known safe software. When finished, it locks the SDMA to prevent further updates to RAM and selected registers. More details described in Locked Mode.

66.6.1 Locked Mode

The LOCK bit in the SDMA_LOCK register provides support for SDMA scripts to freeze RAM contents after the initial bootload routine to prevent future unauthorized updates to SDMA RAM.

After initial RAM contents are uploaded, ARM platform software can set the LOCK bit to secure the RAM contents to prevent future updates by an unauthorized. After the LOCK bit is written with a '1', the SDMA is "locked" until reset.

The LOCK bit can be read in the SDMA's internal memory map in the LOCK register (see Section SDMA LOCK (SDMAARM_SDMA_LOCK)). SDMA scripts which load information into RAM can check the value of the LOCK bit to determine if an upload to RAM is allowed. If not allowed, the script can refuse to allow the request to copy data into the RAM to continue. The exact use of the LOCK bit in SDMA scripts for security control will be described in SDMA software documentation (see SDMA Scripts).

While SDMA is locked, attempts to write to the SDMA_LOCK, CHN0ADR, ILLINSTADDR, and ONCE_ENB registers will be ignored. All registers remain readable. Writes to other registers are still allowed.

Once the SDMA is locked, the LOCK bit can only be cleared by a reset. A hardware reset will always clear the LOCK bit. A software reset initiated by writing to the RESET register will only clear the LOCK bit if the SRESET_LOCK_CLR bit in the SDMA_LOCK register is set. Since SDMA_LOCK register cannot be updated if SDMA is locked, the SRESET_LOCK_CLR bit must be configured before setting the LOCK bit. The SREST_LOCK_CLR bit will also be cleared by resets that clear the LOCK bit.

The SDMA RISC core uses the ILLINST and CHN0ADDR registers as pointers to determine where to jump to after an illegal instruction or upon boot after a reset. The LOCK bit prevents updates to these registers to protect against unauthorized changes to these pointers.

While SDMA is locked, the ONCE_ENB register cannot be written to prevent the OnCE under ARM platform control from being used to gain access to SDMA internal memory. If ARM platform control of the OnCE is enabled before setting the LOCK bit, the ARM platform can use the ONCE for debug purpose after LOCK is set.

66.7 OnCE and PCU Debug States

The SDMA has two different debug modes in which the OnCE performs debug instructions.

Refer to Figure 66-4 for an example of the PCU states in debug. The following are the two debug states:

- When a channel is running (that is, when CCR and CCPRI are different from 0, which can be read in the PSW register), SDMA can execute a SoftBkpt instruction from the channel script or receive a debug request. When either happens, the SDMA enters its "Classical" *Debug* state, which is described in OnCE and Real-Time Debug.
- When a channel is not running, the SDMA can be in *Sleep* state or in *Sleep after Reset* state. If a debug request is sent to the core, it enters its *Debug in Sleep* state. This debug mode works similarly to the "Classical" *Debug* state, except it returns to the original state (*Sleep* or *Sleep after Reset*) when the debug mode is left via the exec_core instruction of the OnCE. From this *Debug in Sleep* state, the SDMA can execute a program whereas no channel is running. If a new debug request is sent to the core or if a SoftBkpt is executed, it comes back to this *Debug in Sleep* state.

The OnCE is provided with several instructions that can be executed when the core is in either debug state. The following table summarizes the behavior of these OnCE debug instructions. There exists other secondary OnCE instructions that are described in OnCE and Real-Time Debug.

Table 66-6. SDMA in Debug Mode

Instruction	Debug	Debug in Sleep
exec_once	exec_once <instruction> SDMA executes the <instruction> and returns to the *Debug* state. The Program Counter (PC) is not incremented. This command must not be used with an instruction that modifies the PC value.	exec_once <instruction> SDMA executes the <instruction> and returns to the *Debug in Sleep* state. The Program Counter (PC) is not incremented. This command must not be used with an instruction that modifies the PC value.
run_core	run_core <instruction> SDMA executes the <instruction>, leaves the *Debug* state and continues executing the channel script from the position where it stopped. This command must not be used with an instruction that modifies the PC value.	run_core <instruction> SDMA executes the <instruction> and returns to its *Sleep* or *Sleep after Reset* initial state. This command must not be used with an instruction that modifies the PC value.
exec_core	exec_core <instruction> It is similar to run_core except it requires an instruction that changes the PC value (jump, branch...): the SDMA jumps to the new PC value, leaves the *Debug* state and starts executing instructions from this new PC value.	exec_core <instruction> If the previous state was *Sleep after Reset*, the SDMA returns to this state, and Chn0Addr value overrides the PC value. Otherwise, the SDMA jumps to the new PC value and starts executing instructions from this new PC.

NOTE

The feature exec_core in *Debug in Sleep* after *Sleep after Reset* was added for the Channel boot (channel 0) to allow the debugger to return to *Sleep after Reset* state with a new PC

value. The SDMA will be ready to boot at the Chn0Addr address.

66.8 SDMA Clocks and Low Power Modes

The SDMA receives several root clocks from the SoC clock controller block and performs adaptive clock gating to optimize its power consumption. From a user standpoint, clock gating and power mode selection are fully automatized inside the SDMA.

Root clock control is available from the SoC clock controller block.

There are numerous clock sources that are used in the SDMA. They belong to one of two possible clock domains listed in the following table, and have frequency constraints within each domain. Clocks are considered asynchronous between domains.

Within the ARM platform/SDMA clock domain, all clocks must come from the same DPLL. The ARM platform DMA interfaces (peripheral DMA and burst DMA) receive their clock from the ARM platform DMA clock source whose frequency can be once or twice the frequency of the SDMA core clock. The DMA interfaces are designed to work at the ARM platform DMA frequency, but the SDMA core is physically limited to a maximum 104 MHz frequency. Since this is lower than the maximum ARM platform DMA frequency, the SDMA core clock is tied to the ARM platform peripheral clock frequency.

The ARM platform Peripheral Bus Clock source must be an exact sub-frequency of the SDMA Core clock source (any integer value greater or equal to 1).

Table 66-7. Clocking Scheme

Clock Domain	Source Clock	Comments
ARM platform	SDMA core (SDMA main core)	Source clock for the core and all its operations; this clock is thus used by most of the SDMA sub-blocks.
	ARM platform DMA	DMA interface for the peripheral DMA and the burst DMA. It is balanced with the main clock source, and its frequency is either once or twice the main clock frequency.
	ARM platform peripheral	Connection to the ARM platform peripheral bus. It is a sub-frequency of the main clock frequency.
JTAG	TCK	Clock for JTAG access, limited to maximum of 1/8 of the SDMA core clock frequency.

The JTAG clock is sampled by the SDMA main clock to determine its rising edge. This simplifies design and clock management, but it also adds a ratio constraint between those two clocks. It is guaranteed the JTAG interface works properly when the frequency of TCK is lower than 1/8th of the frequency of the SDMA main clock (which is about 8 MHz when the SDMA core clock frequency is 66 MHz).

66.8.1 Clock Gating and Low Power Modes

The SDMA automatically performs power saving without requiring user involvement. It implements two levels of automatic clock gating.

66.8.1.1 Coarse Clock Gating

Every sub-block clock comes from one of the five available sources, and is gated with the sub-block specific enabling condition.

The following table displays the sub-block clocks and their source. It also indicates the relationships that may exist between different sub-blocks clock enables.

Table 66-8. Sub-blocks Clocks

Sub-block	Source Clocks	Enabling Condition and Comments	Related Enabling Conditions
Core	SDMA Main Core	The core sub-block clock is running when the core is not in one of its sleep states (Sleep or Sleep after Reset) or there is a pending channel. Typically, the core sub-block clock is stopped once all the channels are processed and the core enters its sleep state. A new pending channel awakes the core sub-block clock.	None
Memories	SDMA Main Core	The clock activation only occurs during a core access.	Disabled when Core sub-block clock is disabled or no memory access in progress
Scheduler	SDMA Main Core	Its clock only runs when scheduling is needed: for example, when there are pending channels, upon reception of a DMA request, and anytime the ARM platform modifies the channel running conditions.	None
ARM platform Control	SDMA Main Core & ARM platform peripheral	The ARM platform peripheral clock is solely used to determine the frequency ratio with the SDMA main clock. The control registers' clock is based on *SDMA main clock*; it is active when the ARM platform or the SDMA modifies the contents of one of these registers.	None
CRC	SDMA Main Core	The CRC clock is based on *SDMA main clock* and is only active during data processing.	Disabled when Core sub-block clock is disabled

Table continues on the next page...

Table 66-8. Sub-blocks Clocks (continued)

Sub-block	Source Clocks	Enabling Condition and Comments	Related Enabling Conditions
Burst DMA	SDMA Main Core & ARM platform DMA	The burst DMA has two clocks: The first clock is derived from the SDMA main core clock and drives registers that are connected to the FUBUS. The second clock is derived from the ARM platform DMA clock and drives registers that are connected to the ARM platform DMA bus outside the SDMA. Both clocks are enabled during active phases of data transfers (for example, these clocks are turned off when the burst DMA is not used by the running channel script).	Disabled when Core sub-block clock is disabled
Peripheral DMA	SDMA Main Core & ARM platform DMA	The peripheral DMA has two clocks: The first clock is derived from SDMA main clock and drives registers that are connected to the FUBUS. The second clock is derived from the ARM platform DMA clock and drives registers that are connected to the ARM platform DMA bus outside the SDMA. Both clocks are enabled during active phases of data transfers (for example, these clocks are turned off when the peripheral DMA is not used by the running channel script).	Disabled when Core sub-block clock is disabled
OnCE	SDMA Main Core	The OnCE clock is derived from main source clock. It is disabled by default. In order to use the OnCE, its clock must be explicitly turned on, either by enabling the OnCE access from the ARM platform peripheral bus (register ONCE_ENB), or by driving the clk_gating_off input pin high. This is a SDMA input whose driver depends on the SoC implementation (typically a JTAG controller). The OnCE also receives the TCK input, which is the JTAG clock. It does not use it as a functional clock; the TCK input is sampled instead. Refer to Synchronization Implementation.	When enabled, all other clocks are systematically on (clock gating is off)

66.8.1.2 Refined Clock Gating

The SDMA implements a second level of clock gating on a register-per-register basis.

Unlike the first level that covers all the SDMA flip-flops, except the synchronizers (only five flip-flops are always running), the second level is only available for eligible registers, which amounts to about 90% of the SDMA flip-flops.

These gated registers are only clocked when the hardware logic detects a new data loading. This additional gating further reduces dynamic power consumption.

66.8.1.3 Low Power Modes and User Control

Power savings are automatically managed by the SDMA hardware without any user involvement; however, one can distinguish three different power modes: SLEEP, RUN, and DEBUG.

The following table describes these modes, and shows how to switch from one mode to another.

Table 66-9. Power Modes

Power Mode	Sub-blocks								Comments
	Core	Memories	Scheduler	ARM platform Control	CRC	Burst DMA	Peripheral DMA	OnCE	
SLEEP	off[1]	off	wait[2]	wait	off	off	off	off	Set when the PCU state is either *Sleep* or *Sleep after Reset* and the SDMA is not in DEBUG mode. This is the default mode after reset.
RUN	on[3]	wait	wait	wait	wait	wait	wait	off	Set for the other PCU states that are reachable out of debug: *Program*, *Data*, *Change of Flow*, *Error in Loop*, *Debug*, *Functional Unit*, *Save,* or *Restore*.
DEBUG	on	on	on	on	on	on	on	on	Set regardless of the PCU state when clock gating is turned off to use the OnCE features (either *clk_gating_off* pin high or ONCE_ENB[0] set).

1. *off*: no clock
2. *wait*: only clocked when accessed or stimulated
3. *on*: clock is always running

It is possible to control the SDMA power mode. The procedures to force the SDMA into either mode are described in SLEEP Mode.

66.8.1.3.1 SLEEP Mode

This is the default mode after reset; therefore, resetting the SDMA forces this mode.

However, the common procedure is as follows:

- Ensure the *clk_gating_off* pin is low and ONCE_ENB[0] is cleared.
- Disable all channels (via the STOP_STAT control register, and the HO, DO, EO if necessary).
- Wait for the active channels to complete or force a reschedule via the reschedule bit in the RESET register.
- The SDMA is in SLEEP mode making it possible to completely shut off its clock from the chip level clock controller using the procedure described in Stop Mode Response.

66.8.1.3.2 RUN Mode

This is the default mode when a channel is running:

- Ensure the *clk_gating_off* pin is low and ONCE_ENB[0] is cleared.
- Activate at least one channel (via the HSTART control registers, a DMA request, and/or the HO, DO, EO register bits).

66.8.1.3.3 DEBUG Mode

The DEBUG mode must be set when one needs to use the debugging facilities of the SDMA.

- Ensure the SDMA clocks are running from the CCM.
- Set the *clk_gating_off* pin high or use the SDMA to set ONCE_ENB[0].

66.8.1.4 Stop Mode Response

The SDMA receives a stop request from the chip level clock controller. This request may be asserted when the chip enters the stop low power mode.

If the SDMA is running when the request is received, then the SDMA will complete all pending channels before returning to the SLEEP state. The SDMA sends an acknowledgement to the clock controller when the SLEEP state is entered indicating that the SDMAs clocks can be turned off.

66.8.2 Reset

After reset (either received from the reset block or a software reset required by the ARM platform), the SDMA is in IDLE mode. It will start its boot code located at address 0 once a channel is activated.

Activating a channel can be done by the ARM platform after programming a positive priority and setting the channel bit in the EVTPEND register.

There will not be a context RESTORE for the first channel (bootload channel) called after a reset because the context data in RAM has not been initialized. Static context mode should be used for the first channel called after reset to ensure that the all context RAM for that channel is initialized. Subsequent calls to the same channel or different channels may use any of the dynamic context modes

66.9 Software Interface

Appendix A fully describes the SDMA Application Programming Interface (API).

66.10 Initialization Information

This section discusses the following:

66.10.1 Hardware Reset

After reset, the program RAM, context RAM, data RAM, and RAM containing the channel enable registers (CHNENBLn) have unpredictable contents.

The active register set is assigned to channel 0 and the PC is initialized to all zeros. However, since the channel enable register is all zeros, there are no active channels and the SDMA is halted waiting for the boot channel to start.

The ARM platform will have to setup the SDMA in order to boot it. The CONFIG register must be initialized to determine the DMA/core clock ratio (1 or 2). Channel Enable Registers must also be initialized.

To start up the SDMA, the ARM platform first creates some channel control blocks (CCB) and buffer descriptors (BD) in ARM platform memory for the boot channel (channel 0) and then initializes the channel 0 pointer register (SDMA_MC0PTR) to the address of the first control block. Data Structures for Boot Code and Channel Scripts provides an overview of the data structure for the CCB and BD's. The SDMA_HSTART, SDMA_HOSTOVR and SDMA_EVTOVR registers are then configured according to Runnable Channels Evaluation to allow channel 0 to run.

Upon being enabled, the SDMA begins executing the script located at the address indicated by the Channel 0 Boot Address register (SDMA_CHN0ADDR) in the program memory. The reset value of SDMA_CHN0ADDR points to the default bootload script in ROM. This ROM script will read the channel 0 pointer register (SDMA_MC0PTR) to determine the location of the Channel Control Block (SDMA_CCB) in ARM platform memory. The script will then begin fetching by DMA the first channel control block which contains a pointer to the location channel 0 Buffer Descriptor chain which is also fetched via DMA. If the buffer descriptor contains a valid command, the script interprets the command in each buffer descriptor and proceeds to implement the command and move on to the next buffer descriptor control block. The buffer descriptor commands for channel zero are typically set up to load SDMA's program RAM, Data RAM, and initial values for the channel contexts. Some channel scripts expect particular parameters to be passed

There are two ways to make the SDMA boot on a user-defined script. The OnCE (either via its JTAG interface or its ARM platform Control interface) can be used to download any code in the SDMA RAM and force the SDMA to boot on that code. Also, the SDMA_CHN0ADDR register in the ARM platform programming model can be modified to point to user code in RAM which would need to either have been loaded via the ONCE or default bootload routine (ex before a S/W reset).

66.10.2 Channel Script Execution

The execution of an SDMA script depends on both the instructions that make up the script, the data context upon which it operates, and commands or parameters passed in the buffer descriptor or. All these items must be initialized before the script is allowed to execute.

Each of the 32 channels has a separate context, but may share scripts and locations in data RAM.

Channels are initialized by the ARM platform by using channel 0 to download any required scripts and data values and the channels initial context. The context contains all the initial values of the SDMA core registers. This includes the Program Counter (PC) which is set to the start of the desired script in SDMA program memory.

The ARM platform selects which trigger conditions that must occur for the channel to start by configuring the SDMA_CHNENBL, SDMA_HOSTOVR and SDMA_EVTOVR registers. The trigger events include ARM platform setting HE (SDMA_HSTART) or a hardware DMA request asserts an event input to SDMA. The channel can become active according to its priority compared with other runnable channels when the selected trigger(s) cause the condition described in Runnable Channels Evaluation to evaluate as true.

The specific parameters to be passed to each script in the buffer descriptor or context are documented in the software documentation for each script. Please refer to SDMA Scripts for complete script documentation. Buffer Descriptor Format provides an overview of the buffer descriptor format.

66.10.3 Initialization and Script Execution Setup Sequence

To summarize, the following steps are minimally required to setup SDMA and run channel scripts.

- Perform Hardware Reset. The program RAM, context RAM, data RAM and SDMA_CHNENBLn registers have unpredictable contents after this reset

- Initialize SDMA_CHNENBLn registers to map DMA request events to desired channels.
- Configure SDMA_CHNPRIn registers to select priority for runnable channels, A non-zero priority is required for the channel to run.
- Configure the SDMA_CONFIG register to select DMA to SDMA core clock ratio .
- Set up channel control blocks and buffer descriptors in ARM platform to specify the loading of SDMA program RAM and channel contexts for each SDMA channel to be used. Reference Data Structures for Boot Code and Channel Scripts.
- Configure SDMA_MC0PTR register with base address of ARM platform Channel Control Block base address.
- Initialize SDMA_CHNENBLn registers to map DMA request events to associated channel. Reference Mapping DMA Requests to Pending Channels.
- Configure SDMA_CHNPRIn registers to set priority for each channel to be run.
- For each channel to be run, configure SDMA_HOSTOVR (HO) and SDMA_EVTOVR (EO) registers to select which events (hardware and/or software trigger events) must occur for the channel to be runnable. Reference Runnable Channels Evaluation.
- Set bit 0 of the SDMA_HSTART register to set HE[0] and allow Channel 0 to run (assumes EO[0] andDO[0] were both set in previous step). This will cause SDMA to load the program RAM and channel contexts configured previously.
- Wait for Channel 0 to finish running. This is indicated by HI[0]=1 in the SDMA_SDMA_INTR register, or by optional interrupt to the ARM platform.
- Set the LOCK bit in the SDMA_SDMA_LOCK register to prevent un-authorized uploads of data to SDMA RAM.
- Additional channel scripts can now be run by enabling the selected software or hardware trigger event according to Runnable Channels Evaluation.

66.11 SDMA Programming Model

The following section describes the programming model for the SDMA RISC engine, including its processor, memory, and internal control registers.

All addresses are related to the internal SDMA memory map, which is completely different from the ARM platform memory maps. The ARM platform processor has no access to any hardware resource described, except when those resources are described in ARM Platform Memory Map and Control Register Summary. .

66.11.1 State and Registers Per Channel

The SDMA can be seen as a set of 32 identical devices that are able to perform one data transfer channel each. Only one channel can work at a time, but every channel state is available at any time.

This chapter lists the components of every channel state.

66.11.2 General Purpose Registers

Each channel has eight general purpose registers of 32 bits for use by scripts. General register 0 has a dedicated function for the loop instruction, but otherwise can be used for any purpose.

66.11.3 Functional Unit State

Each channel context has some state that is part of the functional units.

The specific allocation of this state is part of the functional unit definition that is described in Burst DMA Unit Programming, Peripheral DMA Unit Programming .

This state must be saved/restored on context switches.

66.11.3.1 Program Counter Register (PC)

The PC is 14 bits. Since instructions are 16 bits in width and all memory in the SDMA is 32 bits in width, the low order bit of the PC selects which half of the 32-bit word contains the current instruction.

A low order bit of zero selects the most significant half of the word.[1]

66.11.3.2 Flags

Each channel has the following four flags:

- The T bit reflects the status of some arithmetic and test instructions. It is set when the result of an addition or a subtraction is zero and cleared otherwise. It is also the copy of the tested bits. Finally, it can also be set when the loop counter (GReg0) reaches

1. For example, big-Endian.

zero. When the last instruction of the hardware loop is an operation that can modify the T flag, its effect on T is discarded and replaced by the GReg0 status.

- Two additional bits, SF and DF, are used to indicate error conditions resulting from loading data sources and storing to destinations, respectively. Access errors set these bits, and successful transactions clear them. They can also be cleared by specific instructions (CLRF and loop). The source fault (SF) is updated by the loads LD and LDF; the destination fault (DF) is updated by the stores ST and STF.
- Access errors are caused by several conditions including writing to the ROM, writing to a read-only memory mapped register, accessing an unmapped address, or any transfer error received by a peripheral when it is accessed.

  The SF and DF flags have a major impact on the behavior of the hardware loop: If SF or DF is set when starting a hardware loop and it is not masked by the loop instruction, the loop body will not be executed. Inside the loop body, if a load or store sets the corresponding SF or DF flag, the loop exits immediately. Testing the status of the T flag at the end of the loop (as well as testing both SF and DF) tells if the loop exited abnormally as any anticipated exit prevents GReg0 from reaching the zero value and thus setting the T flag. This is also valid if the fault occurs at the last instruction of the last loop.

- The last flag is the loop mode flag, LM, which is composed of two bits. The most significant bit indicates when the processor is currently operating in loop mode. It is set by the loop instruction and is cleared after execution of the last instruction of the last loop. The least significant bit is set when the program counter points to the last instruction of a loop on the last path. It is used for a channel that is restored with this configuration to know that the next program counter is EPC. As with the dynamic context switch Greg0, which indicates when the program must get out of the loop, it can be restored only on the last instruction of the loop. This, however, is too late to fetch the next instruction after the loop.

66.11.3.3 Return Program Counter (RPC)

The RPC is 14 bits. It is set by the jump to the subroutine instructions and used by the return from the subroutine instructions.

Instructions are available to transfer its contents to and from a general register.

66.11.3.4 Loop Mode Start Program Counter (SPC)

The SPC is 14 bits. It is set by the loop instruction to the location immediately following it.

66.11.3.5 Loop Mode End Program Counter (EPC)

The EPC is 14 bits. It is set by the loop instruction to the location of the next instruction after the loop.

66.11.4 Context Switching-Programming

Each channel has a separate context consisting of the eight general purpose registers and additional registers representing the state of the functional units.

The active registers and functional units contain the context of the active channel. The context of inactive channels is stored in SDMA RAM, which is part of the SDMA address space.

In a function of the selected context switching mode (Context Switching), modified registers by the program can be saved in the channel RAM space while the program is going on. In every cycle, a write access to the RAM is possible.

On a done or yield(ge) instruction, SDMA goes into "real" context switching. In one of the dynamic modes, modified registers not previously saved, as well as the PC-Loop registers, are stored into the context area of the channel that will be closed. The new PC-Loop registers are loaded from the context area of the new channel. All other registers are restored while the program is executed, giving priority to registers used by the decoded instruction. Therefore, in the best case, only the PC and Loop registers should be saved and restored during this context-switching phase, which only requires five SDMA cycles.

In static mode, the context switch stores all registers in the old channel RAM space, and restores all registers from the new channel RAM space. It requires 26 SDMA cycles.

The address of the context memory for channel *i* is *CONTEXT_BASE* + 24**i* or *CONTEXT_BASE* + *32*i w*here CONTEXT_BASE equals 0x0800. The table below presents the layout of a channel context in memory:

Table 66-10. Layout of a Channel Context in Memory for SDMA

OFFSET	31	30	29-16	15	14	13-0
0	SF	-	RPC	T	-	PC
1	LM		EPC	DF	-	SPC
2	GR0					
3	GR1					
4	GR2					

Table continues on the next page...

Table 66-10. Layout of a Channel Context in Memory for SDMA (continued)

5	GR3
6	GR4
7	GR5
8	GR6
9	GR7
10	MDA (burst DMA)
11	MSA (burst DMA)
12	MS (burst DMA)
13	MD (burst DMA)
14	PDA (peripheral DMA)
15	PSA (peripheral DMA)
16	PS (peripheral DMA)
17	PD (peripheral DMA)
18	CA (CRC)
19	CS (CRC)
20	Reserved[1]
21	Reserved[1]
22	Reserved[1]
23	Reserved[1]
24	Scratch RAM (optional)
25	Scratch RAM (optional)
26	Scratch RAM (optional)
27	Scratch RAM (optional)
28	Scratch RAM (optional)
29	Scratch RAM (optional)
30	Scratch RAM (optional)
31	Scratch RAM (optional)

66.11.5 Address Space

The SDMA has four internal buses, as follows:

- The Instruction bus reads instructions from the memory. Its address map is described in Instruction Memory Map.
- The Data bus (DMBUS) accesses the same memories as those visible on the Instruction bus, some memory-mapped registers (scheduler status and OnCE

registers), and up to 14 peripherals. Its address map is described in Data Memory Map.

- The Functional Units bus (FUBUS) accesses the burst DMA, peripheral DMA, and CRC internal registers. The addressing mechanism is further detailed in Functional Units Programming Model.
- The Context Switch bus reads/writes registers into context-switch RAM space. It is a 64-bit bus dedicated for accessing this RAM space for updating the context of the running channel. While the program is going on, this bus has the lowest priority compared to the Instruction and Data buses, except for restoring a register needed for the decoded instruction to be executed. On the save part of a context switch (when the PCU is in its slave state), this is the only one used. On the restore part, the Instruction bus has the priority to read the next instruction at the restored PC and otherwise the Context Switch bus is used. It is not possible to control the actual data transfers that occur on this bus.

66.11.5.1 Instruction Memory Map

The instruction memory map is based on a 14-bit address bus and a 16-bit data (instruction) bus. Each address corresponds to a 16-bit data location.

Instructions are fetched from either program ROM or program RAM. An SDMA script is able to change the contents of the program RAM, which is also visible from the data bus.

The first two instruction locations (at 0 and 1) are special. Location 0 is where the PC is set on reset. Location 1 is where the PC is set upon the execution of an illegal instruction. It is expected that both of these locations will contain a jmp to handle routines.

Table 66-11. SDMA Instruction Memory Space

Device	SDMA Address (Hex)	Base Address Label	Block Name	WS	Description
ROM	0x0000 ↓ 0x07FF	SDMA_IBUS_ROM_ADDR	-	0	4 Kbyte internal ROM with boot code and standard routines.
RAM	0x1000 ↓ 0x1FFF	SDMA_IBUS_RAM_ADDR	-	0	8 Kbyte internal RAM with channels context and user data/routines.

66.11.5.2 Data Memory Map

All of the data accessible to SDMA scripts make up the data memory space of the SDMA.

This address space has several components:

- ROM (also visible on the Instruction bus)
- RAM (also visible on the Instruction bus)
- Shared Peripherals Registers
- SDMA Internal Registers (scheduler, OnCE, and registers that are also accessible by the ARM platform)

SDMA scripts can read and write to the context RAM, data RAM, shared peripheral registers, and internal registers.

The address range is 16 bits and the data width is 32 bits. Each address corresponds to a 32-bit data word. When accessing peripheral registers (USB and so on), the data width may be different. The exact address map for the peripherals depends on the project (as presented in each respective chapter).

The SDMA can perform only 32-bit access to shared peripheral registers. For this device, shared peripherals registers are aliased as if byte addressed. This is a consequence of connections through the SPBA shared peripheral bus outside of the SDMA. The result is that although address space 4 Kwords (16Kbyte) is allocated for each peripheral, only the first 4 Kbyte of the peripheral's register space can accessed. For example, the shared peripheral register at address 0x3000 is mapped also to addresses 0x3001, 0x3002, 0x3003. A read or write access to any of any of these 4 addresses will respond as if the access was to address 0x3000.

Data access is performed with *ld* and *st* instructions that take the address from a general purpose register in the core (GReg*n*). The mapping between the general purpose register contents and the address bus is given in the following table:

Table 66-12. GReg*n* to DMBUS Address Mapping

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sz	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
address															

Grayed bits are simply discarded but they must be cleared to ensure forward-script compatibility.

- sz (bit 31) indicates the peripheral data width: 0 is used for a 32-bit peripheral and 1 is used for a 16-bit peripheral.
- address (bits 15 down to 0) is the address of the accessed resource (internal memory, internal register, or shared peripheral).

Table 66-13. SDMA Data Memory Space

Device	SDMA Address (Hex)	Size	Description
ROM	0x0000 → 0x03FF	4 Kbyte	4 Kbyte internal ROM with boot code and standard routines
Reserved	0x0400 → 0x07FF	4 Kbyte	4 Kbyte Reserved
RAM	0x0800 → 0x0FFF	8 Kbyte	8 Kbyte internal RAM with channels contexts and user data/routines
per1	0x1000 → 0x1FFF	16 Kbyte	*peripheral 1* memory space (4 Kbyte peripheral's address space)
per2	0x2000 → 0x2FFF	16 Kbyte	*peripheral 2* memory space (4 Kbyte peripheral's address space)
per3	0x3000 → 0x3FFF	16 Kbyte	*peripheral 3* memory space (4 Kbyte peripheral's address space)
per4	0x4000 → 0x4FFF	16 Kbyte	*peripheral 4* memory space (4 Kbyte peripheral's address space)
per5	0x5000 → 0x5FFF	16 Kbyte	*peripheral 5* memory space (4 Kbyte peripheral's address space)
per6	0x6000 → 0x6FFF	16 Kbyte	*peripheral 6* memory space (4 Kbyte peripheral's address space)
Registers	0x7000 → 0x7FFF	16 Kbyte	Memory mapped registers
per7	0x8000 → 0x8FFF	16 Kbyte	*peripheral 7* memory space (4 Kbyte peripheral's address space)
per8	0x9000 → 0x9FFF	16 Kbyte	*peripheral 8* memory space (4 Kbyte peripheral's address space)
per9	0xA000 → 0xAFFF	16 Kbyte	*peripheral 9* memory space (4 Kbyte peripheral's address space)
per10	0xB000 → 0xBFFF	16 Kbyte	*peripheral 10* memory space (4 Kbyte peripheral's address space)
per11	0xC000 → 0xCFFF	16 Kbyte	*peripheral 11* memory space (4 Kbyte peripheral's address space)
per12	0xD000 → 0xDFFF	16 Kbyte	*peripheral 12* memory space (4 Kbyte peripheral's address space)
per13	0xE000 → 0xEFFF	16 Kbyte	*peripheral 13* memory space (4 Kbyte peripheral's address space)
per14	0xF000 → 0xFFFF	16 Kbyte	*peripheral 14* memory space (4 Kbyte peripheral's address space)

66.12 SDMA Initialization

Appendix A describes the setup of the SDMA . This section provides a quick description of several initialization procedures.

NOTE

There may be differences with the actual implementation in the API.

66.12.1 Hardware Reset-SDMA

After reset, the RAM that holds contexts, data, scripts, and the DMA request-channels matrix has unpredictable content.

The core registers are all reset to 0, including the PC; the PCU state is *Sleep after Reset*. No channel can be activated because all of the priorities are also reset to 0.

66.12.2 Standard Boot Sequence

The following is the standard boot sequence:

1. Initialize the CONFIG register-detailed in Configuration Register (SDMAARM_CONFIG)-to determine the ARM platform DMA/core clock ratio (1 or 2)
2. Initialize the DMA request-channels matrix (seeChannel Enable RAM (SDMAARM_SDMA.CHNENBL*n*)).
3. Program the channel control registers-Channel Event Override (SDMAARM_EVTOVR), Channel BP Override (SDMAARM_DSPOVR), Channel BP Override (SDMA_HOSTOVR), and Channel Event Pending (SDMAARM_EVTPEND)-according to the channel allocation.
4. Perform any necessary setup as required by the standard boot script in ROM (this is described in Appendix A).
5. Trigger channel 0 with the Channel Start (SDMAARM_HSTART) register, which starts the execution of the ROM script starting at address 0. This boot downloads channel scripts and contexts in RAM.

66.12.3 User-Defined Boot Sequence

The following is a user-defined boot sequence:

1. Initialize the Configuration Register (SDMAARM_CONFIG)Channel Enable RAM (SDMAARM_SDMA.CHNENBL*n*), Channel Event Override (SDMAARM_EVTOVR), Channel BP Override (SDMAARM_DSPOVR), Channel ARM platform Override (SDMAARM_HOSTOVR), and Channel Event Pending (SDMAARM_EVTPEND).
2. Use the OnCE (either via its JTAG interface or its ARM platform control registers) to download any code in the SDMA RAM. Accessing the Memory describes how to write data to the RAM via the OnCE.
3. Use the OnCE instructions to make the PC default value point to the new boot script start address, or rely on the ROM startup script, which first jumps to the address in Channel 0 Boot Address (SDMAARM_CHN0ADDR). (This register default address points to the standard boot script.)

66.12.4 Script Loading and Context Initialization

The execution of an SDMA script depends on both the instructions that make up the script and the data context upon which it operates. Both must be initialized before the script is allowed to execute.

Each of the 32 channels has a separate data context, but may share scripts and locations in the data RAM.

The ARM platform manages the space in program RAM and data RAM. It also manages the assignment of SDMA channels to the device drivers that need them. Channels are initialized by the ARM platform via the channel 0 boot script. The boot channel downloads any required scripts with their data and the channels' initial contexts. Every context contains all the initial values of the registers, including the PC. Then the ARM platform can enable any channel that becomes active and begins fetching and executing instructions from its script.

66.13 Instruction Description

The following sections introduce the instruction of the SDMA. Instruction set details are available in Instruction Set.

66.13.1 Scheduling Instructions

The following are scheduling instructions:

- done-The instruction causes certain scheduling or interrupt bits to be set or cleared, which may cause a change in the schedule-ability of the running channel. Then the instruction causes the SDMA to evaluate the current scheduling priorities and to choose the highest priority ready channel. If this channel is not the current channel, a context switch will take place. If there are no runnable channels, the SDMA will enter the stopped mode. The done 5 has a special usage reserved for debug, as explained in Debug Instructions.
- yield-These instructions are special cases of the done instruction. They do not modify the scheduling bits, but allow the highest pending channel (if it exists) to preempt the current channel if the pending channel priority is strictly greater than the current channel priority.
- yieldge-These instructions are special cases of the done instruction. They do not modify the scheduling bits, but allow the highest pending channel (if it exists) to

preempt the current channel if the pending channel priority is strictly greater or equal to the current channel priority.
- notify-The notify instruction affects the scheduling bits, but does not cause rescheduling.

66.13.2 Conditional Branch Instructions

The conditional branch instructions of an 8-bit displacement, which is sign-extended and added to the current PC (which points to the next instruction) if the condition is satisfied.

Otherwise, control passes to the next sequential instruction.

- BF-Branch if False. The branch is taken if the T bit in the processor status is zero (false).
- BT-Branch if True. The branch is taken if the T bit in the processor status is one (true).
- BSF-Branch if Source Fault. The branch is taken if the SF bit in the processor status is one.
- BDF-Branch if Destination Fault. The branch is taken if the DF bit in the processor status is one.

66.13.3 Unconditional Jump Instructions

There are two varieties of unconditional control transfers: an absolute transfer and a through-register transfer.

Absolute transfers have a 14-bit address field that replaces the current PC.

- JMP-Jump. Causes the processor to jump to an absolute address encoded in the instruction itself.
- JSR-Jump to Subroutine. Causes the processor to jump to a subroutine, the address of which is encoded in the instruction itself.
- JMPR-Jump through Register. Causes the processor to jump to an absolute address contained in a General register. This instruction is meant to be used when more than one level of subroutines are required.
- JSRR-Jump to Subroutine through Register. Causes the processor to jump to a subroutine, the address of which is contained in a General register. This instruction is meant to be used when more than one level of subroutines are required.

66.13.4 Subroutine Return Instructions

The following are subroutine return instructions:

- RET-Return from Subroutine. The RET restores the contents of RPC to PC.
- LDRPC-Load from RPC to Register. THe LDRPC instruction is meant to be used when more than one level of subroutines are required. It stores the contents of RPC in any General register.

66.13.5 Loop Instruction

The following is a loop instruction:

LOOP-Enters Loop Mode. Before entering loop mode, the loop instruction can optionally clear the fault flags (SF and/or DF) based on a 2-bit field in the instruction. This feature is linked to the fact that setting SF or DF in loop mode will cause an immediate exit of the loop.

66.13.6 Miscellaneous Instructions

The following are miscellaneous instructions:

- CLRF-Clear Fault Flags. This instruction clears any combination of SF and DF.
- MOV r,s-This moves data from GReg[s] to GReg[r].
- LDI r,immediate-This loads GReg[r] with a zero-extended immediate value.

66.13.7 Logic Instructions

The following are logic instructions:

- XORr,s-This performs an exclusive or between GReg[r] and GReg[s], and stores the result in GReg[r].
- XORIr,immediate-This performs an exclusive or between GReg[r] and a zero-extended immediate value, and stores the result in GReg[r].
- ORr,s-This performs an or between GReg[r] and GReg[s], and stores the result in GReg[r].
- ORIr,immediate-This performs an or between GReg[r] and a zero-extended immediate value and, stores the result in GReg[r].
- ANDNr,s-This performs an and between GReg[r] and the negated GReg[s], and stores the result in GReg[r].

- ANDNIr,immediate-This performs an and between GReg[r] and the negated zero-extended immediate value, and stores the result in GReg[r].
- ANDr,s-This performs an and between GReg[r] and GReg[s], and stores the result in GReg[r].
- ANDIr,immediate-This performs an and between GReg[r] and a zero-extended immediate value, and stores the result in GReg[r].

66.13.8 Arithmetic Instructions

Arithmetic instructions modify the T bit in the processor status according to the result of the operation. The T bit is set if the result is zero, otherwise it is cleared.

- ADD r,s-This performs the addition of GReg[r] and GReg[s], and stores the result in GReg[r].
- ADDI r,immediate-This performs the addition of GReg[r] and a zero-extended immediate value, and stores the result in GReg[r].
- SUB r,s-This performs the subtraction of GReg[s] from GReg[r], and stores the result in GReg[r].
- SUBIr,immediate-This performs the subtraction of a zero-extended immediate value from GReg[r], and stores the result in GReg[r].

66.13.9 Compare Instructions

Compare instructions modify the T bit in the processor status according to the result of the operation. The T bit is set if the comparison is true, otherwise it is cleared.

NOTE

Only one version of the immediate form is implemented. Non-equality comparisons to immediate values will require two instructions.

- CMPEQ r,s-This sets T when registers GReg[r] and GReg[s] are equal.
- CMPEQIr,immediate-This sets T when register GReg[r] and the zero-extended immediate value are equal.
- CMPLTr,s-This sets T when register GReg[r] is less than and not equal to GReg[s]. The comparison is signed.
- CMPHS r,s-This sets T when register GReg[r] is greater than or equal to GReg[s]. The comparison is signed.

66.13.10 Test Instructions

Test instructions modify the T bit in the processor status according to the result of the operation. The T bit is set if any bit in the result is one, otherwise it is cleared.

- TSTr,s-This performs an and between GReg[r] and GReg[s], and sets T if the result is not zero.
- TSTIr,immediate-This performs an and between GReg[r] and a zero-extended immediate value, and sets T if the result is not zero.

66.13.11 Byte Permutation Instructions

These instructions shuffle the bytes in a register. For the purpose of describing these instructions, have the bytes in a register be numbered from the most significant as b_3, b_2, b_1, b_0.

- RORBr-The rotate right byte. The result is b_0, b_3, b_2, b_1.
- REVBr-The reverse bytes in word. The result is b_0, b_1, b_2, b_3.
- REVBLOr-The reverse, two low-order bytes. The result is b_3, b_2, b_0, b_1.

66.13.12 Bit Shift Instructions

The following are bit shift instructions:

- ROR1r-The rotate right 1 bit. This instruction does a circular right shift of 1 bit.
- LSR1r-The logical shift right 1 bit. This instruction shifts all bits to the right by 1. The high order bit is replaced by a 0.
- ASR1r-The arithmetic shift right 1 bit. This instruction shifts all bits to the right by 1. The high order bit is replaced by itself.
- LSL1r-The logical shift left 1 bit. This instruction shifts all bits to the left by 1. The low order bit is replaced by zero.

66.13.13 Bit Manipulation Instructions

- BCLRIr,n-The bit clear is immediate; clears bit number *i* in register *r*.
- BSETIr,n-The bit set is immediate; sets bit number *i* in register *r*.
- BTSTIr,n-The bit test is immediate; tests bit number *i* in register *r* (T becomes equal to the selected register bit).

66.13.14 SDMA Memory Access Instructions

All memory accesses are 32 bits.

Any memory location that is implemented with less than 32 bits (for example, peripheral registers) causes unimplemented bits to be read as 0s.

All memory accesses will cause either the SF or DF flags in the processor status to be set if they cause a fault.

What constitutes a fault, especially when accessing peripheral registers, is a property of the memory location.

- LDr,(b,d)-The load instruction creates an address by adding the displacement field (d) to the contents of the base register (b). The SDMA location at the resulting address is read and placed in the destination register (r).
- STr,(b,d)-The store instruction creates an address in the same manner as the load instruction. The register (r) is stored in the SDMA location at the resulting address.

66.13.15 Functional Unit Instructions

The functional unit instructions have an 8-bit field that is placed on the functional unit bus.

Some of these bits are used to select which functional unit should be involved in the transfer. The remaining bits are decoded by the selected functional unit so their specific use depends on the functional unit. See Functional Units Programming Model.

There are two functional unit instructions, as follows:

- LDFr,fub-The 8-bit field is placed on the functional unit bus and a read is issued to the selected functional unit. As a result of this instruction, the SF may be set in the processor status.
- STFr,fub-The 8-bit field is placed on the functional unit bus and a write is issued to the selected functional unit. As a result of this instruction, the DF may be set in the processor status.

66.13.16 Illegal Instructions

All instruction encodings that are illegal cause the following actions:

- The current PC (which points to one beyond the offending instruction) is put in the EPC register.

- The loop mode bit is cleared.
- The PC is set to the value stored in the Illegal Instruction Trap Address (SDMAARM_ILLINSTADDR) register (the default value is 0x0001).

ILLEGAL-Although any instruction other than those indicated in the SDMA specification will trigger the illegal instruction mechanism, the ILLEGAL instruction code is preferred as it will always be kept as *illegal* in the possible future versions of the SDMA core.

66.13.17 Debug Instructions

The following are debug instructions:

- SOFTBKPT-The software breakpoint instruction causes the core to stop and enter debug mode. The core can then be accessed and started by the OnCE debug block only.
- done 5-This instruction is used for debugging, as it copies the contents of the PCU registers and flags to the context memory. Information on this instruction is described in Saving the Context.
- CpShReg-This instruction copies the context memory into the PCU registers and flags. Modifying the corresponding memory location before executing this instruction enables you to have the channel continue from a new instruction address. This instruction is described in Restoring the Context.

66.14 Functional Units Programming Model

The functional unit instructions cause an 8-bit code, found in the low eight bits of the instruction, to be asserted on the functional unit control bus.

Some of these bits are used to select one of several functional units. Functional units which can be selected include SDMA registers such as MSA and MSD which are not mapped in the SDMA memory map, and are accessible only through the functional unit bus. These Functional Unit Registers are listed in the following table. In order to establish

a programming convention, assume the selection bits are some number of the most significant bits of the 8-bit code. Furthermore, some number of the least significant bits is decoded by a given functional unit to establish the type of operation to perform.

Table 66-14. Functional Unit Registers

Functional Unit	Register	Register Name	Section/Page
Burst DMA Unit Programming	SDMSA	Memory Source Address Register	Memory Source Address Register (MSA)
	MDA	Memory Source Address Register	Memory Destination Address Register (MDA)
	MD	Memory Data Buffer Register	Memory Data Buffer Register (MD) (Write) Burst DMA Write (stf) (Read) Burst DMA Read (ldf)
	MS	Memory State Register	State Register (MS)
Peripheral DMA Unit Programming	PSA	Peripheral Source Address Register	Peripheral Source Address Register (PSA)
	PDA	Peripheral Source Address Register	Peripheral Destination Address Register (PDA)
	PD	Peripheral Data Buffer Register	Peripheral Data Register (PD) (Write) Peripheral DMA Write (stf)-Write Mode (Read) Peripheral DMA Read (ldf)-Read Mode
	PS	Peripheral State Register	Peripheral State Register (PS)
CRC Unit	CA	Polynomial Register	Polynomial Register (CA)
	CS	Accumulator Register	Accumulator Register (CS)

More information regarding the functional units can be found in CRC Calculation Unit,Peripheral DMA Unit, and Burst DMA Unit.

66.14.1 Burst DMA Unit Programming

The DMA instructions control the DMA state machine and may cause a DMA cycle on the associated memory bus.

There are four registers associated with the burst DMA unit: a Memory Source Address register (MSA), a Memory Destination Address register (MDA), a Memory Data buffer (MD), and a state register (MS). The burst DMA has two different uses:

- A data transfer between External Memory Interface and SDMA general register
- A data transfer in copy mode where blocks of data are transferred from the source address to the destination address

66.14.1.1 Memory Source Address Register (MSA)

The source address register contains the pointer into EXTMC memory associated with the next read data transfer. It has byte granularity.

Reading the register with the ldf instruction has no side effects, and gives the address value in the EXTMC memory of the next data that is read by the SDMA during an ldf MD instruction.

Writing the source address register has two side effects: If the prefetch bit is set, a DMA read cycle (8-word read access) is issued with the new address. Any data still located in the buffer is lost. If there is valid write data in the buffer, it is necessary to force the DMA to completely flush it out before modifying MSA to guarantee all the data is effectively written to memory.

The MSA register has two modes of programming:

- Frozen-In frozen mode, the MSA register is not modified after DMA accesses.
- Incremented (default mode)-In incremental mode, MSA is incremented by the number of bytes transferred during read cycles.

66.14.1.2 Memory Destination Address Register (MDA)

The destination address register contains the pointer into EXTMC memory associated with the next write data transfer. It has byte granularity.

Reading the MDA register with the ldf instruction has no side effects. It gives the address value in the EXTMC memory where the next SDMA data (stf r,MD instruction) is stored when MD FIFO is flushed.

Writing the destination address register has one side effect. Any data still located in the buffer is lost. If there is valid write data in the buffer, it is necessary to force the DMA to completely flush it out before modifying MDA to guarantee all the data is effectively written to memory.

The MDA register has two modes of programming:

- Frozen-In frozen mode, the MDA register is not modified after DMA accesses.
- Incremented (default mode)-The MDA register is incremented by the number of bytes transferred during write cycles.

66.14.1.3 Memory Data Buffer Register (MD)

The data buffer register consists of a bank of 36 bytes that behave like FIFO.

This FIFO stores the eight words received when a read burst is triggered by the DMA (DMA is in read mode).

The MD register is in write mode after a writing in MDA or after an stf MD instruction.

In that case, a burst write access is automatically triggered when there are more than eight words in MD. For bandwidth optimization, any transfers between DMA and the EXTMC controller are based on burst accesses.

An ldf r,MD|SIZE instruction that reads the data buffer may cause a DMA cycle, as follows:

- If there are less bytes in the FIFO than the size parameter of the instruction. For instance, if only two bytes are available in MD and a 4-byte read is requested, a burst read access is executed to complete the two bytes.
- If the prefetch bit is set, and after reading there is enough space in the FIFO to store a full burst, a burst read access is triggered.

An stf r,MD|SIZE instruction that writes to the data buffer may cause a DMA cycle if the number of written bytes in MD is higher than 32 (eight words) or if the flush bit is set.

When DMA is used for data transfer between SDMA and EXTMC (reading or writing), no immediate error is possible because the block manages a data misalignment issue; therefore, it is allowed to read/write a word to/from a half-word address. However, the addresses (source or destination) must belong to the EXTMC memory mapping. The only potential error, in this mode, would be the error sent back by the EXTMC controller when an access to a super-user page is detected. The whole transfer on the DMA associated bus will be considered successful when there are no errors seen on the bus during the transfer. In copy mode, an immediate error could be returned to SDMA as described in Burst DMA Unit Error Management.

66.14.1.4 State Register (MS)

The state register contains the DMA state-machine value. It can be accessed in case of an error received during a transfer. MS is also accessed to set-up the conditional yielding feature.

The initialization value of this register is 0 and it consists of the following:

Table 66-15. SDMA_MS Structure

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	spriv	stype	0	0	dpriv	dtype
W																
R	0	0	0	0	y	d	e		0	0	n					
W																

Table 66-16. SDMA_MS Field Descriptions

Field	Description
31-22	Reserved
21 spriv	The spriv value is ignored for this device. 0 = valid value 1 = Reserved
20 stype	Source Mode. Indicates if MSA has to be incremented (or not) during accesses. 0 Frozen-MSA is not modified. 1 Incremented-MSA is incremented by the number of transferred bytes during read access.
19-18	Reserved
17 dpriv	The dpriv value is ignored for this device. 0 = valid value 1 = Reserved
16 dtype	Destination Mode. Indicates if MDA has to be incremented (or not) during accesses. 0 Frozen-MDA is not modified. 1 Incremented-MDA is incremented by the number of transferred bytes during write access.
15-12	Reserved
11 y	Conditional Yielding selector. When selected, theyield/yieldge instructions will not switch channels if the Burst DMA is in Write Mode, and it has less than four bytes in its FIFO. This is aimed at reducing the number of inefficient FIFO flushes due to context switches. 0 Always yields 1 Yields conditionally (when there are less than four bytes in the FIFO in write mode)

Table continues on the next page...

Table 66-16. SDMA_MS Field Descriptions (continued)

Field	Description
10 d	Access Direction or DMA Mode. DMA is in write mode when data was written into MD by stf MD instructions, or if a previous DMA cycle on the external bus was a write access. Writing MDA or MSA changes the DMA mode to the respective value. DMA is in read mode when a previous DMA cycle was a read access, and DMA stays in read mode when data is read by SDMA with an ldf MD instruction. Reading MDA or MSA does not change the DMA mode. 0 Read Mode 1 Write Mode
9-8 e	Error. Indicates if the previous access was acknowledged with a bus error. 00 No error was received. 01 *reserved* 10 Error mode 11 error read burst
7-6	Reserved
5-0 n	Number of bytes in the MD FIFO.

66.14.1.5 Burst DMA Write (stf)

When received from a stf instruction, the function code bits are interpreted as follows, depending on the addressed register:

Table 66-17. STF Code Bits

Register	7	6	5	4	3	2	1	0
MSA	s		p	freeze	r			spriv
MDA								dpriv
MD			f	cpy			sz	
MS								

Table 66-18. STF Code Bit Field Descriptions

Field	Description
7-6 s	Functional Unit selector 00 for Burst DMA
5 p (MSA)	Prefetch Flag 0 No prefetch 1 Prefetch required from new MSA

Table continues on the next page...

Table 66-18. STF Code Bit Field Descriptions (continued)

Field	Description
5 f (MD)	Forced Flush Flag 0 Automatic flush 1 FIFO contents are flushed (including the new written data).
4 freeze (MSA/MDA)	Address Freeze Mode 0 Address is normally incremented. 1 Address is frozen.
4 cpy (MD)	Copy Mode selection 0 Write Mode 1 Copy Mode
3-2 r	Register selection 00 MSA 01 MDA 10 MD 11 MS
1-0 sz (MD/MS)	Transfer Size 00 size 0 (no data stored in the FIFO) 01 byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)
0 spriv (MSA)	The spriv value is ignored for this device. 0 = valid value 1 = Reserved
0 dpriv (MDA)	The dpriv value is ignored for this device. 0 = valid value 1 = Reserved

The possible write instructions are listed in the table below (unused bits should always be cleared).

Table 66-19. Burst DMA STF Instruction List

Binary	Assembly	Comments
00_0_0_00_00	stf r,MSA	Writes content of the SDMA general register (r) to the source address register. MSA is in incremented mode.
00_0_1_00_00	stf r,MSA|FR	Writes content of the SDMA general register (r) to the source address register. MSA is in frozen mode.
00_1_0_00_00	stf r,MSA|PF	Writes content of the SDMA general register (r) to the source address register, and starts a read burst access. MSA is in incremented mode.

Table continues on the next page...

Table 66-19. Burst DMA STF Instruction List (continued)

Binary	Assembly	Comments
00_1_1_00_00	stf r,MSA|PF|FR	Writes content of the SDMA general register (r) to the source address register, and starts a read burst access.
00_0_0_01_00	stf r,MDA	Writes content of the SDMA general register (r) to the destination address register. MDA is in incremented mode.
00_0_1_01_00	stf r,MDA|FR	Writes content of the SDMA general register (r) to the destination address register. MDA is in frozen mode.
00_1_0_10_00	stf r,MD|SZ0|FL	No data transfers between the SDMA and MD, but all valid written data of the MD is flushed to the memory. An acknowledge or error is sent back to the SDMA core on transfer completion.
00_0_0_10_01	stf r,MD|SZ8	8-bit (byte) transfer to write buffer MD
00_1_0_10_01	stf r,MD|SZ8|FL	8-bit (byte) transfer to write buffer MD and flush after transfer. All valid written data of the MD is flushed to memory.
00_0_0_10_10	stf r,MD|SZ16	16-bit (half-word) transfer to write buffer MD
00_1_0_10_10	stf r,MD|SZ16|FL	16-bit (half-word) transfer to write buffer MD and flush after transfer. All valid written data of the MD is flushed to memory.
00_0_0_10_11	stf r,MD|SZ32	32-bit (word) transfer to write buffer MD
00_1_0_10_11	stf r,MD|SZ32|FL	32-bit (word) transfer to write buffer MD and flush after transfer. All valid written data of MD is flushed to memory.
00_0_1_10_00	stf r,MD|CPY	No data transfer between SDMA and MD but starts a copy transfer whose length is given by the 4 LSB of *r* register. (Maximum burst length is eight words.)
00_0_0_11_11	stf r,MS	32-bit (word) transfer to status register MS
00_0_0_11_00	stf r,MS|SZ0	Clears the error flag (if set). Other MS bits are unchanged; this instruction is also known as clref MS.

NOTE

When a flush bit is set, the SDMA flushes the FIFO including the newly written data. An acknowledge is sent to the core before the flush completes (except if size 0 is used). The goal of this flush bit is to force a flush, but it is recommended to use it only when needed (for example, when finishing a row of pixels during 2D data transfers). Indeed, if this bit is omitted and if there are more than 32 bytes in the FIFO, a burst write access is automatically triggered.

Since all the stf r,MD instructions (including the copy mode) acknowledge the SDMA core before the store is effective (except if size 0 is used), it is recommended to perform an ldf from MS before terminating a channel in order to check the final error status. (The ldf from MS will stall the core until all the data was flushed out and the transfer status is known.)

After every stf MD instruction, the MDA is incremented by the number of bytes that are written in MD, except when it is programmed in frozen mode.

66.14.1.6 Burst DMA Read (ldf)

When received from an ldf instruction, the function code bits are interpreted as follows, depending on the addressed register:

Table 66-20. LDF Code Bits

Register	7	6	5	4	3	2	1	0
MSA	s				r			
MDA								
MD			p				sz	
MS								

Table 66-21. LDF Code Bit Field Descriptions

Field	Description
7-6 s	Functional Unit selector 00 for Burst DMA
5 p (MD)	Prefetch Flag 0 no prefetch 1 automatic prefetch
3-2 r	Register selection 00 MSA 01 MDA 10 MD 11 MS
1-0 sz (MD)	Transfer Size 00 reserved 01 byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)

The table below lists the possible write instructions (unused bits should always be cleared).

Table 66-22. Burst DMA LDF Instruction List

Binary	Assembly	Comments
00_0_0_00_00	ldf r,MSA	Copies the source address register value into an SDMA general register. It gives the memory address of the next data that will be read with an ldf MD instruction.
00_0_0_01_00	ldf r,MDA	Copies the destination address register value into an SDMA general register. It gives the memory address where the next incoming data will be flushed.
00_0_0_10_01	ldf r,MD|SZ8	8-bit (byte) read
00_1_0_10_01	ldf r,MD|SZ8|PF	8-bit (byte) read. If after this reading and the MD FIFO is empty, a burst read access at the MSA address is triggered.
00_0_0_10_10	ldf r,MD|SZ16	16-bit (half-word) read
00_1_0_10_10	ldf r,MD|SZ16|PF	16-bit (half-word) read. If after this reading, and the MD FIFO is empty, a burst read access at the MSA address is triggered.
00_0_0_10_11	ldf r,MD|SZ32	32-bit (word) read
00_1_0_10_11	ldf r,MD|SZ32|PF	32-bit (word) read. If after this reading and the MD FIFO is empty, a burst read access at the MSA address is triggered.
00_0_0_11_00	ldf r,MS	Copy the status register value into an SDMA general register.

NOTE

Read data is 0-extended before writing in the SDMA general registers. When reading the MD register, the DMA takes data from the FIFO if it is available. If part or whole data is not in the FIFO, an external burst read access is performed to provide the missing data. The SDMA is stalled as long as the required read data is not complete.

After every reading, MSA is incremented by the number of read bytes from MD FIFO, except when MSA is programmed in frozen mode.

66.14.1.7 Prefetch/Flush and Auto-Flush Management-Burst DMA Unit

The prefetch and auto-flush management enables the SDMA RISC machine to go on while a DMA access is performed.

When the RISC core requires a prefetch (p = 1) to the Burst DMA, it will receive an immediate transfer acknowledge before the DMA has finished the external access. This enables the RISC core to do other things like accessing another DMA machine.

The basic principle in prefetch mode is for the DMA to anticipate data reads from the SDMA RISC engine by fetching external bursts of data as soon as there is enough space in the DMA FIFO to store it. If ever the RISC engine required data that is not available in the FIFO, the read acknowledge is delayed until the data is available, but it does not have to wait until the burst completes.

The auto-flush basic principle is similar: An automatic flush is triggered every time there are eight words to be written in the FIFO. If the FIFO is full and the RISC engine requires another write, it is stalled until the burst has started and enough space was freed in the FIFO to store that new data. This means the SDMA RISC engine does not have to wait for the completion of a burst to receive its acknowledge and continue its processing.

In particular, an auto-flush is executed when DMA is in write mode and if the following is true:

- If the FIFO is empty and the first write is to a word-aligned address of any size (ex: the 2 LSB of MDA[1:0]= 0x0), the auto-flush is triggered immediately after the write of the 32'nd byte.
- If the FIFO is empty, and if MDA is an odd byte address (1, 3, 5, 7,...) and an stf MD|SZ8 is executed, the byte is flushed to memory. Once MDA increments to a word aligned address, the auto-flush will be triggered every 32 bytes.
- If the FIFO is empty, and if MDA is a half-word address (2, 6, 0xA,...) and an stf MD|SZ16 is executed, the two bytes of the incoming data are flushed to memory. Once MDA increments to a word aligned address, the auto-flush will be triggered every 32 bytes.
- If the FIFO is empty, and if MDA is not a word-aligned address (ex 1, 2, 3, 5, 6, 7, 9,...), and an stf MD|SZ32 is executed, the first 1 to 3 bytes will be flushed up to the next word aligned address. Afterwards, an auto-flush will be triggered each time the FIFO receives 32-bytes.
- Therefore, if an stf MD|SZ32 is executed with MDA equal to 0x1 and with an empty MD FIFO, the bytes located at addresses 1, 2, and 3 are flushed, and the byte located at address 4 remains in MD FIFO. This solves the misalignment issue. Additionally, the next write instructions (stf) complete the FIFO until it contains eight words; then a burst write is executed by the DMA to empty the FIFO. Protocol on the external bus does not support bursts of different data types (byte, half-word, or word).

  For example, consider the case where data is written using a byte access, stf MD|SZ8. The value of MDA during the very first byte write determines when the auto-flush will occur as follows:

- If MDA=0x0, the flush occurs following the write of byte 32
- If MDA=0x1, the flush occurs following the write of byte 1, byte 3 and byte 35.
- If MDA=0x2, the flush occurs following the write of byte 2 and byte 34.

- If MDA=0x3, the flush occurs following the write of byte 1 and byte 33.
- If MDA=0x4, the flush occurs following the write of byte 32

The flush command forces the DMA to flush all MD valid bytes to the EXTMC controller. An acknowledge is sent immediately to the SDMA, and any potential error is reported on a future access. It is thus essential to conclude a transfer with a last read from MS, which will stall the core until all data was flushed out and returned to the transfer status (acknowledge or error).

NOTE

During this kind of auto-flush (which occurs only at the beginning of a misaligned write transfer) no acknowledge is sent back to the SDMA, which is stalled until a flush is completed.

66.14.1.8 Data Alignment and Endianness-Burst DMA Unit

66.14.1.8.1 Burst DMA in Read Mode

For every read access to MD, the data returned to the SDMA core and the new FIFO state depends on the MSA status and the access size.

The FIFO is considered as a stack of 36 bytes: Data is fetched externally on a 32-bit bus, but the valid bytes only are stored in the FIFO and left-aligned (for a transfer of consecutive words, it is only the first word that may be truncated). The following table shows the FIFO byte alignment strategy and the corresponding MSA, the returned data, and the new FIFO state for any access size of an internal read from MD.

Table 66-23. FIFO Read Configuration

Before read		Internal read access size	Read data	After read	
MSA[1:0]	FIFO state			MSA[1:0]	FIFO state
00	x0 x1 x2 x3 y0 y1 y2 y3 z0 z1 z2 z3 and so on...	sz8	00 00 00 x0	01	x1 x2 x3 y0 y1 y2 y3 z0
		sz16	00 00 x0 x1	10	x2 x3 y0 y1 y2 y3 z0 z1
		sz32	x0 x1 x2 x3	00	y0 y1 y2 y3 z0 z1 z2 z3

Table continues on the next page...

Table 66-23. FIFO Read Configuration (continued)

Before read		Internal read access size	Read data	After read	
MSA[1:0]	FIFO state			MSA[1:0]	FIFO state
01	x1 x2 x3 y0 y1 y2 y3 z0 z1 z2 z3 t0 and so on...	sz8	00 00 00 x1	10	x2 x3 y0 y1 y2 y3 z0 z1
		sz16	00 00 x1 x2	11	x3 y0 y1 y2 y3 z0 z1 z2
		sz32	x1 x2 x3 y0	01	y1 y2 y3 z0 z1 z2 z3 t0
10	x2 x3 y0 y1 y2 y3 z0 z1 z2 z3 t0 t1 and so on...	sz8	00 00 00 x2	11	x3 y0 y1 y2 y3 z0 z1 z2
		sz16	00 00 x2 x3	00	y0 y1 y2 y3 z0 z1 z2 z3
		sz32	x2 x3 y0 y1	10	y2 y3 z0 z1 z2 z3 t0 t1
11	x3 y0 y1 y2 y3 z0 z1 z2 z3 t0 t1 t2 and so on...	sz8	00 00 00 x3	00	y0 y1 y2 y3 z0 z1 z2 z3
		sz16	00 00 x3 y0	01	y1 y2 y3 z0 z1 z2 z3 t0
		sz32	x3 y0 y1 y2	11	y3 z0 z1 z2 z3 t0 t1 t2

66.14.1.8.2 Burst DMA in Write Mode

For every write access to the MD, the new FIFO state depends on the MDA status and the access size.

The FIFO is considered as a stack of 36 bytes: Data is stored in the FIFO according to the internal access size and the former MDA value. The following table shows the FIFO byte alignment strategy corresponding to MDA, as well as the new FIFO state for any access size of an internal write to MD.

Table 66-24. FIFO Write Configuration

Before write		Internal write access size	Written data	After write	
MDA[1:0]	FIFO state			MDA[1:0]	FIFO state
00	tt uu vv ww ?? ?? ?? ?? ?? ?? ?? ?? and so on...	sz8	?? ?? ?? x0	01	tt uu vv ww x0 ?? ?? ?? ?? ?? ?? ??
		sz16	?? ?? x0 x1	10	tt uu vv ww x0 x1 ?? ?? ?? ?? ?? ??
		sz32	x0 x1 x2 x3	00	tt uu vv ww x0 x1 x2 x3 ?? ?? ?? ??
01	tt uu vv ww xx ?? ?? ?? ?? ?? ?? ?? and so on...	sz8	?? ?? ?? x0	10	tt uu vv ww xx x0 ?? ?? ?? ?? ?? ??
		sz16	?? ?? x0 x1	11	tt uu vv ww xx x0 x1 ?? ?? ?? ?? ??
		sz32	x0 x1 x2 x3	01	tt uu vv ww xx x0 x1 x2 x3 ?? ?? ??
10	tt uu vv ww xx yy ?? ?? ?? ?? ?? ?? and so on...	sz8	?? ?? ?? x0	11	tt uu vv ww xx yy x0 ?? ?? ?? ?? ??
		sz16	?? ?? x0 x1	00	tt uu vv ww xx yy x0 x1 ?? ?? ?? ??
		sz32	x0 x1 x2 x3	10	tt uu vv ww xx yy x0 x1 x2 x3 ?? ??

Table continues on the next page...

Table 66-24. FIFO Write Configuration (continued)

Before write		Internal write access size	Written data	After write	
MDA[1:0]	FIFO state			MDA[1:0]	FIFO state
11	tt uu vv ww xx yy zz ?? ?? ?? ?? ?? and so on...	sz8	?? ?? ?? x0	00	tt uu vv ww xx yy zz x0 ?? ?? ?? ??
		sz16	?? ?? x0 x1	01	tt uu vv ww xx yy zz x0 x1 ?? ?? ??
		sz32	x0 x1 x2 x3	11	tt uu vv ww xx yy zz x0 x1 x2 x3 ??

NOTE

If the FIFO mode changes from a write to a read mode, all remaining written bytes in MD are lost but no error is returned. Typically, this happens if an ldf MD is executed after stf MD instructions. Before a mode change, it is recommended to force the flush of a potential remaining byte by a stfMD|SZ0|FL instruction. In the same way, if a FIFO mode changes from a read to a write mode, all prefetched data present in the FIFO is lost and no error is returned.

66.14.1.8.3 Endianness-Burst DMA Unit

Big and Little Endian are supported by the Burst DMA, but data is always stored in MD in Big Endian.

Byte manipulation is performed when data is exchanged with an Burst controller (for example, during read or write burst accesses).

66.14.1.9 Burst DMA Unit Copy Mode

A mechanism is available to perform fast ARM-to-ARM transfers.

Data does not flow through the SDMA core: It is kept in the DMA FIFO. This mechanism is selected when writing MD with a special option in the instruction code (copy flag).

It is possible to transfer up to eight words in one SDMA instruction (this does not mean in one cycle). In this mode, every time an stf MD|CPY is executed, a read burst is executed and directly followed by a write burst transfer. Burst transfers are limited to eight words. The size of the transfer (in words)-given by the SDMA general register (4 LSB)-is also limited to eight. The following SDMA code shows how 100 bytes could be copied from the MSA address to the MDA address. This is sample code only.

Burst DMA copy mode example

```
          ldi r0,@src
          stf r0,MSA                          // Source address setup
          ldi r1,@dst
          stf r1,MSA                          // Destination address setup
          ldi r0,0x64                         // data transfer counter
          ldi r1,0x8
MAIN_XFER:
          cmphs r0,r1                         // Is r0 >= 0x8
          bf LAST_XFER                      // If not, jump to last transfer label
          stf r1,MD|CPY                        // Copy 8 words from MSA to MDA address.
          subi r0,0x8                         // Decrement counter
          jmp MAIN_XFER                     // return to main transfer loop
LAST_XFER:
          stf r0,MD|CPY
```

The main transfer loop is executed 12 times; then r0 equals 4 and the last transfer loop is run.

In this mode, an acknowledge is transmitted to the core as soon as the read burst can start; thus, a first copy instruction returns an immediate acknowledge and subsequent copy instructions will be acknowledged as soon as the previous copy has finished.

66.14.1.10 Burst DMA Unit Error Management

Another point to consider is the management of errors.

Because the DMA immediately sends an acknowledge to the RISC core (except for the stf MS|SZ0|FLS instruction), it assumes no error will occur. If an error occurs, it is flagged (transfer error acknowledge) for the following DMA access.

This should not be a problem if the DMA is used properly. The MD accesses are meant to stall the SDMA as little as possible to optimize throughput and hide calculation time. Therefore, final access to MS should be performed before closing a channel. This access waits until any pending operation is finished in the burst DMA and gather any remaining error.

In copy mode, an error could be immediately returned to the SDMA on execution of the ldf copy or stf copy instruction. It happens when MSA or MDA are not word addresses (for example, 0[4]). This is because copy mode must only be used for transferring a large packet of aligned data.

When an error is received during a *read* transfer to the external bus, which may occur during the burst accesses, the MD FIFO contains the valid beats of the burst, and the error flag of MS is set to 2'b11 (error read burst). It is possible to read MS ("n" field) to know how much valid data remains in MD and when MD is empty (after ldf instructions). The next read MD instruction sets the MS error flag to 2'b10 (error mode), and an error is sent back to the SDMA core. In error mode, it is possible to read MSA, which gives the address of the error data. Any attempt to read or write MD, or to modify MDA or MSA in error mode, gives rise to an error; therefore, an error flag must be reset by clearing MS at the end of the SDMA code section responsible for error management.

In "error read burst" mode, writing MDA, MSA, or MD, or starting a copy transfer by a stf MD|COPY instruction will cancel the error mode. The following table shows when an immediate error is sent back according to the executed instruction.

Table 66-25. Possibilities in ERROR READ BURST Mode

DMA Instruction	Immediate Error	Comments
stf rn, MD stf rn, MSA (|U |PF) stf rn, MDA stf rn,MD|COPY	NO	Error mode is reset. MSA, MDA, or MD are updated and a DMA cycle may start. For the stf MD|COPY, a copy loop is executed.
stf rn, MS	NO	MS is updated.
ldf rn, MS ldf rn, MSA ldf rn, MDA	NO	MS, MSA, and MDA could be read in ERROR READ mode without any side effects (for example, no DMA cycle is triggered).
ldf rn, MD	YES/NO	Immediate error if there is no more data available for read in the FIFO.

When an error is received during a *write* transfer, the error is reported to the next DMA access. In this case, an error is sent to the SDMA core and the DMA goes to its error mode. Reading MS gives the number of bytes that remain in MD; reading MDA gives the address of the error data. Any attempt to read or write MD, or to modify MDA or MSA in error mode, give rise to an error; therefore, an error flag must be reset by clearing MS at the end of the SDMA code section responsible for error management.

Table 66-26. Possibilities in ERROR Mode

DMA Instruction	Immediate Error	Comments
stf rn, MD stf rn, MSA stf rn, MDA	Yes	Any attempt to modify MD, MSA, MDA will raise an immediate error and burst DMA remains in error mode. When address registers are write-accessed, an error is returned.
stf rn, MS	No	This is the only way to exit error mode. MS[9:8] must be reset by an stf MS|SZ0 instruction.
ldf rn, MS ldf rn, MSA ldf rn, MDA	No	MS, MSA, and MDA could be read in error mode without any side effects (for example, no DMA cycle is triggered).
ldf rn, MD	Yes	Whatever the DMA direction (read or write), an ldf rn triggers an immediate error.

66.14.1.11 Conditional Yielding-Burst DMA Unit

The standard SDMA transfer is based upon a hardware loop that has the following structure:

Hardware Loop

```
loop
load Rn,source            // can be ldf or ld
<computation>            // can be done through functional units
store Rn,dest              // can be st or stf
done 0               // yield
```

This structure needs to be kept independent of the functional units' particularities regarding the context switch. However, there can be variations in the context switch's efficiency, which can depend on the number of data received up to that point, and on the data itself.

The DMA, with its 8-word burst capability, has a preferable context switch period when its address register is 8-word aligned: It is the only moment that occurs once every eight loops when the succession of bursts is not broken by the context switch. When this is not the case, a context switch requires the storing (or loading) of less than eight words, which requires separate accesses and is far less efficient. The rest of the 8-word packet is stored (or loaded) after the context restore, and this is done as separate accesses.

The proposed solution is a conditional yielding, which occurs only when the DMA is in an optimum state. It does not require any modification to the scripts. The condition is decided at the DMA level.

The DMA can be programmed in two modes-conditional or always-true-for every channel, which provides complete flexibility. By default, the DMA is not in conditional mode.

The DMA condition is computed from the FIFO fill level and the various modes, as follows:

- When copy mode is selected, regardless of the transfer direction ('read' or 'write'), the condition is always true.
- In read mode, the condition is always true.
- In write mode, the condition is true when there are four bytes or less in the FIFO; it is false when there are more than four bytes. The 4-byte limit comes from the possibility of saving those bytes as MD with absolutely no impact on the bus accesses.

The aim at conditional yielding is to avoid splitting bus accesses (especially bursts).

66.14.2 Peripheral DMA Unit Programming

The peripheral DMA unit is connected to the Multi-Layer DMA Crossbar Switch of the ARM platform. Its goal is to perform data transfers between any blocks connected to the DMA bus of this platform.

These blocks are either peripherals or memories. The peripheral DMA could be seen as the ARM platform DMA controller.

The DMA performs data transfers in three modes:

- Read mode, where data is read from peripherals or from memory connected to the ARM platform and copied in a SDMA general register.
- Write mode, where data of a general register has to be written in a peripheral or a memory.
- Copy mode, where data is read from a peripheral (or memory) at a source address (PSA) and automatically written to a peripheral (or memory) at a destination address (PDA).

In copy mode, no SDMA general register is involved as transferred data only goes through the data register of the DMA.

The peripheral DMA has three addressing modes: frozen, incremented, and decremented, as follows:

- Frozen mode-When source or destination addresses are frozen, their value is not modified after a transfer. This mode is typically used for addressing peripheral FIFOs located at a fixed address.

- Incremented mode-When source or destination addresses are in incremented mode, after every transfer they are incremented by the number of bytes transferred.
- Decremented mode-In decremented mode, addresses are decremented by the number of bytes transferred.

The peripheral DMA registers are as follows:

- Two, 32-bit address registers (PSA and PDA) that respectively contain the source address for a read access and the destination address for a write access
- A 32-bit status register (PS) that contains information on the peripheral DMA configuration, such as the number of valid bytes in the data register, the error flag, the source and destination address mode, and so on.
- A 32-bit data register (PD) that stores data involved in a data transfer

66.14.2.1 Peripheral Source Address Register (PSA)

The source address register contains a pointer to a source peripheral or a memory associated with the next read data transfer. It has byte granularity.

It is based on the following:

- A 32-bit register (PSA) to store the address value
- A 2-bit register (stype) to store the source address mode (frozen, incremented, or decremented)
- A 2-bit register (ssize) to store the source target data path size (byte, half-word, or word)

Reading the register with the ldf instruction has no side effects and gives the address value of the next data that will be read by the SDMA during an ldf MD instruction. Writing the source address register may have side effects. If there is valid write data in the data register and the source address is changed, the write data is discarded. If the prefetch bit is set, a DMA read cycle is issued with the new address.

When PSA is to be written, you must specify the source target address mode, providing its size (byte, half-word, or word). This enables omission of the size field in all ldf MD instructions. When DMA performs a read cycle, its size is given by the value of the PSA source size register (ssize). If source is a memory in incremented mode, first programmed in word mode (stf PSA|SZ32|I), and if an SDMA script needs to read bytes from this memory, the size of the source target must be updated before executing new accesses. The source address mode and its size are given by labels added to the stf PSA instruction as described in the write section. The ssize and stype registers are part of the DMA status register (PS).

Writing to PSA may issue an immediate error if the source size is not compatible with the value to be written into the PSA register. For instance, writing a 2 in PSA and specifying that it is memory-accessed in word mode creates an immediate error.

66.14.2.2 Peripheral Destination Address Register (PDA)

The destination address register contains a pointer to a source peripheral or a memory associated with the next write data transfer. It has byte granularity.

It is based on the following:

- A 32-bit register (PDA) to store the address value
- A 2-bit register (dtype) to store the destination address mode (frozen, incremented, or decremented)
- A 2-bit register (dsize) to store the destination target data path size (byte, half-word, or word)

Reading the register with the ldf instruction has no side effects, and gives the address value of the next data that will be written by SDMA during an stfMD instruction. Writing the destination register has no side effect. Similar to the PSA register, the destination address mode and source are specified in the stf PDA instruction and may also generate an error in case of incorrect programming.

66.14.2.3 Peripheral Data Register (PD)

The data register of the peripheral DMA is a 32-bit register. When the destination address is correctly set up, any writing to PD will automatically flush the new input data.

The number of SDMA bytes that will be transferred is given by the PDA size register. Unlike other SDMA DMAs, PD is not a FIFO: It is not used to accumulate bytes that from the SDMA and must be packed before being sent to external memories. In read mode, and if the source address is correctly set up, an ldf instruction will empty PD. If a prefetch is required along with the instruction, the DMA will initiate a new read transfer.

Reading PD in prefetch mode only stalls the SDMA when the prefetched data is not yet available. Writing PD only stalls the SDMA if the previous write operation was not completed. As soon as the previous operation is over, the acknowledge is sent back to the SDMA RISC engine.

An error flag-part of PS-is set when an external access fails. The error is thus reported to the next SDMA instruction that involves the peripheral DMA.

66.14.2.4 Peripheral State Register (PS)

The state register contains the DMA state-machine value. It can be accessed in case of an error received during a transfer.

Although all PS fields can be written by an stf instruction, it is recommended to access only the error bit (to reset it). Modifying other PS fields will provide an un-guaranteed DMA behavior.

The initialization value of PS is 0, and it consists of the following structure:

Table 66-27. PS Structure

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	ssize		stype		dsize		dtype	
W																
R	0	0	0	0	0	d	e		0	0	0	0	0	n		
W																

Table 66-28. PS Field Descriptions

Field	Description
31-24	Reserved
23-22 ssize	Source Target Size. Determines the size of the read transfers on the external bus. It should match the accessed device characteristics. 00 *reserved* 01 Byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)
21-20 stype	Source address Mode. Determines whether PSA is incremented, decremented, or kept unmodified after every read from the external bus. 00 Frozen Mode 01 Incremented Mode 10 Decremented Mode 11 *reserved*
19-18 dsize	Destination Target Size. Determines the size of the write transfers on the external bus. It should match the accessed device characteristics. 00 *reserved* 01 Byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)

Table continues on the next page...

Table 66-28. PS Field Descriptions (continued)

Field	Description
17-16 dtype	Destination address Mode. Determines whether PDA is incremented, decremented, or kept unmodified after every write on the external bus. 00 Frozen Mode 01 Incremented Mode 10 Decremented Mode 11 *reserved*
15-11	Reserved
10 d	Direction Flag or DMA Mode. DMA is in write mode when data was written into PD by stf PD instructions, or if a previous DMA cycle on the external bus was a write access. Writing PDA or PSA does not change the DMA mode. DMA is in read mode when a previous DMA cycle was a read access, and DMA stays in read mode when data is read by the SDMA with an ldf PD instruction. Reading PDA or PSA does not change the DMA mode. 0 Read Mode 1 Write Mode
9-8 e	Error. Indicates if the previous access was acknowledged with a bus error. 00 No error was received. 01 *reserved* 10 Error mode 11 Error read
7-3	Reserved
2-0 n	number of bytes in PD

NOTE

dtype, dsize, stype, and ssize are updated when PSA and PDA are written.

66.14.2.5 Peripheral DMA Write (stf)-Write Mode

When written by an stf instruction, the function code bits are interpreted as follows:

Table 66-29. STF Code Bits

Register	7	6	5	4	3	2	1	0
PSA	s		p	ar	am		sz	
PDA								
PD			pdsel					
PS			pssel					

Table 66-30. STF Code Bits Field Descriptions

Field	Description
7-6 s	Functional Unit selector 11 for Peripheral DMA
5 p (PSA)	Prefetch Flag 0 no prefetch 1 automatic prefetch
4 ar (PSA/PDA)	Address Register Selector 0 PSA 1 PDA
3-2 am (PSA/PDA)	Address Mode. Determines how PSA or PDA is modified after every read or write access to the PD. 00 Frozen-Address registers are not modified after the transfer. 01 Incremented-Address registers are incremented by the number of transferred bytes. 10 Decremented-Address registers are decremented by the number of transferred bytes. 11 Updated-PSA and PDA are not modified. Either address mode is not modified, but the width of the data path is updated by the sz field.
1-0 sz	Transfer Size 00 *reserved* 01 byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)
5-0 pdsel	PD access selector 001000 is the only valid option
5-0 pssel	PS access selector 111111 writes to PS 001100 only clears the error flag in PS

Due to the large number of possible stf instructions, the following table provides only a short list of all the possible write instructions:

Table 66-31. Peripheral DMA STF Instruction List

Binary	Assembly	Comments
11_00_00_01 11_00_00_10 11_00_00_11	stf Rn, PSA|SZ8 |F stf Rn, PSA|SZ16|F stf Rn, PSA|SZ32|F	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is frozen.
11_10_00_01 11_10_00_10 11_10_00_11	stf Rn,PSA|SZ8 |F|PF stf Rn,PSA |SZ16|F|PF stf Rn,PSA |SZ32|F|PF	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • 1, 2, or 4 bytes are *fetched* from the peripheral source. • Source address is frozen.

Table continues on the next page...

Table 66-31. Peripheral DMA STF Instruction List (continued)

Binary	Assembly	Comments
11_00_01_01 11_00_01_10 11_00_01_11	stf Rn, PSA|SZ8 |I stf Rn, PSA|SZ16|I stf Rn, PSA|SZ32|I	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is in incremented mode: PSA = PSA + 1,2 or 4 after read PD.
11_10_01_01 11_10_01_10 11_10_01_11	stf Rn, PSA|SZ8 |I|PF stf Rn, PSA|SZ16|I|PF stf Rn, PSA|SZ32|I|PF	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is in incremented mode: PSA = PSA + 1, 2, or 4 after read PD. • 1, 2, or 4 bytes are *fetched* from the peripheral source.
11_00_10_01 11_00_10_10 11_00_10_11	stf Rn, PSA|SZ8 |D stf Rn, PSA|SZ16|D stf Rn, PSA|SZ32|D	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is in incremented mode: PSA = PSA-1,2, or 4 after read PD.
11_10_10_01 11_10_10_10 11_10_10_11	stf Rn, PSA|SZ8 |D|PF stf Rn, PSA|SZ16|D|PF stf Rn, PSA|SZ32|D|PF	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is in incremented mode: PSA = PSA-1,2, or 4 after read PD. • 1, 2, or 4 bytes are *fetched* from the peripheral source.
11_00_11_01 11_00_11_10 11_00_11_11	stf Rn, PSA|SZ8 |U stf Rn, PSA|SZ16 |U stf Rn, PSA|SZ32 |U	• *Update* source pointer to memory, which becomes a pointer to a memory accessed in byte, half-word, or word. • PSA value is not modified by Rn. • Bytes present in PD are lost.
11_10_11_01 11_10_11_10 11_10_11_11	stf Rn, PSA|SZ8 |PF|U stf Rn, PSA|SZ16 |PF|U stf Rn, PSA|SZ32 |PF|U	• *Update* source pointer, which becomes a pointer to a target accessed in byte, half-word, or word. • PSA value is not modified by Rn. • Bytes present in PD are lost. • 1, 2, or 4 bytes are *fetched* from the memory source.
11_01_00_01 11_01_00_10 11_01_00_11	stf Rn, PDA|SZ8 |F stf Rn, PDA|SZ16|F stf Rn, PDA|SZ32|F	• Destination is a byte, half-word, or word target at the Rn address, and any further PD write instructions will trigger byte, half-word, or word access to the destination. • Destination address is frozen.
11_01_01_01 11_01_01_10 11_01_01_11	stf Rn, PDA|SZ8 |I stf Rn, PDA|SZ16|I stf Rn, PDA|SZ32|I	• Destination is a byte, half-word, or word target at the Rn address, and any further PD write instructions will trigger byte, half-word, or word access to the destination. • Destination address is in incremented mode: PDA = PDA + 1, 2, or 4 after write PD.
11_01_10_01 11_01_10_10 11_01_10_11	stf Rn, PDA|SZ8 |D stf Rn, PDA|SZ16|D stf Rn, PDA|SZ32|D	• Destination is a byte, half-word, or word target at the Rn address, and any further PD write instructions will trigger byte, half-word, or word access to the destination. • Destination address is in incremented mode: PDA = PDA-1, 2, or 4 after write PD.
11_01_11_01 11_01_11_10 11_01_11_11	stf Rn, PDA|SZ8 |U stf Rn, PDA|SZ16 |U stf Rn, PDA|SZ32 |U	• *Update* destination pointer to memory, which becomes a pointer to a memory accessed in byte, half-word, or word. • PDA value is not modified by Rn • bytes present in PD are lost

Table continues on the next page...

Table 66-31. Peripheral DMA STF Instruction List (continued)

Binary	Assembly	Comments
11_00_10_00	stf Rn, PD	• Write "dsize" bytes of Rn in PD and automatically flush to destination target
11_11_11_11	stf Rn, PS	• Write status register
11_00_11_00	stf Rn,clrefPS	• Clear error flag if set

NOTE

When writing PD, size information is not important: It is embedded in the dsize field of PDA register. If dsize is 1, 2, or 4, then one, two, or four bytes from Rn is written to the PD register, and automatically flushed out to the destination target.

66.14.2.6 Peripheral DMA Read (ldf)-Read Mode

When received from an ldf instruction, the function code bits are interpreted as follows.

Table 66-32. LDF Code Bits

Register	7	6	5	4	3	2	1	0
PSA	s			ar	a			
PDA								
PD			p	cpy				
PS			pssel					

Table 66-33. LDF Code Bits Descriptions

Field	Description
7-6 s	Functional Unit selector 11 for Peripheral DMA
5 p (PD)	Prefetch Flag 0 no prefetch 1 automatic prefetch
4 ar (PSA/PDA)	Address Register Selector 0 PSA 1 PDA
4 cpy (PD)	Copy Mode 0 standard access 1 copy mode access

Table continues on the next page...

Table 66-33. LDF Code Bits Descriptions (continued)

Field	Description
3 a	Register Set selection 0 PSA or PDA 1 PD or PS
5-0 pssel	PS access selector 111111 is the only valid option to read PS

Table 66-34. Peripheral DMA LDF Instruction List

Binary	Assembly	Comments
11_0_0_0_000	ldf Rn, PSA	Reads 32-bit of PSA value
11_0_1_0_000	ldf Rn, PDA	Reads 32-bit of PDA value
11_0_0_1_000	ldf Rn, PD	Reads programmed source size bytes of PD (0-extended)
11_1_0_1_000	ldf Rn, PD|PF	Reads programmed source size bytes of PD (0-extended), and starts a prefetch at PSA address.
11_0_1_1_000	ldf Rn, PD|COPY	Starts a copy transfer from the source target at the PSA address to the destination target at the PDA address. No data transmits through Rn, but Rn contents are lost (Rn is loaded with PD temporary contents that are *not* the copied data).
11_111111	ldf Rn, PS	Reads 32-bit of PS value

NOTE

When reading PD, size information is not important: It is embedded in the ssize field of the PSA register. If ssize is 1, 2, or 4, the one, two, or four bytes is transferred from PD to Rn. Read data is 0-extended.

66.14.2.7 Peripheral DMA Unit Copy Mode

Like burst DMA, the peripheral DMA unit has a copy mode that is used when data transfers do not involve SDMA general registers.

Data is read from the source target at a PSA address, stored in PD, and then automatically flushed to the destination target at the PDA address. Copy mode is only available for transfers that involve two targets of the same data path width.

Since copy mode is invoked with an ldf instruction, the *loaded* general purpose register loses its previous contents. (However, the new contents are unpredictable as they depend on temporary values that are seen on the external DMA bus.)

66.14.2.8 Error Management

Peripheral DMA generates two kinds of errors: the immediate error that sanctioned incorrect register programming; and the error triggered by the previous access and stored in the error flag of PS until a DMA instruction is executed.

66.14.2.8.1 Immediate Errors

The following table lists all incorrect DMA register setups.

Table 66-35. Immediate Errors with Peripheral DMA

Rn[1:0] values	DMA instruction	Comments
0x01 0x11	stf Rn, PSA|SZ16|F stf Rn, PSA|SZ16|I stf Rn, PDA|SZ16|F stf Rn, PDA|SZ16|I	If PSA points to a half-word peripheral or to a half-word address in memory, its value must be 0 modulo 2.
0x01 0x10 0x11	stf Rn, PSA|SZ32|F stf Rn, PSA|SZ32|I stf Rn, PDA|SZ32|F stf Rn, PDA|SZ32|I	If PSA points to a word peripheral or to a word address in memory, its value must be 0 modulo 4.
PSA[1:0]-PDA[1:0]	DMA instruction	Comments
0x01 0x10 0x11	stf Rn, PSA|SZ32|U stf Rn, PDA|SZ32|U	When PDA or PSA is updated and becomes a pointer to a word address in memory, its content must be 0 modulo 4.
0x01 0x11	stf Rn, PSA|SZ16|U stf Rn, PDA|SZ16|U	When PDA or PSA is updated and becomes a pointer to a half-word address in memory, its content must be 0 modulo 2.
Read/Write PD instruction	Comments	
stf Rn,PD ldf Rn,PD	If PDA size (dsize) has never been set up before an stf PD instruction (dsize=0) If PSA size (ssize) has never been set up before an ldf PD instruction (ssize=0)	
ldf Rn,PD|CPY	Copy mode is possible only between two targets whose data path width is identical. It is P8↔P8, P16↔P16, or P32↔P32 regardless of the way the address registers are incremented.	

66.14.2.8.2 Data Transfer Errors

When PSA and PDA are correctly set up, the only error that may arise for an ldf PD or stf PD instruction would be the error of the previous DMA cycle.

Error handling is driven by a single consideration: When an error occurred during a data read on the DMA interface, this error should appear as a transfer error to the core when the core attempts to retrieve the data that was not successfully read from the accessed device (memory or peripheral).

When an error occurred during a write access to the DMA interface, the data is still available in PD and should not be destroyed by subsequent core accesses: The core must be warned about the error issue.

There are three error handling mechanisms for each case: Read Error (First Phase), Write Error and Read Error (Second Phase), and Copy Mode Errors handling.

66.14.2.8.3 Read Error (First Phase)

If an error occurred during a prefetch command, the peripheral DMA enters its ERROR READ mode (PS[9:8]=11). In this mode, the error is reported on the next ldf PD instruction and writing PSA, PDA, or PD will cancel the error flag.

The block returns no error mode and instructions are normally executed (a DMA cycle may be triggered). Similarly, initiating a copy transfer will reset the error flag and start a copy transfer. The following table details which instructions can be executed in this mode.

Table 66-36. Possibilities in ERROR READ Mode

DMA Instruction	Immediate Error	Comments
stf rn, PD stf rn, PSA (IU IPF) stf rn, PDA ldf rn,PD|COPY	NO	Error mode is reset, PSA or PDA are updated, or a write cycle is started. For the ldf PD|COPY, a copy loop is executed.
stf rn, PS	NO	PS is updated.
ldf rn, PS ldf rn, PSA ldf rn, PDA	NO	PS, PSA, and PDA could be read in ERROR READ mode without any side effects (for example, no DMA cycle is triggered).
ldf rn, PD	YES	Error of the previous read access is reported here and the peripheral DMA enters its ERROR mode.

66.14.2.8.4 Write Error and Read Error (Second Phase)

The peripheral DMA enters its ERROR mode (PS[9:8]=10) when the previous DMA write cycle failed, or, as explained in Read Error (First Phase), when an ldf PD is executed while the block is in ERROR READ mode. When a DMA cycle failed, address registers (PSA, PDA) are not modified and continue to point to the problematic address. In ERROR mode, stf instructions may raise an immediate error, and ldf instructions will not (as detailed in the table below).

Table 66-37. Possibilities in ERROR Mode

DMA Instruction	Immediate Error	Comments
stf rn, PD stf rn, PSA stf rn, PDA	YES	Any attempt to modify PD, PSA, or PDA will raise an immediate error, and the peripheral DMA stays in ERROR mode. When address registers are write accessed, an error is returned.
stf rn, PS	NO	This is the only way to exit the ERROR mode. PS[3] must be reset by an stf PS instruction.
ldf rn, PS ldf rn, PSA ldf rn, PDA	NO	PS, PSA, and PDA could be read in ERROR mode without any side effects (for example, no DMA cycle is triggered).
ldf rn, PD	YES	Whatever the DMA direction (read or write), an ldf rn, PD instruction will show an immediate error.

66.14.2.8.5 Copy Mode Errors

Because copy mode is a write access that follows a read access, there are two possible cases of bus error.

When the read access incurs a bus error, the peripheral DMA behaves exactly as described in Read Error (First Phase) and Write Error and Read Error (Second Phase) : It enters its ERROR READ mode, and so on.

When the error occurred during the write access of the copy transfer, the DMA enables the core to retrieve the data that was read because it is assumed the read from the peripheral removed the data from its source device. Therefore, the data to be flushed is still in PD. Any subsequent access to PD triggers an error to the core, which should execute its error handling procedure.

Once the ERROR mode is left (after writing to PS), it is possible for the core to retrieve the data in PD with an ldf instruction or try to flush PD contents once again (for example, when the error was due to a full FIFO and the script waited for the FIFO to be emptied) with another ldf instruction in copy mode. This latter instruction detects that there is valid data in PD, tries to flush it, and thus skips the read phase of the copy instruction. This is a different behavior from the usual stf PD instruction that overwrites PD with the selected General Purpose register contents. The same mechanism can be used any time PD holds data that is not written because of a bus error on the DMA interface; when the data was written via a copy instruction, or via the usual stf PD instruction.

66.14.2.8.6 Error Check Example

The following code illustrates an example checking for both immediate and data transfer errors on a store to the PD register. The first bdf instruction checks for an immediate error, but if a data transfer error occurred it is reported until the next instruction to access the Peripheral DMA. A second check of the error flags is done after the ldf PS

instruction. The value of PS here can be ignored. The act of reading any register in Peripheral DMA while it is in an error mode that returns the error to the core to set either the SF or DF flag. Any error returned on an ldf command sets the SF flag and any error returned on an stf instruction sets the DF flag. This can create a situation as shown in the example where a bus error during a DMA write which would normally be considered as a destination fault is reported as a source fault because the error was reported to the SDMA core during an ldf instruction.

Peripheral DMA Error Check

```
    clrf    0            // Clear SF and DF flags
 stf    R4, PD        // Write data to memory
bdf    error_routine    // Check for immediate error from write to PD.
ldf    r3, PS        // Read PS (PS value in R3 can be ignored)
    bsf    error_routine    // Check for bus error from "stf R4,PD"
            // SF is set because it is a ldf instruction, even though
            // the original error was a destination fault
```

66.14.2.9 Peripheral DMA Unit Prefetch/Flush Management

There is no flush bit because every time data is stored in PD by a stf PD instruction-assuming PDA is correctly programmed-it is automatically flushed to the destination.

An acknowledge is returned in the cycle of the DMA instruction, and the SDMA is only stalled by an instruction that addresses the peripheral DMA when the previous DMA access is not over.

66.14.3 CRC Unit

The Cyclic Redundancy Check (CRC) unit is connected to the SDMA core via the FUBUS. This unit can perform a CRC calculation for a set of given polynomials from degree 8 to 32.

When all CRC unit registers are loaded, the unit can process one byte of data every clock cycle. When loading new data to compute the CRC, the SDMA can perform 32-bit, 16-bit, or 8-bit accesses.

Two 32-bit registers comprise the unit:

- The CRC algorithm CA that describes the polynomial
- The CRC checksum CS to accumulate the data after each processing

66.14.3.1 Polynomial Register (CA)

This register defines the CRC algorithm currently used in the calculation, and the management of data ordering to/from the data register CS.

Before starting any CRC calculation, it must be loaded with the chosen polynomial reference number and data ordering mode as indicated in the following figure and table.

Table 66-38. CA Structure

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
R	0	0	0	0	0	0	0	0	0	0	0	ri	ro	p		
W																

Table 66-39. CA Descriptions

Field	Description
31-5	Reserved
4 ri	Reverse bit order of data in 0 Must be used when the peripheral receives bytes as bit streams in LSB-first order (UART case). 1 Selects the reverse mode, which must be used when the peripheral receives bytes as bit streams in MSB-first order (MMC case).
3 ro	Reverse bit order of data out 0 Must be used when the peripheral transmits bytes as bit streams in LSB-first order (UART case). 1 Selects the reverse mode, which must be used when the peripheral transmits bytes as bit streams in MSB-first order (MMC case).
2-0 p	Polynomial selection 000 CRC32 ($X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$) 001 CRC16 ($X^{16}+X^{15}+X^2+1$) 010 CCITT16 ($X^{16}+X^{12}+X^5+1$) 011 IS136 ($X^{12}+X^{10}+X^8+X^5+X^4+X^3+1$) 100 CRC10 ($X^{10}+X^9+X^5+X^4+X+1$) 101 CRC8 ($X^8+X^2+X+1$) 110 Parity ($X^8+1$) 111 Reserved

66.14.3.2 Accumulator Register (CS)

The accumulator register accumulates the division remainder during the CRC processing.

When writing the accumulator register (process bit is not set) if the *ro* bit is set, the write data has its order reversed whatever the data length (the length is specified by the selected polynomial).

When computing new CRC (writing CS and process bit set) if the *ri* bit is set in CA, any byte of write data will have its bit order reversed before storage and calculation. In the first byte, bit 7 is replaced by bit 0, bit 6 is replaced by bit 1, and so on. In the second byte, bit 7 is replaced by bit 0, bit 6 is replaced by bit 1, and so on, which means in the write data, bit 15 is replaced by bit 8, bit 14 will replaced by bit 9, and so on.

If the *ro* bit is set in CA, any read data will have its bit order reversed whatever the data length. If the valid data length is 16 bits, bit 15 is replaced by bit 0, bit 14 is replaced by bit 1, and so on.

When loading new data to compute the CRC, the SDMA can perform 32-bit, 16-bit, or 8-bit accesses. The CRC is computed in four clock cycles when performing a 32-bit access, in two clock cycles when performing a 16-bit access, and in one clock cycle when performing an 8-bit access. In all cases, an immediate acknowledge is sent back to the SDMA. A wait state is inserted if the SDMA tries to perform a new access before the end of the computation.

When performing a multi-bit access, the first byte used to compute the CRC is the most significant byte of the valid data.

66.14.3.3 Write Instruction (stf)

The following table shows the stf instructions.

Table 66-40. Stf Instructions for CRC

Byte Command								Description
1	0	0	0	0	0	0	0	Write polynomial encoded in the General Register (CA)
1	0	0	0	0	1	0	0	Write accumulator register (right-aligned)
1	0	0	1	0	1	0	0	Compute the CRC with the new incoming byte (b0)
1	0	0	1	0	1	0	1	Compute the CRC with the new incoming byte (b0)
1	0	0	1	0	1	1	0	Compute the CRC with the new incoming halfword (b1 b2) The bytes are used in the following order: b1 and b0.
1	0	0	1	0	1	1	1	Compute the CRC with the new incoming word (b3 b2 b1 b0) The bytes are used in the following order: b3, b2, b1, and b0.

66.14.3.4 Read Instruction (ldf)

The following table shows the ldf instructions.

Table 66-41. Ldf Instructions for CRC

Byte Command								Description
1	0	0	0	0	0	0	0	read the polynomial register encoded
1	0	0	0	0	1	0	0	read the accumulator register, right aligned

66.14.3.5 Operating Mode

The following is the operating mode example:

- Load the polynomial register to select the algorithm and preset the accumulator, if required.
- Data is right-aligned in the general register.
- Input data is fed byte-by-byte into the accumulator through stf instructions.
- When all data is fed into the CRC unit, the CRC checksum is read with ldf ca, sz.

16-bit CCITT CRC with $P(X) = X^{16}+X^{12}+X^{5}+1$

```
          .equ rL,0 // GReg[0] = rL; loop count register
.equ rA,1 // GReg[1] = rA; CCITT16 polynomial
.equ rP,2 // GReg[2] = rP; initial CRC value
.equ rT,3 // GReg[3] = rT; transferred byte register
.equ aT,4 // GReg[4] = aT; transferred byte address
// (MMC DAT_RX - 0xA003)
.equ CA,8 // CRC unit CA register
.equ CS,9 // CRC unit CS register
ldi rA,2 // init register with CCITT16 polynomial
stf rA,CA // updates the selected polynomial
ldi rP,0xff // initializes register with 0x000000ff
stf rP,CS,0 // presets the accumulator with 0x000000ff
ldi aT,0xA0 // initializes aT with 0xA003: aT = 0x000000A0
revblo aT // initializes aT with 0xA003: aT = 0x0000A000
ori aT,0x03 // initializes aT with 0xA003: aT = 0x0000A003
ldi rL,200 // expects to read 200 bytes from MMC DAT_RX
loop 2 // executes 200 times the next 2 instructions
```

```
ld rT,(aT,0) // loads next byte from MMC DAT_RX

stf rT,CS,1 // process checksum with new incoming byte

ldf rT,CS // read the final checksum at the end
```

66.14.4 OnCE and Real-Time Debug

The On-Chip Emulation block (OnCE) is the debug interface to the SDMA. It supports the access to all core internal devices (registers, memory, and so on), and provides a set of mechanisms that control the core.

The OnCE is accessed by JTAG ports at the chip's board level, or by the host via its peripheral bus.

To reduce the size of the hardware material involved, all tasks supported by the OnCE are performed on the SDMA core. The architecture of the SDMA OnCE is relatively simple and very flexible.

The commands supported by the SDMA OnCE are listed in the following sections.

66.14.4.1 Memory and Register Access

A set of mechanisms is provided to access SDMA memory and register locations. Both reading and writing are allowed. The access is supported if the processor is in debug mode.

Those registers can also be accessed through the ARM platform Control interface when the OnCE is controlled by the ARM platform, as described in the "Using BP" section.

66.14.4.2 Hardware Breakpoints

An event detection unit is implemented to support memory breakpoints. The unit watches the data exchanged between the SDMA memory bus and the core.

A debug request is sent to the core when matching conditions occur. The unit supports mixed conditions based on address range, access type, and data value. Event detection unit configuration registers are memory mapped in the SDMA space (see ARM platform Channel 0 Pointer (SDMAARM_MC0PTR)): You can modify them through a regular memory access or the ARM platform control interface.

66.14.4.3 Watchpoints

One output pin is provided to monitor matching trigger conditions that are defined in the event detection unit.

66.14.4.4 Software Breakpoints

The SDMA instruction set contains a software breakpoint. Upon executing a software breakpoint instruction, the core suspends normal execution and enters debug mode.

No hardware step execution mode is implemented in the OnCE, but this feature may be implemented at the software level with this instruction.

66.14.4.5 Core Control

Commands are provided to monitor and control processor activity. You can halt the core, rerun the core from another address location, and get processor status.

Any hardware breakpoint on the instruction bus is not supported, but this feature may be implemented by inserting a software breakpoints program.

66.15 The OnCE Controller

The OnCE controller receives commands from the ARM platform or from the JTAG controller. Each command is interpreted before being sent to the core.

66.15.1 OnCE Commands

A small set of commands supports the communication between the OnCE and the external world. This command set enables you to perform any of the following tasks: control processor activity, save core context, and execute an SDMA instruction from the OnCE.

Combined together, these tasks perform more complex commands.

A full OnCE command contains a 4-bit instruction (the OnCE command opcode) and a variable length data field (the OnCE data). During command execution, the OnCE data is transferred in a OnCE internal register before being exchanged with the SDMA. Some

data values are also exported. This mechanism creates a link between the processor and the external world. Nine commands are defined: The following table presents their formats.

Table 66-42. OnCE Command Opcode Values

Instruction Opcode	Name	Action	Register	Data Field Size	Mode
0000	rstatus	Reads the OnCE status register	STATUS	16-bit	normal/debug
0001	dmov	Updates general register GReg1	GREG1	32-bit	debug
0010	exec_once	Runs the instruction from the SDMA instruction register	INSTRUCTION	16-bit	debug
0011	run_core	Returns to normal execution	BYPASS	1-bit	debug
0100	exec_core	Returns to normal execution via a jump instruction that specifies the new address	INSTRUCTION	16-bit	debug
0101	debug_rqst	Stops the core after execution of current instruction	BYPASS	1-bit	normal
0110	rbuffer	Reads the real time buffer	RTB	32-bit	normal/debug
0111-1110	reserved	Reserved	BYPASS	1-bit	normal/debug
1111	bypass	Bypasses TARM platform controller	BYPASS	1-bit	normal/debug

Each instruction corresponds to a specific action performed on the OnCE. The nature of the associated data field is clearly identified. The dmov command is followed by a 32-bit data value (which is a data value for the SDMA); the exec_once and the exec_core commands are followed by a 16-bit data value (which is an instruction for the SDMA); the rstatus command is followed by a 16-bit control value (which is the content of the OnCE status register); the rbuffer command is followed by a 32-bit data value. The debug_rqst and the run_core commands are followed by a single bit data field (this is a bypass value). Finally, the bypass instruction enables the SDMA JTAG TAP controller to be daisy-chained with another JTAG TAP controller. This is a JTAG-only feature. The set of commands is simple, but enables you to perform any possible task on the SDMA during a debug process.

66.15.2 Sending Commands to the OnCE Controller

The JTAG access is the standard access to the OnCE, but sometimes the JTAG is not available to fix some bugs (if the chip is in production for instance), an additional access is then required. Therefore, one ARM platform access to the OnCE is provided.

66.15.2.1 Using the JTAG Interface

A serial access is performed through the five JTAG pins TCK, TRST, TMS, TDI, and TDO. A Test Access Port controller is provided to decode the TMS control signal.

It produces shift-enable signals (shift_ir and shift_dr), and updates enable signals (update_ir and update_dr). It is fully compliant with the IEEE 1149.1 testability (JTAG) standard.

During the shift_ir state, the command opcode is shifted into the OnCE controller (for example, the signal from the TDI pin is shifted into the command register and the TDO pin receives the signal shifted out). After transferring the four bits of the command, an update_ir signal is asserted and the command is decoded. The target data register is now clearly identified and the corresponding control signal is produced, as follows: bypass enable signal (bp_en), instruction enable signal (inst_en), data enable (data_en), and status enable signal (stat_en).

During the shift_dr state, the TDI signal is shifted into one of the following target registers: bypass register (1 bit), SDMA instruction register (16 bits), SDMA data register (32 bits), or OnCE status register (16 bits). The TDO pin is connected to the output of the selected register to receive the signals shifted out.

The JTAG access is disabled when the ARM platform access is enabled.

66.15.2.2 Using the ARM platform

The ARM platform access to the OnCE is not the standard access, but it is required if the JTAG is not available.

For example, if the SDMA ROM is out of use on a chip in production, and the ARM platform needs to download new code and restart the SDMA, the OnCE can easily perform this operation. This type of debug operation justifies the use of an ARM platform access to the OnCE.

To drive the OnCE, the ARM platform uses some registers contained in the ARM platform Control block of the SDMA. These registers are accessed through the ARM platform peripheral bus. Most of these registers are connected to another register in the OnCE controller. Thus, accessing one of these registers is equivalent to accessing the associated register in the OnCE controller.

The set of registers in the ARM platform Control block is listed below:

- ONCE_ENB register (1 bit, read/write)-This 1-bit register enables the ARM platform access to the OnCE. When this bit is set, the signals from the JTAG are ignored.

When it is cleared, all writing operations to the following registers through the Host Control interface are ignored. This register is reset on a JTAG reset.

- ONCE_CMD register (4 bits, read/write)-This 4-bit register receives the command opcode. It is connected to the command register in the controller. A write access to this register causes the associated command to be executed on the OnCE. For example, after writing "0001" in this register, a dmov command is executed.

NOTE

On the ARM platform side, the rstatus and bypass commands are not supported. This register is reset on a JTAG reset.

- ONCE_DATA register (32 bits, read/write)-This 32-bit register is connected to the SDMA data register. This register is used when executing a dmov or rbuffer command.

NOTE

Before requesting a dmov command, the 32-bit data to transfer must be written in the ONCE_DATA register. At the end of the execution, the register is updated with GReg1 former value. This register is reset on a JTAG reset.

- ONCE_INSTR register (16 bits, read/write)-This 16-bit register is connected to the SDMA instruction register. This register is used when executing an exec_core or an exec_once command.

NOTE

Before requesting an exec_core or an exec_once command, the appropriate instruction must be written in the ONCE_INSTR register. This register is reset on a JTAG reset.

- ONCE_STAT register (16 bits, read only)-A read access to the ONCE_STAT register returns the content of the OnCE status register (OSTAT). This register is read only.
- The bypass register is not useful when the ARM platform controls the OnCE, therefore no register is defined in the ARM platform Control block to access the bypass register.

66.15.2.3 Conflicts Between the JTAG and the ARM platform Accesses

When ARM platform access to the SDMA OnCE is enabled (that is, when the bit in the ONCE_ENB register is set), the JTAG access is disabled. This guarantees that the block is not accessed at the same time on both sides.

It is possible to check whether the JTAG access to the SDMA OnCE is enabled from the JTAG port. When the JTAG access is disabled, the SDMA TDO always returns 1. The check requires the following steps:

- Execute a dmov command from debug mode (with neither 0xffffffff nor 0x0 as dmov value: 0x5a5a5a5a is good).
- Execute another dmov command (the value here is not important).
- The returned value from the latter dmov command should be the original one if the JTAG access is enabled; if it is 0xffffffff instead of the original input value, this means the JTAG access is disabled.

66.15.3 Executing a Command from the OnCE

All the commands defined in OnCE Commands can be accessed through the JTAG. The ARM platform can access all these commands except the rstatus command.

On the ARM platform side, the OnCE status is directly accessed by reading the ONCE_STAT register.

66.15.3.1 Nature of the Commands

Two types of commands may be distinguished. First, there are two commands that do not interact with the core: rstatus and rbuffer. Those commands may be requested at any time: They do not depend on the core status.

NOTE

Each of these commands exports a data value or a status value from the SDMA.

There are also commands that interact with the core: dmov, run_core, exec_core, exec_once, and debug_rqst. These commands are core status dependent, as follows:

- During user mode only the debug_rqst is taken into account.
- During debug mode, all these commands are taken into account except the debug_rqst. For example, an exec_once command requested while not in debug mode has no effect.

66.15.3.2 Execution Request

The SDMA starts executing a task in debug mode when requested by the OnCE controller. The execution starting time depends on the type of access used to communicate with the OnCE.

If the JTAG is used, the request is send after decoding the update_dr state in the TAP controller. Therefore, always cross this state when sending a command through the JTAG. If the OnCE is driven from the ARM platform side, the request is sent after detecting a write access to the ONCE_CMD register. All the registers involved in this operation must be loaded first.

The following is an example of an exec_core command execution from the ARM platform side: After writing '010' in the ONCE_CMD register, the OnCE controller asks the SDMA to execute the instruction contained in the ONCE_INSTR register. The instruction involved should be available in the ONCE_INSTR register before the beginning of the execution.

66.15.3.3 Command Execution

The following list shows the commands and details how each command is executed:

- rstatus command execution-The rstatus command exports the content of the OnCE status register (OSR). If the JTAG is used, the status information is captured in the OnCE status register during the capture_dr state, and shifted out after 16 TCK clock cycles in the shift_dr state. The rstatus command is not supported on the ARM platform side, but a status register is provided instead. The rstatus may be performed in both debug and user modes.
- dmov command execution-The dmov command accesses SDMA internal registers. Executing a dmov instruction exchanges the 32-bit data values between the SDMA data register and the general register GReg[1].
- If the JTAG is used, the content of GReg1 is captured in the SDMA data register during the capture_dr state, then it is shifted out after 32 TCK clock cycles in the shift_dr state. During the update_dr state, GReg1 is updated with the new, shifted-in 32-bit data value. If the OnCE is driven from the ARM platform side, the data values contained in GReg1 and the SDMA data register are exchanged after detecting a write access to the ONCE_CMD register. The ONCE_DATA register must therefore be loaded first.
- exec_once command execution-The exec_once command executes the instruction loaded in the SDMA instruction register. The command may only be requested from debug mode. The SDMA returns to debug mode at the end of the execution.

- Change of flow instructions as well as instructions that may cause a context switch are not supported: The comprehensive list comprises done/yield/yiedge (except done 5), BF, BT, BSF, BDF, JMP, JSR, JMPR, JSRR, RET, and LOOP, as well as all the illegal instructions.

  No other command should be requested before the SDMA returns to debug mode. The SDMA status (for example, whether it is in debug mode or not) can be detected by polling with the rstatus OnCE command, monitoring the debug_mode pin, or checking the OnCE Status Register (SDMAARM_ONCE_STAT) register via the ARM platform control interface.

  NOTE

  Most of the instructions are single-cycle, which omits the step of polling the status. Loads and stores to DMA units are typical instructions that might require this polling.

  If the JTAG is used, the 16-bit instruction is shifted in the SDMA instruction register after 16 TCK clock cycles in the shift_dr state. A request is sent to the core when the update_dr state is decoded in the TAP controller. If the OnCE is driven from the ARM platform side, the request is sent to the SDMA when detecting a write access to the ONCE_CMD register. The ONCE_INSTR register must be therefore be loaded first.

- run_core command execution-The run_core command leaves debug mode and resume normal program execution. The next instruction executed is the last instruction decoded before entering debug mode. Be sure to restore core context before re-running the core. This procedure is detailed in Restoring the Context.
- If the JTAG is used, a 1-bit bypass value is shifted in the bypass register in the shift_dr state. The SDMA is rerun when the update_dr state is decoded in the TAP controller. If the OnCE is driven from the ARM platform side, the core is rerun when detecting a write access to the ONCE_CMD register.

- exec_core command execution-The exec_core command resumes program execution from any address. The 16-bit instruction provided with the exec_core overwrites the last instruction decoded before entering debug mode. This command is designed to support change of flow instructions, so that a program execution can be restarted from any address. After executing an exec_core command, the SDMA leaves debug mode. The exec_core command is usually used with a jmp instruction.
- If the JTAG is used, the 16-bit branch instruction is shifted in the SDMA instruction register after 16 TCK clock cycles in the shift_dr state. The SDMA is rerun when the update_dr state is decoded in the TAP controller. If the OnCE is driven from the ARM platform side, the SDMA reruns when detecting a write access to the

ONCE_CMD register. The ONCE_INSTR register must therefore be loaded first. For example, to restart the SDMA from the program address 0x100, the instruction loaded should be a jump to address 0x100 instruction.

- debug_rqst command execution-The debug_rqst command puts the SDMA in debug mode. If the JTAG is used, a 1-bit bypass value is shifted in the bypass register during the shift_dr state. A debug request is sent to the SDMA when the update_dr state is decoded in the TAP controller. If the OnCE is driven from the ARM platform side, the debug request is sent when detecting a write access to the ONCE_CMD register. When the SDMA is already in debug mode, this command is simply ignored.
- rbuffer command execution-The rbuffer command exports the content of the real time buffer (RTB). If the JTAG is used, the content of the real time buffer (RTB) is captured in the SDMA data register during the capture_dr state. The register is completely shifted out after maintaining the shift_dr state during 32 TCK clock cycles. If the OnCE is driven from the ARM platform side, the content of the RTB is captured in the ONCE_DATA register after detecting a write access to the ONCE_CMD register.
- bypass command execution-This command is only available from the JTAG interface. It enables daisy-chaining of the SDMA JTAG TAP controller with other JTAG TAP controllers. This command does not change the SDMA state and can be executed in any mode (run, debug, or sleep). It selects the bypass register of the TAP controller.

66.15.4 Registers Descriptions

See SDMACORE, and SDMAARM, for detailed information on each register.

66.15.4.1 Event Cell Counter Register (ECOUNT)

The event cell counter register is a 16-bit register that contains the number of times minus one that an event detection occurs before generating a debug request.

This register should be written before attempting to use the event detection counter during an event detection process. The event cell counter register is cleared on a JTAG reset.

66.15.4.2 Event Cell Address Registers (EAA or EAB)

The event cell contains two address registers-the event cell address register (a), called EAA, and the event cell address register (b), called EAB. Every address register is a 16-bit register that stores a user-defined address value. This value computes one of the following address conditions: addra_cond or addrb_cond. Every address register is cleared on a JTAG reset.

66.15.4.3 Event Cell Address Mask Register (EAM)

The event cell address mask register is a 16-bit register that contains a user-defined address mask value. This mask is applied to the address value latched from the memory address bus before comparing addresses.

NOTE

There is a common address mask value for the two address comparators. If bit *i* of this register is set, then bit *i* of the address value latched from the memory bus does not influence the result of the address comparison. The event cell address mask register is cleared on a JTAG reset.

66.15.4.4 Event Cell Data Register (ED)

The event cell data register is a 32-bit register that contains a user-defined data value. This data value is an input for the data comparator, which generates the data_cond condition.

The event cell data register is cleared on a JTAG reset.

66.15.4.5 Event Cell Data Mask Register (EDM)

The event cell data mask register is a 32-bit register that contains a user-defined data mask value. This mask is applied to the data value latched from the memory bus before comparing data.

Setting bit *i* of the event cell data mask register means that bit *i* of the data value latched from the address bus does not influence the result of the data comparison. The event cell data mask register is cleared on a JTAG reset.

66.15.4.6 Real Time Buffer Register (RTB)

The real Time Buffer register is a 32-bit register that stores and retrieves run-time information without putting the SDMA in debug mode.

Refer to Real Time Buffer for more details.

66.15.4.7 Event Control Register (ECTL)

The event cell control register is a 16-bit register that defines cell event occurrence conditions.

The event cell control register is cleared on a JTAG reset. See also OnCE Event Detection Unit for more details.

66.15.4.8 Trace Buffer (TB)

The Trace Buffer register retrieves the information in the Trace Buffer.

See Trace Buffer for more details.

66.15.4.9 OnCE Status Register (OSTAT)

The OnCE status register is a 16-bit register that contains processor and event detection unit status. The OSTAT is a read-only register.

Refer to OnCE Status Register (SDMAARM_ONCE_STAT) for detailed description of the individual fields in the OSTAT register.

The following figure shows the OSTAT structure.

Table 66-43. OnCE Status Register (OnCE)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PST[3:0]				RCV	EDR	ODR	SWB	MST					ECDR[2:0]		

Where PST[3:0] is the SDMA core state, RCV is set when the real-time buffer (RTB) is modified. EDR, ODR, and SWB are set, respectively, when the SDMA has entered debug mode because of an external debug request, a OnCE debug_rqst command, or a software breakpoint. MST is set when the OnCE is controlled from the ARM platform control interface, and when ECDR is a three-flag set that shows the event cell condition(s) that put the core in debug mode. The OSTAT never provides more than one reason for entering debug mode.

There are two ways of accessing OSTAT content, as follows:

1. Send an rstatus command to the OnCE controller through the JTAG, or read the ONCE_STAT register through the ARM platform access. Executing the rstatus command through the JTAG can be performed in both user and debug modes.
2. Perform an SDMA read access to the location in the SDMA core memory map (OSTAT register) debug mode using the exec_once command. With this method of access, the SDMA state reflected by the PST (processor status bit) is always DATA.

The register may also be accessed by a running application.

66.15.5 JTAG Interface Requirements

Because the signals received from the JTAG (running on TCK) are transferred to the OnCE controller (running on the SDMA clock), a synchronization mechanism is required.

66.15.5.1 TCK Speed Limitation

In the JTAG top-level layer, the TDO signal is always captured on a TCK falling edge. To guarantee a stable TDO signal from the SDMA during this operation, a falling edge detection is performed on TCK.

Before being latched in the *I* flip-flop (see Figure 66-12) on TCK falling edge, the TDO signal must be stable at the input of the flip-flop. This condition is verified if the TCK period is superior to the following delay:

worst-case edge detection delay + negative-edge signal propagation delay + JTAG top-level logic propagation delay

The frequency relationship, TCK < CLK/8, limitation guarantees that all operations are performed as expected.

66.15.5.2 Synchronization Implementation

The following figure shows the synchronization mechanism. Flip-flops tck0, tck1, and tck2 perform falling- and rising-edge detections on TCK. They generate the posedge_detected and negedge_detected nets that are used to sample the TDI and TMS inputs into the respective tdi and tms flip-flops, and update the tdo flip-flop to yield the

TDO output. In the design, the only signal that might go metastable is the output of the tck0 flip-flop. This signal is captured in the tck1 flip-flop and no logical operation is performed on it to minimize a metastability propagation risk.

The TDI and TMS flip-flops also cannot go metastable: The propagation time of the rising-edge detection signal through tck0, tck1, and tck2 guarantees that the TDI and TMS inputs are stable when captured in the TDI and TMS flip-flops.

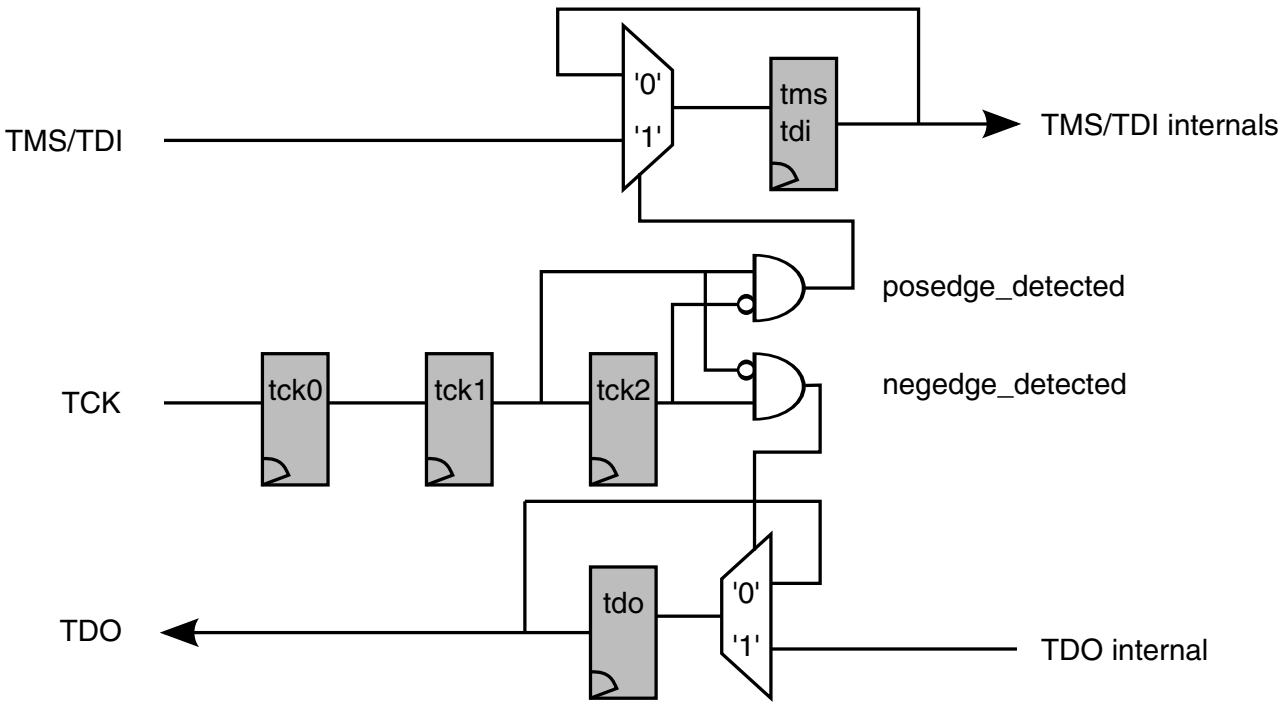


Figure 66-12. OnCE Synchronization Layer

The following figure shows synchronization timings. It takes three CLK clock cycles to synchronize TDI on the SDMA clock.

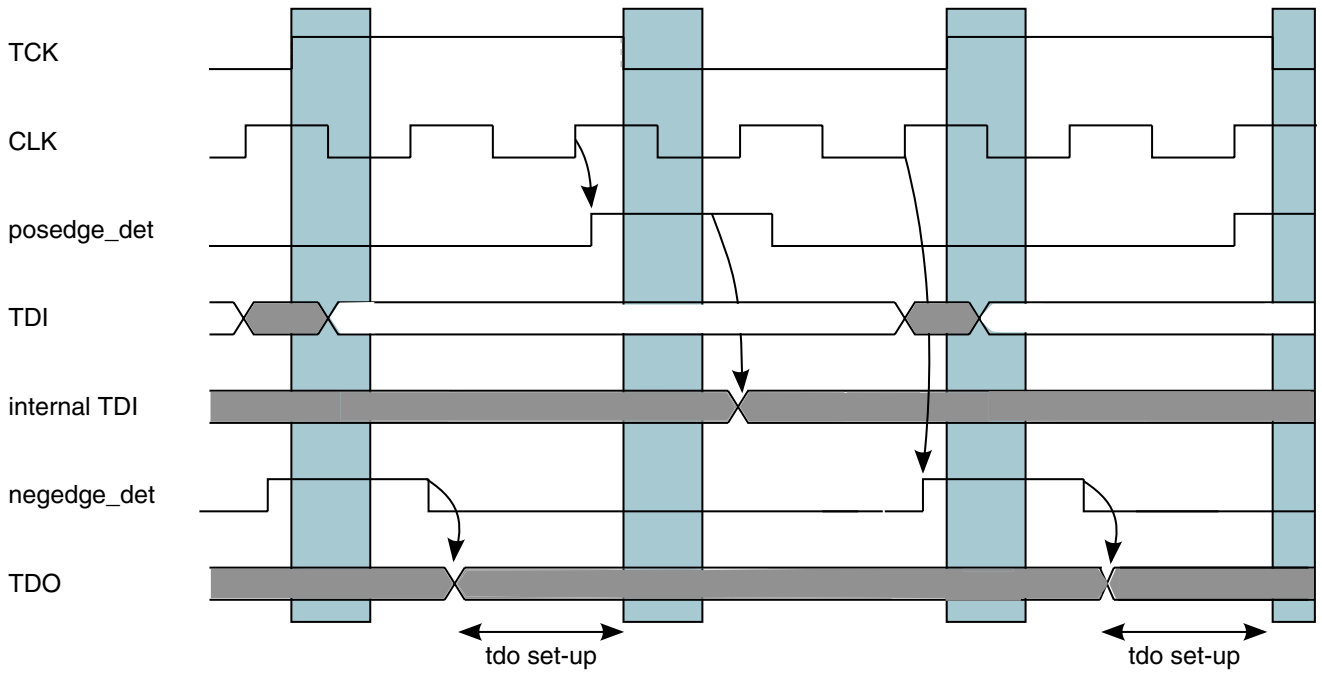


Figure 66-13. Synchronization Timings

66.15.5.3 JTAG Controller Start-Up Recommended Procedure

To ensure correct TAP controller initialization, it is recommended to use the following procedure:

1. Assert JTAG reset TRSTB (for example, set low).
2. Set TMS low.
3. Wait for 1 TCK clock.
4. Release JTAG reset TRSTB (for example, set high).
5. Wait for a minimum of five TCK cycles.

66.16 Using the OnCE

This section provides the elements necessary to run the OnCE during a debug process. In addition to the basic set of commands described in OnCE Commands, more complex commands can be built to meet users' requirements.

66.16.1 Activating Clocks in Debug Mode

For power consumption issues, some clocks in the SDMA are disabled when not needed. This is the case for instances when the SDMA is in sleep mode. Clock gating management depends on the interface used to control the OnCE.

- For the JTAG access, the SDMA clock gating must be turned off via the clk_gating_off input.
- For the ARM platform access, the SDMA clock gating is automatically turned off when the ARM platform access is enabled (see OnCE Enable (SDMAARM_ONCE_ENB)).

66.16.2 Getting the Current Status

Most of the commands the OnCE supports have an impact on the status of the SDMA.

It is not permissible to request the execution of an instruction on the SDMA from the OnCE while the SDMA is not in debug mode. Such a violation may cause unpredictable behavior, and it might be necessary to reset the SDMA.

Therefore, the value of the PST bits provided in the OnCE status register should always be checked before sending any request to the SDMA.

66.16.3 Methods of Entering Debug Mode

A debug request may be asserted at any time, but it is not always taken into account immediately. Debug mode cannot be entered in the middle of an instruction, or during the save or restore states of a context switch.

The request is ignored when the core is already in debug mode. Refer to Figure 66-4, which shows all possible transitions to the debug state, as there are several ways to enter debug mode.

66.16.3.1 External Debug Request During Reset

To enter debug mode after exiting reset, the external debug line has to be maintained high. This line is handled by the JTAG top-level block.

NOTE

The SDMA detects the debug requests only if the SDMA clock is running (see Activating Clocks in Debug Mode). The debug

request line should be not be maintained high when the SDMA is in debug mode.

NOTE

The debug_rqst command (from the OnCE command set) is not supported during system reset.

66.16.3.2 Debug Request During Normal Activity

During normal activity, the SDMA enters debug mode when the following is true:

1. If the debug request line from the JTAG top-level is asserted, or
2. If the OnCE controller receives a debug_rqst command.

   The debug_rqst command can be sent by the JTAG access or by an access on the ARM platform side (if the ARM platform access is enabled).

66.16.3.3 Software Breakpoint Instruction

The SDMA enters debug mode at the end of the execution of a software breakpoint instruction. This instruction must be inserted in program flow executed by the core.

66.16.3.4 Event Detection Unit Matching Condition

If the event detection is enabled, a debug request is sent to the core after detecting a matching condition on the SDMA memory bus.

See OnCE Event Detection Unit for more details.

66.16.4 Executing Instructions in Debug Mode

The OnCE supports a mechanism to execute instructions in debug mode. If the SDMA is in debug mode, then the exec_once command can be used to execute an SDMA instruction from the OnCE controller. The SDMA returns to debug mode at the end of each execution.

Some instructions are not supported by the exec_once command: done/yield/yiedge (except done 5), BF, BT, BSF, BDF, JMP, JSR, JMPR, JSRR, RET, and LOOP, as well as all the illegal instructions are not supported.

NOTE

While instructions are executed in debug mode from the OnCE, the program counter of the SDMA is not incremented.

66.16.5 Command Sequences Examples

This section provides examples of command sequences that run the SDMA in debug mode. These sequences are available for both the ARM platform and JTAG accesses.

The following presents the syntax used in this section. The data field provided with each command is put in parenthesis with the command name. A '-' is used if the data field provided is a *don't care* value.

```
my_command(data_field);                 // executing my_command with a data field
my_command(-);                        // executing my_command with a don't care data field
```

The value returned by the command (if there is one) is referred by an assignment. In case the value returned by the command is not used, the assignment is omitted. For an ARM platform access, the value returned (it is always a data value) is obtained by reading back into the SDMA data register.

```
data_out = my_command(data_in); // returning a data value
```

To clarify the syntax, the instructions' opcodes are referred to by their names. In practice, use the corresponding 16-bit encoding.

66.16.5.1 Getting the SDMA Status

NOTE

Before executing any command that affects the SDMA (like dmov or exec_once), check that the SDMA is in debug mode.

Use the following snippet:

```
rstatus();        // read SDMA status until the SDMA is in debug mode
...
rstatus();
```

If the SDMA is not in debug mode, then a debug request must be generated. In this case, the SDMA enters debug mode at the end of the execution of the current instruction. Use this snippet:

```
debug_rqst(-);   // debug request
```

In the following sections, it is assumed that the SDMA was successfully put into debug mode.

66.16.5.2 Saving the Context

The first debug task is to save the SDMA context, which is the content of the eight general-purpose registers, the loop and PC-related registers, and the flags. Use the general register GReg[1] as an intermediate register to export the entire context of the SDMA.

The following example shows how to save GReg[0], GReg[1], GReg[2] and GReg[3]. The sequence of commands used to export additional general registers is very similar to this.

Save GReg[0], GReg[1], GReg[2], and GReg[3]

```
GReg1_data = dmov(-);                          // the value exported is the content of
GReg[1]
exec_once("mov GReg1,GReg0");                      // puts the content of GReg[0] into
GReg[1]
GReg0_data = dmov(-);                          // the value exported is the content of
GReg[0]
exec_once("mov GReg1, GReg2");                     // puts the content of GReg[2] into
GReg[1]
GReg2_data = dmov(-);                          // the value exported is the content of
GReg[2]
exec_once("mov GReg1, GReg3");                     // puts the content of GReg[3] into
GReg[1]
GReg3_data = dmov(-);                          // the value exported is the content of
GReg[3]
```

Get the value of the internal flags (SF, DF, T, and LM), of the loop related registers (EPC and SPC), and of the PC-related registers (PC and RPC). Use a done 5, which is the formatting instruction dedicated to the debug. This instruction formats the flags and the values contained in the registers. It also writes the resulting values into the channel context memory. It should not be used when entering debug from the IDLE state (for example, with no active channel script running on the SDMA), because it will update a channel context that may belong to any channel.

```
exec_once("done 5");                    // formatting the value of flags and registers
```

At this point, the channel context should be up-to-date in memory, and debug operations should now be possible. However, the context can be exported with the following instructions:

Exporting the Context

```
dmov(ctx_base_addr);                              // loading GReg[1] with the channel
context base address
exec_once("ld GReg0,(GReg1,0)");                      // get RPC-PC into GReg0
exec_once("ld GReg1, (GReg1,1)");                     // get SPC-EPC into GReg1
Loop_data = dmov(-);                           // read back the value of Loop registers
exec_once("mov GReg1, GReg0");                     // puts the PC info into GReg1
PC_data = dmov(-);                             // reads back the content of the PC registers
```

After this sequence of operations, the entire SDMA context is exported via the OnCE

66.16.5.3 Restoring the Context

At this point in the operation, restore the context of the SDMA. It can be different from the original context located in memory, and the content previously saved into the debugging application via the OnCE.

The following example shows how it is possible to modify the current channel context:

Modifying the Current Channel Context

```
dmov(Loop_data);                                  // put Loop former value into GReg[1]
exec_once("mov GReg0, GReg1");                         // copy to GReg[0]
dmov(PC_data);                                    // put PC former value into GReg[1]
exec_once("mov GReg2, GReg1");                         // copy to GReg[2]
dmov(ctx_base_addr);                                  // put channel context base address into
GReg[1]
exec_once("st GReg0, (GReg1,1)");                          // restore Loop context
exec_once("st GReg2, (GReg1,0)");                          // restore PC context
```

Once the context in memory is the desired context (with or without applying the previous instruction sequence), it can be restored to the *real* PC and loop registers in the SDMA core:

```
exec_once("cpShReg");                    // restore flags and PC & loop related registers
```

After this command, the SDMA core PC, RPC, SPC, EPC registers, as well as the flags contain the same data as what is stored in the context RAM for the current channel.

The following example shows how to restore the context of general registers GReg[0], GReg[1], GReg[2] and GReg[3].

Restoring the General Register Context

```
dmov(GReg3_data);                    // put GReg[3] restore value in GReg[1]
exec_once("mov GReg3, GReg1"); // restore GReg[3]
dmov(GReg2_data);                    // put GReg[2] restore value in GReg[1]
exec_once("mov GReg2, GReg1"); // restore GReg[2]
dmov(GReg0_data);                    // put GReg[0] restore value in GReg[1]
exec_once("mov GReg0, GReg1"); // restore GReg[0]
dmov(GReg1_data);                    // restore GReg[1]
```

At this point, it is possible to restart the normal program execution.

NOTE

Every SDMA core general register value can be modified by a mov instruction, which makes modification of these registers easy during debug. Unfortunately, there is no such instruction as a mov to directly modify the contents of either PCU register or flag (PC, RPC, SPC, EPC, T, LM, SF, or DF). The cpShReg instruction is meant to provide a means for changing these register contents via the context memory.

66.16.5.4 Accessing the Memory

In the following example, it is assumed that the SDMA context is entirely saved. If true, it is permissible to modify the general purpose registers during debugging activity.

To perform a memory read access, the target address is stored via the OnCE in GReg[1], then the load instruction is executed on the SDMA (the data loaded from the memory overwrites the address contained in GReg[1]), and then the result value is read back via the OnCE.

```
macro READ:             dmov(target_addr);                                  // put the target
address in GReg[1]
                    exec_once("ld GReg1,(GReg1,0)");                          // execute the
load instruction
                    res_data = dmov(-);                                 // exports the result
data value
```

For a memory write access, the target address is written in GReg[0], and the value to store is written in GReg[1]. Then the store instruction is executed on the SDMA.

```
macro WRITE:             dmov(target_addr);                                   // puts the
target address in GReg[1]
                    exec_once("mov GReg0,GReg1");                          // puts the target
address in GReg[0]
                    dmov(target_data);                                  // puts the target
data in GReg[1]
                    exec_once("st GReg1,(GReg0,0)");                           // performs the
store operation
```

This sequence is shown as an example; however, many other sequences are possible.

NOTE

This sequence of commands can also be applied to memory-mapped registers.

66.16.5.5 Resuming Program Execution

Before resuming program execution, it is assumed that the SDMA context is properly restored. There are two ways to restart the SDMA.

Start by executing the last instruction fetched before entering debug mode, as follows.

```
run_core(-);                          // resume execution from where we stopped before
```

If necessary, restart the execution from a different address. In this case, use the exec_core command. The data field provided with this command must be the encoding of a jump instruction.

```
exec_core("jmp start_addr"); // rerun the SDMA from another address
```

In these two examples, the SDMA exits debug mode and keeps executing the code fetched from the memory.

66.16.5.6 Single Stepping in RAM

To execute a program step-by-step from the RAM, insert software breakpoints in the program flow at appropriate places so that the SDMA only executes one instruction before returning to debug mode.

First, read the next instruction to execute in the RAM. Then, depending on the value of this instruction, compute the address where a software breakpoint instruction should be inserted. The instruction at the corresponding address must be saved, and, the software breakpoint instruction is inserted. After restarting the SDMA, there is only one instruction executed before meeting the software breakpoint.

The following example shows the macro functions READ and WRITE, which correspond to the sequence of commands (described above) used to access the memory.

NOTE

The data read from the memory are 32-bit values, while the instructions are 16-bit values only. This is why it is best to only use addresses divided by two when accessing the memory.

READ and WRITE Macro Functions

```
next_instr = READ(run_addr/2);                          // read the next instruction to execute
// the tool now has to compute the address where the breakpoint
// instruction should be inserted, this address is the "bkpt_addr"
instr_save = READ(bkpt_addr/2);                          // save the instruction before
overwriting
STORE("bkpt instruction",bkpt_addr/2);                        // store the bkpt instruction
in memory
exec_core("jmp run_addr");                              // rerun the SDMA
rstatus(-);                                           // wait for the SDMA to enter debug mode
...
rstatus(-);
STORE(instr_save,bpkt_addr/2);                           // restore the instruction
overwritten
```

In case of branched conditional instructions, a breakpoint instruction should be written at the two possible target addresses.

66.16.5.7 Single Stepping in ROM

No single-step mechanism is supported in ROM. The program code can be loaded in the RAM, where the single-step mechanism can be executed.

66.16.6 OnCE Event Detection Unit

The event detection unit watches signals from the data memory bus (DMBUS), which the SDMA core uses to access its RAM, ROM, and memory mapped registers.

A debug request is sent to the OnCE controller when user-defined conditions on address and/or data values are true.

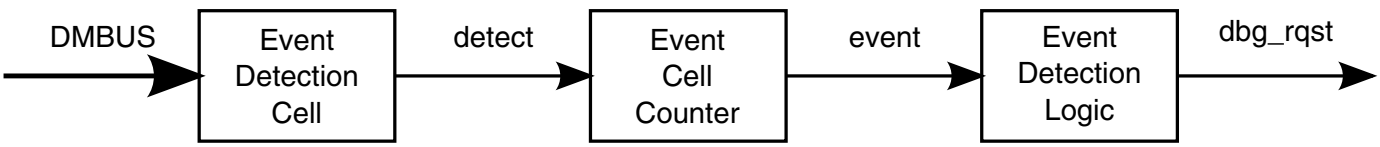


Figure 66-14. Event Detection Unit

A counter, provided with the detection cell, is decreased after an event detection. A debug request is sent to the core only when the counter reaches the value of 0. It is possible to disable the use of the counter if a debug request has to be generated after each event detection.

The event cell is the basic block that supports hardware breakpoints on an address value and/or data values coming from the SDMA memory bus. The trigger condition that generates the debug request is a mixed condition based on those values.

The following figure shows the event cell architecture. The event cell contains the address (stored in the memory address register) and the data (stored in the memory data register) used during the last memory access. There are some user-defined reference values located in memory mapped registers-the event cell addresses, the event cell address mask, the event cell data, and the event cell data mask. These registers are accessed by standard load/store instructions just like regular memory locations.

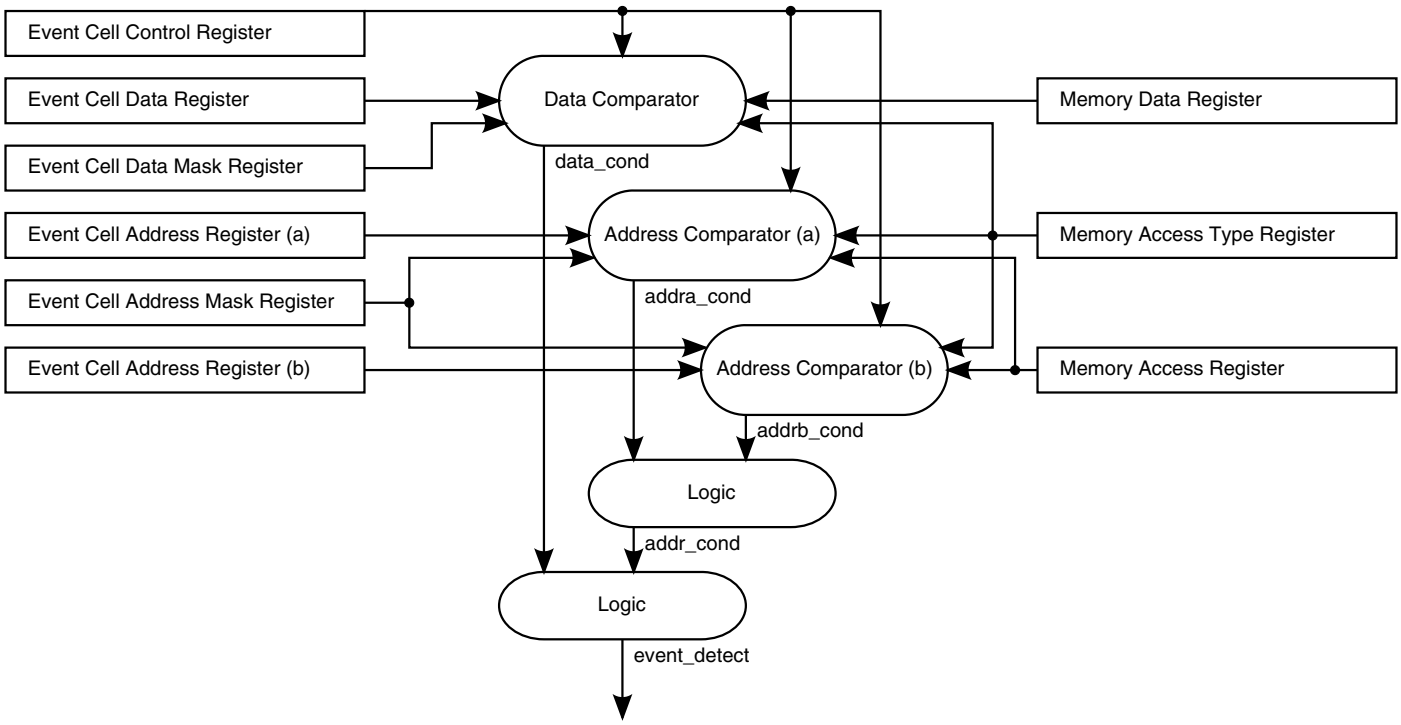


Figure 66-15. Event Cell Architecture

To define a memory breakpoint, three conditions are taken into account: The first two conditions are comparisons of the current memory address with user-defined reference addresses (these conditions are called addressA and addressB). The third condition consists of a comparison between the data received on the DMBUS and a user-defined reference data (this condition is called data). An intermediate address condition is set to express a dependency between addressA and addressB conditions.

66.16.7 Clock Gating and Reset

This section details how to use the clocks and handle the reset signals.

66.16.7.1 Clocks

Because the SDMA uses clock gating to save power, it is necessary to disable the clock gating and force the clocks to be enabled when using the OnCE.

When the OnCE is accessed through its JTAG interface, clock gating must be disabled outside the SDMA via a dedicated SDMA input port clk_gating_off. The reason why detection is not performed automatically by the SDMA internal hardware is that it would cost power to monitor activity on the JTAG interface.

This clk_gating_off input is controlled by a control bit in the System JTAG Controller. Refer to the System JTAG Controller chapter for more information.

When the OnCE is accessed through the ARM platform Control interface, clock gating is automatically turned off. This is done when bit 0 of the ONCE_ENB register (see OnCE Enable (SDMAARM_ONCE_ENB)) is set. A write access to this register is possible even when the OnCE clock is not running. If the ARM platform access is used, the bit in the ONCE_ENB register must be set before any attempt to access any other OnCE register.

66.16.7.2 Resets

The OnCE reset is different from the SDMA main reset. The OnCE reset is connected to the TRST_B pin.

Normally, activating the SDMA reset while keeping the OnCE reset inactive (when possible) enables you to reset the core without having to reprogram the OnCE.

66.16.8 Real Time Features

To rebuild the skeleton of a program execution, it is necessary to store the addresses of the program instructions where jumps are taken: A trace buffer is therefore provided. A real time buffer has also been added to receive data values written during a program execution.

The content of this register may be exported through JTAG ports without stopping the core.

66.16.8.1 Trace Buffer

The Trace Buffer is a 32-stage buffer that contains appropriate information to identify the 32 last changes of flow detected during a program execution.

The following figure shows an overview of the Trace Buffer.

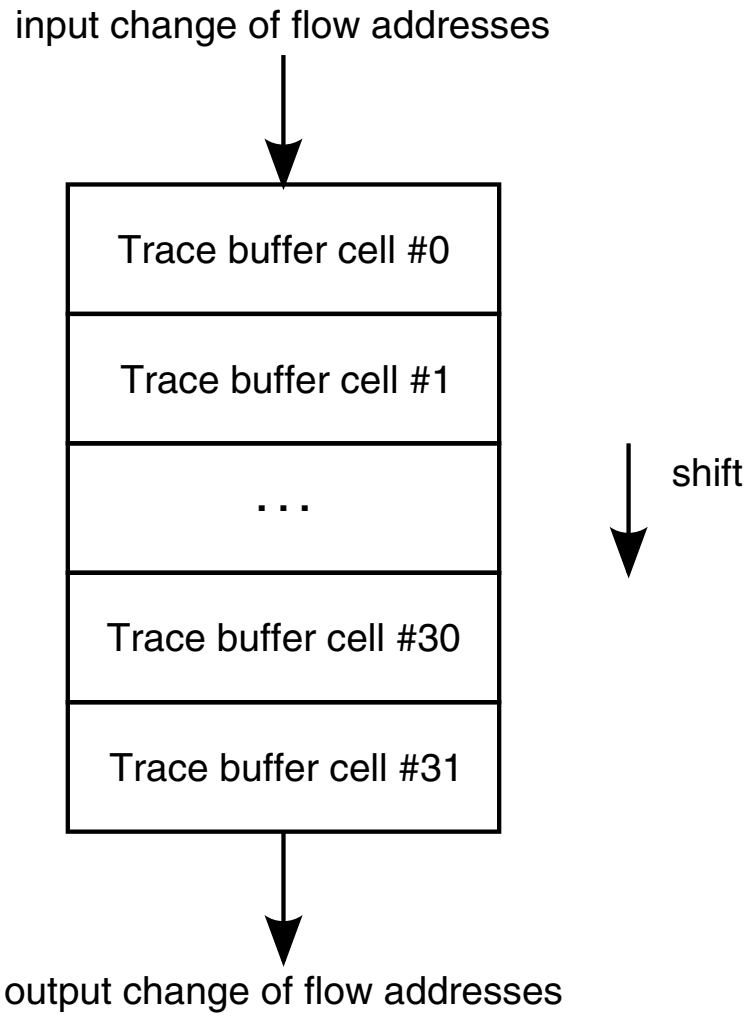


Figure 66-16. Trace Buffer

Each cell of the trace buffer contains two reference addresses and a flag. The flag is set when the addresses stored in the cell correspond to a valid change of flow; otherwise, the flag is cleared. The three most significant bits are unused.

After every change of flow detection, the address of current instruction and the address of the target instruction are stored at the top of the Trace Buffer (cell #0). The flag in the cell is set to indicate that a valid change of flow was detected. Former cell values are shifted one level down. The Trace Buffer contains the 32 last changes of flow. All the flags are reset on a software or a hardware reset, and after each transition from debug mode to user mode.

A memory mapped register of SDMA core, the Trace Buffer register (TB), is provided to read the content of the Trace Buffer. This operation should be done in debug mode. Performing a read access to the Trace Buffer register returns the content of the bottom of the Trace Buffer (cell #31). After every read access, the trace buffer is shifted one level down, and the flag at the top of the trace buffer is cleared.

A typical OnCE command sequence that retrieves the oldest change-of-flow information is a follows:

```
exec_once("mov r1, TB");                              // stores the oldest change-of-flow in
GReg1
dmov(-);                                          // retrieves GReg1 contents
```

This sequence requires the SDMA to be put in debug mode.

66.16.8.2 Real Time Buffer

The Real Time Buffer register (RTB) is a memory mapped register that can be accessed as a regular memory location by the SDMA core during program execution. This register is located in the OnCE.

Executing ar rbuffer command (see The OnCE Controller for further details) exports the content of this register through JTAG ports.

When a write access is performed at the memory location corresponding to the RTB, the receive flag (for example, the RCV bit) is set in the OnCE Status Register (OSR). This flag is cleared at the end of the execution of a rbuffer command.

NOTE

Every write access to the RTB memory location updates the RTB register even if the RCV flag is set. The RTB is cleared on a JTAG reset.

66.16.8.3 Emulation Pin

The debug_matched_event emulation pin reflects the matching condition status detected by the Event Detection Unit.

Since it can be necessary to detect conditions without triggering debug requests, it is possible to disable the generation of debug requests by the Event Detection Unit and still have the matching condition available on the emulation pin. This can be done by clearing the EN flag in the ECTL register.

66.16.8.4 Real-Time Debug Outputs

The folloing table shows the debug signals that are available at the SDMA boundaries. Their availability at chip boundaries depends on the project.

Table 66-44. Real-Time Debug Output Pins

Pin	Description
debug_core_state[3:0]	The core_state bits reflect the state of the SDMA core. • The "Program" state is the usual instruction execution cycle. • The "Data" state is inserted when there are wait-states during a load or a store on the data bus (ld or st). • The "Change of Flow" state is the second cycle of any instruction that breaks the sequence of instructions (jumps and channel switching instructions). • The "Change of Flow in Loop" state is used when an error causes a hardware loop exit. • The "Debug" state means the SDMA is in debug mode. • The "Functional Unit" state is inserted when there are wait-states during a load or a store on the functional units bus (ldf or stf). • In "Sleep" modes, no script is running (this is the core idle state); the "after Reset" is slightly different because no context restoring phase will happen when a channel is triggered: The script located at address 0 is executed (boot operation). • The "in Sleep" states are the same as above except they do not have any corresponding channel: they are used when entering debug mode after reset; the reason is that it is necessary to return to the "Sleep after Reset" state when leaving debug mode. 0 Program 1 Data 2 Change of Flow 3 Change of Flow in Loop 4 Debug 5 Functional Unit 6 Sleep 7 Context Switch Saving Channel 8 Program in Sleep 9 Data in Sleep 10 Change of Flow in Sleep 11 Change of Flow in Loop in Sleep 12 Debug in Sleep 13 Functional Unit in Sleep 14 Sleep after Reset 15 Context Switch Restoring Channel
debug_yield	Pulse that is active when a yield (done 0) or a yieldge (done 1) instruction is executed. 0 - 1 yield/yieldge executed

Table continues on the next page...

Table 66-44. Real-Time Debug Output Pins (continued)

Pin	Description
debug_core_run	Active when the SDMA core is executing instructions. 0 Debug or sleep mode 1 Run mode
debug_event_channel_sel	Indicates if debug_event_channel displays current channel or last received event 0- debug_event_channel[5:0] gives the number of the current channel 1- debug_event_channel[5:0] gives the number of the last received event
debug_event_channel[5:0]	Gives the number of any DMA request as soon as it is received or the number of the current channel. The value of debug_event_channel_sel indicates if debug_event_channel displays the current channel or last received event. The signal debug_event_channel_sel must be observed to determine what information is provided on debug_event_chanel at any given time.
debug_pc[13:0]	Program Counter value; it has a meaning when the core is in run mode.
debug_mode	Set when the core is in debug. 0 - 1 Core is in debug
debug_bus_error	Set when an error was received during a load or a store (ld, st, ldf, or stf instruction) and registered in SF or DF flag. 0 No error during last load/store 1 Error during last load/store

Table continues on the next page...

Table 66-44. Real-Time Debug Output Pins (continued)

Pin	Description
debug_bus_device[4:0]	Indicates the device or functional unit that is accessed by the current instruction. The debug_bus_device output is always valid when in sleep mode, debug mode, or executing any instruction that does not access the functional units or the memory mapped devices, "no access" is output. 0 No access 1 MSA 2 MDA 3 MD 4 MS 5 PSA 6 PDA 7 PD 8 PS 9 RESERVED 10 RESERVED 11 RESERVED 12 RESERVED 13 CA 14 CS 15 Reserved 16 Memory (RAM or ROM) 17 Memory mapped register 18 Peripheral #1 19 Peripheral #2 20 Peripheral #3 21 Peripheral #4 22 Peripheral #5 23 Peripheral #6 24 Peripheral #7 25 Peripheral #8 26 Peripheral #9 27 Peripheral #10 28 Peripheral #11 29 Peripheral #12 30 Peripheral #13 31 Peripheral #14

Table continues on the next page...

Table 66-44. Real-Time Debug Output Pins (continued)

Pin	Description
debug_bus_rwb	Indicates the direction of the access given by debug_bus_device 0 Write access (st or stf) 1 Read access (ld or ldf)
debug_matched_dmbus	Pulse indicating the OnCE event detection unit has detected a match on the data bus during an access to memory (RAM or ROM), a memory mapped register or a peripheral that is hooked to the SDMA. 0 - 1 data bus match detected
debug_rtbuffer_write	Pulse indicating when the real-time buffer is written by the core. 0 - 1 RTB was modified
debug_evt_chn_lines[7:0]	Eight lines that generate short pulses when DMA requests are received or channels are (re)started. Every line is controlled through two parameters defined in registers Cross-Trigger Events Configuration Register 1 (SDMAARM_XTRIG_CONF1) (as described in SDMAARM). The following two parameters are available for every line: • CNF-Indicates what is monitored on the line: 0 for a channel start, 1 for a DMA request reception • NUM[5:0]-Gives the number of the DMA request or channel to monitor

The matched_event emulation pin reflects the matching condition status detected by the Event Detection Unit. Because it can be necessary to detect conditions without triggering debug requests, it is possible to disable the generation of debug requests by the Event Detection Unit and still have the matching condition available on the emulation pin. This can be done by clearing the EN flag in the ECTL register.

All real-time debug outputs are disabled by default (for example, they are stuck to 0) to avoid power consumption when they are not used. They are enabled when bit 11 (RTDOBS) of the Configuration Register (SDMAARM_CONFIG) is set. Signals provided to the system JTAG controller for SDMA debug mode status will also be enabled when the *clk_gating_off* input is asserted.

66.17 Instruction Set

66.17.1 Instruction Encoding

This section presents a short summary of the instruction codes. All context switch instructions are listed for information only; they cannot function properly out of the context switch routine.

```
x...x - don't care

rrr - destination/source general register

sss - additional source general register

bbb - general register used as address base register

ddddd - address displacement

nnnnn - bit number
uuuuuuuu - function unit command bits

pppppppp - branch displacement (signed)

iiiiiiii - 8-bit immediate

jjj - control bit to clear

ff - flag to clear
00000jjj00000000        - done (done,yield,wait)
00000jjj00000001        - notify
00000xxx00000010        - reserved
00000xxx00000011        - reserved
00000xxx00000100        - reserved
0000000000000101     - softBkpt
0000000100000101     - reserved
0000001000000101     - reserved
0000001100000101     - reserved
0000010000000101     - reserved
0000010100000101     - reserved
0000011000000101     - reserved
0000011100000101     - reserved
0000000000000110     - ret
0000000100000110     - reserved
0000001000000110     - reserved
0000001100000110     - reserved
0000010000000110     - reserved
0000010100000110     - reserved
0000011000000110     - reserved
0000011100000110     - reserved
000000ff00000111       - clrf ff
0000010000000111     - reserved
0000010100000111     - reserved
0000011000000111     - reserved
0000011100000111     - illegal
00000rrr00001000        - jmpr r
00000rrr00001001        - jsrr
00000rrr00001010        - ldrpc r
00000rrr00001011        - reserved
00000rrr000011xx          - reserved
00000rrr00010000        - revb
00000rrr00010001        - revblo
00000rrr00010010        - rorb
00000rrr00010011        - reserved
00000rrr00010100        - ror1
00000rrr00010101        - lsr1
00000rrr00010110        - asr1
00000rrr00010111        - lsl1
00000rrr001nnnnn        - bclri r,n
00000rrr010nnnnn        - bseti r,n
00000rrr011nnnnn        - btsti r,n
00000xxx10000xxx           - reserved
00000rrr10001sss           - mov
00000rrr10010sss           - xor
00000rrr10011sss           - add
00000rrr10100sss           - sub
00000rrr10101sss           - or
00000rrr10110sss           - andn
```

```
00000rrr10111sss           - and
00000rrr11000sss           - tst
00000rrr11001sss           - cmpeq
00000rrr11010sss           - cmplt
00000rrr11011sss           - cmphs
0000011011100000     - reserved
0000011011100001     - reserved
0000011011100010     - cpShReg
0000011011100011     - reserved
0000011011100100     - reserved
0000011011100101     - reserved
0000011011100110     - reserved
0000011011100111     - reserved
00000xxx11101xxx           - reserved
00000xxx11110xxx           - reserved
00000xxx11111xxx           - reserved
00001rrriiiiiiii                - ldi r,i
00010rrriiiiiiii                - xori r,i
00011rrriiiiiiii                - addi r,i
00100rrriiiiiiii                - subi r,i
00101rrriiiiiiii                - ori r,i
00110rrriiiiiiii                - andni r,i
00111rrriiiiiiii                - andi r,i
01000rrriiiiiiii                - tsti r,i
01001rrriiiiiiii                - cmpeqi r,i
01010rrrddddbbbb         - ld r,(d,b)
01011rrrddddbbbb         - st r,u
01100rrruuuuuuuu         - ldf r,u
01101rrruuuuuuuu         - stf r,u
011100xxxxxxxxxx               - reserved
011101xxxxxxxxxx               - reserved
011110ffnnnnnnnn       - Loop ff flags are reset
01111100pppppppp     - bf pc=pc+signed(pppppppp)+1
01111101pppppppp     - bt pc=pc+signed(pppppppp)+1
01111110pppppppp     - bsf pc=pc+signed(pppppppp)+1
01111111pppppppp     - bdf pc=pc+signed(pppppppp)+1
10aaaaaaaaaaaaaa     - jmp absolute
11aaaaaaaaaaaaaa     - jsr absolute
```

66.17.2 SDMA Instruction Set

This section describes all the useful instructions from the SDMA set.

Table 66-45. SDMA Instruction List

Instruction	Description	Page
ADD	Addition	ADD (Addition)
ADDI	Add with Immediate Value	ADDI (Add with Immediate Value)
AND	Logical AND	AND (Logical AND)
ANDI	Logical AND with Immediate Value	ANDI (Logical AND with Immediate Value)
ANDN	Logical AND NOT	ANDN (Logical AND NOT)

Table continues on the next page...

Table 66-45. SDMA Instruction List (continued)

Instruction	Description	Page
ANDNI	Logical AND with Negated Immediate Value	ANDNI (Logical AND with Negated Immediate Value)
ASR1	Arithmetic Shift Right by 1 Bit	ASR1 (Arithmetic Shift Right by 1 Bit)
BCLRI	Bit Clear Immediate	BCLRI1 (Bit Clear Immediate)
BDF	Conditional Branch if Destination Fault	BDF (Conditional Branch if Destination Fault)
BF	Conditional Branch if False	Functional Units Programming Model
BSETI	Bit Set Immediate	BSETI (Bit Set Immediate)
BSF	Conditional Branch if Source Fault	BSF (Conditional Branch if Source Fault)
BT	Conditional Branch if True	BT (Conditional Branch if True)
BTSTI	Bit Test immediate	BTSTI (Bit Test immediate)
CLRF	Clear ARM platform flags	CLRF (Clear ARM platform flags)
CMPEQ	Compare for Equal	CMPEQ (Compare for Equal)
CMPEQI	Compare with Immediate for Equal	CMPEQI (Compare with Immediate for Equal)
CMPHS	Compare for Higher or Same	CMPHS (Compare for Higher or Same)
CMPLT	Compare for Less Than	CMPLT (Compare for Less Than)
cpShReg	Update Context of PCU Registers and Flags	cpShReg (Update Context of PCU Registers and Flag)
DONE	DONE, Yield	DONE (DONE, Yield)
ILLEGAL	ILLEGAL Instruction	ILLEGAL (ILLEGAL Instruction)
JMP	Unconditional Jump Immediate	JMP (Unconditional Jump Immediate)
JMPR	Unconditional Jump	JMPR (Unconditional Jump)
JSR	Unconditional Jump to Subroutine Immediate	JSR (Unconditional Jump to Subroutine Immediate)
JSRR	Unconditional Jump to Subroutine	JSRR (Unconditional Jump to Subroutine)

Table continues on the next page...

Table 66-45. SDMA Instruction List (continued)

Instruction	Description	Page
LD	Load Register	LD (Load Register)
LDF	Load Register from Functional Unit	LDF (Load Register from Functional Unit)
LDI	Load Register with Immediate Value	LDI (Load Register with Immediate Value)
LDRPC	Load from RPC to Register	LDRPC (Load from RPC to Register)
LOOP	Hardware Loop	LOOP (Hardware Loop)
LSL1	Logical Shift Left by 1 Bit	LSL1 (Logical Shift Left by 1 Bit)
LSR1	Logical Shift Right by 1 Bit	LSR1 (Logical Shift Right by 1 Bit)
MOV	Logical Move	MOV (Logical Move)
NOTIFY	Notify to ARM platform	NOTIFY (Notify to ARM platform)
OR	Logical OR	OR (Logical OR)
ORI	Logical OR with Immediate Value	ORI (Logical OR with Immediate Value)
RET	Return from Subroutine	RET (Return from Subroutine)
REVB	Reverse Byte Order	REVB (Reverse Byte Order)
REVBLO	Reverse Low Order Bytes	Reverse Low Order Bytes(REVBLO)
ROR1	Rotate Right by 1 Bit	ROR1 (Rotate Right by 1 Bit)
RORB	Rotate Right by 1 Byte	RORB (Rotate Right by 1 Byte)
SOFTBKPT	Software Breakpoint	SOFTBKPT (Software Breakpoint)
ST	Store Register	ST (Store Register)
STF	Store Register in Functional Unit	STF (Store Register in Functional Unit)
SUB	Subtract	SUB (Subtract)
SUBI	Subtract with Immediate	SUBI (Subtract with Immediate)
TST	Test with Zero	TST (Test with Zero)
TSTI	Test Immediate	TSTI (Test Immediate)

Table continues on the next page...

Table 66-45. SDMA Instruction List (continued)

Instruction	Description	Page
XOR	Logical Exclusive OR	XOR (Logical Exclusive OR)
XORI	Exclusive OR with Immediate	XORI (Exclusive OR with Immediate)

66.17.2.1 ADD (Addition)

Operation:

```
GReg[r] ← GReg[s] + GReg[r]

T ← (GReg[r] == 0)
```

Assembler:

```
Syntax: add r,s

Example: add 0,3
```

ADD GReg[3] and GReg[0] and store the result in GReg[0]

CPU Flags: T

Cycles: 1

Description: Performs the ADDition of the source general register *s* and the destination general register *r*, and stores the result in the destination general register *r*. The T flag is set if the result of the operation is 0. It is cleared if the result is not 0.

Instruction Format:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	0	1	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]
```

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]

66.17.2.2 ADDI (Add with Immediate Value)

Operation:

GReg[r] ← GReg[r] + immediate

T ← (GReg[r] == 0)

Assembler:

Syntax: addi r,immediate

Example: add 6,112

ADD GReg[6] and decimal value 112 and store the result in GReg[6]

CPU Flags: T

Cycles: 1

Description: Adds a 0-extended immediate value to a general register; stores the result in the general register. The flag T is set when the result of the operation is 0; otherwise, it is cleared. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

```
111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

 ...

11111110 - 254

11111111 - 255
```

66.17.2.3 AND (Logical AND)

Operation:

```
GReg[r] ← GReg[s] & GReg[r]
```

Assembler:

```
Syntax: and r,s

Example: and 1,2
```

AND GReg[1] and GReg[2] and store the result in GReg[1]

CPU Flags: Unaffected

Cycles: 1

Description: Performs the AND of the source general register *s* and the destination general register *r*, and stores the result in the destination general register *r*.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	1	1	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]
```

110 - GReg[6]

111 - GReg[7]

66.17.2.4 ANDI (Logical AND with Immediate Value)

Operation:

GReg[r] ← GReg[r] & immediate

Assembler:

Syntax: andi r,immediate

Example: andi 7,45

AND GReg[7] and decimal value 45 and store the result in GReg[7]

CPU Flags: unaffected

Cycles: 1

Description: Performs an AND between a 0-extended immediate value and a general register; stores the result in the general register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]

iiiiiiii - immediate value:

00000000 - 0

```
00000001 - 1

...

11111110 - 254

11111111 - 255
```

66.17.2.5 ANDN (Logical AND NOT)

Operation:

```
GReg[r] ← ~GReg[s] & GReg[r]
```

Assembler:

```
Syntax:andn r,s

Example: andn 3,4
```

AND GReg[3] and NOT GReg[4] (bit inverted) and store the result in GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Performs the AND of the negation of the source general register *s* and the destination general register *r*, and stores the result in the destination general register *r*.

Instruction Format:

Table 66-46. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	1	1	0	s	s	s

Instruction Fields:

rrr /sss - destination register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.6 ANDNI (Logical AND with Negated Immediate Value)

Operation:

```
GReg[r] ← GReg[r] & ~immediate
```

Assembler:

```
Syntax: andni r,immediate

Example: andni 0,2
```

AND GReg[0] and decimal value -3 (inverted 32-bit value 2) and store the result in GReg[0]

CPU Flags: unaffected

Cycles: 1

Description: Performs an AND between the negation of a 0-extended 8-bit immediate value and a general register; stores the result in the general register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	0	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1
```

```
...

11111110 - 254

11111111 - 255
```

66.17.2.7 ASR1 (Arithmetic Shift Right by 1 Bit)

Operation:

```
GReg[r]:{b31,b30,...,b1,b0} ← GReg[r]:{b31,b31,b30,...,b1}
```

Assembler:

```
Syntax: asr1 r

Example: asr1 3
```

divide by 2 the signed value of GReg[3] and store the result in GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Shift the bits of any general register to the right and keep the same sign: The left bit (bit 31) is kept untouched.

Instruction Format:

Table 66-47. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	1	1	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.8 BCLRI1 (Bit Clear Immediate)

Operation:

```
GReg[r]:{b31,...,b(i+1),0,b(i-1),...,b0} ← GReg[r]:{b31,...,b(i+1),b(i),b(i-1),...,b0}
```

Assembler:

```
Syntax: bclri r,i

Example: bclri 1,12
```

clear bit 12 in GReg[1]

CPU Flags: Unaffected

Cycles: 1

Description: Clear the bit of register r specified by the 5-bit immediate field

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	1	i	i	i	i	i

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiii - immediate value:

```
00000 - 0

00001 - 1

...

11110 - 30

11111 - 31
```

66.17.2.9 BDF (Conditional Branch if Destination Fault)

Operation:

```
if (DF == 1) PC ← PC + 1 + displacement else PC ← PC + 1
```

Assembler:

```
Syntax:bdf label

Example: bdf LLL
```

Jump to LLL if DF is set, or go to the next instruction if DF is cleared; the displacement value is calculated by the assembler.

CPU Flags: Unaffected

Cycles: 2 when the branch is done, 1 otherwise

Description: If flag DF is set, jump to the new address that is calculated by adding the sign-extended 8-bit displacement to the next PC address. If flag DF is cleared, no jump is performed: The next instruction is located at the next PC address.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	p	p	p	p	p	p	p	p

Instruction Fields:

pppppppp - signed displacement field:

```
00000000 - 0

00000001 - 1

...

01111110 - 126

01111111 - 127

10000000 - (-128)

10000001 - (-127)

...

11111110 - (-2)

11111111 - (-1)
```

66.17.2.10 BF (Conditional Branch if False)

Operation:

```
if (T == 0)

PC ← PC + 1 + displacement

else

PC ← PC + 1
```

Assembler:

```
Syntax: bf label

Example: bf LLL
```

Jump to LLL if T is cleared, or go to the next instruction if T is set. The displacement value is calculated by the assembler.

CPU Flags: Unaffected

Cycles: 2 when the branch is done, 1 otherwise

Description: Conditional branch: If flag T is cleared, jump to the new address that is calculated by adding the sign-extended 8-bit displacement to the next PC address. If flag T is set, no jump is performed: The next instruction is located at the next PC address.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	0	0	p	p	p	p	p	p	p	p

Instruction Fields:

pppppppp - signed displacement field:

```
00000000 - 0

00000001 - 1

...

01111110 - 126

01111111 - 127

10000000 - (-128)

10000001 - (-127)

...

11111110 - (-2)

11111111 - (-1)
```

66.17.2.11 BSETI (Bit Set Immediate)

Operation:

```
GReg[r]:{b31,...,b(i+1),1,b(i-1),...,b0} ← GReg[r]:{b31,...,b(i+1),b(i),b(i-1),...,b0}
```

Assembler:

```
Syntax: bseti r,i

Example: bseti 6,5
```

Set bit 5 in GReg[6]

CPU Flags: Unaffected

Cycles: 1

Description: Sets bit number i in the selected General Register.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	1	0	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiii - bit number field:

```
00000 - 0

00001 - 1

...

11110 - 30

11111 - 31
```

66.17.2.12 BSF (Conditional Branch if Source Fault)

Operation:

```
if (SF == 1) PC ← PC + 1 + displacement else PC ← PC + 1
```

Assembler:

```
Syntax: bsf label

Example: bsf LLL
```

Jump to LLL if SF is set, or go to the next instruction if SF is cleared. The displacement value is calculated by the assembler.

CPU Flags: Unaffected

Cycles: 2 when the branch is done, 1 otherwise

Description: Conditional branch: If flag SF is set, jump to the new address that is calculated by adding the sign-extended 8-bit displacement to the next PC address. If flag SF is cleared, no jump is performed: The next instruction is located at the next PC address.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	0	p	p	p	p	p	p	p	p

Instruction Fields:

pppppppp - signed displacement field:

```
00000000 - 0

00000001 - 1

...

01111110 - 126

01111111 - 127

10000000 - (-128)

10000001 - (-127)

...

11111110 - (-2)

11111111 - (-1)
```

66.17.2.13 BT (Conditional Branch if True)

Operation

```
if (T == 1)

PC ← PC + 1 + displacement

else

PC ← PC + 1
```

Assembler

```
Syntax: bt label

bt LLL
```

Jump to LLL if T is set, or go to the next instruction if T is cleared. The displacement value is calculated by the assembler.

CPU Flags: Unaffected

Cycles: 2 when the branch is done, 1 otherwise

Description: Conditional branch: If flag T is set, jump to the new address that is calculated by adding the sign-extended 8-bit displacement to the next PC address. If flag T is cleared, no jump is performed: The next instruction is located at the next PC address.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	0	1	p	p	p	p	p	p	p	p

pppppppp - signed displacement field:

```
00000000 - 0

00000001 - 1

...

01111110- 126

01111111 - 127

10000000 - (-128)

10000001 - (-127)

...

11111110 -  (-2)

11111111 - (-1)
```

66.17.2.14 BTSTI (Bit Test immediate)

Operation:

```
T ← GReg[r]:b(i)
```

Assembler:

```
Syntax: btsti r,i

Example: btsti 2,29
```

Test bit 29 in GReg[2] and copy its value in flag T

CPU flags: T

Cycles: 1

Description: T is loaded with the value of bit number i from the selected general register.

Instruction Format:

Table 66-48. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	1	1	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiii - bit number field:

```
0000 - 0

0001 - 1

...

11110 - 30
```

```
11111 - 31
```

66.17.2.15 CLRF (Clear ARM platform flags)

Operation:

```
 if (ff%2 == 0)

SF ← 0

if (ff/2 == 0)

DF ← 0
```

Assembler:

```
Syntax: clrf ff

Example: clrf 2
```

Clear flag SF and keep flag DF unchanged

CPU Flags: SF, DF

Cycles: 1

Description: Clears a selection of the ARM platform fault flags: SF, DF, both SF and DF or none can be cleared.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	f	f	0	0	0	0	0	1	1	1

Instruction Fields:

ff - flags field:

```
00 - clear SF and clear DF

01 - clear DF

10 - clear

SF 11 - no clear
```

66.17.2.16 CMPEQ (Compare for Equal)

Operation:

```
T ← (GReg[s] == GReg[r])
```

Assembler:

```
Syntax: cmpeq r,s

Example: cmpeq 7,5
```

Compare GReg[7] and GReg[5] and set flag T if they are equal

CPU flags: T

Cycles: 1

Description: Subtracts the destination general register *r* from the source general register *s*, and sets T if the result is 0, clears T if the result is not 0.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	1	0	0	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.17 CMPEQI (Compare with Immediate for Equal)

Operation:

```
T ← (GReg[r] == immediate)
```

Assembler:

```
Syntax: cmpeqi r,immediate

Example: cmpeqi 2,13
```

Compare GReg[2] and decimal value 13 and set flag T if they are equal

CPU Flags: T

Cycles: 1

Description: Subtracts the 0-extended 8-bit immediate value from the general register, and sets T if the result is 0, clears T if the result is not 0. The immediate value is the low-order byte of the instruction.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - destination register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

66.17.2.18 CMPHS (Compare for Higher or Same)

Operation:

```
T ← (GReg[r] ≥ GReg[s])
```

Assembler:

```
Syntax: cmphs r,s

Example: cmphs 0,1
```

Compare GReg[0] and GReg[1] and set flag T if GReg[0] is higher than or equal to GReg[1]

CPU Flags: T

Cycles: 1

Description: Compares the destination general register *r* and the source general register *s*, and sets T if the destination general register *r* is higher than or equal to the source general register *s*, clears T otherwise. The comparison is unsigned.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	1	0	1	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.19 CMPLT (Compare for Less Than)

Operation:

```
T ← (GReg[r] < GReg[s])
```

Assembler:

```
Syntax: cmplt r,s

Example: cmplt 7,4
```

Compare GReg[7] and GReg[4] and set flag T if GReg[7] is lower than GReg[4]

CPU Flags: T

Cycles: 1

Description: Compares the destination general register *r* and the source general register *s*, and sets T if the destination general register *r* is lower than the source general register *s*, clears T otherwise. The comparison is signed.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	1	0	1	0	s	s	s

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.20 cpShReg (Update Context of PCU Registers and Flag)

Assembler:

```
Syntax: cpShReg
```

CPU Flags: none

Cycles: 1

Description: SF, RPC, T, PC,LM, EPC, DF, and SPC registers are updated according to the value of their corresponding bits in the context memory. This instruction must only be used in debug mode via the OnCE. It reverses the done *5* operation.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	0	1	1	1	0	0	0	1	0

66.17.2.21 DONE (DONE, Yield)

Operation:

```
if (jjj&6 == 2) HE[CCR] ← 0

if (jjj == 3) HI[CCR] ← 1

if (jjj == 4) EP[CCR] ← 0


if ((jjj == 0) && (NCP > CCP)) CCR ← NCR

else if ((jjj == 1) && (NCP >= CCP))

CCR ← NCR

else

CCR ← NCR
```

(CCR stands for Current Channel Register; NCR stands for Next Channel Register)

Assembler:

```
Syntax: done jjj

Example: done 3
```

Clear HE bit for the current channel, send an interrupt to the ARM platform for the current channel and reschedule.

CPU Flags: Unaffected

Cycles: Variable if a context switch is done, 1 otherwise

Description: Clears one of the channel enabling bits (HE or EP for the corresponding channel number) if required. Sends an interrupt to the corresponding ARM platform by setting the appropriate flag, if required (HI for the corresponding channel number). Reschedules according to the mode and the NCP (Next Channel Priority) and CCP (Current Channel Priority) values. According to the scheduling decision, the NCR (Next Channel Register) is copied to the CCR (Current Channel Register) and channel contexts are switched. If several channels with the same highest priority are pending, they are ordered by their number from 31 down to 0. The higher number is selected (for example, channel 26 is selected if channels 3, 12, 14, and 26 with the same highest priority are pending). If no flag is modified, the reschedule can allow the replacement of the current channel by another channel with a priority strictly greater than the current channel priority (yield). Or, it can allow the replacement of the current channel by another channel with a priority greater than or equal to the current channel priority (yieldge). In the latter case, the selected channel will always be the first one with the same priority, starting from channel number 31 down to channel 0 (the current channel does not belong to the set of selectable channels).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	j	j	j	0	0	0	0	0	0	0	0

jjj - Channel Flags field:

000 - No channel flags affected: Reschedule only if the next channel priority is greater than current channel priority (yield)

001 - No channel flags affected: Reschedule only if the next channel priority is greater than or equal to the current channel priority (yieldge)

010 - Clear HE for the current channel and reschedule 011 - Clear HE, set HI for the current channel and reschedule 100 - Clear EP for the current channel and reschedule

101 - Reserved for debug to copy relevant registers into context memory

110 - RESERVED

111 - RESERVED

For the scheduling rules, refer to Scheduler Functional Description. Every possible done instruction is further described as follows:

- done 0/yield is executed by a channel script when it accepts preemption by a higher priority channel;
- done 1/yieldge is executed by a channel script when it accepts preemption by a higher priority channel and it also accepts a roll-up with other channels that have the same priority;
- done 2 is executed by a channel script that was triggered by a ARM platform start via the Channel Start (SDMAARM_HSTART) register, when its task is completed and it requires termination;
- done 3 is executed by a channel script that was triggered by a ARM platform start via the Channel Start (SDMAARM_HSTART) register, when its task is completed, it requires termination and it needs to trigger an interrupt to the ARM platform upon closure;
- done 4 is executed by a channel script that was triggered by a DMA request, when its task is completed and it requires termination;
- done 5 is used in debug mode only; it copies the PCU registers and flags to the context memory of the current channel;

66.17.2.22 ILLEGAL (ILLEGAL Instruction)

Operation:

```
PC ← 0001
```

Assembler:

```
Syntax: illegal
```

CPU Flags: Unaffected

Cycles: 2

Description: Jumps to the Illegal instruction routine located at address 0001. All unauthorized instructions result in an Illegal instruction behavior; however, the ILLEGAL instruction must be used to guarantee software compatibility with future versions of the SDMA.

Instruction Format

Table 66-49. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

66.17.2.23 JMP (Unconditional Jump Immediate)

Operation:

```
PC ← absolute_address
```

Assembler:

```
Syntax: jmp label

Example: jmp LLL
```

The assembler translates the label to the exact address

CPU Flags:Unaffected

Cycles: 2

Description: Jumps to the absolute address contained the lower 14 bits of the instruction (the PC is a 14-bit register).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	a	a	a	a	a	a	a	a	a	a	a	a	a	a

aaaaaaaaaaaaa - address field:

```
00000000000000 - 0

00000000000001 - 1
```

```
...

11111111111110 - 16382

11111111111111 - 16383
```

66.17.2.24 JMPR (Unconditional Jump)

Operation:

```
PC ← GReg[r]
```

Assembler:

```
Syntax: jmpr r

Example: jmpr 0
```

Jump to address stored in GReg[0]

CPU Flags: Unaffected

Cycles: 2

Description: Jumps to the absolute address contained in a General Register.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	0	1	0	0	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.25 JSR (Unconditional Jump to Subroutine Immediate)

Operation:

```
RPC ← PC + 1

PC ← absolute_address
```

Assembler:

```
Syntax: jsr r

Example:jsr LLL
```

Jumps to subroutine starting at LLL; the assembler translates the label to exact address

CPU Flags: Unaffected

Cycles: 2

Description: Jumps to the subroutine located at the absolute address contained the lower 14 bits of the instruction (the PC is a 14-bit register).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	a	a	a	a	a	a	a	a	a	a	a	a	a	a

aaaaaaaaaaaaaa - address field:

```
00000000000000 - 0

00000000000001 - 1

...

11111111111110 - 16382

11111111111111 - 16383
```

66.17.2.26 JSRR (Unconditional Jump to Subroutine)

Operation:

```
RPC ← PC + 1

PC ← GReg[r]
```

Assembler:

```
Syntax: jsrr r

Example:jsrr 5
```

Jumps to subroutine located at address stored in GReg[5]

CPU Flags: Unaffected

Cycles: 2

Description: Jumps to the subroutine at address contained in a General Register

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	0	1	0	0	1

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.27 LD (Load Register)

Operation:

```
GReg[r] ← [GReg[b] + displacement]

if (transfer_error)

SF ← 1

else

SF ← 0
```

Assembler:

```
Syntax: ld r,(b,displacement)

Example: ld 1,(2,23)
```

Loads data into GReg[1]; the data is located at address obtained by adding decimal value 23 to GReg[2]

CPU Flags: SF

Cycles: 2+n where n is 0 for ROM, RAM or memory mapped registers, and n is the number of wait-states of the peripheral for a peripheral access

Description: Adds a 5-bit 0-extended displacement to a base address in General Register b; the result is the address of the data to fetch on the DM bus. The data received from the bus is stored in the destination General Register r. If an error occurs during the transfer, the flag SF is set, else it is cleared.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	0	r	r	r	d	d	d	d	d	b	b	b

rrr / bbb - register field:

```
000 - GReg[0]

001 - GReg[1]

...

111 - GReg[7]
```

ddddd - displacement value:

```
00000 - 0

00001 - 1

...

11111 - 31
```

66.17.2.28 LDF (Load Register from Functional Unit)

Operation:

```
GReg[r] ← [fu_address]

if (transfer_error)

SF ← 1

else

SF ←0
```

fu_address is an 8-bit field and depends on addressed functional unit

Assembler:

```
Syntax: ldf r,fu_address

Example: ldf 0,13
```

Loads data coming from the Burst DMA register MD into GReg[0]; it is a 32-bit access with no prefetch

CPU Flags: SF

Cycles: 1+n where n is the number of wait-states that may be inserted by the functional unit

Description: Sends an 8-bit address on the Functional Unit Bus (FU bus) and stores the data received from the bus in the destination General Register r. If an error occurs during the transfer, the flag SF is set, else it is cleared.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	r	r	r	f	f	f	f	f	f	f	f

See the following sections for more details of the LDF instruction usage with each functional unit:

- Burst DMA Read (ldf) for Burst DMA
- Peripheral DMA Read (ldf)-Read Mode for Peripheral DMA
- Read Instruction (ldf) for CRC unit

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

ffffffff - functional unit source register and action (unspecified values are reserved):

```
00000000 - MSA

00000100 - MDA

00001001 - MD byte

00001010 - MD halfword

00001011 - MD word
```

```
00001100 - MS

00101001 - MD byte - prefetch

00101010 - MD halfword - prefetch

00101011 - MD word - prefetch

01000000 - DSA


10000000 - CA

10000100 - CS (CRC checksum)

11000000 - PSA

11001000 - PD

11010000 - PDA

11011000 - PD in copy mode (rrr contents are lost)

11101000 - PD - prefetch next data

11111111 - PS
```

66.17.2.29 LDI (Load Register with Immediate Value)

Operation:

```
GReg[r] ← immediate
```

Assembler:

```
Syntax: ldi r,immediate

Example: ldi 6,1
```

loads decimal value 1 into GReg[6]

CPU Flags: Unaffected

Cycles: 1

Description: Stores a 0-extended immediate value in a General Register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

66.17.2.30 LDRPC (Load from RPC to Register)

Operation:

```
GReg[r] ← RPC
```

Assembler:

```
Syntax: ldrpc r

Example: ldrpc 3
```

copies RPC to GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Stores the contents of the RPC in a General Register. That instruction may be used to have more than one level of subroutines.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	0	1	0	1	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.31 LOOP (Hardware Loop)

Operation:

```
if (ff%2 == 0)
   SF ← 0
if (ff/2 == 0)
  DF ← 0
if ((GReg[0] == 0) || (SF == 1) || (DF == 1))
   PC ← PC + loop_size + 1
else
   {
      SPC ← PC + 1
      EPC ← PC + loop_size + 1
      LM ← 1
      PC ← PC + 1
   }
```

during every instruction execution in the loop:

```
if ((SF == 1) || (DF == 1))
 {
    LM ← 0
    PC ← EPC
  }
else if ((PC + 1) == EPC)
```

```
 {
    GReg[0] ← GReg[0] - 1
    if (GReg[0] == 0)
      {
         LM ← 0
         PC ← EPC
       }
     else
         PC ← SPC
   }
else
  PC ← nextPC(instruction)
```

after the execution of the last instruction of the loop body:

```
if (GReg[0] == 0)
   T ← 1
else
   T ← 0
```

Assembler:

```
Syntax: loop n{,ff}
Example: loop 3,1
```

Executes GReg[0] times the instructions comprised between PC+1 and PC+3 (included); ff=1 clears the DF flag before starting the loop. When omitted, the ff field is set to 0 (clearing both SF and DF).

CPU Flags: LM[1:0], T

Cycles: 2 when the loop count (GReg[0]) is 0 or SF or DF is set at loop start, 1+1 when the loop starts but exits abnormally (SF or DF set inside the loop which adds 1 cycle to the offending load or store to jump to EPC), 1 when the loop is executed normally

Description: The loop instruction executes a sequence of instructions several times. The number of times is given by the contents of GReg[0], the loop counter. SDMA will jump to the first instruction after the end of the loop if the value in GReg[0] is 0. Otherwise the SDMA enters loop mode. It sets the most significant bit of the LM flag that will only be reset once the last instruction of the last loop is executed. The instructions in the loop are executed GReg[0] times. The management of fault flags (SF and DF) is as follows. When entering the hardware loop, SF and DF can be cleared according to the ff field of the

instruction. After that operation, if any flag is still set the loop will not be executed. The SDMA will jump to the first instruction after the end of the loop without entering loop mode. During the execution of the loop, if any fault flag is set by a LD, LDF, ST, or STF instruction, the SDMA will immediately exit loop mode and jump to the first instruction after the end of the loop. In that case, GReg0 is not decremented for that last piece of the loop body execution (even if the SF or DF flag is set at the last instruction of the loop body). The T flag reflects the state of GReg[0] after the end of the loop, which is an indicator of the complete execution of the loop. If the loop exited because of an error (SF or DF set), GReg[0] will not be 0 at the end of the loop, hence T will be cleared. If the loop executes without fault, GReg[0] will be 0 at the end of the loop, hence T will be set. The boundary case when a source or destination fault occurs at the last instruction of the last loop is considered as an anticipated exit of the loop, which causes the T flag to be cleared. If the last instruction executed before leaving the hardware loop also tries to modify the T flag, the flag is updated according to the value of GReg[0], NOT according to the result of the last executed instruction.

Limitations:

1. 1. Jump instructions (JMP, JMPR, JSR, JSRR, BF, BT, BSF, BDF) are not allowed inside the hardware loop.
2. 2. GReg[0] cannot be written to inside the hardware loop (it can be read).
3. 3. The empty loop (0 instruction in the body) is forbidden.
4. 4. If GReg[0] == 0 at the start of the loop, which causes a jump to EPC, the T flag is not updated.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	0	f	f	n	n	n	n	n	n	n	n

Instruction Fields:

```
ff - flags field:

00 - clear SF and clear DF

01 - clear DF

10 - clear SF

11 - no clear
```

nnnnnnnn - loop size

```
00000000 - empty loop: forbidden value

00000001 - 1 instruction in the loop

00000010 - 2 instructions in the loop
```

```
...

11111111 - 255 instructions in the loop
```

66.17.2.32 LSL1 (Logical Shift Left by 1 Bit)

Operation:

```
GReg[r]:{b30,...,b1,b0,0} ← GReg[r]:{b31,b30,...,b1,b0}
```

Assembler:

```
Syntax: lsl1 r

Example: lsl1 2
```

multiplies by 2 the value in GReg[2]

CPU Flags: Unaffected

Cycles: 1

Description: Shift the bits of any General Register to the left. The right bit (bit 0) is set to 0. No overflow is detected by the hardware.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	1	1	1

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.33 LSR1 (Logical Shift Right by 1 Bit)

Operation:

GReg[r]:{0,b31,b30,...,b1} ← GReg[r]:{b31,b30,...,b1,b0}

Assembler:

Syntax: lsr1 r

Example: lsr1 4

divides by 2 the unsigned value contained in GReg[4]

CPU Flags: Unaffected

Cycles: 1

Description: Shift the bits of any General Register to the right. The left bit (bit 31) is set to 0.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	1	0	1

Instruction Fields:

rrr - destination register field:

000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]

66.17.2.34 MOV (Logical Move)

Operation:

GReg[r] ← GReg[s]

Assembler:

Syntax: mov r,s

Example: mov 4,0

copies GReg[0] to GReg[4]

CPU Flags: Unaffected

Cycles: 1

Description: Move the contents of the source General Register s to the destination General Register r.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	0	0	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.35 NOTIFY (Notify to ARM platform)

Operation:

```
 if (jjj & 4 == 0)
 {
   if (jjj&2 == 2)
      HE[CCR] ← 0
   if (jjj&1== 1)
       HI[CCR] ← 1
  }
else if (jjj == 4)
   EP[CCR] ← 0
else
```

(CCR stands for Current Channel Register)

Assembler:

```
Syntax: notify jjj

Example: notify 3
```

clears the HE bit for the current channel and sends an interrupt to the Host for the current channel

CPU Flags: Unaffected

Cycles: 1

Description: Clears one of the channel enabling bits (HE or EP for the corresponding channel number) if required, sends an interrupt to the corresponding ARM platform by setting the appropriate flag if required (HI for the corresponding channel number).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	j	j	j	0	0	0	0	0	0	0	1

jjj - Channel Flags field:

```
000 - unused

001 - set HI for the current channel

010 - clear HE for the current channel

011 - clear HE, set HI for the current channel

100 - clear EP for the current channel

101 - RESERVED

110 - RESERVED

111 - RESERVED
```

66.17.2.36 OR (Logical OR)

Operation:

```
GReg[r] ← GReg[s] | GReg[r]
```

Assembler:

```
Syntax: or r,s

Example: or 3,6
```

ORs GReg[3] and GReg[6] and stores the result in GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Performs the OR of the source General Register s and the destination General Register r, and stores the result in the destination General Register r.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	1	0	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.37 ORI (Logical OR with Immediate Value)

Operation:

```
GReg[r] ← GReg[r] | immediate
```

Assembler:

```
Syntax: ori r,immediate

Example: ori 1,56
```

ORs GReg[1] and the decimal value 56 and stores the result in GReg[1]

CPU Flags: unaffected

Cycles: 1

Description: Performs an OR between a 0-extended 8-bit immediate value and a General Register; stores the result in the General Register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

66.17.2.38 RET (Return from Subroutine)

Operation:

```
PC ← RPC
```

Assembler:

```
Syntax: ret
```

CPU Flags: Unaffected

Cycles: 2

Description: Return from subroutine.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

66.17.2.39 REVB (Reverse Byte Order)

Operation:

```
GReg[r]:{B3,B2,B1,B0} ← GReg[r]:{B0,B1,B2,B3}
```

Assembler:

```
Syntax: revb r

Example:  revb 5
```

reverses bytes order in GReg[5]

CPU Flags: Unaffected

Cycles: 1

Description: Reverse the byte order of any General Register.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	0	0	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.40 Reverse Low Order Bytes(REVBLO)

Operation:

```
GReg[r]:{B3,B2,B0,B1} ← GReg[r]:{B3,B2,B1,B0}
```

Assembler:

```
Syntax: revblo r

Example: revblo 0
```

reverses low order bytes in GReg[0]

CPU Flags: Unaffected

Cycles: 1

Description: Reverse both low order bytes of any General Register.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	0	0	1

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.41 ROR1 (Rotate Right by 1 Bit)

Operation:

```
GReg[r]:{b0,b31,b30,...,b1} ← GReg[r]:{b31,b30,...,b1,b0}
```

Assembler:

```
Syntax: ror1 r

Example: ror1 3
```

rotates bits to the right in GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Rotate the bits of any General Register to the right.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	1	0	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.42 RORB (Rotate Right by 1 Byte)

Operation:

```
GReg[r]:{B0,B3,B2,B1} ← GReg[r]:{B3,B2,B1,B0}
```

Assembler:

```
Syntax: rorb r

Example: rorb 2
```

rotates bytes to the right in GReg[2]

CPU Flags: Unaffected

Cycles: 1

Description: Rotate the bytes of any General Register to the right.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	0	1	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.43 SOFTBKPT (Software Breakpoint)

Operation:

Stops the current script and enters debug mode

Assembler:

```
softbkpt
```

CPU Flags: Unaffected

Description: When the core executes this instruction, it has the same effect as receiving a debug request from the OnCE or via the external debug request input: the script execution halts, the PCU enters its debug state and waits for the OnCE commands that are described in OnCE and Real-Time Debug.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

66.17.2.44 ST (Store Register)

Operation:

```
[GReg[b] + displacement] ← GReg[r]

if (transfer_error)

 DF ← 1

 else

DF ← 0
```

Assembler:

```
Syntax: st r,(b,displacement)

Example: st 7,(0,9)
```

stores the value from GReg[7] into memory at address obtained by adding decimal value 9 to GReg[0]

CPU Flags: DF

Cycles: 2+n where n is 0 for ROM, RAM or memory mapped registers, and n is the number of wait-states of the peripheral for a peripheral access

Description: Adds a 5-bit 0-extended displacement to a base address in General Register b; the result is the address of the data to store on the DM bus. The data sent on the bus comes from the source General Register r. If an error occurs during the transfer, the flag DF is set, else it is cleared.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	1	r	r	r	d	d	d	d	d	b	b	b

Instruction Fields:

rrr / bbb - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]
```

```
111 - GReg[7]
```

ddddd - displacement value:

```
00000 - 0

00001 - 1

...

11111 - 31
```

66.17.2.45 STF (Store Register in Functional Unit)

Operation:

```
[fu_address] ← GReg[r] 0

if (transfer_error) 0

DF ← 1 0

else 0

DF ← 0
```

fu_address is an 8-bit field

Assembler:

```
Syntax: stf r,fu_address

Example: stf 3,0x2B
```

stores the 32-bit contents of GReg[3] to the Burst DMA register MD; waits until the flush to external memory is completed

CPU Flags: DF

Cycles: 1+n where n is the number of wait-states that may be inserted by the functional unit

Description: Sends an 8-bit address on the Functional Unit Bus (FU bus) and sends the contents of the source General Register r on the bus. If an error occurs during the transfer, the flag DF is set, else it is cleared.

Table 66-50. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	1	r	r	r	f	f	f	f	f	f	f	f

See the following sections for more details of the STF instruction usage with each functional unit:

- Burst DMA Write (stf) for Burst DMA
- Peripheral DMA Write (stf)-Write Mode for Peripheral DMA
- Write Instruction (stf) for CRC

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

ffffffff - functional unit destination register and action (unspecified values are reserved):

00000000 - MSA in incremented mode

00000100 - MDA in incremented mode

00001001 - MD byte

00001010 - MD halfword

00001011 - MD word

00001100 - clear MS error flag

00001111 - MS

00010000 - MSA in frozen mode

00010100 - MDA in frozen mode

00011000 - MD in copy mode - number of words in rrr

00100000 - MSA in incremented mode - start prefetch

00101000 - MD no data - flush

00101001 - MD byte - flush

00101010 - MD halfword - flush

00101011 - MD word - flush

00110000 - MSA in frozen mode - start prefetch

10000000 - CA

10000100 - init CRC accumulator

10010100 - CS byte - compute CRC

10010101 - CS byte - compute CRC

10010110 - CS halfword - compute CRC

10010111 - CS word - compute CRC

11000001 - PSA in frozen mode - 8-bit data width

11000010 - PSA in frozen mode - 16-bit data width

11000011 - PSA in frozen mode - 32-bit data width

11000101 - PSA in incremented mode - 8-bit data width

11000110 - PSA in incremented mode - 16-bit data width

11000111 - PSA in incremented mode - 32-bit data width

11001000 - PD

11001001 - PSA in decremented mode - 8-bit data width

11001010 - PSA in decremented mode - 16-bit data width

11001011 - PSA in decremented mode - 32-bit data width

11001100 - clear PS error flag

11001101 - PSA data width becomes 8-bit

11001110 - PSA data width becomes 16-bit

11001111 - PSA data width becomes 32-bit

11010001 - PDA in frozen mode - 8-bit data width

11010010 - PDA in frozen mode - 16-bit data width

11010011 - PDA in frozen mode - 32-bit data width

11010101 - PDA in incremented mode - 8-bit data width

11010110 - PDA in incremented mode - 16-bit data width

11010111 - PDA in incremented mode - 32-bit data width

11011001 - PDA in decremented mode - 8-bit data width

11011010 - PDA in decremented mode - 16-bit data width

11011011 - PDA in decremented mode - 32-bit data width

11011101 - PDA data width becomes 8-bit

11011110 - PDA data width becomes 16-bit

11011111 - PDA data width becomes 32-bit

11100001 - PSA in frozen mode - 8-bit data width - prefetch data

11100010 - PSA in frozen mode - 16-bit data width - prefetch data

11100011 - PSA in frozen mode - 32-bit data width - prefetch data

11100101 - PSA in incremented mode - 8-bit data width - prefetch data

11100110 - PSA in incremented mode - 16-bit data width - prefetch data

11100111 - PSA in incremented mode - 32-bit data width - prefetch data

11101001 - PSA in decremented mode - 8-bit data width - prefetch data

11101010 - PSA in decremented mode - 16-bit data width - prefetch data

11101011 - PSA in decremented mode - 32-bit data width - prefetch data

11101101 - PSA data width becomes 8-bit - prefetch data

11101110 - PSA data width becomes 16-bit - prefetch data

11101111 - PSA data width becomes 32-bit - prefetch data

11111111- PS

66.17.2.46 SUB (Subtract)

Operation:

```
GReg[r] ← GReg[r] - GReg[s]

T ← (GReg[r] == 0)
```

Assembler:

```
Syntax: sub r,s

Example: sub 4,7
```

SUBtracts GReg[7] from GReg[4] and stores the result in GReg[4]

CPU Flags: T

Cycles: 1

Description: Subtracts the source General Register s from the destination General Register r, and stores the result in the destination General Register r. The T flag is set if the result of the operation is 0; it is cleared if the result is not 0.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	1	0	0	s	s	s

Instruction Fields:

rrr / sss - register fields:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.47 SUBI (Subtract with Immediate)

Operation:

```
GReg[r] ← GReg[r] - immediate

T ← (GReg[r] == 0)
```

Assembler:

```
Syntax: sub r,immediate

Example: sub 1,255
```

SUBtracts decimal value 255 from GReg[1] and stores the result in GReg[1]

CPU Flags: T

Cycles: 1

Description: Subtracts a 0-extended 8-bit immediate value from a General Register; stores the result in the General Register. The flag T is set when the result of the operation is 0; otherwise, it is cleared. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

66.17.2.48 TST (Test with Zero)

Operation:

```
T ← ((GReg[s] & GReg[r]) != 0)
```

Assembler:

```
Syntax: tst r,s

Example: tst 2,3
```

ANDs GReg[2] and GReg[3] and sets T if the result is non-null

CPU Flags: T

Cycles: 1

Description: Performs the AND of the source General Register s and the destination General Register r, and sets T if the result is not 0, clears T if the result is 0.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	1	0	0	0	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.49 TSTI (Test Immediate)

Operation:

```
T ← ((GReg[r] & immediate) != 0)
```

Assembler:

```
Syntax: tsti r,immediate

Example: tsti 5,13
```

ANDs GReg[5] and decimal value 13 and sets T if the result is non-null

CPU Flags: T

Cycles: 1

Description: Performs the AND of a 0-extended 8-bit immediate value and the destination General Register r, and sets T if the result is not 0, clears T if the result is 0. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - destination register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

66.17.2.50 XOR (Logical Exclusive OR)

Operation:

```
GReg[r] ← GReg[s] ^ GReg[r]
```

Assembler:

```
Syntax: xor r,s

Example: xor 0,3
```

XORs GReg[0] and GReg[3] and stores the result in GReg[0]

CPU Flags: Unaffected

Cycles: 1

Description: Performs the eXclusive OR of the source General Register s and the destination General Register r, and stores the result in the destination General Register r.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	0	1	0	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

66.17.2.51 XORI (Exclusive OR with Immediate)

Operation:

```
GReg[r] ← GReg[r] ^ immediate
```

Assembler:

```
Syntax: xori r,immediate

Example: xor 7,5
```

XORs GReg[5] and decimal value 5 and stores the result in GReg[7]

CPU Flags: Unaffected

Cycles: 1

Description: Performs an eXclusive OR between a 0-extended 8-bit immediate value and a General Register; stores the result in the General Register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	0	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

66.17.2.52 YIELD, YIELDGE (DONE, Yield)

By default, unsupported assembler syntax. Can be aliased to the corresponding done instructions (yield = done 0; yieldge = done 1). Refer to the done instruction description DONE (DONE, Yield) .

66.18 Software Restrictions

66.18.1 Unsupported Burst DMA Access Sequence

The SDMA does not support triggering a pre-fetch followed by a flush of the Burst DMA without reading or writing any data. If the flush occurs while the background pre-fetch DMA operation is still in progress, it could result in un-defined behavior.

An example of the sequence which could result in undefined results is shown in the following example:

Instruction sequence not supported

```
          stf r1, MSA|PF              ; Update source address, triggers data pre-fetch in the
background
          mov R0,R0                ; Execute multiple assembly instructions, none of which
read
          mov R0,R0                ; or write data to/from MD
          stf MD|SZ0|FL               ; Flush FIFO without writing data. If the pre-fetch is
still in
                                  ; progress when this instruction is executed, there could be
```

; undefined operation.

A work-around to avoid any undesirable results is to first read MD to ensure the pre-fetch is complete before the flush is attempted.

Work-Around to previous example

```
          stf r1, MSA|PF              ; Update source address, triggers data pre-fetch
          mov R0,R0                ; Execute multiple assembly instructions, none of which
read
          mov R0,R0                ; or write data to/from MD
          ldf r2, MD                  ; dummy read of MD to ensure pre-fetch is complete before
the
                                  ; next instruction
```

stf MD|SZ0|FL ; Flush FIFO without writing data.

66.19 Application Notes

66.19.1 Data Structures for Boot Code and Channel Scripts

SDMA boot code downloads the different channel contexts and the scripts that will be executed on SDMA channels during the application. The boot code is run after reset when channel 0 is started by the ARM platform.

The boot code is also known as channel 0 script.

The boot code is based on the Channel Control Block (CCB) and Buffer Descriptor (BD) mechanisms that are data structures located into the ARM platform memory space. With these data structures, it is possible to instruct SDMA to download scripts and contexts but also to dump a context or a script to a destination data buffer. Channel scripts also use the CCB and BD data structures to pass instructions and/or pointers to data to be copied.

The format, processing, and field definition of the CCB and BD are defined and performed entirely by the software script rather than the SDMA hardware. An overview of the format and structure is provided here, but for complete details refer to the SDMA software documentation (see SDMA Scripts).

The CCB and BD data structures are accessed by SDMA using DMA and processed by the SDMA scripts. The ROM contains common sub-routines for processing these data structures which may be called by the bootload and channel scripts.

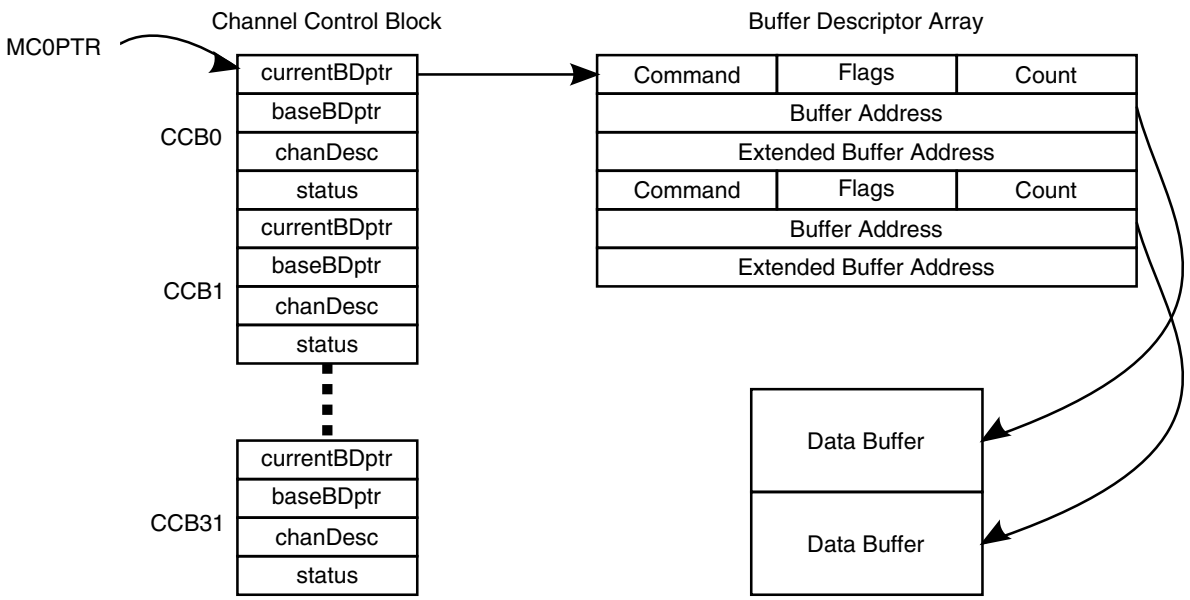


Figure 66-17. Data Structures Layout

The previous figure shows an example how these data structures are linked to pass command and pointers to data buffers. The SDMA's MC0PTR register holds the base address of the Channel 0 Control Block (CCB0). The Channel 0 control block holds a pointer to the array of buffer descriptors. The buffer descriptors are used to tell the channel 0 (boot channel) what to do as described Buffer Descriptor Format.

66.19.1.1 Buffer Descriptor Format

Buffer descriptors are three longs (32-bit words) in size as, shown in the following figure. A buffer descriptor describes the properties of the data buffer it points to. The buffer descriptors can be used for linear or circular data buffers in the ARM platform processor memory. The CCB contains a pointer to the base BD as well as the current BD.

Table 66-51. Buffer Descriptor

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Command								-	-	L	R	I	C	W	D	Count															
Buffer Address																															
Extended Buffer Address																															

Table 66-52. Buffer Descriptor Field Descriptions

Field	Description
31-24 Command	Command. The command field is used to differentiate operations performed within a script when the script accesses this particular buffer descriptor. The use of this field can be defined by the script. The command values defined for the bootload script are defined in Buffer Descriptor Commands for Bootload scripts. Refer to the individual script definition in script library documents in SDMA Scripts for command field definitions for other scripts.
23	Reserved
22	Reserved
21 L	Last Buffer Descriptor: This bit is set in SDMA IPC scripts to indicate to the receiving Core that the transfer has ended. Whenever the source finishes transferring the count it wanted to transfer, it sets LAST_BIT in the destination BD, to let the destination know that transfer is over.This bit also tells the destination software that when it processes the destination BDs, they need not process any BD after the BD with the LAST_BIT set.For example, when the DSP prepares a single buffer descriptor with count equals to 25 and ARM platform prepares a single buffer descriptor with count equals 100. When 25 bytes have been transferred from DSP to ARM platform, the DSP buffer descriptor is normally closed while the ARM platform buffer descriptor will have the L bit set and the byte count updated to 25.
20 R	erroR. Indicates an error occurred on the channel's buffer descriptor requested command. Some scripts may overwrite the command field with an error code indicating the source of the error. 0 No Error 1 Error
19 I	Interrupt. When SDMA has finished to process data transfer attached to this buffer descriptor, send an interrupt to the ARM platform. 0 No Interrupt 1 Interrupt the processor when BD is complete
18 C	CONTinuous. This buffer is allowed to receive multiple transmit buffers or is allowed to transmit to multiple receive buffers.The Continuos bit is decoded at the end of the processing of a BD to determine if the SDMA script must open a new BD to potentially continue the data transfer. 0 No further buffer descriptors 1 SDMA should move to the next Buffer descriptor after this one

Table continues on the next page...

Table 66-52. Buffer Descriptor Field Descriptions (continued)

Field	Description
17 W	Wrap. Indicates if this buffer descriptor is the last one for the channel control block. When encountering this bit set, the SDMA scripts updates the CurrentBD pointer to point to the first Buffer Descriptor of the array. This bit is set if the ARM platform wants to organize the array of BD in a circular way (like a ring). When all BD have been processed and if Wrap bit and CONtinuous bit are set in the last BD, the SDMA script will wrap around and it will try to re-open the first BD. 0 No Error 1 Wrap to first buffer descriptor after this one is processed.
16 D	D - "Done": bit 16: indicates the "ownership" of the buffer descriptor. When D=0 the host owns the buffer descriptor; when D=1 SDMA owns the buffer descriptor. In the case of the channel 0, D=1 indicates the SDMA has not yet processed this buffer, D=0 indicates the SDMA has processed this buffer. 0 ARM platform owns the buffer. 1 SDMA owns the buffer
15-0 Count	Count. the count field (bit 15-0) indicates the size of the data to be transmitted, the size of the data buffer pointed to by the buffer descriptor. The SDMA memory structure is different for program memory (16-bits shorts/ half-words) and data memory (32-bits long). For channel 0 buffer descriptors, Count is expressed in 16-bit half-words when PM is addressed and in 32-bit words when DM is addressed. Count is typically expressed in bytes for other channel scripts, but the unit is dependant on the script.
31-0	Buffer address. Address pointer to the data buffer.
31-0	Extended buffer address. Additional pointer or other information required by some scripts.

The buffer descriptors form an array of programmable size. If the last buffer descriptor is marked by the Wrap flag-bit W=1, the array of buffer descriptor is treated as a ring with some logically continuous portion owned by the ARM platform with D=0, and the remainder owned by the SDMA with D=1. The count field of the buffer descriptor indicates how much data has been transmitted.

If ARM platform has prepared 3 buffers to be filled by the SDMA script, it has also prepared 3 BD, one for each buffer. The *Cont* and *Wrap* bits are used to organize the buffers in a circular way. For example, *CONT*inous bit is set to 1 in the 2 first BDs and *Wrap* is set in the 3rd BD. The SDMA script opens and processes BD#1. Since *CONT*inous bit is set for this BD, the SDMA will open the second BD and it will process it. Each time a BD is processed, its *Done* bit is reset by the SDMA. After the 3rd BD, if *CONT*inous is not set but if *Wrap* is set, the SDMA script stops here and the next time the channel will be triggered, the script will open the BD pointed by the currentBDptr pointer of the CCB and it will correspond to the first buffer descriptor.

If the *CONT*inous bit and *Wrap* bits are both set in the 3rd BD, the script will close it and it will try to open the first BD. An error may occur at this point if the BD#1 has already been processed and its *Done* bit is 0. The SDMA script cannot process a BD with a *Done* bit to 0. It means the BD is not ready to be processed. To avoid this situation, the *CONT*inous bit should not be set for the last BD if *Wrap* is set, and the Interrupt flag must set for the last BD. It will warn the owner of the BD that all the BDs have been processed and it has to re-set to 1 the *Done* bit of all the BD's if it desi

fill them again. Basically, if the ARM platform expects the SDMA to fill up the buffers in a circular fashion, then it's the responsibility of the ARM platform to set the *Done* bit of a buffer descriptor at an appropriate time.

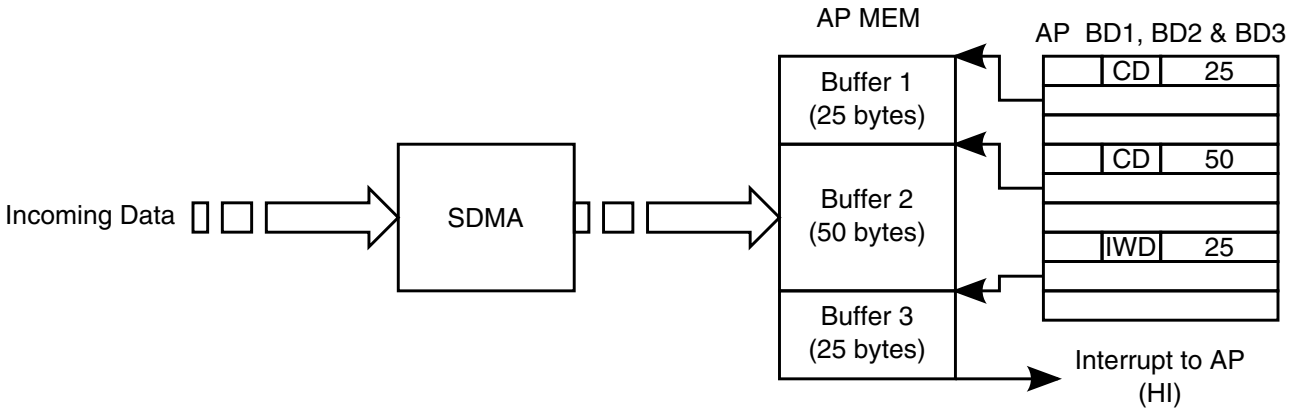


Figure 66-18. Buffer Descriptor Flow

The previous figure shows an example buffer descriptor flow. When the incoming data is stored and fills the first buffer of 25 bytes, the SDMA script opens the second BD because the CONTinuous bit was set. Then next incoming data is put in the second buffer. After receiving 50 bytes, the second buffer descriptor is also closed. The Done bit is reset and the third BD is opened. After receiving another 25 bytes, the third buffer is full and an interrupt is sent to the ARM platform because the Interrupt flag is set in the 3rd BD. The CONTinuous flag is not present the transfer is over. The next time the script will be triggered, the BD to be opened will be the first buffer descriptor since the Wrap flag was set in the 3rd BD. It is the ARM platform responsibility to set the Done bit of all the BD if it wants to use the same buffers.

66.19.1.2 Buffer Descriptor Commands for Bootload scripts

The command field of the buffer descriptor is defined separately for each script.

The following table lists the buffer descriptor commands defined for the channel 0 bootload script.

Table 66-53. Channel Zero Buffer Descriptor Commands

Command Field (binary)	Command	Description	Buffer Address	Extended Buffer Address
0000_0001 (0x01)	C0_SET_DM	Load SDMA data memory (RAM) from ARM platform memory buffer	ARM platform memory source address	SDMA memory destination address
0000_0010 (0x02)	C0_GET_DM	Copy SDMA data memory (RAM) to ARM platform memory buffer	ARM platform memory destination address	SDMA memory source address
0000_0100 (0x04)	C0_SET_PM	Load SDMA program memory (RAM) from ARM platform memory buffer	ARM platform memory source address	SDMA memory destination address
0000_0110 (0x06)	C0_GET_PM	Copy SDMA program memory (RAM) to ARM platform memory buffer	ARM platform memory destination address	SDMA memory source address
cccc_c111 (0x07 | CHN)	C0_SETCTX	Load Context for channel cccc into SDMA RAM from ARM platform memory buffer	ARM Platform memory source address	-
cccc_c011 (0x03 | CHN)	C0_GETCTXT	Copy Context for channel ccccc from SDMA RAM to ARM platform memory buffer	ARM platform memory destination address	-

The Channel 0 bootload commands are summarized as follows:

- C0_SET_[PM-DM]: load the buffer descriptor data in the SDMA local memory at the address pointed to by the "extended buffer address" field. The SDMA RAM can be seen as a Program Memory (PM, 16-bit address) or Data Memory (DM 32-bit address). When C0_SET_PM is used, the count field is expressed in "shorts" (16-bit half words), this command can be used to download scripts. When C0_SET_DM is used, the count field is expressed in "long" (32-bit words), this command can be used to download channel contexts to the context channel area in RAM.
- C0_GET_[PM-DM]: write to the buffer descriptor's data buffer the content of the SDMA local memory from the address pointed to by the "extended buffer address" field for the length defined by the count in the buffer descriptor. C0_GET_PM is used to dump some part of the Program Memory (may be used to dump context of a channel), therefore count is expressed in "shorts"; while C0_GET_DM is used to dump to the buffer descriptor's data buffer, so the count field is in "longs."
- C0_SETCTX: load a context into the SDMA context page area. The handling script decodes the channel number from the 5 MSB of the command field of the buffer descriptor. Using the channel number the script computes the offset of the context data pointer for the channel relative to the context page base to use as the destination

address in SDMA memory. Then the C0_SET_DM command explained above is invoked to load SDMA RAM from memory. The counter indicates the size in words of the context structure.

- Command value: (in binary) cccc c111, where ccccc is the channel number (5 bits). For instance, 0x0F means set context for channel 1, 0xFF means set context for channel 31.

- C0_GETCTX: write to the buffer descriptor's data buffer the content of the SDMA context page area. The handling script decodes the channel number from the 5 MSB of the command field of the buffer descriptor. Using this channel number, the script computes the offset of the context data pointer for the channel relative to the context page base to use as the source address for the copy. Then the C0_GET_DM command explained above is invoked to copy the context to memory. The counter indicates the size in words of the context structure.
- Command value: (in binary): cccc c011, where ccccc is the channel number (5 bits). For instance, 0x03 means get context of channel 1, 0xFB means get context of channel 31.

NOTE

To download channel context, C0_SETDM and C0_SETCTXT command can be used but the second one is easier because the channel number is embedded into the command field, whereas with the C0_SETDM, the pointer to the channel context area must be written into the extended buffer address field of the buffer descriptor.

66.19.1.3 Example of Buffer Descriptors for Channel 0.

Figure 66-20 illustrates the buffer descriptors that must be set in ARM platform memory space, before execution of boot code, to download contexts and scripts of channels 1, 4, and 10. After boot code execution, SDMA memory will be populated with the different contexts and scripts as presented in the following figure.

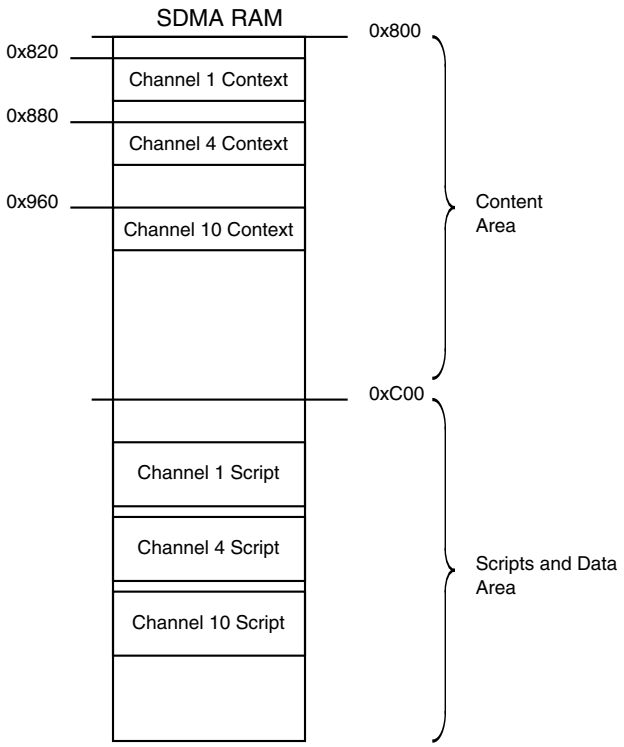
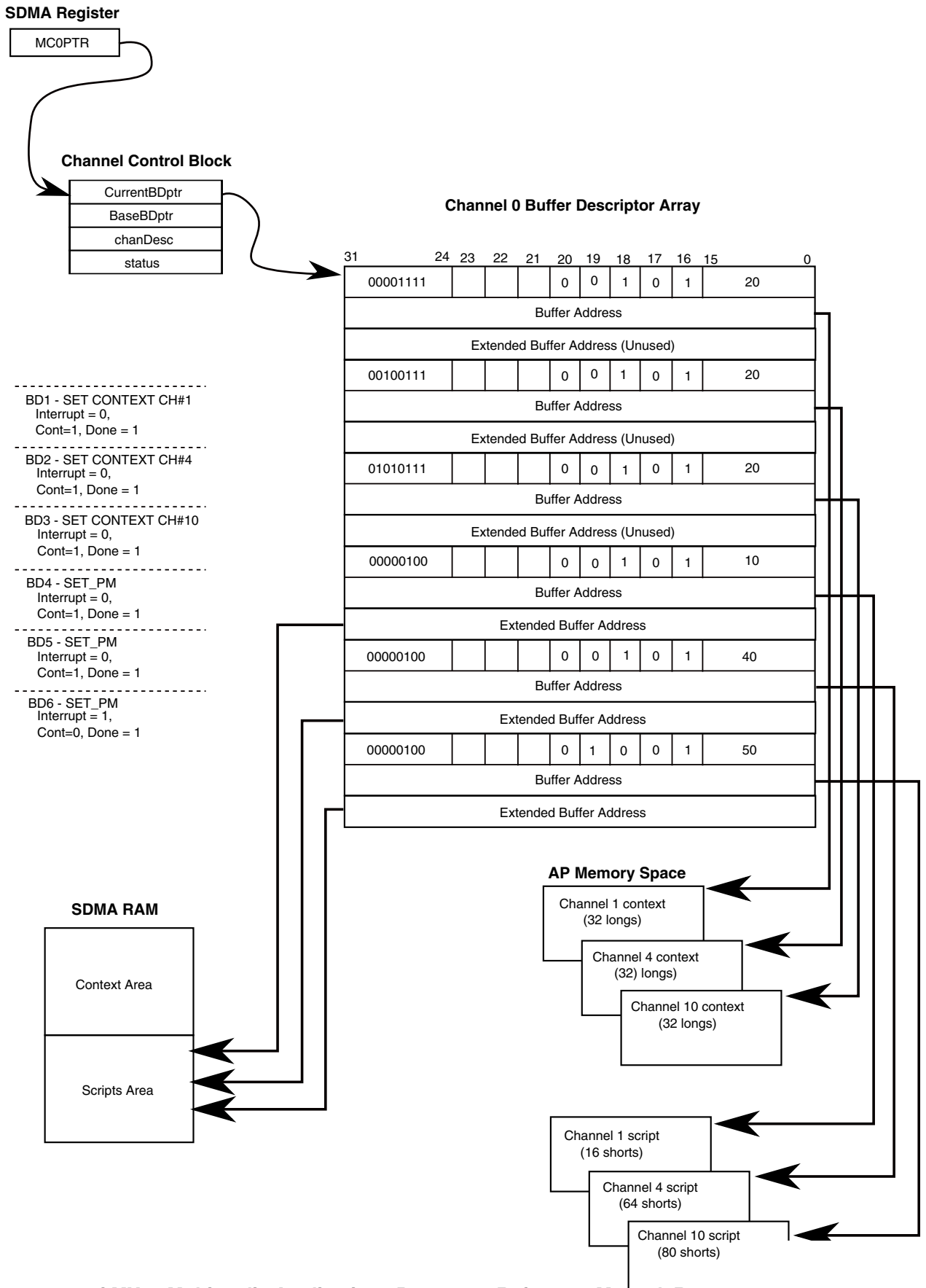


Figure 66-19. Example of SDMA RAM After Boot Session


SDMA Register
MC0PTR
Channel Control Block
CurrentBDptr
BaseBDptr
chanDesc
status
Channel 0 Buffer Descriptor Array
31 24 23 22 21 20 19 18 17 16 15 0
00001111 0 0 1 0 1 20
Buffer Address
Extended Buffer Address (Unused)
00100111 0 0 1 0 1 20
Buffer Address
Extended Buffer Address (Unused)
01010111 0 0 1 0 1 20
Buffer Address
Extended Buffer Address (Unused)
00000100 0 0 1 0 1 10
Buffer Address
Extended Buffer Address
00000100 0 0 1 0 1 40
Buffer Address
Extended Buffer Address
00000100 0 1 0 0 1 50
Buffer Address
Extended Buffer Address
BD1 - SET CONTEXT CH#1
Interrupt = 0,
Cont=1, Done = 1
BD2 - SET CONTEXT CH#4
Interrupt = 0,
Cont=1, Done = 1
BD3 - SET CONTEXT CH#10
Interrupt = 0,
Cont=1, Done = 1
BD4 - SET_PM
Interrupt = 0,
Cont=1, Done = 1
BD5 - SET_PM
Interrupt = 0,
Cont=1, Done = 1
BD6 - SET_PM
Interrupt = 1,
Cont=0, Done = 1
SDMA RAM
Context Area
Scripts Area
AP Memory Space
Channel 1 context
(32 longs)
Channel 4 context
(32) longs)
Channel 10 context
(32 longs)
Channel 1 script
(16 shorts)
Channel 4 script
(64 shorts)
Channel 10 script
(80 shorts)

66.19.1.4 Channel Context

There are 32 channel context memory structures pointed to by the local save area pointer. These channel context memory structures are fixed.

The script in the SDMA computes the memory offset for a given channel based on the structure length and channel number. Figure below shows the structure of the channel context as it is saved in the SDMA local memory (RAM).

A channel context consists in 24 words, one per register. A total of 32 words are reserved for every channel. The additional 8 words are called scratch ram and they are dedicated to each channel. This memory area is commonly used for stack management.

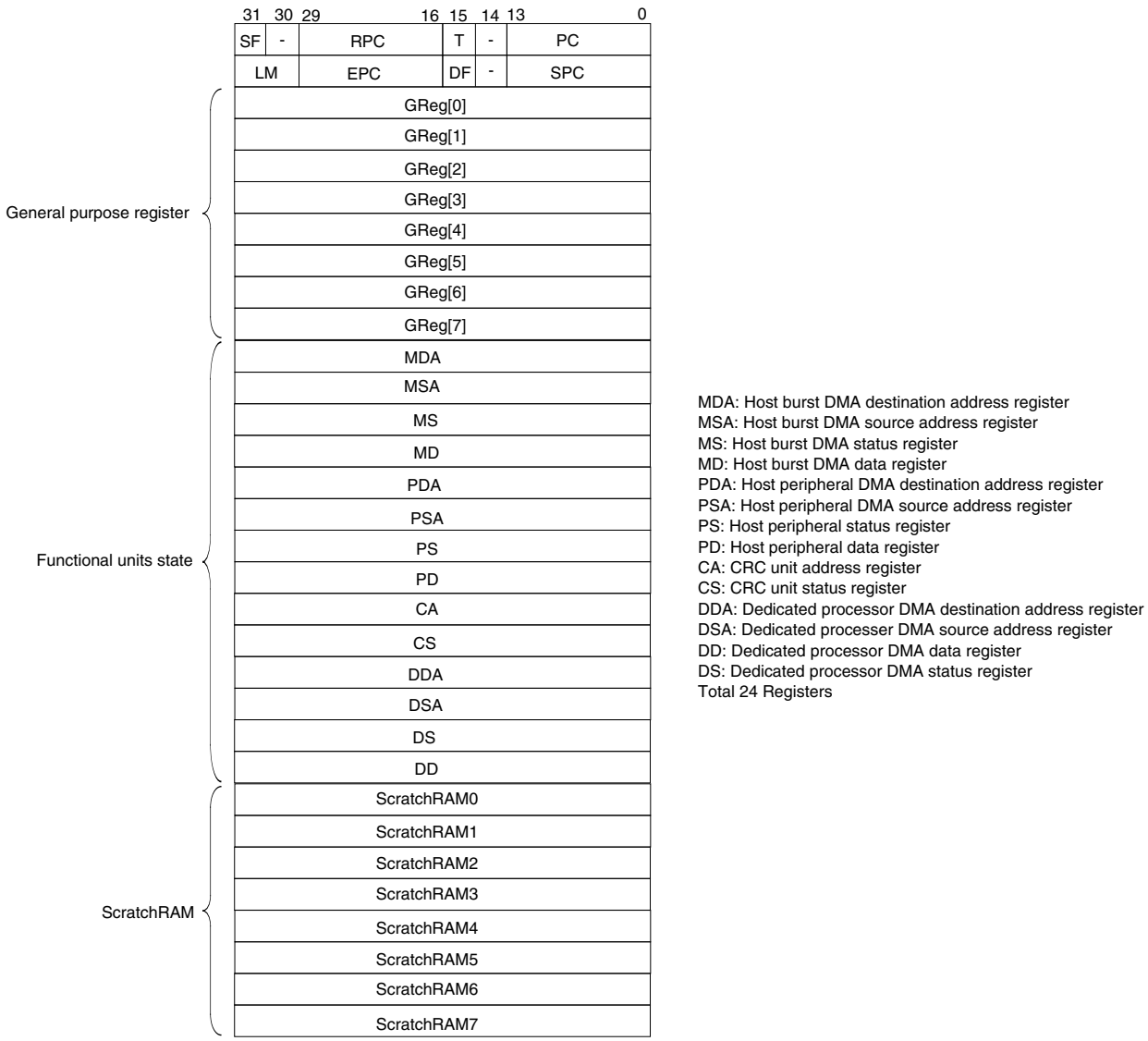


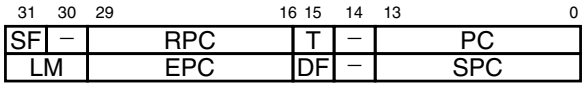
Figure 66-21. Channel Context Memory Structure

The structure is divided in 4 areas:

- Channel status registers
- General purpose registers
- Functional units state registers reflecting the state of the ARM platform DMAs (Burst and Peripheral DMA) and the CRC unit.
- Scratch RAM

The details of the channel context status registers are described in the following figure.

The PC field of the first long register must point to the SDMA RAM address where the script that will be executed on the channel is located and this value equals the one stored in the extended buffer address of the buffer descriptor with C0_SETPM command.



SF: Source fault while loading data
RPC: Return program counter
T: Test bit: status of arithmetic and test instructions
PC: Program counter
LM: Loop mode
EPC: Loop end program counter
DF: Destination fault while storing data
SPC: Loop Start program counter

Figure 66-22. SDMA State Registers (ShPC, ShLoop)

66.19.2 Typical Data Transfer Supported by SDMA DMA Units

This section presents a library of SDMA scripts that perform data transfers through the peripheral DMA and the burst DMA units.

The ARM platform memory and peripherals are devices that either the peripheral DMA or the burst DMA can access. The scripts are given for a peripheral DMA whose address registers are programmed in incremented mode when internal memory is involved. See the following table for the summary.

Table 66-54. Typical Data Transfers Summary

Data Transfer	Peripheral DMA	Burst DMA	Comments
ARM platform External Memory ↔ ARM platform External Memory		3	Copy mode Script example, see Burst DMA Unit Copy Mode and External Memory to External Memory.
ARM platform Peripheral ↔ ARM platform Peripheral	3		Copy mode if same data path width Script example, see Peripheral to Peripheral Transfer.
ARM platform External Memory ↔ ARM platform Peripheral	3	3	Data transit through SDMA Script example, see Transfer Between Peripheral and External Memory.
ARM platform External Memory ↔ ARM platform Internal Memory		3	Copy mode Script example, see Transfer Between External Memory and Internal Memory.

Table continues on the next page...

Table 66-54. Typical Data Transfers Summary (continued)

Data Transfer	Peripheral DMA	Burst DMA	Comments
ARM platform Internal Memory ↔ ARM platform Internal Memory		3	Copy mode Script example, see Internal Memory to Internal Memory.
ARM platform Internal memory ↔ ARM platform Peripheral	3		Data transit through SDMA Script example, see Transfer Between Peripheral and Internal Memory.

NOTE

These scripts are provided as examples of how to use DMA blocks to perform required data transfers: They are not "official" programs.

66.19.2.1 External Memory to External Memory

This section describes the SDMA script that performs data moves in external memory.

For this particular data transfer, only the burst DMA is used. It is programmed in copy mode, so no data transmits through an SDMA general register.

The SDMA core only monitors data transfer status. It is assumed source and destination address values are already present in two SDMA general registers (r1 and r2). For this example, it is also assumed that a 32-bit word-to-move for source-to-destination address is present in r0 and equals 64.

Data Moves in External Memory

```
1          stf r1,MSA                          // Source address setup

2          stf r2,MDA                          // Destination address setup

3          ldi r0,0x64                         // 64 words must be transferred from MSA to
MDA

4          ldi r1,0x8

MAIN_XFER:

5          cmphs r0,r1                         // Is r0 >= 0x8

6          bf LAST_XFER                      // If not, jump to last transfer label

7          stf r1,MD|CPY                        // Copy 8 words from MSA to MDA address.

8          subi r0,0x8                         // Decrement counter

9          jmp MAIN_XFER                     // return to main transfer loop

LAST_XFER:

10         stf r0,MD|CPY                        // perform last transfer
```

All instructions are performed in one cycle (jumps excepted). Instruction 7 triggers a copy transfer: A read burst access of 8-word starts, data is staged in MD and then a write burst of 8 words is executed. Instruction 8, 9, 5, and 6 are executed while the burst access is in progress. If this access is not complete when instruction 7 is executed a second time, SDMA stalls on this instruction as long as the previous copy transfer is not over. In this case, the instruction is no longer a one-cycle instruction.

During the main loop (MAIN_XFER), r1 always equals 8, so burst lengths are 8 words. On the last ldf |CPY instruction (10), r1 equals the reminder of r0 divided by 8; therefore, the length of bursts triggered in copy mode equal r1 value, which is between 1 and 7.

66.19.2.2 Peripheral to Peripheral Transfer

For this data transfer, only the peripheral DMA is used. It is programmed in copy mode, so no data will transmit through the SDMA general register used in the ldf instruction, but the contents of the general register are lost.

The SDMA core only monitors the transfer.

66.19.2.2.1 Source and Destination Target Have the Same Data Path Width

When the source and destination target have the same data path width, the following is true:

- Source target is a *half-word* (16-bit) peripheral located at address 0x1002.
- Destination is a *half-word* (16-bit) peripheral located at address 0x2006.

It is assumed the address values are already present in two SDMA general registers (r1, r2). The script for a transfer of 10 half-word is as follows:

Same Data Path Width for Source and Destination

```
//SETUP SECTION
1         stf r1, PSA|SZ16|F                    //r1=0x1002 Source address register setup
2         stf r2, PDA|SZ16|F                    //r2=0x2006 Destination address register
setup
3         bdf ERROR_ADDR_SETUP
4         ldi r0,0xa                          //loop counter is 10
//MAIN LOOP TRANFER
copy_loop:

5         loop 2,0
6           ldf r7,PD|CPY                     //Reads 1 half-word from src and writes to
dest.
7           yield
8         bdf ERROR_DURING_XFER
ERROR_ADDR_SETUP:
                    //correction of PSA/PDA setup and jumps to main loop transfer
ERROR_DURING_XFER:
//flag error is set,
//PS can be read to know if error occurs during read or write access.
```

If a data transfer must occur between two word peripherals, only the setup section should be updated. The transfer itself is always performed by the hardware loop instruction.

All instructions are executed in one cycle (change of flow excepted). On instruction 6, a single read access is triggered, read data is staged in PD, and a write-to-destination is executed. When the transfers are in progress, the SDMA can execute he next instructions in parallel. If instruction 6, which performs the copy transfer, is executed while the previous access is not over, SDMA is stalled and instruction ldf is a multi-cycle instruction.

66.19.2.2.2 Source and Destination Target Have a Different Data Path Width

When the source and destination target have a different data path width, copy mode cannot be used, and any attempt to initiate a copy transfer immediately raises an error, which is stored in the SF flag.

The following example shows the SDMA code that could transfer 10 words from a *word* (32-bit) peripheral to a *half-word* peripheral whose addresses are preliminary and stored in r1 and r2.

Different Data Path Width for Source and Destination

```
//SETUP SECTION
1          stf r1, PSA|SZ32|F|PF                    //r1=0x1000 and prefetch data
2          stf r2, PDA|SZ16|F                      //r2=0x2006
3          bdf ERROR_ADDR_SETUP
4          ldi r0,0xa                             //loop counter is 10
//MAIN LOOP TRANFER
main_loop_xfer_16_16:
5          loop 6,0
6            ldf r7,PD                          //copy 32-bit of PD in r7
7            stf r7,PD                           //store 16 LSB of r7 in PD and a flush is
executed
8            rorb r7
9            rorb r7                           //16 MSB --> 16 LSB
10           stf r7,PD                           //store 16 LSB of r6 in PD and a flush.
11           yield
```

On instruction 1, when the source address register is programmed and a data prefetch is required, a read access is executed. In parallel, the SDMA executes instructions 2 to 5. On instruction 6, the SDMA tries to read data that was fetched by instruction 1. If data is ready, the ldf will be a one cycle instruction; otherwise, the SDMA is stalled as long as the read access is not finished. Then, the 16 LSB of the read data is stored in PD and automatically flushed to the destination peripheral. In parallel, the SDMA executes the rotation instructions (8, 9), and stores the 16 MSB of the read data into PD. If a previous write access is finished, instruction 10 will be a one-cycle instruction.

The main loop transfer may appear inefficient, but due to wait states imposed to the peripheral DMA each time an external access is performed, a software pipeline is in place. During the time needed to flush PD, the SDMA executes the move and rotation operations. SDMA executes instructions in parallel with DMA accesses.

66.19.2.3 Transfer Between Peripheral and External Memory

66.19.2.3.1 Peripheral to External Memory Transfer

A transfer from a peripheral to the external memory controller involves the peripheral DMA and the burst DMA.

The code for transferring 100 word from word peripheral to the external memory would be as follows:

Peripheral to External Memory Transfer

```
//SETUP SECTION source and destination addresses are already in r1 and r2
1         stf r1, PSA|SZ16|F|PF                  //r1=0x1000 and prefetch 32-bit data
2         stf r2, MDA                          //r2=0x2000, setup burst DMA destination
address
3         bdf ERROR_ADDR_SETUP
4         ldi r0,0x64                           //loop counter is 100
5
          //MAIN LOOP TRANFER
6         loop 3,0
7           ldf r1,PD|PF                        // read 32 bits of PD and initiate a new read
access.
8           stf r1,MD|32                         // store 32 bits of r1 in the MD fifo.
9           yield
10        ldf r1,PD                            // last word data is read
11        stf r1,MD|32|FL                        // to flush all remaining bytes of MD
```

On instruction 1, the source address register of the peripheral DMA is programmed and data is fetched. This data is stored in PD and the SDMA reads PD during instruction 7, which is a one-cycle instruction that is read-access finished. On the same instruction (7), a data prefetch is required and a read access to the source peripheral is executed. In parallel, the SDMA stored the previous read data into the data register of MD. When MD (which is an eight-word FIFO) is full, a burst write access is executed to empty the FIFO. As long as the next SDMA instructions do not access the burst DMA, they will be one-cycle instructions. The following figures show how the peripheral DMA and burst DMA work in parallel.

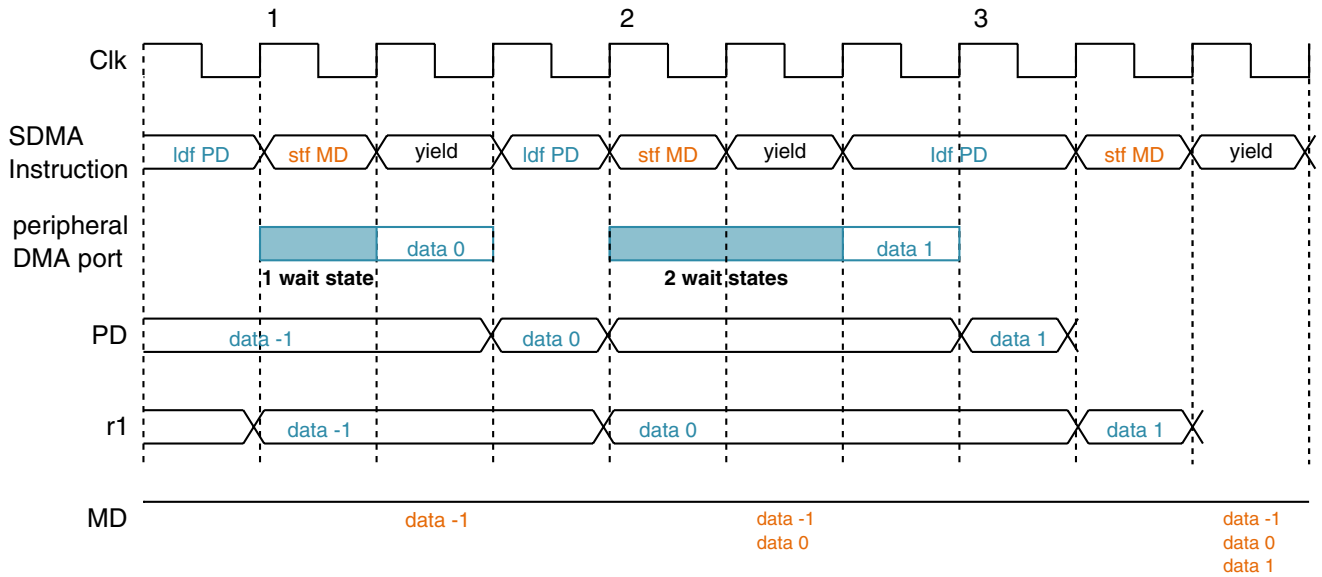


Figure 66-23. Peripheral to External Memory Example (1)

As seen in the figure above, the read access triggered by the ldf PD instruction is symbolized by the blue bar when in progress. After wait states, the read data (data 0, data 1) is stored in PD on the clk rising edge. On edge 2, data 0 is available in PD so it can be transferred to the SDMA general register r1, and then stored in MD FIFO. On edge 3, data 1 is not in PD; therefore, SDMA is stalled on the ldf instruction, which lasts two cycles. The figure below shows an example of when MD FIFO is full with data.

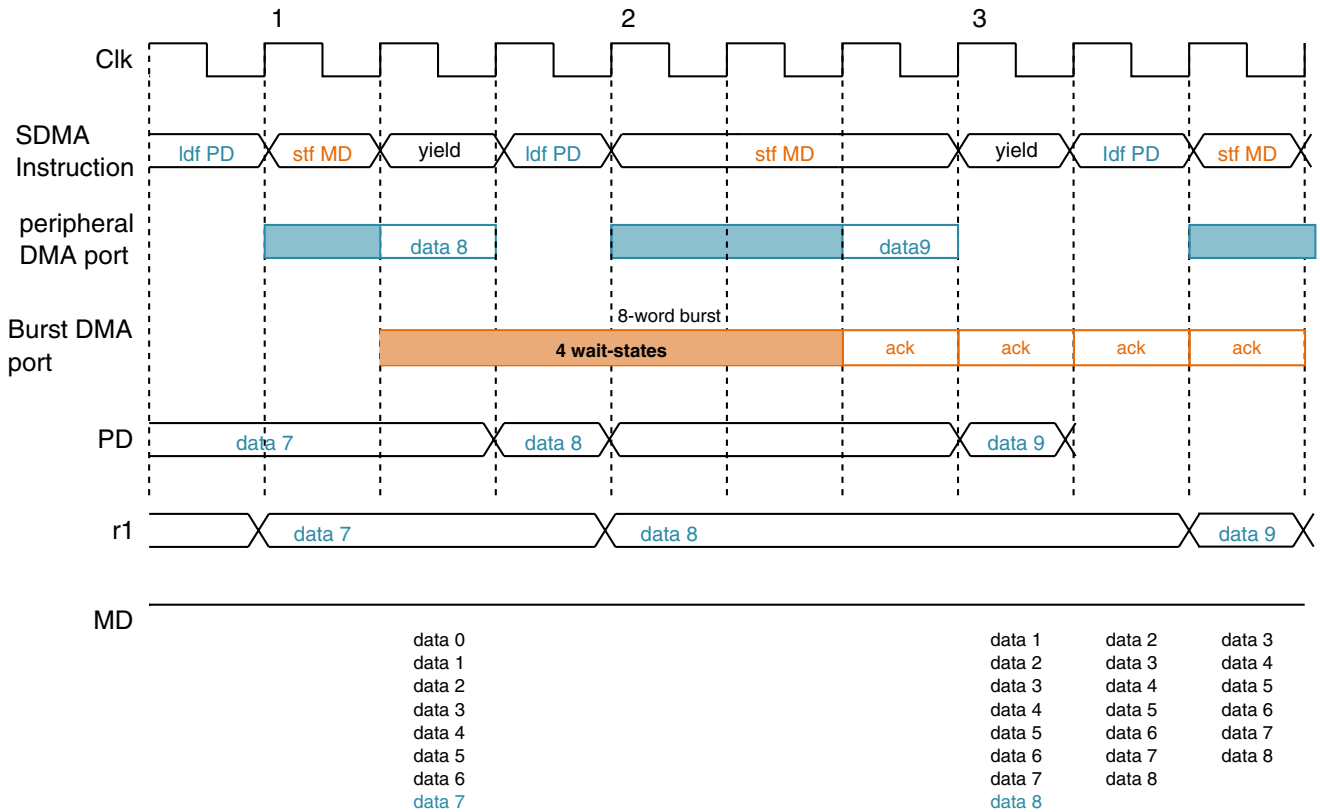


Figure 66-24. Peripheral to External Memory Example (2)

In the previous figure, the write bar means the burst DMA is performing a write burst access. The latency to have the first write acknowledge is four cycles. SDMA is stalled on instruction stf because no acknowledge was received, MD FIFO is full, and there is no empty slot to store data 9. When an acknowledge is sampled by the burst DMA, FIFO is shifted and data 8 is written. As long as there is at least one empty slot in MD FIFO, the stf MD instruction lasts one cycle.

66.19.2.3.2 External Memory to Peripheral Transfer

A transfer from the external memory to a peripheral involves the peripheral DMA and the burst DMA.

The code for transferring 100 word from external memory to a word peripheral would be as follows:

External Memory to Peripheral Transfer

```
//SETUP SECTION source and destination addresses are already in r1 and r2
1         stf r1, MSA|PF              //r1=0x1000 and starts a 8-word read burst
2         stf r2, PDA|SZ32|P           //r2=0x2010, setup peripheral DMA destination address
3         bdf ERROR_ADDR_SETUP
4         ldi r0,0x64                 //loop counter is 100
//MAIN LOOP TRANFER
6         loop 3,0
```

```
7           ldf r1,MD|32|PF          // read 32 bits of MD and initiate a new read access
                                  // if MD is empty after this reading.
8           stf r1,PD               // store 32 bits of r1 in the PD.
9           yield
          10 ldf r1,MD|32            // last word data is read
11        stf r1,PD                 // last write access
```

On instruction 1, a read burst of 8 words begins. Read data is staged into MD. On instruction 7 (and if data is available in MD), 32 bits are copied into r1. Then instruction 8 writes them into PD and an automatic flush is executed. The SDMA core, peripheral DMA, and burst DMA can work in parallel as long as no SDMA instruction tries to start a new write access on the peripheral DMA while the previous access is still in progress, or as long as there is data in MD when the SDMA tries to read it.

66.19.2.4 Transfer Between External Memory and Internal Memory

Since the internal memory (ARM platform RAM) is accessed via the peripheral DMA and the external memory is accessed via the burst DMA, the SDMA scripts that are described in Transfer Between Peripheral and External Memory can be reused. The exception is that the peripheral DMA address registers (PSA or PDA, depending on the script) should be programmed in incremented mode rather than frozen mode.

66.19.2.4.1 Internal Memory to Internal Memory

The internal memory can only be accessed via the peripheral DMA, so the script described in Peripheral to Peripheral Transfer can be reused with a different programming of the peripheral DMA address registers.

66.19.2.4.2 Transfer Between Peripheral and Internal Memory

For this transfer, the peripheral DMA is also used in copy mode.

The SDMA script is very similar to the one described in Peripheral to Peripheral Transfer, except for the peripheral DMA address registers programming.

66.20 ARM Platform Memory Map and Control Register Definitions

The ARM platform controls the SDMA by means of several interface registers. Those registers are described in the current section.

All registers are clocked with the SDMA clock (which means the ARM platform must ensure that the SDMA clock is running when it wants to access any register).

SDMAARM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FB_0000	ARM platform Channel 0 Pointer (SDMAARM_MC0PTR)	32	R/W	0000_0000h	66.20.1/ 4230
63FB_0004	Channel Interrupts (SDMAARM_INTR)	32	w1c	0000_0000h	66.20.2/ 4231
63FB_0008	Channel Stop/Channel Status (SDMAARM_STOP_STAT)	32	w1c	0000_0000h	66.20.3/ 4231
63FB_000C	Channel Start (SDMAARM_HSTART)	32	R/W	0000_0000h	66.20.4/ 4231
63FB_0010	Channel Event Override (SDMAARM_EVTOVR)	32	R/W	0000_0000h	66.20.5/ 4232
63FB_0014	Channel BP Override (SDMAARM_DSPOVR)	32	R/W	FFFF_ FFFFh	66.20.6/ 4232
63FB_0018	Channel ARM platform Override (SDMAARM_HOSTOVR)	32	R/W	0000_0000h	66.20.7/ 4233
63FB_001C	Channel Event Pending (SDMAARM_EVTPEND)	32	w1c	0000_0000h	66.20.8/ 4233
63FB_0024	Reset Register (SDMAARM_RESET)	32	R	0000_0000h	66.20.9/ 4234
63FB_0028	DMA Request Error Register (SDMAARM_EVTERR)	32	R	0000_0000h	66.20.10/ 4234
63FB_002C	Channel ARM platform Interrupt Mask (SDMAARM_INTRMASK)	32	R/W	0000_0000h	66.20.11/ 4235
63FB_0030	Schedule Status (SDMAARM_PSW)	32	R	0000_0000h	66.20.12/ 4235
63FB_0034	DMA Request Error Register (SDMAARM_EVTERRDBG)	32	R	0000_0000h	66.20.13/ 4236
63FB_0038	Configuration Register (SDMAARM_CONFIG)	32	R/W	0000_0003h	66.20.14/ 4237

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FB_003C	SDMA LOCK (SDMAARM_SDMA_LOCK)	32	R/W	0000_0000h	66.20.15/ 4238
63FB_0040	OnCE Enable (SDMAARM_ONCE_ENB)	32	R/W	0000_0000h	66.20.16/ 4239
63FB_0044	OnCE Data Register (SDMAARM_ONCE_DATA)	32	R/W	0000_0000h	66.20.17/ 4239
63FB_0048	OnCE Instruction Register (SDMAARM_ONCE_INSTR)	32	R/W	0000_0000h	66.20.18/ 4240
63FB_004C	OnCE Status Register (SDMAARM_ONCE_STAT)	32	R	0000_E000h	66.20.19/ 4240
63FB_0050	OnCE Command Register (SDMAARM_ONCE_CMD)	32	R/W	0000_0000h	66.20.20/ 4242
63FB_0058	Illegal Instruction Trap Address (SDMAARM_ILLINSTADDR)	32	R/W	0000_0001h	66.20.21/ 4242
63FB_005C	Channel 0 Boot Address (SDMAARM_CHN0ADDR)	32	R/W	0000_0050h	66.20.22/ 4243
63FB_0060	DMA Requests (SDMAARM_EVT_MIRROR)	32	R	0000_0000h	66.20.23/ 4244
63FB_0064	DMA Requests 2 (SDMAARM_EVT_MIRROR2)	32	R	0000_0000h	66.20.24/ 4244
63FB_0070	Cross-Trigger Events Configuration Register 1 (SDMAARM_XTRIG_CONF1)	32	R/W	0000_0000h	66.20.25/ 4245
63FB_0074	Cross-Trigger Events Configuration Register 2 (SDMAARM_XTRIG_CONF2)	32	R/W	0000_0000h	66.20.26/ 4246
63FB_0100	Channel Priority Registers (SDMAARM_SDMA_CHNPRI0)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0104	Channel Priority Registers (SDMAARM_SDMA_CHNPRI1)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0108	Channel Priority Registers (SDMAARM_SDMA_CHNPRI2)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_010C	Channel Priority Registers (SDMAARM_SDMA_CHNPRI3)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0110	Channel Priority Registers (SDMAARM_SDMA_CHNPRI4)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0114	Channel Priority Registers (SDMAARM_SDMA_CHNPRI5)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0118	Channel Priority Registers (SDMAARM_SDMA_CHNPRI6)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_011C	Channel Priority Registers (SDMAARM_SDMA_CHNPRI7)	32	R/W	0000_0000h	66.20.27/ 4247

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FB_0120	Channel Priority Registers (SDMAARM_SDMA_CHNPRI8)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0124	Channel Priority Registers (SDMAARM_SDMA_CHNPRI9)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0128	Channel Priority Registers (SDMAARM_SDMA_CHNPRI10)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_012C	Channel Priority Registers (SDMAARM_SDMA_CHNPRI11)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0130	Channel Priority Registers (SDMAARM_SDMA_CHNPRI12)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0134	Channel Priority Registers (SDMAARM_SDMA_CHNPRI13)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0138	Channel Priority Registers (SDMAARM_SDMA_CHNPRI14)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_013C	Channel Priority Registers (SDMAARM_SDMA_CHNPRI15)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0140	Channel Priority Registers (SDMAARM_SDMA_CHNPRI16)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0144	Channel Priority Registers (SDMAARM_SDMA_CHNPRI17)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0148	Channel Priority Registers (SDMAARM_SDMA_CHNPRI18)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_014C	Channel Priority Registers (SDMAARM_SDMA_CHNPRI19)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0150	Channel Priority Registers (SDMAARM_SDMA_CHNPRI20)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0154	Channel Priority Registers (SDMAARM_SDMA_CHNPRI21)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0158	Channel Priority Registers (SDMAARM_SDMA_CHNPRI22)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_015C	Channel Priority Registers (SDMAARM_SDMA_CHNPRI23)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0160	Channel Priority Registers (SDMAARM_SDMA_CHNPRI24)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0164	Channel Priority Registers (SDMAARM_SDMA_CHNPRI25)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0168	Channel Priority Registers (SDMAARM_SDMA_CHNPRI26)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_016C	Channel Priority Registers (SDMAARM_SDMA_CHNPRI27)	32	R/W	0000_0000h	66.20.27/ 4247

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FB_0170	Channel Priority Registers (SDMAARM_SDMA_CHNPRI28)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0174	Channel Priority Registers (SDMAARM_SDMA_CHNPRI29)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0178	Channel Priority Registers (SDMAARM_SDMA_CHNPRI30)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_017C	Channel Priority Registers (SDMAARM_SDMA_CHNPRI31)	32	R/W	0000_0000h	66.20.27/ 4247
63FB_0200	Channel Enable RAM (SDMAARM_SDMA.CHNENBL0)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0204	Channel Enable RAM (SDMAARM_SDMA.CHNENBL1)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0208	Channel Enable RAM (SDMAARM_SDMA.CHNENBL2)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_020C	Channel Enable RAM (SDMAARM_SDMA.CHNENBL3)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0210	Channel Enable RAM (SDMAARM_SDMA.CHNENBL4)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0214	Channel Enable RAM (SDMAARM_SDMA.CHNENBL5)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0218	Channel Enable RAM (SDMAARM_SDMA.CHNENBL6)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_021C	Channel Enable RAM (SDMAARM_SDMA.CHNENBL7)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0220	Channel Enable RAM (SDMAARM_SDMA.CHNENBL8)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0224	Channel Enable RAM (SDMAARM_SDMA.CHNENBL9)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0228	Channel Enable RAM (SDMAARM_SDMA.CHNENBL10)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_022C	Channel Enable RAM (SDMAARM_SDMA.CHNENBL11)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0230	Channel Enable RAM (SDMAARM_SDMA.CHNENBL12)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0234	Channel Enable RAM (SDMAARM_SDMA.CHNENBL13)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0238	Channel Enable RAM (SDMAARM_SDMA.CHNENBL14)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_023C	Channel Enable RAM (SDMAARM_SDMA.CHNENBL15)	32	R/W	0000_0000h	66.20.28/ 4248

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FB_0240	Channel Enable RAM (SDMAARM_SDMA.CHNENBL16)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0244	Channel Enable RAM (SDMAARM_SDMA.CHNENBL17)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0248	Channel Enable RAM (SDMAARM_SDMA.CHNENBL18)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_024C	Channel Enable RAM (SDMAARM_SDMA.CHNENBL19)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0250	Channel Enable RAM (SDMAARM_SDMA.CHNENBL20)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0254	Channel Enable RAM (SDMAARM_SDMA.CHNENBL21)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0258	Channel Enable RAM (SDMAARM_SDMA.CHNENBL22)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_025C	Channel Enable RAM (SDMAARM_SDMA.CHNENBL23)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0260	Channel Enable RAM (SDMAARM_SDMA.CHNENBL24)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0264	Channel Enable RAM (SDMAARM_SDMA.CHNENBL25)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0268	Channel Enable RAM (SDMAARM_SDMA.CHNENBL26)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_026C	Channel Enable RAM (SDMAARM_SDMA.CHNENBL27)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0270	Channel Enable RAM (SDMAARM_SDMA.CHNENBL28)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0274	Channel Enable RAM (SDMAARM_SDMA.CHNENBL29)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0278	Channel Enable RAM (SDMAARM_SDMA.CHNENBL30)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_027C	Channel Enable RAM (SDMAARM_SDMA.CHNENBL31)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0280	Channel Enable RAM (SDMAARM_SDMA.CHNENBL32)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0284	Channel Enable RAM (SDMAARM_SDMA.CHNENBL33)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0288	Channel Enable RAM (SDMAARM_SDMA.CHNENBL34)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_028C	Channel Enable RAM (SDMAARM_SDMA.CHNENBL35)	32	R/W	0000_0000h	66.20.28/ 4248

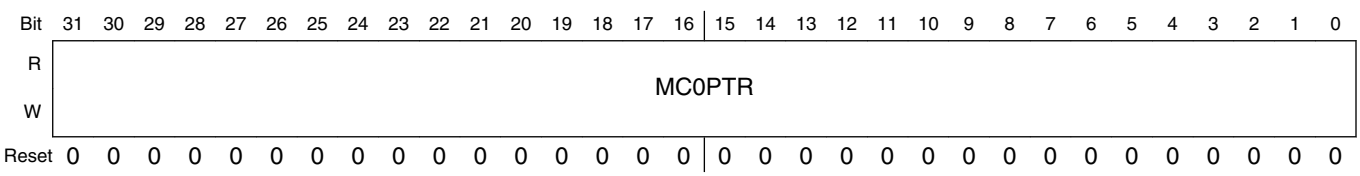
Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FB_0290	Channel Enable RAM (SDMAARM_SDMA.CHNENBL36)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0294	Channel Enable RAM (SDMAARM_SDMA.CHNENBL37)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_0298	Channel Enable RAM (SDMAARM_SDMA.CHNENBL38)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_029C	Channel Enable RAM (SDMAARM_SDMA.CHNENBL39)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_02A0	Channel Enable RAM (SDMAARM_SDMA.CHNENBL40)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_02A4	Channel Enable RAM (SDMAARM_SDMA.CHNENBL41)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_02A8	Channel Enable RAM (SDMAARM_SDMA.CHNENBL42)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_02AC	Channel Enable RAM (SDMAARM_SDMA.CHNENBL43)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_02B0	Channel Enable RAM (SDMAARM_SDMA.CHNENBL44)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_02B4	Channel Enable RAM (SDMAARM_SDMA.CHNENBL45)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_02B8	Channel Enable RAM (SDMAARM_SDMA.CHNENBL46)	32	R/W	0000_0000h	66.20.28/ 4248
63FB_02BC	Channel Enable RAM (SDMAARM_SDMA.CHNENBL47)	32	R/W	0000_0000h	66.20.28/ 4248

66.20.1 ARM platform Channel 0 Pointer (SDMAARM_MC0PTR)

Address: SDMAARM_MC0PTR is 63FB_0000h base + 0h offset = 63FB_0000h



SDMAARM_MC0PTR field descriptions

Field	Description
31–0 MC0PTR	**Channel 0 Pointer** contains the 32-bit address, in ARM platform memory, of channel 0 control block (the boot channel). Appendix A fully describes the SDMA Application Programming Interface (API). The ARM platform has a read/write access and the SDMA has a read-only access.

66.20.2 Channel Interrupts (SDMAARM_INTR)

Address: SDMAARM_INTR is 63FB_0000h base + 4h offset = 63FB_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HI[31:0]																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_INTR field descriptions

Field	Description
31–0 HI[31:0]	The ARM platform Interrupts register contains the 32 HI[i] bits. If any bit is set, it will cause an interrupt to the ARM platform. This register is a "write-ones" register to the ARM platform. When the ARM platform sets a bit in this register the corresponding HI[i] bit is cleared. The interrupt service routine should clear individual channel bits when their interrupts are serviced, failure to do so will cause continuous interrupts. The SDMA is responsible for setting the HI[i] bit corresponding to the current channel when the corresponding `done` instruction is executed.

66.20.3 Channel Stop/Channel Status (SDMAARM_STOP_STAT)

Address: SDMAARM_STOP_STAT is 63FB_0000h base + 8h offset = 63FB_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HE																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_STOP_STAT field descriptions

Field	Description
31–0 HE	This 32-bit register gives access to the ARM platform Enable bits. There is one bit for every channel. This register is a "write-ones" register to the ARM platform. When the ARM platform writes 1 in bit i of this register, it clears the HE[i] and HSTART[i] bits. Reading this register yields the current state of the HE[i] bits.

66.20.4 Channel Start (SDMAARM_HSTART)

Address: SDMAARM_HSTART is 63FB_0000h base + Ch offset = 63FB_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HSTART/HE																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_HSTART field descriptions

Field	Description
31–0 HSTART/HE	The HSTART/HE registers are 32 bits wide with one bit for every channel. When a bit is written to 1, it enables the corresponding channel. Two physical registers are accessed with that address (HSTART and HE), which enables the ARM platform to trigger a channel a second time before the first trigger is processed. • This register is a "write-ones" register to the ARM platform. Neither HSTART[i] bit can be set while the corresponding HE[i] bit is cleared. • When the ARM platform tries to set the HSTART[i] bit by writing a one (if the corresponding HE[i] bit is clear), the bit in the HSTART[i] register will remain cleared and the HE[i] bit will be set. • If the corresponding HE[i] bit was already set, the HSTART[i] bit will be set. The next time the SDMA channel *i* attempts to clear the HE[i] bit by means of a `done` instruction, the bit in the HSTART[i] register will be cleared and the HE[i] bit will take the old value of the HSTART[i] bit. • Reading this register yields the current state of the HSTART[i] bits. This mechanism enables the ARM platform to pipeline two HSTART commands per channel.

66.20.5 Channel Event Override (SDMAARM_EVTOVR)

Address: SDMAARM_EVTOVR is 63FB_0000h base + 10h offset = 63FB_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_EVTOVR field descriptions

Field	Description
31–0 EO	The Channel Event Override register contains the 32 EO[i] bits. A bit set in this register causes the SDMA to ignore DMA requests when scheduling the corresponding channel.

66.20.6 Channel BP Override (SDMAARM_DSPOVR)

Address: SDMAARM_DSPOVR is 63FB_0000h base + 14h offset = 63FB_0014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DO																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

SDMAARM_DSPOVR field descriptions

Field	Description
31–0 DO	This register is reserved. All DO bits should be set to the reset value of 1. A setting of 0 will prevent SDMA channels from starting according to the condition described in Runnable Channels Evaluation.

SDMAARM_DSPOVR field descriptions (continued)

Field	Description
	0 - Reserved 1 - Reset value.

66.20.7 Channel ARM platform Override (SDMAARM_HOSTOVR)

Address: SDMAARM_HOSTOVR is 63FB_0000h base + 18h offset = 63FB_0018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HO																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_HOSTOVR field descriptions

Field	Description
31–0 HO	The Channel ARM platform Override register contains the 32 HO[i] bits. A bit set in this register causes the SDMA to ignore the ARM platform enable bit (HE) when scheduling the corresponding channel.

66.20.8 Channel Event Pending (SDMAARM_EVTPEND)

Address: SDMAARM_EVTPEND is 63FB_0000h base + 1Ch offset = 63FB_001Ch

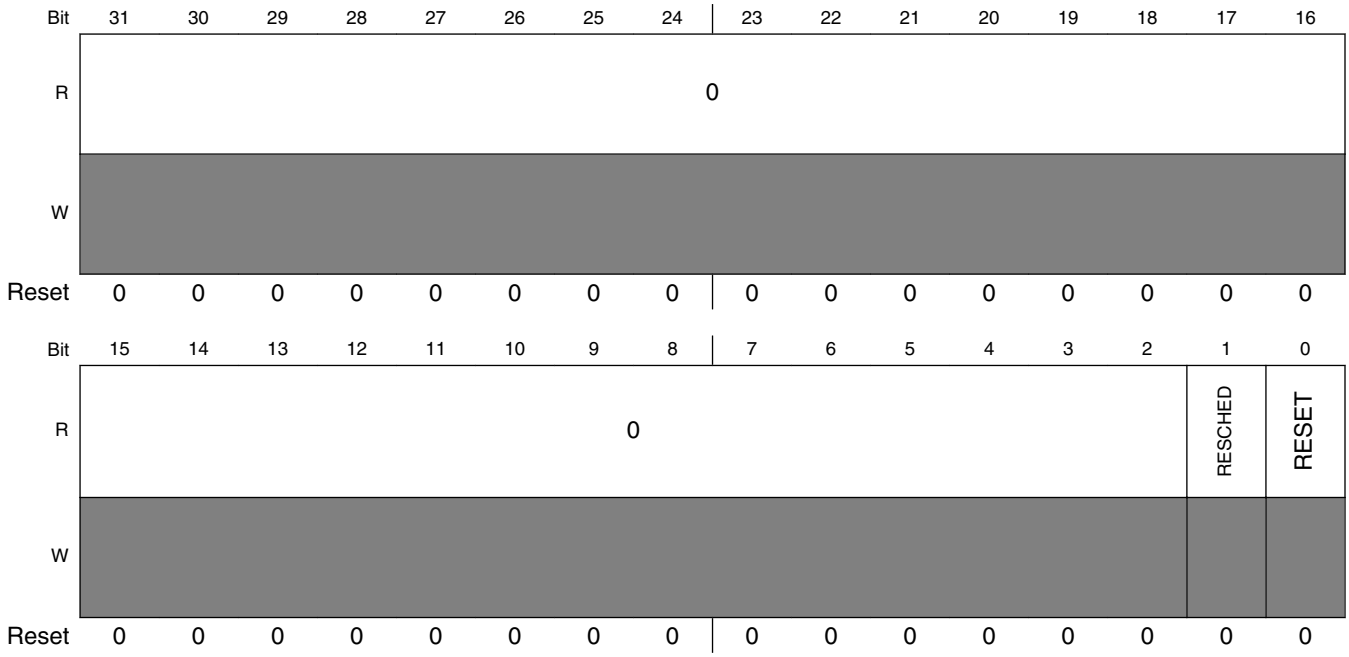
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EP																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_EVTPEND field descriptions

Field	Description
31–0 EP	The Channel Event Pending register contains the 32 EP[i] bits. Reading this register enables the ARM platform to determine what channels are pending after the reception of a DMA request. • Setting a bit in this register causes the SDMA to reevaluate scheduling as if a DMA request mapped on this channel had occurred. This is useful for starting up channels, so that initialization is done before awaiting the first request. The scheduler can also set bits in the EVTPEND register according to the received DMA requests. • The EP[i] bit may be cleared by the `done` instruction when running the channel *i* script. This a "write-ones" mechanism: Writing a '0' does not clear the corresponding bit.

66.20.9 Reset Register (SDMAARM_RESET)

Address: SDMAARM_RESET is 63FB_0000h base + 24h offset = 63FB_0024h

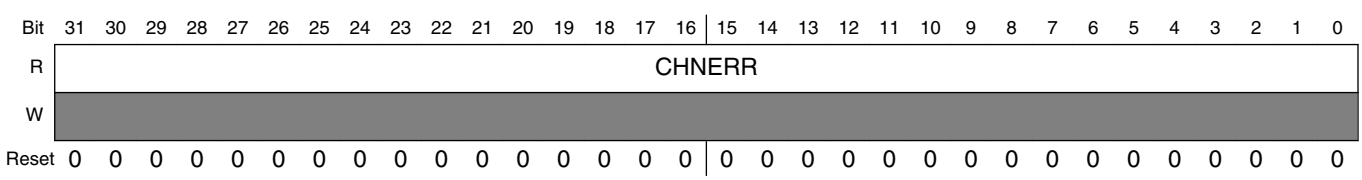


SDMAARM_RESET field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1 RESCHED	When set, this bit forces the SDMA to reschedule as if a script had executed a `done` instruction. This enables the ARM platform to recover from a runaway script on a channel by clearing its HE[i] bit via the STOP register, and then forcing a reschedule via the RESCHED bit. The RESCHED bit is cleared when the context switch starts.
0 RESET	When set, this bit causes the SDMA to be held in a software reset. The internal reset signal is held low 16 cycles; the RESET bit is automatically cleared when the internal reset signal rises.

66.20.10 DMA Request Error Register (SDMAARM_EVTERR)

Address: SDMAARM_EVTERR is 63FB_0000h base + 28h offset = 63FB_0028h



SDMAARM_EVTERR field descriptions

Field	Description
31–0 CHNERR	This register is used by the SDMA to warn the ARM platform when an incoming DMA request was detected and it triggers a channel that is already pending or being serviced. This probably means there is an overflow of data for that channel. • An interrupt is sent to the ARM platform if the corresponding channel bit is set in the INTRMASK register. • This is a "write-ones" register for the scheduler. It is only able to set the flags. The flags are cleared when the register is read by the ARM platform or during SDMA reset. • The CHNERR[i] bit is set when a DMA request that triggers channel *i* is received through the corresponding input pins and the EP[i] bit is already set; the EVTERR[i] bit is unaffected if the ARM platform tries to set the EP[i] bit, whereas, that EP[i] bit is already set.

66.20.11 Channel ARM platform Interrupt Mask (SDMAARM_INTRMASK)

Address: SDMAARM_INTRMASK is 63FB_0000h base + 2Ch offset = 63FB_002Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HIMASK																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_INTRMASK field descriptions

Field	Description
31–0 HIMASK	The Interrupt Mask Register contains 32 interrupt generation mask bits. If bit HIMASK[i] is set, the HI[i] bit is set and an interrupt is sent to the ARM platform when a DMA request error is detected on channel *i* (for example, EVTERR[i] is set).

66.20.12 Schedule Status (SDMAARM_PSW)

Address: SDMAARM_PSW is 63FB_0000h base + 30h offset = 63FB_0030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																NCP[2:0]			NCR[4:0]					CCP[2:0]			CCR[4:0]			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_PSW field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

SDMAARM_PSW field descriptions (continued)

Field	Description
15–13 NCP[2:0]	The Next Channel Priority gives the next pending channel priority. When the priority is 0, it means there is no pending channel and the NCR value has no meaning. 0 No running channel 1 Active channel priority
12–8 NCR[4:0]	The Next Channel Register indicates the number of the next scheduled pending channel with the highest priority.
7–4 CCP[2:0]	The Current Channel Priority indicates the priority of the current active channel. When the priority is 0, no channel is running: The SDMA is idle and the CCR value has no meaning. In the case that the SDMA has finished running the channel and has entered sleep state, CCP will indicate the priority of previous running channel. 0 No running channel 1 Active channel priority
3–0 CCR[4:0]	The Current Channel Register indicates the number of the channel that is being executed by the SDMA. SDMA. In the case that the SDMA has finished running the channel and has entered sleep state, CCR will indicate the previous running channel.

66.20.13 DMA Request Error Register (SDMAARM_EVTERRDBG)

Address: SDMAARM_EVTERRDBG is 63FB_0000h base + 34h offset = 63FB_0034h

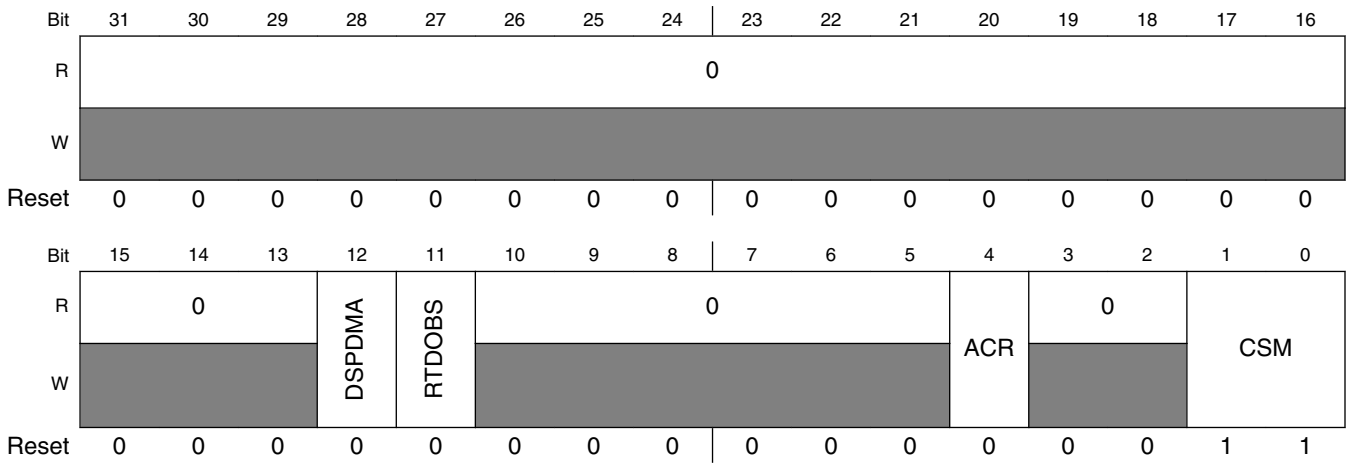
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHNERR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_EVTERRDBG field descriptions

Field	Description
31–0 CHNERR	This register is the same as EVTERR, except reading it does not clear its contents. This address is meant to be used in debug mode. The ARM platform OnCE may check this register value without modifying it.

66.20.14 Configuration Register (SDMAARM_CONFIG)

Address: SDMAARM_CONFIG is 63FB_0000h base + 38h offset = 63FB_0038h



SDMAARM_CONFIG field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 DSPDMA	This bit's function is reserved and should be configured as zero. 0 - Reset Value 1 - Reserved
11 RTDOBS	Indicates if Real-Time Debug pins are used: They do not toggle by default in order to reduce power consumption. 0 RTD pins disabled 1 RTD pins enabled
10–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 ACR	ARM platform DMA / SDMA Core Clock Ratio. Selects the clock ratio between ARM platform DMA interfaces (burst DMA and peripheral DMA) and the internal SDMA core clock. The frequency selection is determined separately by the chip clock controller. This bit has to match the configuration of the chip clock controller that generates the clocks used in the SDMA. 0 ARM platform DMA interface frequency equals twice core frequency 1 ARM platform DMA interface frequency equals core frequency
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 CSM	Selects the Context Switch Mode. The ARM platform has a read/write access. The SDMA cannot modify that register. The value at reset is 3, which selects the dynamic context switch by default. That register can be modified at anytime but the new context switch configuration will only be taken into account at the start of the next restore phase.

Table continues on the next page...

SDMAARM_CONFIG field descriptions (continued)

Field	Description
	NOTE: The first call to SDMA's channel 0 Bootload script after reset should use static context switch mode to ensure the context RAM for channel 0 is initialized in the channel SAVE Phase. After Channel 0 is run once, then any of the dynamic context modes can be used. 0 static 1 dynamic low power 2 dynamic with no loop 3 dynamic

66.20.15 SDMA LOCK (SDMAARM_SDMA_LOCK)

Address: SDMAARM_SDMA_LOCK is 63FB_0000h base + 3Ch offset = 63FB_003Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														SRESET_LOCK_CLR	LOCK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_SDMA_LOCK field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1 SRESET_ LOCK_CLR	The SRESET_LOCK_CLR bit determine if the LOCK bit is cleared on a software reset triggered by writing to the RESET register. This bit cannot be changed if LOCK=1. SREST_LOCK_CLR is cleared by conditions that clear the LOCK bit. 0 Software Reset does not clear the LOCK bit. 1 Software Reset clears the LOCK bit.
0 LOCK	The LOCK bit is used to restrict access to update SDMA script memory through ROM channel zero scripts and through the OnCE interface under ARM platform control. The LOCK bit is set: • The SDMA_LOCK, ONCE_ENB,CH0ADDR, and ILLINSTADDR registers cannot be written. These registers can be read, but writes are ignored. • SDMA software executing out of ROM or RAM may check the LOCK bit in the LOCK register Lock Status Register (SDMACORE_SDMA_LOCK) to determine if certain operations are allowed, such as up-loading new scripts.

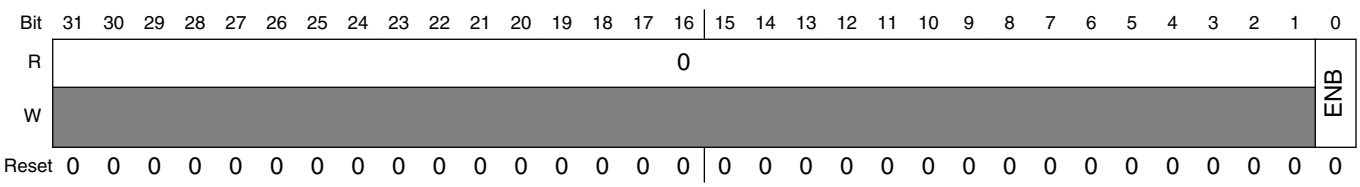
Table continues on the next page...

SDMAARM_SDMA_LOCK field descriptions (continued)

Field	Description
	Once the LOCK bit is set to 1, only a reset can clear it. The LOCK bit is cleared by a hardware reset. LOCK is cleared by a software reset only if SRESET_LOCK_CLR is set. 0 LOCK disengaged. 1 LOCK enabled.

66.20.16 OnCE Enable (SDMAARM_ONCE_ENB)

Address: SDMAARM_ONCE_ENB is 63FB_0000h base + 40h offset = 63FB_0040h

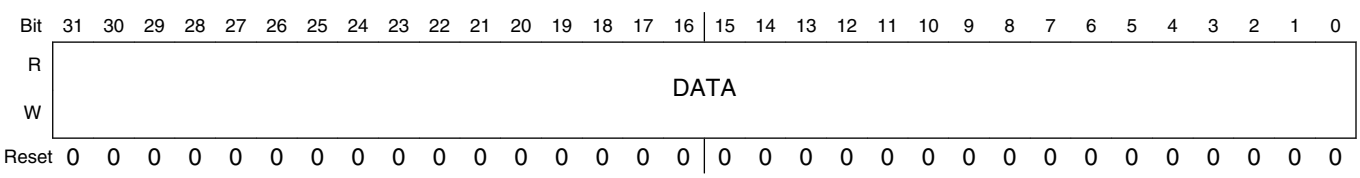


SDMAARM_ONCE_ENB field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 ENB	The OnCE Enable register selects the OnCE control source: When cleared (0), the OnCE registers are accessed through the JTAG interface; when set (1), the OnCE registers may be accessed by the ARM platform through the addresses described, as follows. • After reset, the OnCE registers are accessed through the JTAG interface. • Writing a 1 to ENB enables the ARM platform to access the ONCE_* as any other SDMA control register. • When cleared (0), all the ONCE_xxx registers cannot be written. The value of ENB cannot be changed if the LOCK bit in the SDMA_LOCK register is set.

66.20.17 OnCE Data Register (SDMAARM_ONCE_DATA)

Address: SDMAARM_ONCE_DATA is 63FB_0000h base + 44h offset = 63FB_0044h

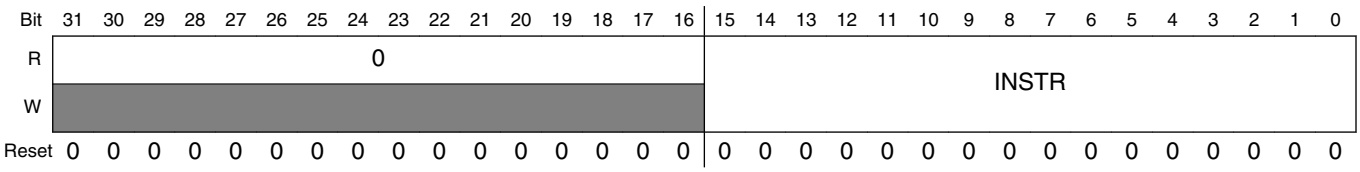


SDMAARM_ONCE_DATA field descriptions

Field	Description
31–0 DATA	Data register of the OnCE JTAG controller. Refer to OnCE and Real-Time Debug for information on this register.

66.20.18 OnCE Instruction Register (SDMAARM_ONCE_INSTR)

Address: SDMAARM_ONCE_INSTR is 63FB_0000h base + 48h offset = 63FB_0048h

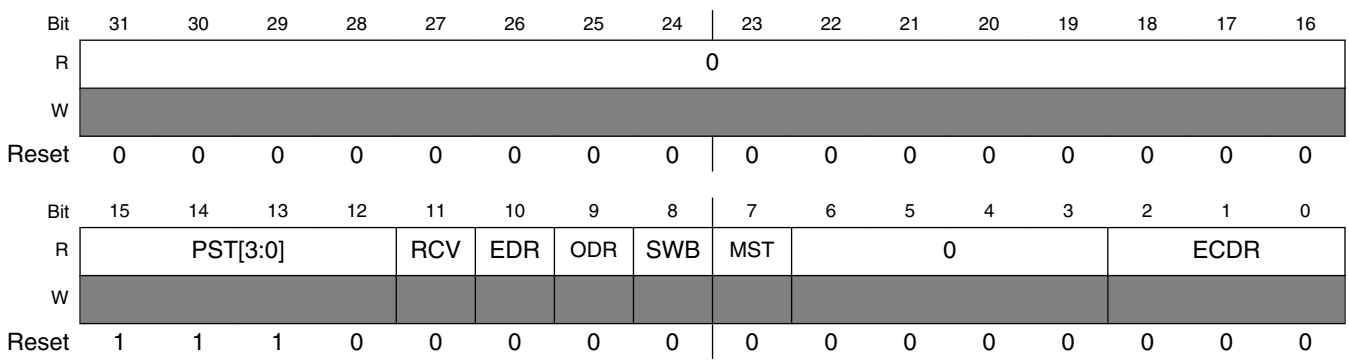


SDMAARM_ONCE_INSTR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–0 INSTR	Instruction register of the OnCE JTAG controller. Refer to OnCE and Real-Time Debug for information on this register.

66.20.19 OnCE Status Register (SDMAARM_ONCE_STAT)

Address: SDMAARM_ONCE_STAT is 63FB_0000h base + 4Ch offset = 63FB_004Ch



SDMAARM_ONCE_STAT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–12 PST[3:0]	The Processor Status bits reflect the state of the SDMA RISC engine. Its states are as follows: • The "Program" state is the usual instruction execution cycle. • The "Data" state is inserted when there are wait-states during a load or a store on the data bus (ld or st). • The "Change of Flow" state is the second cycle of any instruction that breaks the sequence of instructions (jumps and channel switching instructions). • The "Change of Flow in Loop" state is used when an error causes a hardware loop exit. • The "Debug" state means the SDMA is in debug mode.

Table continues on the next page...

SDMAARM_ONCE_STAT field descriptions (continued)

Field	Description
	• The "Functional Unit" state is inserted when there are wait-states during a load or a store on the functional units bus (ldf or stf). • In "Sleep" modes, no script is running (this is the RISC engine idle state). The "after Reset" is slightly different because no context restoring phase will happen when a channel is triggered: The script located at address 0 will be executed (boot operation). • The "in Sleep" states are the same as above except they do not have any corresponding channel: They are used when entering debug mode after reset. The reason is that it is necessary to return to the "Sleep after Reset" state when leaving debug mode. 0 Program 1 Data 2 Change of Flow 3 Change of Flow in Loop 4 Debug 5 Functional Unit 6 Sleep 7 Save 8 Program in Sleep 9 Data in Sleep 10 Change of Flow in Sleep 11 Change Flow in Loop in Sleep 12 Debug in Sleep 13 Functional Unit in Sleep 14 Sleep after Reset 15 Restore
11 RCV	After each write access to the real time buffer (RTB), the RCV bit is set. This bit is cleared after execution of an `rbuffer` command and on a JTAG reset.
10 EDR	This flag is raised when the SDMA has entered debug mode after an external debug request.
9 ODR	This flag is raised when the SDMA has entered debug mode after a OnCE debug request.
8 SWB	This flag is raised when the SDMA has entered debug mode after a software breakpoint.
7 MST	This flag is raised when the OnCE is controlled from the ARM platform peripheral interface. 0 The JTAG interface controls the OnCE. 1 The ARM platform peripheral interface controls the OnCE.
6–3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 ECDR	Event Cell Debug Request. If the debug request comes from the event cell, the reason for entering debug mode is given by the EDR bits. If all three bits of the EDR are reset, then it did not generate any debug request. If the cell did generate a debug request, then at least one of the EDR bits is set (the meaning of the encoding is given below). The encoding of the EDR bits is useful to find out more precisely why the debug request was generated. A debug request from an event cell is generated for a specific combination of the addra_cond, addrb_cond, and data_cond conditions. The value of those fields is given by the EDR bits. 0 1 matched addra_cond

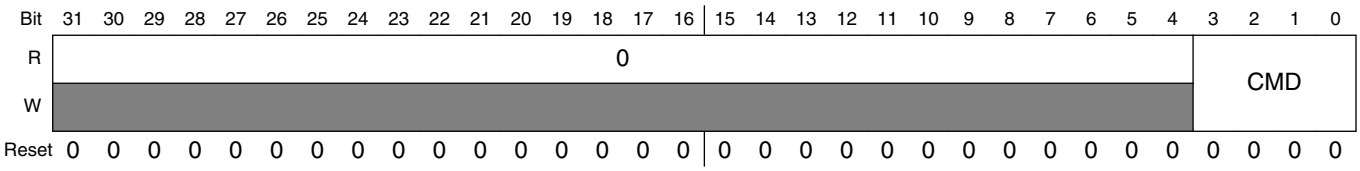
Table continues on the next page...

SDMAARM_ONCE_STAT field descriptions (continued)

Field	Description
	1 1 matched addrb_cond 2 1 matched data_cond

66.20.20 OnCE Command Register (SDMAARM_ONCE_CMD)

Address: SDMAARM_ONCE_CMD is 63FB_0000h base + 50h offset = 63FB_0050h

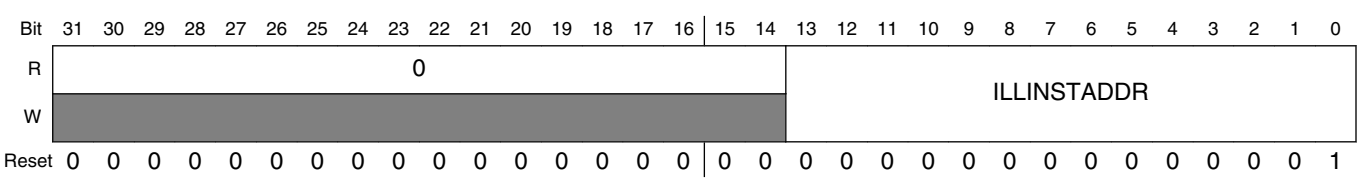


SDMAARM_ONCE_CMD field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value zero. Reserved
3–0 CMD	Writing to this register will cause the OnCE to execute the command that is written. When needed, the ONCE_DATA and ONCE_INSTR registers should be loaded with the correct value before writing the command to that register. For a list of the OnCE commands and their usage, see OnCE and Real-Time Debug. **NOTE:** 7-15 *reserved* 0 rstatus 1 dmov 2 exec_once 3 run_core 4 exec_core 5 debug_rqst 6 rbuffer

66.20.21 Illegal Instruction Trap Address (SDMAARM_ILLINSTADDR)

Address: SDMAARM_ILLINSTADDR is 63FB_0000h base + 58h offset = 63FB_0058h

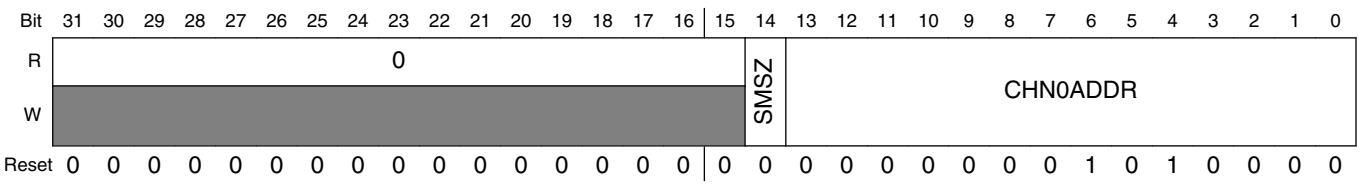


SDMAARM_ILLINSTADDR field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13–0 ILLINSTADDR	The Illegal Instruction Trap Address is the address where the SDMA jumps when an illegal instruction is executed. It is 0x0001 after reset. The value of ILLINSTADDR cannot be changed if the LOCK bit in the SDMA_LOCK register is set.

66.20.22 Channel 0 Boot Address (SDMAARM_CHN0ADDR)

Address: SDMAARM_CHN0ADDR is 63FB_0000h base + 5Ch offset = 63FB_005Ch



SDMAARM_CHN0ADDR field descriptions

Field	Description
31–15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14 SMSZ	The bit 14 (Scratch Memory Size) determines if scratch memory must be available after every channel context. After reset, it is equal to 0, which defines a RAM space of 24 words for each channel. All of this area stores the channel context. By setting this bit, 32 words are reserved for every channel context, which gives eight additional words that can be used by the channel script to store any type of data. Those words are never erased by the context switching mechanism. The value of SMSZ cannot be changed if the LOCK bit in the SDMA_LOCK register is set. 0 24 words per context 1 32 words per context
13–0 CHN0ADDR	This 14-bit register is used by the boot code of the SDMA. After reset, it points to the standard boot routine in ROM (channel 0 routine). By changing this address, you can perform a boot sequence with your own routine. The very first instructions of the boot code fetch the contents of this register (it is also mapped in the SDMA memory space) and jump to the given address. The reset value is 0x0050 (decimal 80). The value of CHN0ADDR cannot be changed if the LOCK bit in the SDMA_LOCK register is set.

66.20.23 DMA Requests (SDMAARM_EVT_MIRROR)

Address: SDMAARM_EVT_MIRROR is 63FB_0000h base + 60h offset = 63FB_0060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EVENTS																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_EVT_MIRROR field descriptions

Field	Description
31–0 EVENTS	This register reflects the DMA requests received by the SDMA for events 31-0. The ARM platform and the SDMA have a read-only access. There is one bit associated with each of 32 DMA request events. This information may be useful during debug of the blocks that generate the DMA requests. The EVT_MIRROR register is cleared following read access. 0 DMA request event not pending 1 DMA request event pending

66.20.24 DMA Requests 2 (SDMAARM_EVT_MIRROR2)

Address: SDMAARM_EVT_MIRROR2 is 63FB_0000h base + 64h offset = 63FB_0064h

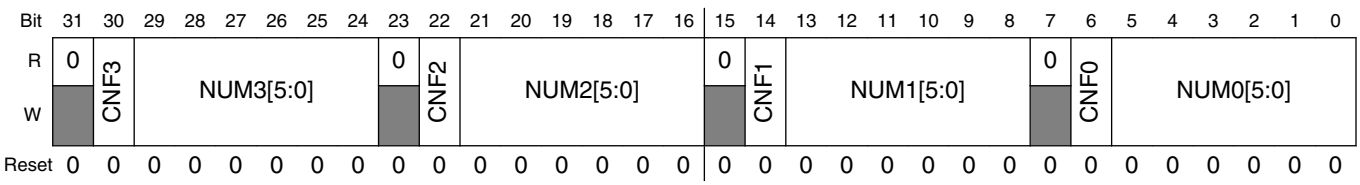
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																EVENTS[47:32]															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_EVT_MIRROR2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–0 EVENTS[47:32]	This register reflects the DMA requests received by the SDMA for events 47-32. The ARM platform and the SDMA have a read-only access. There is one bit associated with each of DMA request events. This information may be useful during debug of the blocks that generate the DMA requests. The EVT_MIRROR2 register is cleared following read access. 0 - DMA request event not pending 1- DMA request event pending

66.20.25 Cross-Trigger Events Configuration Register 1 (SDMAARM_XTRIG_CONF1)

Address: SDMAARM_XTRIG_CONF1 is 63FB_0000h base + 70h offset = 63FB_0070h



SDMAARM_XTRIG_CONF1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30 CNF3	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by the reception of a DMA request or by the starting of a channel script execution. 0 channel 1 DMA request
29–24 NUM3[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22 CNF2	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
21–16 NUM2[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14 CNF1	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
13–8 NUM1[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
7 Reserved	This read-only field is reserved and always has the value zero. Reserved

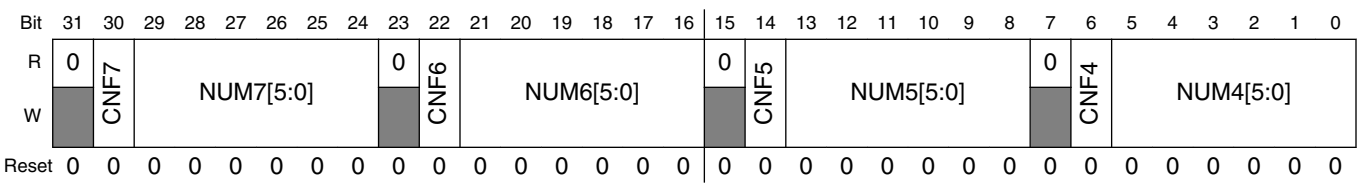
Table continues on the next page...

SDMAARM_XTRIG_CONF1 field descriptions (continued)

Field	Description
6 CNF0	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
5–0 NUM0[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.

66.20.26 Cross-Trigger Events Configuration Register 2 (SDMAARM_XTRIG_CONF2)

Address: SDMAARM_XTRIG_CONF2 is 63FB_0000h base + 74h offset = 63FB_0074h



SDMAARM_XTRIG_CONF2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved
30 CNF7	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
29–24 NUM7[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22 CNF6	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
21–16 NUM6[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved

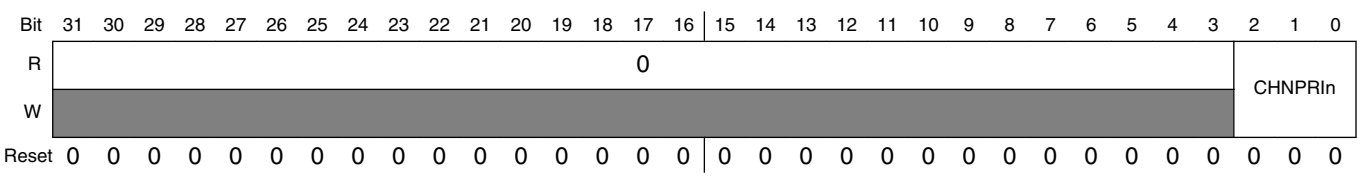
Table continues on the next page...

SDMAARM_XTRIG_CONF2 field descriptions (continued)

Field	Description
14 CNF5	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution 0 channel 1 DMA request
13–8 NUM5[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6 CNF4	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
5–0 NUM4[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.

66.20.27 Channel Priority Registers (SDMAARM_SDMA_CHNPRI*n*)

Addresses: 63FB_0000h base + 100h offset + (4d × *n*), where *n* = 0d to 31d



SDMAARM_SDMA_CHNPRI*n* field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 CHNPRIn	This contains the priority of channel number *n*. Useful values are between 1 and 7; 0 is reserved by the SDMA hardware to determine when there is no pending channel. Reset value is 0, which prevents the channels from starting.

66.20.28 Channel Enable RAM (SDMAARM_SDMA.CHNENBL*n*)

Addresses: 63FB_0000h base + 200h offset + (4d × *n*), where *n* = 0d to 47d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ENBLn																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARM_SDMA.CHNENBL*n* field descriptions

Field	Description
31–0 ENBLn	This 32-bit value selects the channels that are triggered by the DMA request number *n*. If ENBLn[i] is set to 1, bit EP[i] will be set when the DMA request *n* is received. These 48 32-bit registers are physically located in a RAM, with no known reset value. It is thus essential for the ARM platform to program them before any DMA request is triggered to the SDMA, otherwise an unpredictable combination of channels may be started.

66.21 BP Memory Map and Control Register Definitions

The following section describes SDMA control registers available to the BP.

NOTE

These registers are physically implemented in all platforms, but are not accessible when the SDMA BP control port is not connected. Reset values are calculated to allow the system to work when those registers cannot be accessed.

All registers are clocked with the SDMA clock (which means the SDMA clock must be running when the BP wants to access any register).

SDMABP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FB_0000	Channel 0 Pointer (SDMABP_DC0PTR)	32	R/W	0000_0000h	66.21.1/ 4249
63FB_0004	Channel Interrupts (SDMABP_INTR)	32	w1c	0000_0000h	66.21.2/ 4249
63FB_0008	Channel Stop/Channel Status (SDMABP_STOP_STAT)	32	R/W	0000_0000h	66.21.3/ 4250

Table continues on the next page...

SDMABP memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FB_000C	Channel Start (SDMABP_DSTART)	32	R	0000_0000h	66.21.4/ 4250
63FB_0028	DMA Request Error Register (SDMABP_EVTERR)	32	R	0000_0000h	66.21.5/ 4251
63FB_002C	Channel DSP Interrupt Mask (SDMABP_INTRMASK)	32	R/W	0000_0000h	66.21.6/ 4251
63FB_0034	DMA Request Error Register (SDMABP_EVTERRDBG)	32	R	0000_0000h	66.21.7/ 4252

66.21.1 Channel 0 Pointer (SDMABP_DC0PTR)

Address: SDMABP_DC0PTR is 63FB_0000h base + 0h offset = 63FB_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DC0PTR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABP_DC0PTR field descriptions

Field	Description
31–0 DC0PTR	**Channel 0 Pointer** contains the 32-bit address, in BP memory, of the array of channel control blocks starting with the one for channel 0 (the control channel). This register should be initialized by the BP before it enables a channel (for example, channel 0). See the API document i.MX50 SDMA Scripts User Manual for the use of this register. The BP has a read/write access and the SDMA has a read-only access.

66.21.2 Channel Interrupts (SDMABP_INTR)

Address: SDMABP_INTR is 63FB_0000h base + 4h offset = 63FB_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DI																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABP_INTR field descriptions

Field	Description
31–0 DI	The BP Interrupts register contains the 32 DI[i] bits. If any bit is set, it will cause an interrupt to the BP. • This register is a "write-ones" register to the BP. When the BP sets a bit in this register, the corresponding DI[i] bit is cleared.

SDMABP_INTR field descriptions (continued)

Field	Description
	• The interrupt service routine should clear individual channel bits when their interrupts are serviced; failure to do so will cause continuous interrupts. • The SDMA is responsible for setting the DI[i] bit corresponding to the current channel when the corresponding `done` instruction is executed.

66.21.3 Channel Stop/Channel Status (SDMABP_STOP_STAT)

Address: SDMABP_STOP_STAT is 63FB_0000h base + 8h offset = 63FB_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DE																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABP_STOP_STAT field descriptions

Field	Description
31–0 DE	This 32-bit register gives access to the BP (DSP) Enable bits, DE. There is one bit for every channel. • This register is a "write-ones" register to the BP. • When the BP writes 1 in bit *i* of this register, it clears the DE[i] and DSTART[i] bits. • Reading this register yields the current state of the DE[i] bits.

66.21.4 Channel Start (SDMABP_DSTART)

Address: SDMABP_DSTART is 63FB_0000h base + Ch offset = 63FB_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DSTART/DE																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABP_DSTART field descriptions

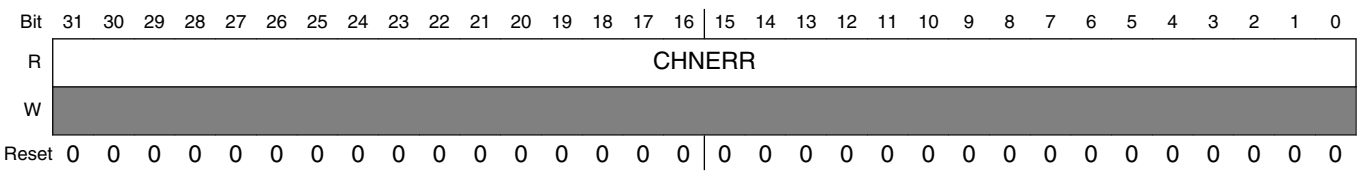
Field	Description
31–0 DSTART/DE	The DSTART/DE registers are 32 bits wide with one bit for every channel. • When a bit is written to 1, it enables the corresponding channel. • Two physical registers are accessed with that address (DSTART and DE), which enables the BP to trigger a channel a second time before the first trigger was processed. • This register is a "write-ones" register to the BP. Neither DSTART[i] bit can be set while the corresponding DE[i] bit is cleared. • When the BP tries to set the DSTART[i] bit by writing a one (if the corresponding DE[i] bit is clear), the bit in the DSTART[i] register will remain cleared and the DE[i] bit will be set. If the corresponding DE[i] bit was already set, the DSTART[i] bit will be set.

SDMABP_DSTART field descriptions (continued)

Field	Description
	• The next time the SDMA channel *i* attempts to clear the DE[i] bit by means of a `done` instruction, the bit in the DSTART[i] register will be cleared and the DE[i] bit will take the old value of the DSTART[i] bit. • Reading this register yields the current state of the DSTART[i] bits. This mechanism enables the BP to pipeline two DSTART commands per channel.

66.21.5 DMA Request Error Register (SDMABP_EVTERR)

Address: SDMABP_EVTERR is 63FB_0000h base + 28h offset = 63FB_0028h

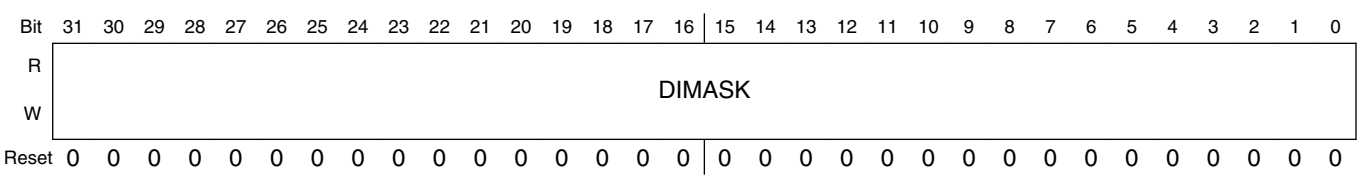


SDMABP_EVTERR field descriptions

Field	Description
31–0 CHNERR	This register is used by the SDMA to warn the BP when an incoming DMA request was detected; it then triggers a channel that is already pending or being serviced, which may mean there is an overflow of data for that channel. An interrupt is sent to the BP if the corresponding channel bit is set in the INTRMASK register. • This is a "write-ones" register for the scheduler. It is only able to set the flags. The flags are cleared when the register is read by the BP or during an SDMA reset. • The CHNERR[i] bit is set when a DMA request that triggers channel *i* is received through the corresponding input pins and the EP[i] bit is already set. The EVTERR[i] bit is unaffected if the BP tries to set the EP[i] bit when that EP[i] bit is already set.

66.21.6 Channel DSP Interrupt Mask (SDMABP_INTRMASK)

Address: SDMABP_INTRMASK is 63FB_0000h base + 2Ch offset = 63FB_002Ch



SDMABP_INTRMASK field descriptions

Field	Description
31–0 DIMASK	The Interrupt Mask Register contains 32 interrupt generation mask bits. If bit DIMASK[i] is set, the DI[i] bit is set and an interrupt is sent to the BP when a DMA request error is detected on channel *i* (for example, EVTERR[i] is set).

66.21.7 DMA Request Error Register (SDMABP_EVTERRDBG)

Address: SDMABP_EVTERRDBG is 63FB_0000h base + 34h offset = 63FB_0034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHNERR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABP_EVTERRDBG field descriptions

Field	Description
31–0 CHNERR	This register is the same as EVTERR except reading it does not clear its contents. This address is meant to be used in debug mode. The BP OnCE may check this register value without modifying it.

66.22 SDMA Internal (Core) Memory Map and Internal Register Definitions

The actual SDMA memory mapped registers are summarized in the following sections; for peripherals' memory maps, refer to the respective chapters.

The following definitions serve as a key for the SDMA internal register summary.

SDMACORE memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FB_0000	ARM platform Channel 0 Pointer (SDMACORE_MC0PTR)	32	R	0000_0000h	66.22.1/ 4253
63FB_0002	Current Channel Pointer (SDMACORE_CCPTR)	32	R	0000_0000h	66.22.2/ 4254
63FB_0003	Current Channel Register (SDMACORE_CCR)	32	R	0000_0000h	66.22.3/ 4254
63FB_0004	Highest Pending Channel Register (SDMACORE_NCR)	32	R	0000_0000h	66.22.4/ 4255
63FB_0005	External DMA Requests Mirror (SDMACORE_EVENTS)	32	R	0000_0000h	66.22.5/ 4256
63FB_0006	Current Channel Priority (SDMACORE_CCPRI)	32	R	0000_0000h	66.22.6/ 4257
63FB_0007	Next Channel Priority (SDMACORE_NCPRI)	32	R	0000_0000h	66.22.7/ 4257

Table continues on the next page...

SDMACORE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FB_0009	OnCE Event Cell Counter (SDMACORE_ECOUNT)	32	R/W	0000_0000h	66.22.8/ 4258
63FB_000A	OnCE Event Cell Control Register (SDMACORE_ECTL)	32	R/W	0000_0000h	66.22.9/ 4258
63FB_000B	OnCE Event Address Register A (SDMACORE_EAA)	32	R/W	0000_0000h	66.22.10/ 4260
63FB_000C	OnCE Event Cell Address Register B (SDMACORE_EAB)	32	R/W	0000_0000h	66.22.11/ 4260
63FB_000D	OnCE Event Cell Address Mask (SDMACORE_EAM)	32	R/W	0000_0000h	66.22.12/ 4261
63FB_000E	OnCE Event Cell Data Register (SDMACORE_ED)	32	R/W	0000_0000h	66.22.13/ 4261
63FB_000F	OnCE Event Cell Data Mask (SDMACORE_EDM)	32	R/W	0000_0000h	66.22.14/ 4261
63FB_0018	OnCE Real-Time Buffer (SDMACORE_RTB)	32	R/W	0000_0000h	66.22.15/ 4262
63FB_0019	OnCE Trace Buffer (SDMACORE_TB)	32	R	0000_0000h	66.22.16/ 4262
63FB_001A	OnCE Status (SDMACORE_OSTAT)	32	R	0000_0000h	66.22.17/ 4263
63FB_001C	Channel 0 Boot Address (SDMACORE_MCHN0ADDR)	32	R	0000_0000h	66.22.18/ 4265
63FB_001D	ENDIAN Status Register (SDMACORE_ENDIANNESS)	32	R	0000_0001h	66.22.19/ 4266
63FB_001E	Lock Status Register (SDMACORE_SDMA_LOCK)	32	R	0000_0000h	66.22.20/ 4267
63FB_001F	External DMA Requests Mirror #2 (SDMACORE_EVENTS2)	32	R	0000_0000h	66.22.21/ 4267

66.22.1 ARM platform Channel 0 Pointer (SDMACORE_MC0PTR)

Address: SDMACORE_MC0PTR is 63FB_0000h base + 0h offset = 63FB_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MC0PTR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_MC0PTR field descriptions

Field	Description
31–0 MC0PTR	Contains the address-in the ARM platform memory space-of the initial SDMA context and scripts that are loaded by the SDMA boot script running on channel 0.

66.22.2 Current Channel Pointer (SDMACORE_CCPTR)

Address: SDMACORE_CCPTR is 63FB_0000h base + 2h offset = 63FB_0002h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																CCPTR															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_CCPTR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–0 CCPTR	Contains the start address of the context data for the current channel: Its value is *CONTEXT_BASE* + 24* *CCR* or *CONTEXT_BASE* + 32* *CCR* where CONTEXT_BASE = 0x0800. The value 24 or 32 is selected according to the programmed channel scratch RAM size in the register shown in Channel 0 Boot Address (SDMAARM_CHN0ADDR) .

66.22.3 Current Channel Register (SDMACORE_CCR)

Address: SDMACORE_CCR is 63FB_0000h base + 3h offset = 63FB_0003h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											CCR				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_CCR field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4–0 CCR	Contains the number of the current running channel whose context is installed. In the case that the SDMA has finished running the channel and has entered sleep state, CCR will indicate the previous running channel. The PST bits in the OSTAT register indicate when the SDMA is in sleep state.

66.22.4 Highest Pending Channel Register (SDMACORE_NCR)

Address: SDMACORE_NCR is 63FB_0000h base + 4h offset = 63FB_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											NCR				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_NCR field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4–0 NCR	Contains the number of the pending channel that the scheduler has selected to run next.

66.22.5 External DMA Requests Mirror (SDMACORE_EVENTS)

This register is very useful in the case of DMA requests that are active when a peripheral FIFO level is above the programmed watermark. The activation of the DMA request (rising edge) is detected by the SDMA logic and it can enable one or several channels. One of the channels accesses the peripheral and reads or writes a number of data that matches the watermark level (for example, if the watermark is four words, the channel reads or writes four words). If the channel is effectively executed long after the DMA request was received, reading or writing the watermark number of data may not be sufficient to reset the DMA request (for example, if the FIFO watermark is four and at the channel execution it already contains nine pieces of data). This means no new rising edge may be detected by the SDMA, although there still remains transfers to perform. Therefore, if the channel were terminated at that time, it would not be restarted, causing potential overrun or underrun of the peripheral.The proposed mechanism is for the channel to check this register after it has performed the "watermark" number of accesses to the peripheral. If the bit for the DMA request that triggers this channel is set, it means there is still another watermark number of data to transfer. This goes on until the bit is cleared. The same script can be used for multiple channels that require this behavior. The script can determine its channel number from the CCR register and infer the corresponding DMA request bit to check. It needs a reference table that is coherent with the request-channel matrix that the ARM platform programmed.

Address: SDMACORE_EVENTS is 63FB_0000h base + 5h offset = 63FB_0005h

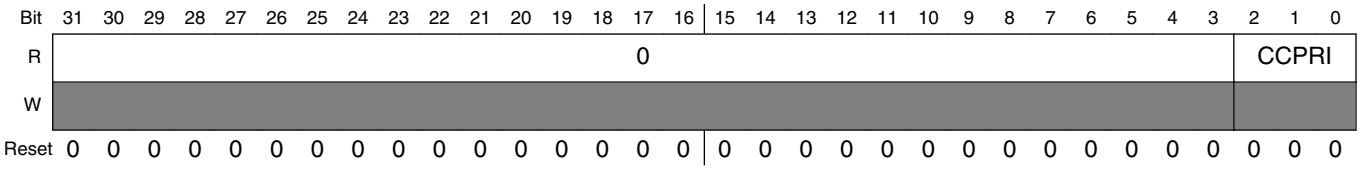
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EVENTS																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_EVENTS field descriptions

Field	Description
31–0 EVENTS	Reflects the status of the SDMA's external DMA requests. It is meant to allow any channel to monitor the states of these SDMA inputs. This register displays EVENTS 0-31. The EVENTS2 register displays events 32-47.

66.22.6 Current Channel Priority (SDMACORE_CCPRI)

Address: SDMACORE_CCPRI is 63FB_0000h base + 6h offset = 63FB_0006h

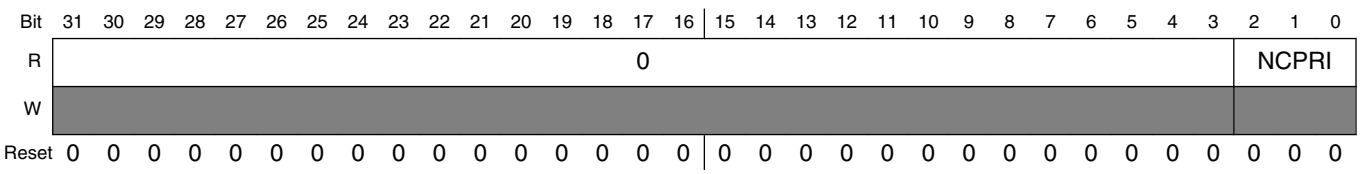


SDMACORE_CCPRI field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 CCPRI	Contains the 3-bit priority of the channel whose context is installed. It is 0 when no channel is running. **NOTE:** 1-7 current channel priority 0 no running channel

66.22.7 Next Channel Priority (SDMACORE_NCPRI)

Address: SDMACORE_NCPRI is 63FB_0000h base + 7h offset = 63FB_0007h

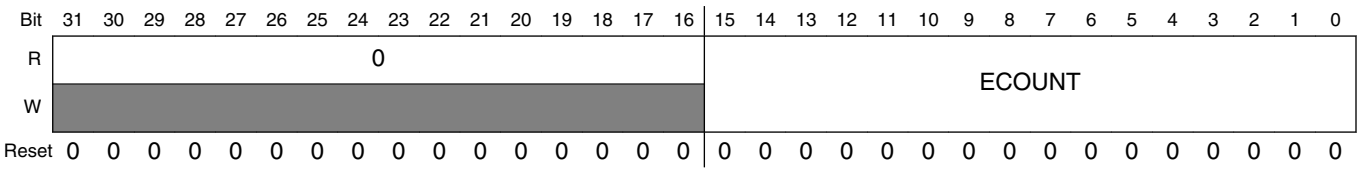


SDMACORE_NCPRI field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 NCPRI	Contains the 3-bit priority of the channel the scheduler has selected to run next. It is 0 when no other channel is pending.

66.22.8 OnCE Event Cell Counter (SDMACORE_ECOUNT)

Address: SDMACORE_ECOUNT is 63FB_0000h base + 9h offset = 63FB_0009h

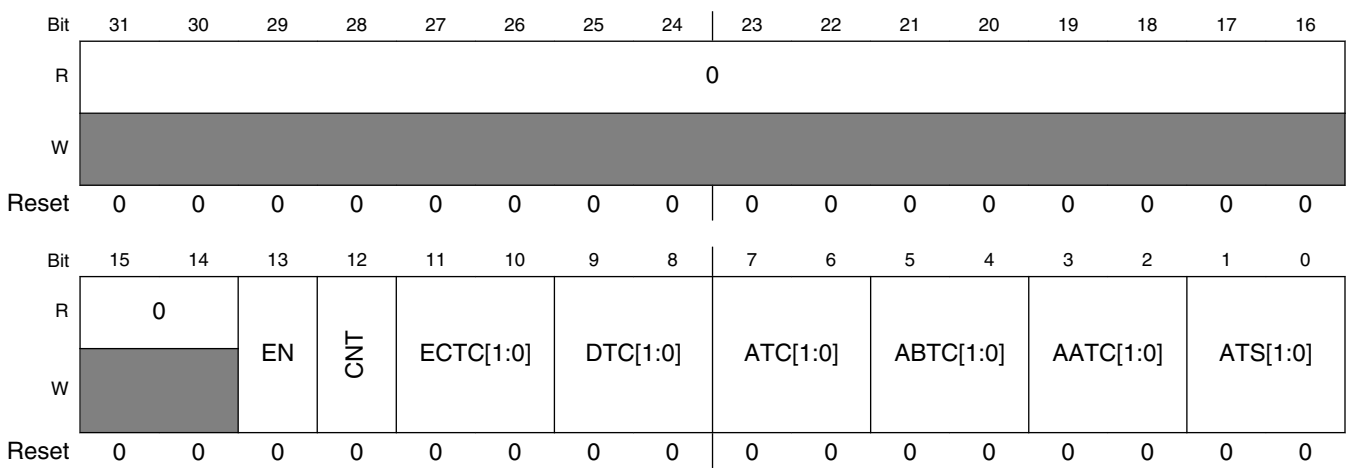


SDMACORE_ECOUNT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–0 ECOUNT	The event cell counter contains the number of times minus one that an event detection must occur before generating a debug request. • This register should be written before any attempt to use the event detection counter during an event detection process. • The counter is cleared on a JTAG reset.

66.22.9 OnCE Event Cell Control Register (SDMACORE_ECTL)

Address: SDMACORE_ECTL is 63FB_0000h base + Ah offset = 63FB_000Ah



SDMACORE_ECTL field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 EN	Event Cell Enable. If the EN bit is set, the event cell is allowed to generate debug requests (the cell is awakened). If it is cleared, the event detection unit is disabled and no hardware breakpoint is generated, but matching conditions are still reflected on the emulation pin.

Table continues on the next page...

SDMACORE_ECTL field descriptions (continued)

Field	Description
	0 Cell is disabled. 1 Cell is enabled.
12 CNT	Event Counter Enable. The event counter enable bit determines if the cell counter is used during the event detection. In order to use the event counter during an event detection process, the event cell counter register should be loaded with a value equal to the number of times minus one that an event occurs before a debug request is sent. After every event detection, the counter is decreased. When the counter reaches the value 0, the event detection cell sends a debug request to the core. The event counter register should be written and the EN bit should be set before each new event detection process uses the event counter. 0 Counter is disabled. 1 Counter is enabled.
11–10 ECTC[1:0]	The event cell trigger condition bits select the combination of address and data matching conditions that generate the final address/data condition. During program execution, if this event cell trigger condition goes to 1, a debug request is sent to the SDMA. The EN bit must be set to enable the debug request generation. 00 address ONLY 01 data ONLY 10 address AND data 11 address OR data
9–8 DTC[1:0]	The data trigger condition bits define when data is considered matching after comparison with the data register of the event detection unit. The operations are performed on unsigned values. 00 equal 01 not equal 10 greater than 11 less than
7–6 ATC[1:0]	The address trigger condition bits select how the two address conditions (addressA and addressB) are combined to define the global address matching condition. The supported combinations are described, as follows. 00 addressA ONLY 01 addrA AND addrB 10 addrA OR addrB 11 reserved
5–4 ABTC[1:0]	The Address B Trigger Condition (ABTC) controls the operations performed by address comparator B. All operations are performed on unsigned values. This comparator B outputs the addressB condition. 00 equal 01 not equal 10 greater than 11 less than
3–2 AATC[1:0]	The Address A Trigger Condition (AATC) controls the operations performed by address comparator A. All operations are performed on unsigned values. This comparator A outputs the addressA condition. 00 equal 01 not equal

Table continues on the next page...

SDMACORE_ECTL field descriptions (continued)

Field	Description
	10 greater than 11 less than
1–0 ATS[1:0]	The access type select bits define the memory access type required on the SDMA memory bus. 00 read ONLY 01 write ONLY 10 read or write 11 -

66.22.10 OnCE Event Address Register A (SDMACORE_EAA)

Address: SDMACORE_EAA is 63FB_0000h base + Bh offset = 63FB_000Bh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																EAA															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_EAA field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–0 EAA	Event Cell Address Register A computes an address A condition. It is cleared on a JTAG reset.

66.22.11 OnCE Event Cell Address Register B (SDMACORE_EAB)

Address: SDMACORE_EAB is 63FB_0000h base + Ch offset = 63FB_000Ch

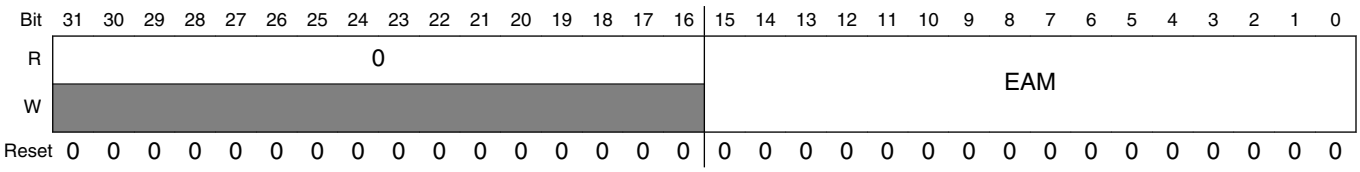
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																EAB															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_EAB field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–0 EAB	Event Cell Address Register B computes an address B condition. It is cleared on a JTAG reset.

66.22.12 OnCE Event Cell Address Mask (SDMACORE_EAM)

Address: SDMACORE_EAM is 63FB_0000h base + Dh offset = 63FB_000Dh

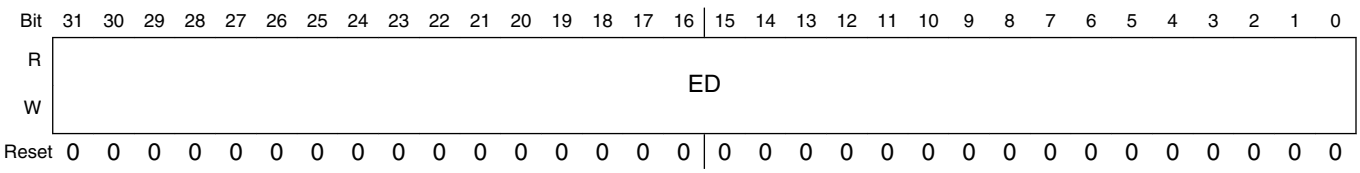


SDMACORE_EAM field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–0 EAM	The Event Cell Address Mask contains a user-defined address mask value. This mask is applied to the address value latched from the memory address bus before performing the address comparison. **NOTE:** There is a common address mask value for both address comparators. If bit *i* of this register is set, then bit *i* of the address value latched from the memory bus does not influence the result of the address comparison. The register is cleared on a JTAG reset.

66.22.13 OnCE Event Cell Data Register (SDMACORE_ED)

Address: SDMACORE_ED is 63FB_0000h base + Eh offset = 63FB_000Eh

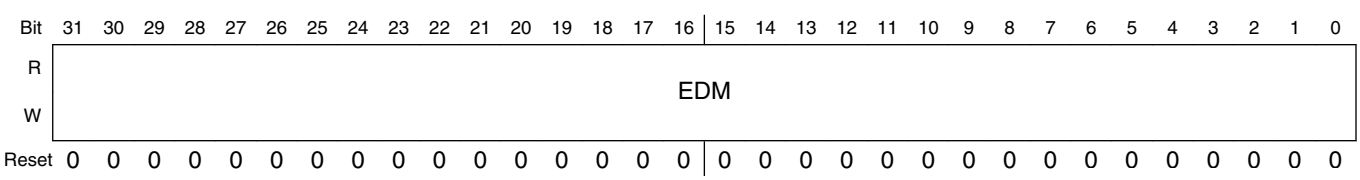


SDMACORE_ED field descriptions

Field	Description
31–0 ED	The event cell data register contains a user defined data value. This data value is an input for the data comparator which generates the data condition. It is cleared on a JTAG reset.

66.22.14 OnCE Event Cell Data Mask (SDMACORE_EDM)

Address: SDMACORE_EDM is 63FB_0000h base + Fh offset = 63FB_000Fh



SDMACORE_EDM field descriptions

Field	Description
31–0 EDM	The event cell data mask register contains the user-defined data mask value. • This mask is applied to the data value latched from the memory bus before performing the data comparison. • Setting bit *i* of the event cell data mask register means that bit *i* of the data value latched from the address bus does not influence the result of the data comparison. • The data mask is cleared on a JTAG reset.

66.22.15 OnCE Real-Time Buffer (SDMACORE_RTB)

Address: SDMACORE_RTB is 63FB_0000h base + 18h offset = 63FB_0018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RTB																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_RTB field descriptions

Field	Description
31–0 RTB	The Real Time Buffer register stores and retrieves run time information without putting the SDMA in debug mode. Writing to that register triggers a pulse on a specific real-time debug pin whose connection depends on the chip implementation. The RTB value can be accessed by the OnCE under ARM platform or JTAG control using the rbuffer command.

66.22.16 OnCE Trace Buffer (SDMACORE_TB)

Address: SDMACORE_TB is 63FB_0000h base + 19h offset = 63FB_0019h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			TBF	TADDR[-2:2]											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TADDR[1:0]			CHFADDR												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_TB field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved
28 TBF	The Trace Buffer Flag is set when the buffer contains the addresses of a valid change of flow. The contents of the buffer should be ignored otherwise. 0 Invalid information 1 Valid information
27–14 TADDR	The target address is the address taken after the execution of the change of flow instruction.
13–0 CHFADDR	The change of flow address is the address where the change of flow is taken when executing a change of flow instruction.

66.22.17 OnCE Status (SDMACORE_OSTAT)

Address: SDMACORE_OSTAT is 63FB_0000h base + 1Ah offset = 63FB_001Ah

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PST[3:0]				RCV	EDR	ODR	SWB	MST	0				ECDR[2:0]		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE_OSTAT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–12 PST[3:0]	The Processor Status bits reflect the state of the SDMA RISC engine. • The "Program" state is the usual instruction execution cycle. • The "Data" state is inserted when there are wait-states during a load or a store on the data bus (ld or st). • The "Change of Flow" state is the second cycle of any instruction that breaks the sequence of instructions (jumps and channel-switching instructions). • The "Change of Flow in Loop" state is used when an error causes a hardware loop exit. • The "Debug" state means the SDMA is in debug mode. • The "Functional Unit" state is inserted when there are wait-states during a load or a store on the functional units bus (ldf or stf).

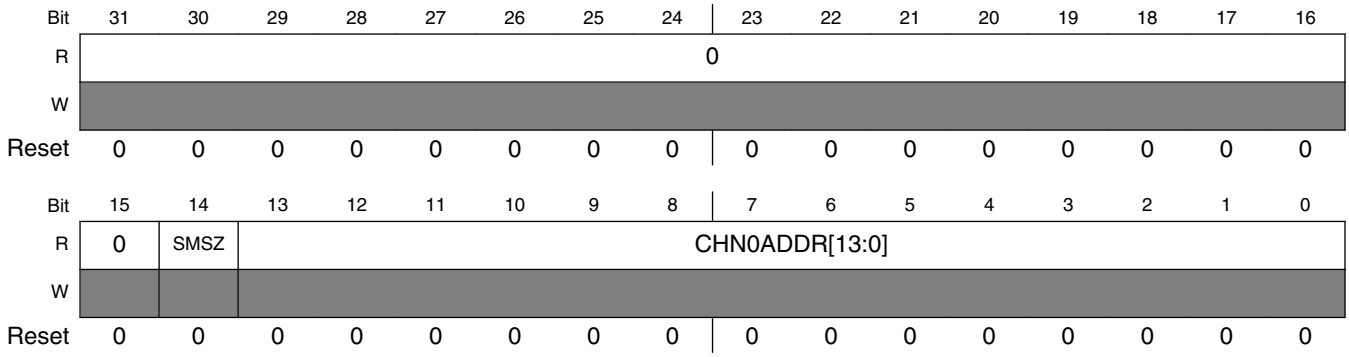
Table continues on the next page...

SDMACORE_OSTAT field descriptions (continued)

Field	Description
	• In "Sleep" modes, no script is running (this is the RISC engine idle state). The "after Reset" is slightly different because no context restoring phase will happen when a channel is triggered: The script located at address 0 will be executed (boot operation). • The "in Sleep" states are the same as above except they do not have any corresponding channel. They are used when entering debug mode after reset; the reason is that it is necessary to return to the "Sleep after Reset" state when leaving debug mode. 0 Program 1 Data 2 Change of Flow 3 Change of Flow in Loop 4 Debug 5 Functional Unit 6 Sleep 7 Save 8 Program in Sleep 9 Data in Sleep 10 Change of Flow in Sleep 11 Change Flow Loop Sleep 12 Debug in Sleep 13 Functional Unit in Sleep 14 Sleep after Reset 15 Restore
11 RCV	After each write access to the real time buffer (RTB), the RCV bit is set. This bit is cleared after execution of an `rbuffer` command and on a JTAG reset.
10 EDR	This flag is raised when the SDMA has entered debug mode after an external debug request.
9 ODR	This flag is raised when the SDMA has entered debug mode after a OnCE debug request.
8 SWB	This flag is raised when the SDMA has entered debug mode after a software breakpoint.
7 MST	This flag is raised when the OnCE is controlled from the ARM platform peripheral interface. 0 JTAG interface controls the OnCE. 1 ARM platform peripheral interface controls the OnCE.
6–3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–0 ECDR[2:0]	Event Cell Debug Request. If the debug request comes from the event cell, the reason for entering debug mode is given by the EDR bits. The encoding of the EDR bits is useful to find out more precisely why the debug request was generated. A debug request from an event cell is generated for a specific combination of the addressA, addressB, and data conditions; the value of those fields is given by the EDR bits. If all three bits of the EDR are reset, then it did not generate any debug request. If the cell did generate a debug request, then at least one EDR bit is set; the meaning of the encoding is as follows: 0 1 matched addressA condition 1 1 matched addressB condition 2 1 matched data condition

66.22.18 Channel 0 Boot Address (SDMACORE_MCHN0ADDR)

Address: SDMACORE_MCHN0ADDR is 63FB_0000h base + 1Ch offset = 63FB_001Ch

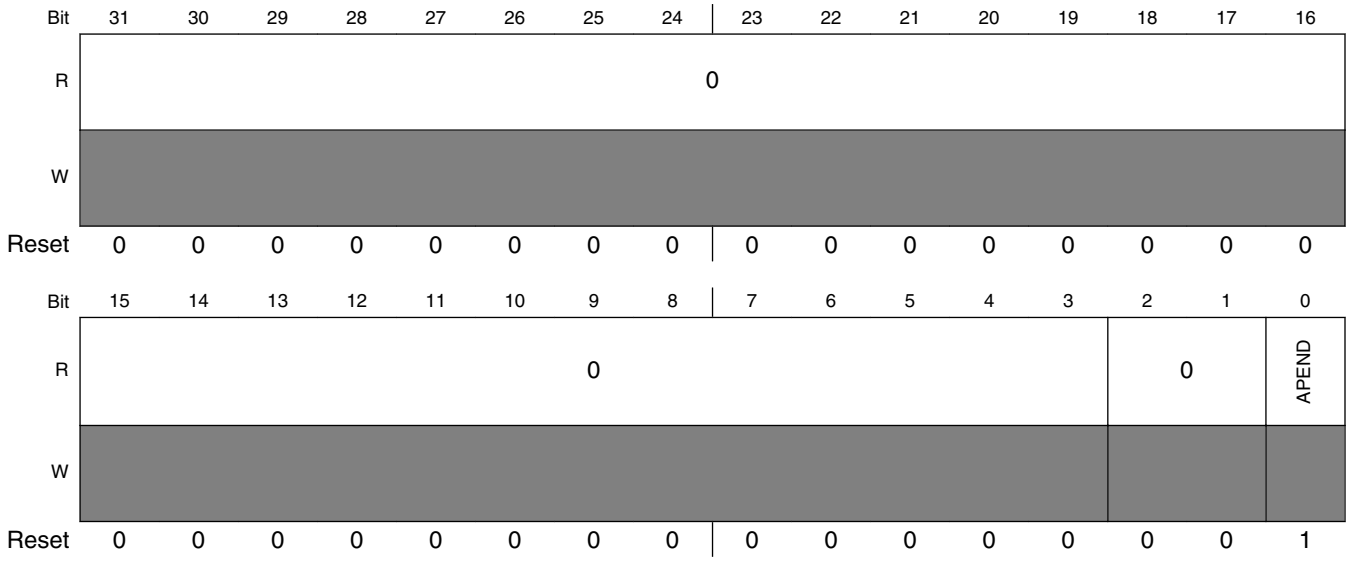


SDMACORE_MCHN0ADDR field descriptions

Field	Description
31–15 Reserved	This read-only field is reserved and always has the value zero. Reserved
14 SMSZ	The bit 14 (Scratch Memory Size) determines if scratch memory must be available after every channel context. After reset, it is equal to 0, which defines a RAM space of 24 words for each channel. All of this area stores the channel context. By setting this bit, 32 words are reserved for every channel context, which gives eight additional words that can be used by the channel script to store any type of data. Those words are never erased by the context switching mechanism. 0 24 words per context 1 32 words per context
13–0 CHN0ADDR[13:0]	Contains the address of the channel 0 routine programmed by the ARM platform; it is loaded into a general register at the very start of the boot and the SDMA jumps to the address it contains. By default, it points to the standard boot routine in ROM.

66.22.19 ENDIAN Status Register (SDMACORE_ENDIANNESS)

Address: SDMACORE_ENDIANNESS is 63FB_0000h base + 1Dh offset = 63FB_001Dh

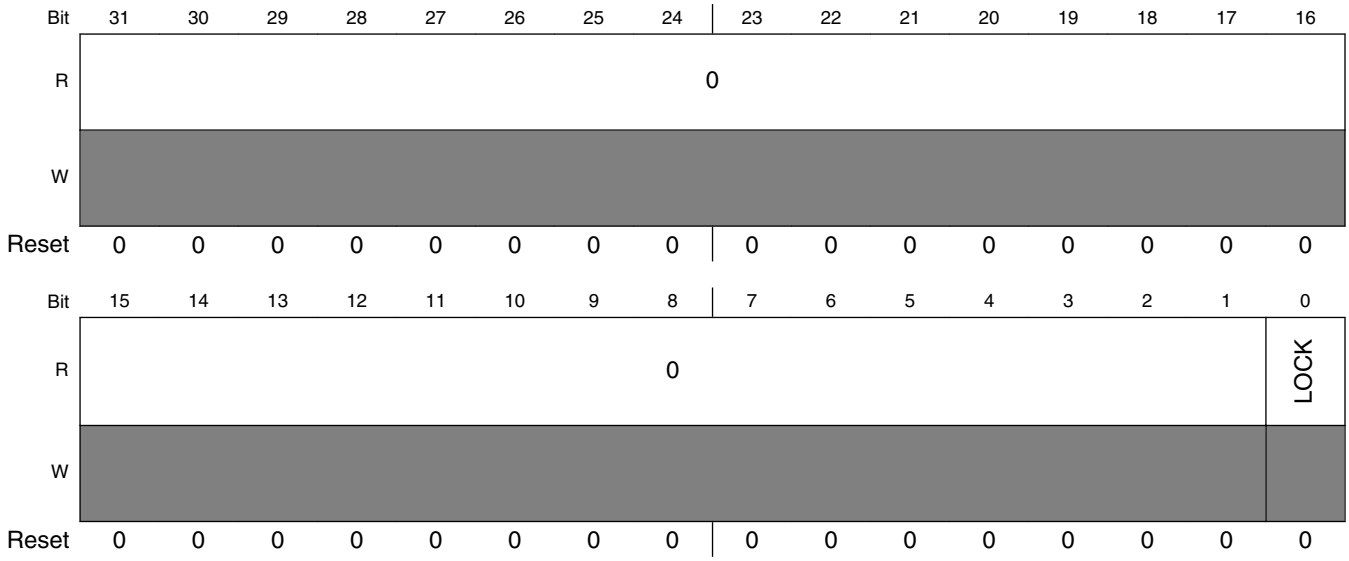


SDMACORE_ENDIANNESS field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value zero. Reserved
2–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 APEND	APEND indicates the endian mode of the Peripheral and Burst DMA interfaces. This bit is tied to logic '1' indicating little-endian mode. 0 - ARM platform is in big-endian mode 1 - ARM platform is in little-endian mode

66.22.20 Lock Status Register (SDMACORE_SDMA_LOCK)

Address: SDMACORE_SDMA_LOCK is 63FB_0000h base + 1Eh offset = 63FB_001Eh

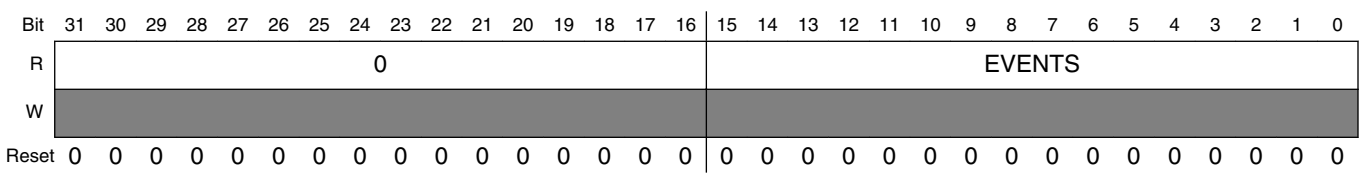


SDMACORE_SDMA_LOCK field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 LOCK	The LOCK bit reports the value of the LOCK bit in the SDMA_LOCK status register. SDMA software may use this value to determine if certain operations such as loading of new scripts is allowed. 0 - LOCK bit clear 1 - LOCK bit set

66.22.21 External DMA Requests Mirror #2 (SDMACORE_EVENTS2)

Address: SDMACORE_EVENTS2 is 63FB_0000h base + 1Fh offset = 63FB_001Fh



SDMACORE_EVENTS2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

SDMACORE_EVENTS2 field descriptions (continued)

Field	Description
15–0 EVENTS	Reflects the status of the SDMA's external DMA requests. It is meant to allow any channel to monitor the states of these SDMA inputs. This register displays EVENTS 32-47. The separate EVENTS register displays events 0-31.

66.23 SDMA Peripheral Registers

Refer to the respective peripherals' chapters more information.

Chapter 67
System JTAG Controller (SJC)

67.1 Introduction

The System JTAG Controller (SJC) provides debug and test control with the maximum security. The test access port (TAP) is designed to support features compatible with the IEEE Standard 1149.1 v2001 (JTAG).

The figure below shows an overview of the JTAG architecture.

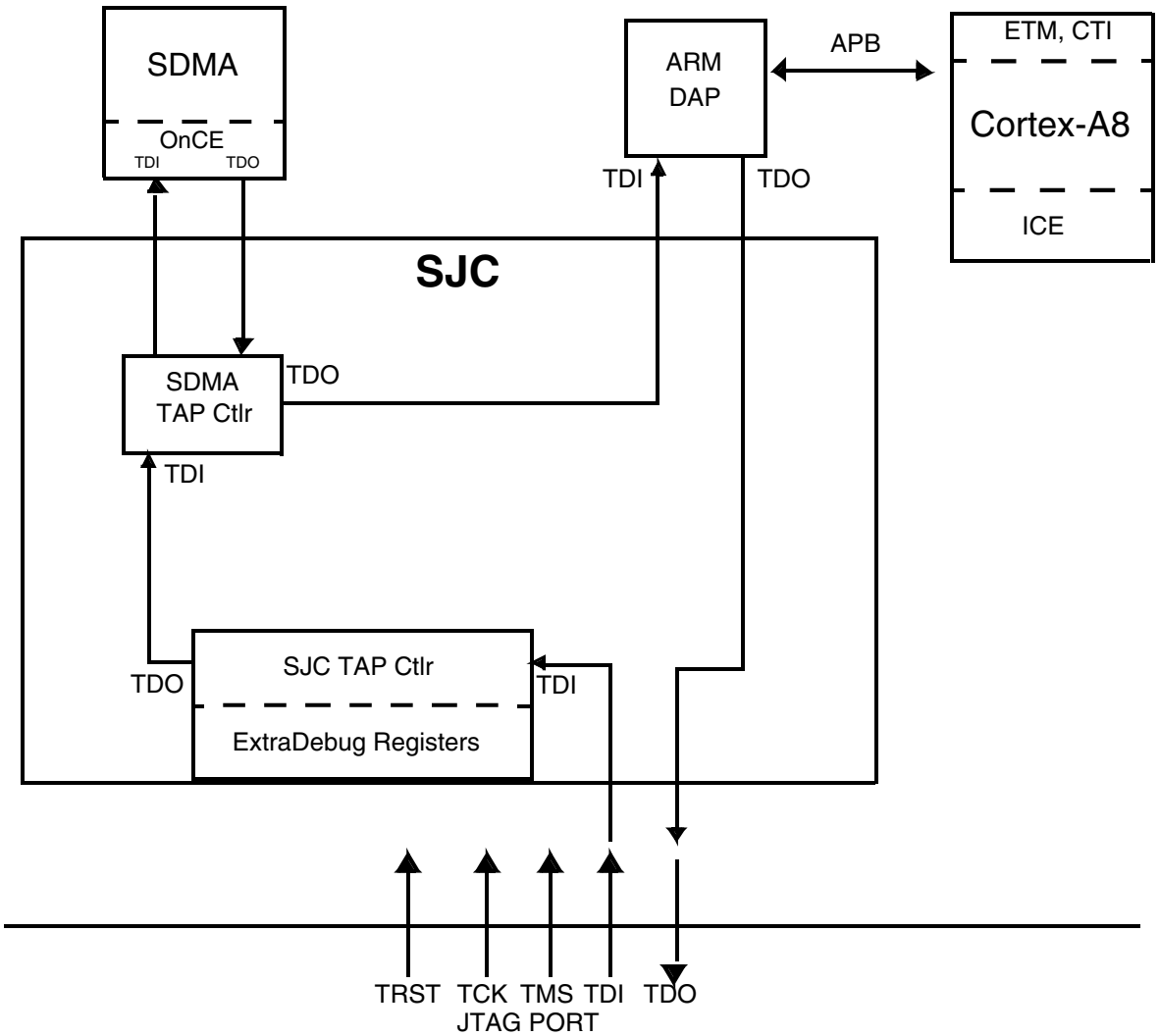


Figure 67-1. System JTAG Controller (SJC) Block Diagram

67.1.1 Overview

The System JTAG Controller (SJC) provides the following capabilities:

- JTAG IEEE1149.1 mandatory instructions, see EXTEST Instruction, SAMPLE/ PRELOAD Instruction, and BYPASS Instruction.
- JTAG IEEE1149.1 optional instructions, see ID_CODE Instruction (IDCODE), and HIGHZ Instruction.
- JTAG IEEE P1149.1 (standard JTAG) interface to off-chip test and development equipment including an SJC-only mode for true IEEE 1149.1 compliance, used primarily for board-level implementation of boundary scan.

- Debug-related control and status, such as putting selected cores into reset and/or debug mode and the ability to monitor individual core status signals via JTAG.
- Provides means for accessing each OnCE/ICE TAP controller independently to control a target system (see Modes of Operation).
- ExtraDebug logic (see ENABLE_ExtraDebug Instruction).
- The maximum clock speed of the SJC is one-eight of the lowest frequency of the accessed OnCE/ICE. For example in normal operation (no core in low-power mode), this frequency is one-eight of the SDMA frequency if this core is present in the TDI-TDO chain (serially connected with other cores or standalone). The user must also consider the 25 MHz frequency limitation on the CE bus.
- Core compliant modes to support standalone core debuggers (see Modes of Operation).
- Multi-cores daisy chained mode (default one) to support multi-core debuggers (see Modes of Operation).

Detailed information about the SJC is provided in the Security Reference Manual. Contact your Freescale representative for information about obtaining this document.

67.2 Modes of Operation

The SJC modes are controlled through both the TAP select register (SJC_TSR) and the "sjc_mod" input port.

The "sjc_mod" port (typically connected to pad of the same name: SJC_MOD) selects between two possible topologies of TAP connections, as seen at SoC level:

- Negating it (this should be the default state) selects all the TAPs (SJC, SDMA and DAP and ARM/ETM) to be connected in the TDI-TDO chain, which is referred to as "daisy chain" mode, throughout this chapter.
- Asserting it only selects the SJC TAP to be connected in the TDI-TDO chain.

IEEE1149.1 standard features are enabled by configuring the SJC input pin: SJC_MOD. Refer to the following table for SJC_MOD settings details:

Table 67-1. SJC Modes

SJC_MOD	Name	Description
0	Daisy chain ALL	For common SW debug (High speed and production)
1	SJC only	IEEE 1149.1 JTAG compliant mode

The following figure shows the SJC mode selection flow. The numbers shown in parenthesis below each block name indicates the TAP's IR length.

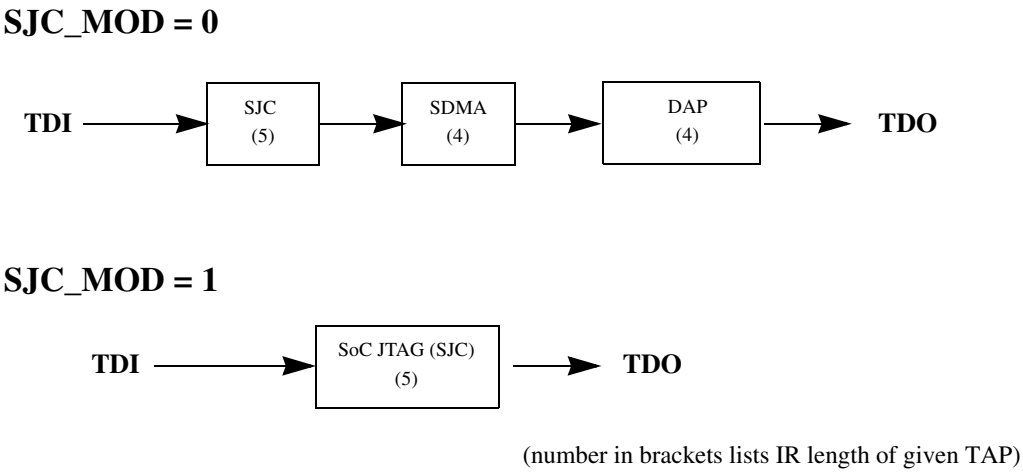


Figure 67-2. SJC Mode Selection Using sjc_mod Pin Sampling

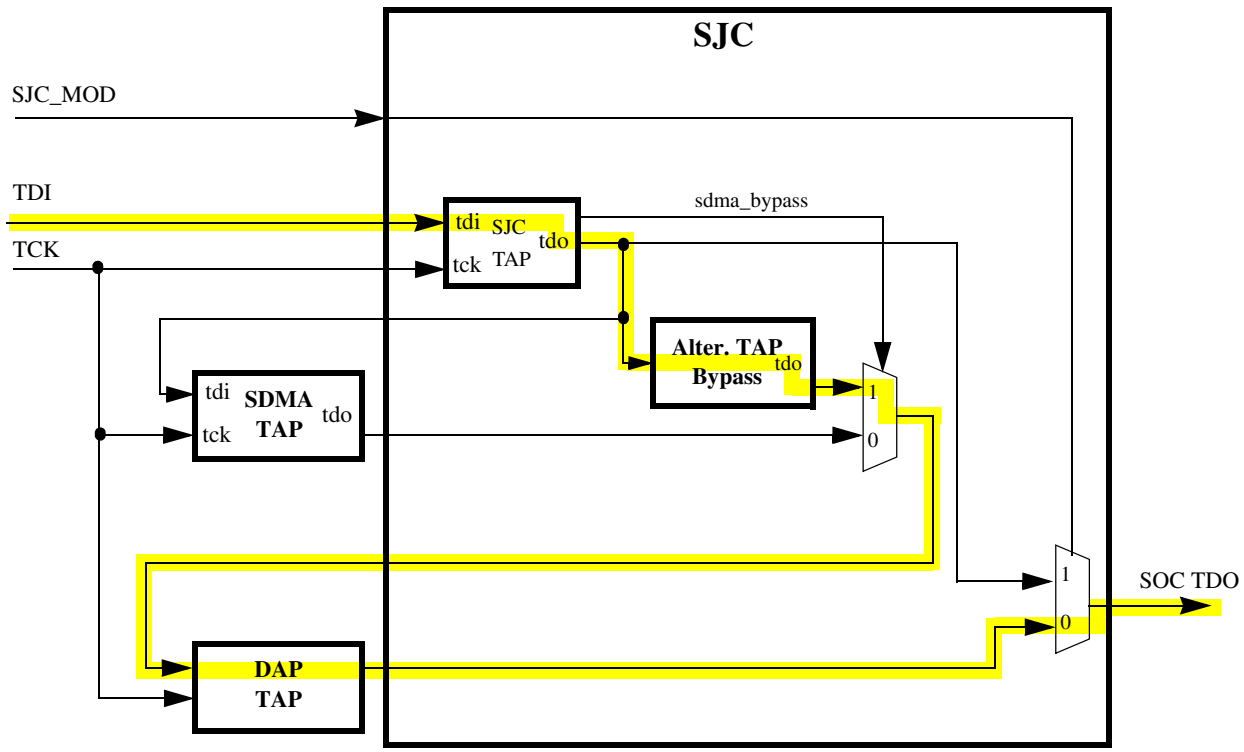
The Connect SDMA bit inside TAP select register controls the SDMA TAP bypass.

- When negated (should be the default state), the SDMA TAP is bypassed with a single D-FF (Flip-flop) during Shift-Dr path
- When asserted SDMA TAP is connected inside the chain
- When taking the SDMA into bypass or out of bypass (by writing to tapsel reg), additional cycle with TMS '0' should be given

The TAP selection block (TSB) provides a simple method of integrating various pieces of IP that have embedded TAPs.

- Provides a way to connect up multiple TAPs within a single SoC
- Identify the SJC TAP as the master TAP which controls the boundary chain (for IEEE 1149.1 standard compliance)
- Follow the state of SJC TAP, and when the Test-Logic-Reset (TLR) state is reached, reset all TAPs

The figure below shows the TAP Selection Block and SOC TAP Chain Scheme.



Note: The default daisy chain connectivity is highlighted in yellow

Figure 67-3. TAP Selection Block and SoC TAP Chain Scheme

NOTE

It is the responsibility of the user to ensure that in any configuration of the TAP controllers chosen, all of the TAPs in the chain comply with the demands of TCK clock frequency as well as the required ratio between TCK clock frequency and that of the core's to which the TAP refers.

67.3 TAP Selection Block (TSB)

As described in Modes of Operation, the SJC can access cores in different modes selected through a TSB.

67.3.1 Select Mode Using Software

Conceptually, the SJC_TSR is a data register which is accessed through Access TSR IR instruction of SJC TAP.

The following figure shows the process of using reserved IR to access the SJC_TSR.

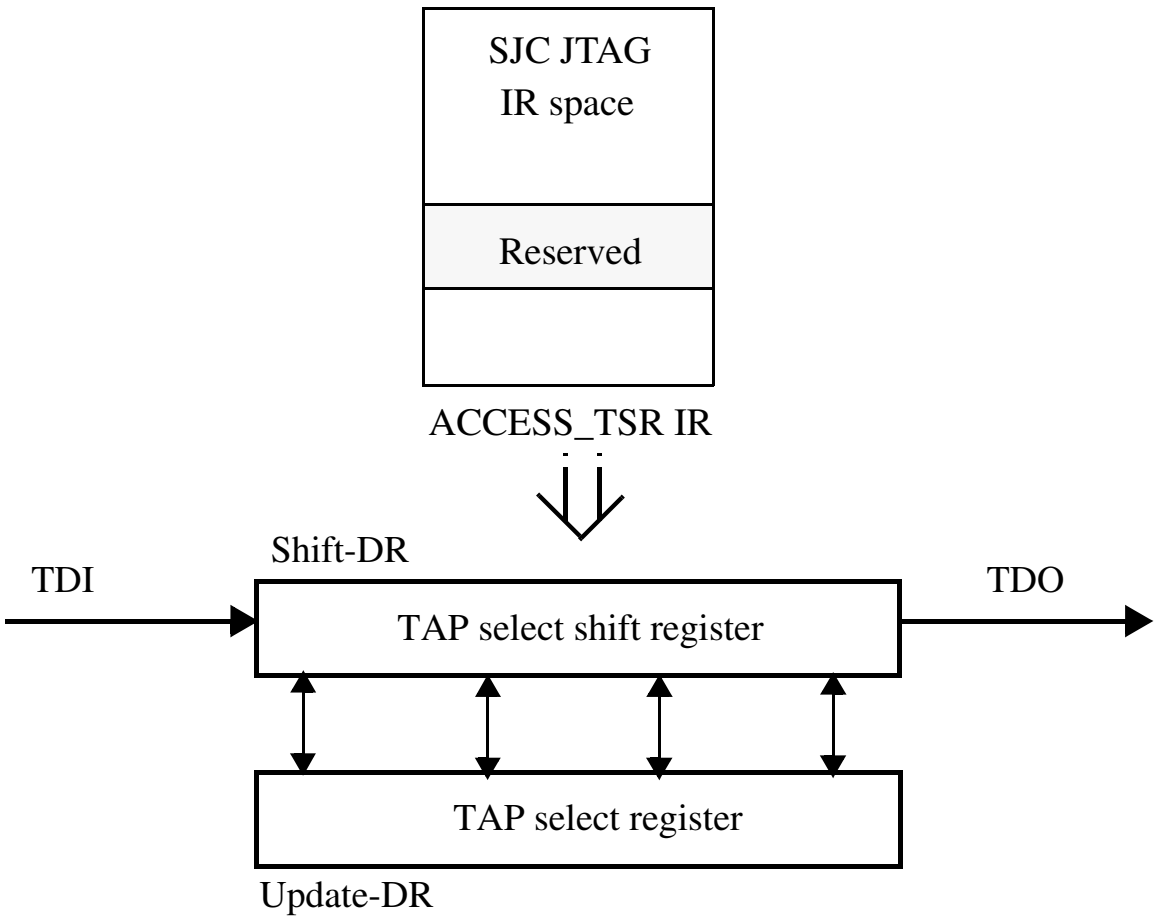


Figure 67-4. Using Reserved IR to Access the TAP Select Register (SJC_TSR)

The SJC_TSR can only be changed during the update-DR state of the TSB JTAG state machine. This is necessary to prevent a TAP that is being selected from losing synchronization with the TSB state machine when the TSB state machine returns to run-test-idle. Therefore, an associated shift register for the SJC_TSR is loaded into the SJC_TSR during the update-DR state (see the figure above). The shift register must also capture the state of the SJC_TSR when in the Capture-DR state for visibility of the contents of the SJC_TSR.

67.4 External Signal Description

67.4.1 External Signal Overview

The SJC provides test and debug control with a minimum number of contacts.

Figure 67-5 shows SJC connections to external contacts and other chip blocks.

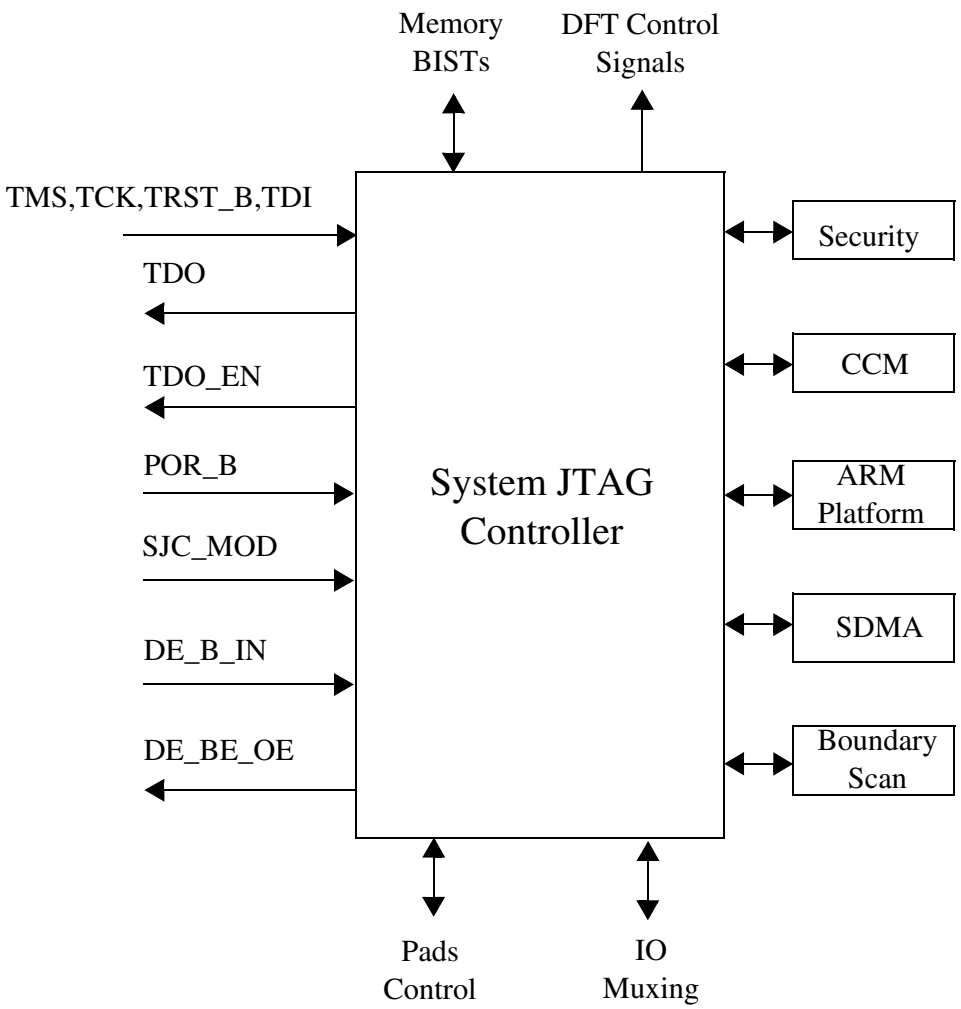


Figure 67-5. SJC Connections

SJC external connections are described along with the signal properties in the table below.

Table 67-2. SJC Signal Properties

Signal	Function	Reset State	Pull up
POR_B	POR reset input. This input can arrive from either pad of IC or from IC's internal logic, for example, Clock Controller.	0	-

Table continues on the next page...

Table 67-2. SJC Signal Properties (continued)

Signal	Function	Reset State	Pull up
TCK	Test Clock (TCK). This is used to synchronize the test logic and includes an internal pull-up resistor	0	-
TDI	Test Data Input (TDI). Serial test instruction and data are received through the test data input (TDI) pin. TDI is sampled on the rising edge of TCK and includes an internal pullup resistor	1	Active
TDO	Test Data Output (TDO). The serial output for test instructions and data. TDO is tri-statable and is actively driven in the shift-IR and shift-DR controller states. TDO changes on the falling edge of TCK	-	-
TDO_EN	Test Data Output Enable (TDO_EN). This signal enables tri-state buffer which output is connected to TDO pad and input connected to IC internal SJC generated TDO signal. Whenever TDO_EN is negated TDO is tri-stated.	-	-
TMS	Test Mode Select (TMS). This is used to sequence the test controller's state machine. TMS is sampled on the rising edge of TCK and includes an internal pullup resistor	1	Active
TRST_B	Test Reset (TRST). This is used to asynchronously initialize the test controller. The TRST pin has an internal pullup resistor	1	Active
SJC_MOD	SJC mode selection. This pin is sampled at TRST reset to determine two possible modes for the TAP connection configuration.	11	Active
DE_IN_B	SoC debug request/acknowledge pin. The DE_IN_B pin is used to propagate an external debug request event to the core(s). This functionality must be enabled first, by set of DE_to_ARM / DE_to_SDMA bits in SJC's DCR register. It is SoC implementation dependent, whether this pin can also be used to reflect the debug acknowledge event back from the cores (in the case where an Open-Drain scheme is used externally).	1	Active

67.4.2 TAP Controller

The TAP controller is responsible for interpreting the sequence of logical values on the TMS signal. It is a synchronous state machine that controls the operation of the JTAG logic. The value shown adjacent to each arc represents the value of the TMS signal sampled on the rising edge of TCK signal. For a description of the TAP controller states, refer to the appropriate IEEE 1149.1 document.

The state machine is shown in the following figure.

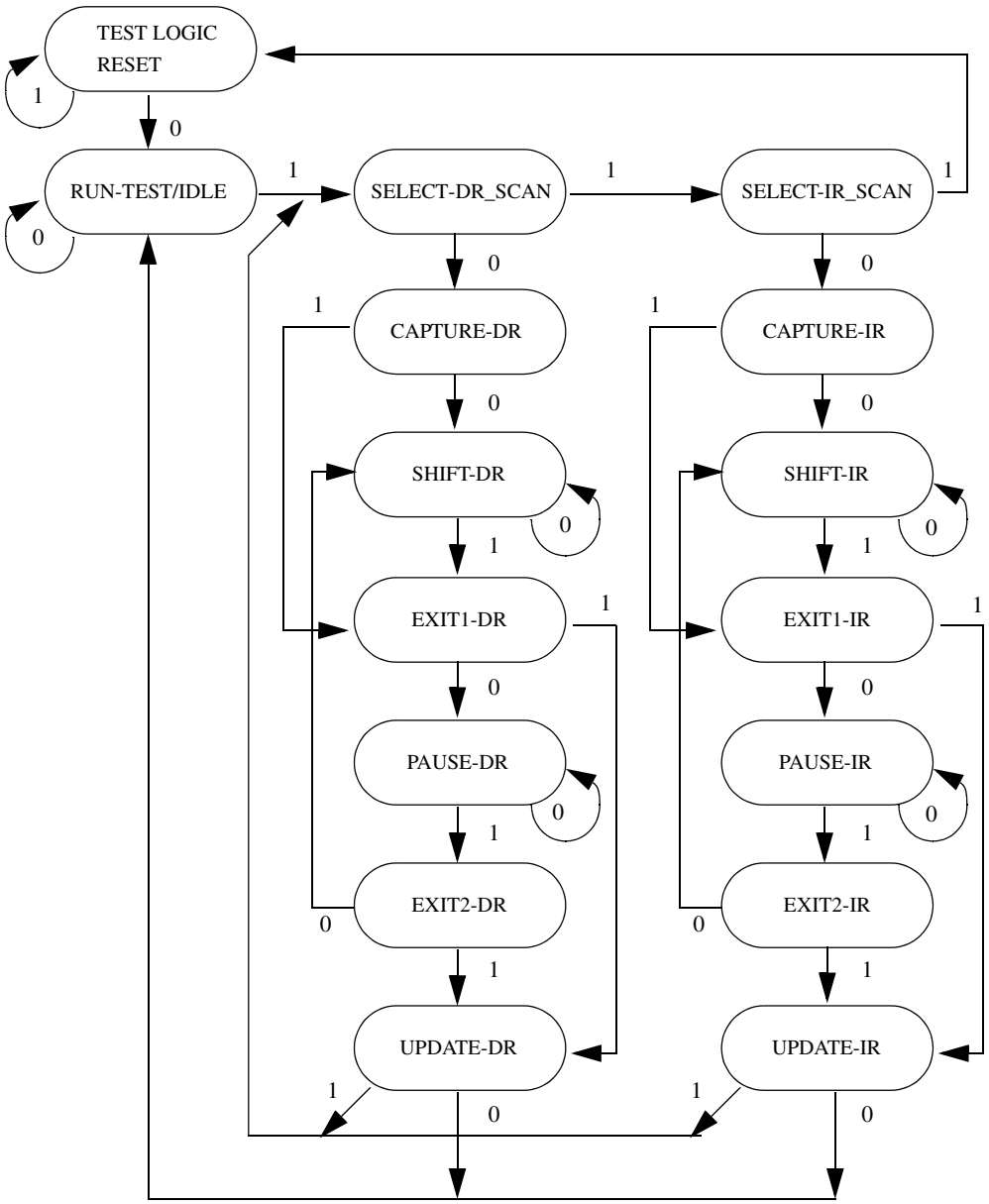


Figure 67-6. TAP Controller State Machine

The change of the JTAG state machine occurs on the rising edge of TCK. TMS and TDI change on the falling edge of TCK. TDO also changes on the falling edge of TCK following entry into the Shift_DR or Shift_IR states (TDO_EN is the enable of the tristate buffer driving the TDO output).

The figure below shows the timings of the SJC signals.

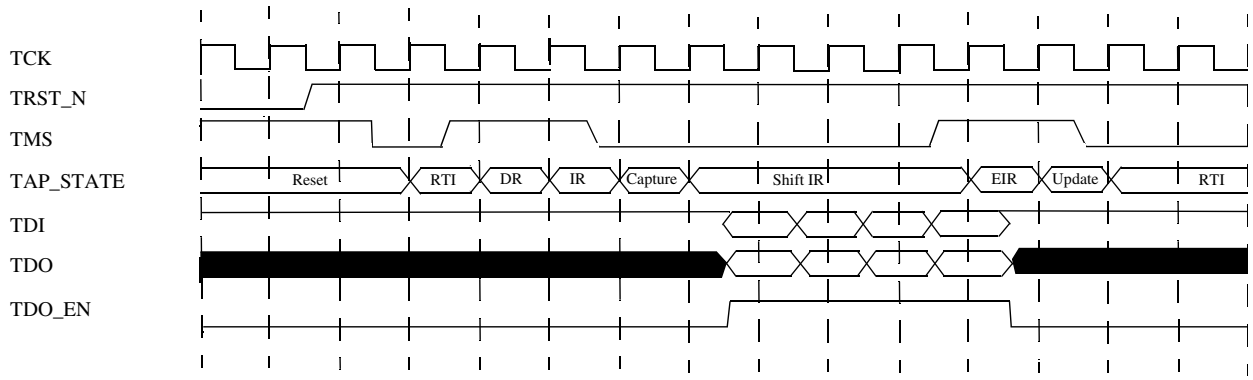


Figure 67-7. SJC Signals Timing Diagram

67.4.3 Accessing ExtraDebug Registers

Accessed through the Select-DR-Scan path, the ExtraDebug shift register consists of 38 bits (maximum) comprising a 32-bit data field (max length, see extradebug register description), a 5 bit address field and read/write bit.

The write actually takes place when the JTAG TAP controller enters the Update-DR state. On a read, the data field is ignored (the user should shift only 5 times to enter Read=1 and the address), the read takes place on the next path through DR at the Capture-DR state, the data is shifted-out during the Shift-DR state.

On the second path for a read access, simultaneous write access is not supported: command converter software shifts in zeros so the TAP decodes a write to the CSR (read-only register) which does not have any effect on the circuit.

The number of shift depends on the width of the accessed register as explained in the following diagrams.

First a write access (one path through Select-DR-Scan):

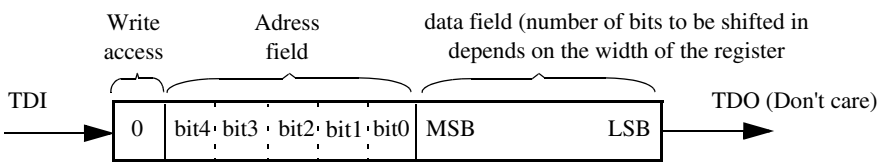
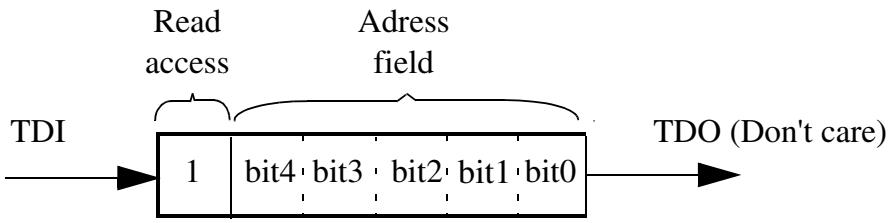


Figure 67-8. TDI/TDO on write access

Then a read access (requires two paths through Jtag DR Scan path):

First path



Second path

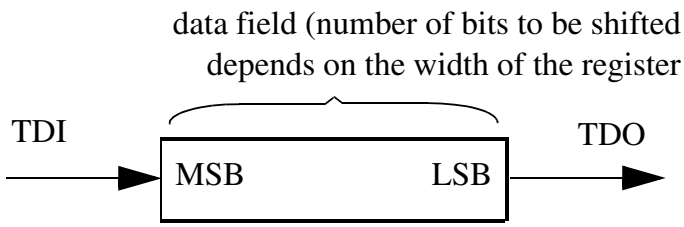


Figure 67-9. TDI/TDO on Read Access

For example, write value 0b1010_1100 to Debug Control Register (address = 0b00110).

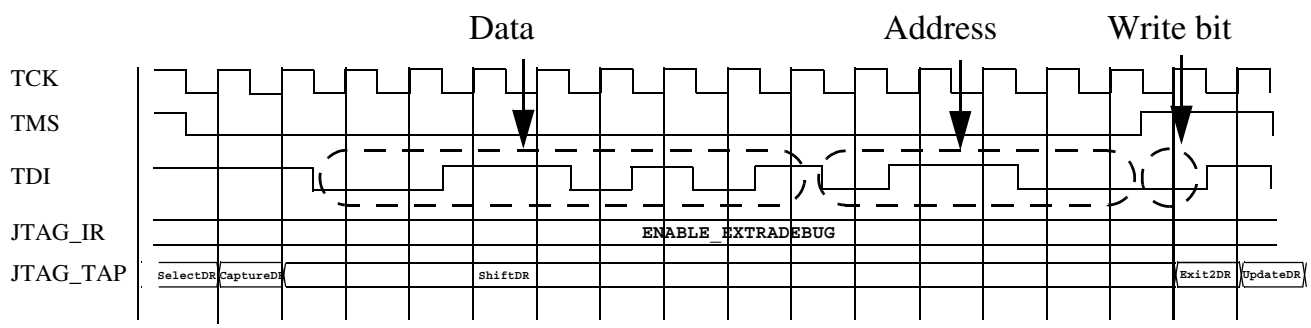


Figure 67-10. Example: Write Access to DCR

The SJC registers have different levels of security:

- Secured- accessible only in mode 2 (supposed correct response entered), mode 3 and mode 4.
- Unsecured- accessible in all modes

The level of security of each register is indicated in its name or description, in "Programmable Registers" section.

A single DE_B pin is dedicated for debug request input/output in bidirectional open drain functionality (including an internal pull-up device).

The main function of the embedded cross trigger is to pass debug events from one core to another. For example this can be programmed to pass debug request from one block as debug request to ARM so that ARM enters debug mode when the debug request is generated. ECT system_debug is connected to SJC and can be programmed for observability on $\overline{DE}$ pad.

Bits 6:4 in DCR register serve as mask bits, controlling the propagation of external debug request to each recipients (ARM Platform, SDMA).

The bits 2:0 define the propagation enable of IR debug request to recipient cores.

Figure 67-11 shows the $\overline{DE}$ Pin Select Logic.

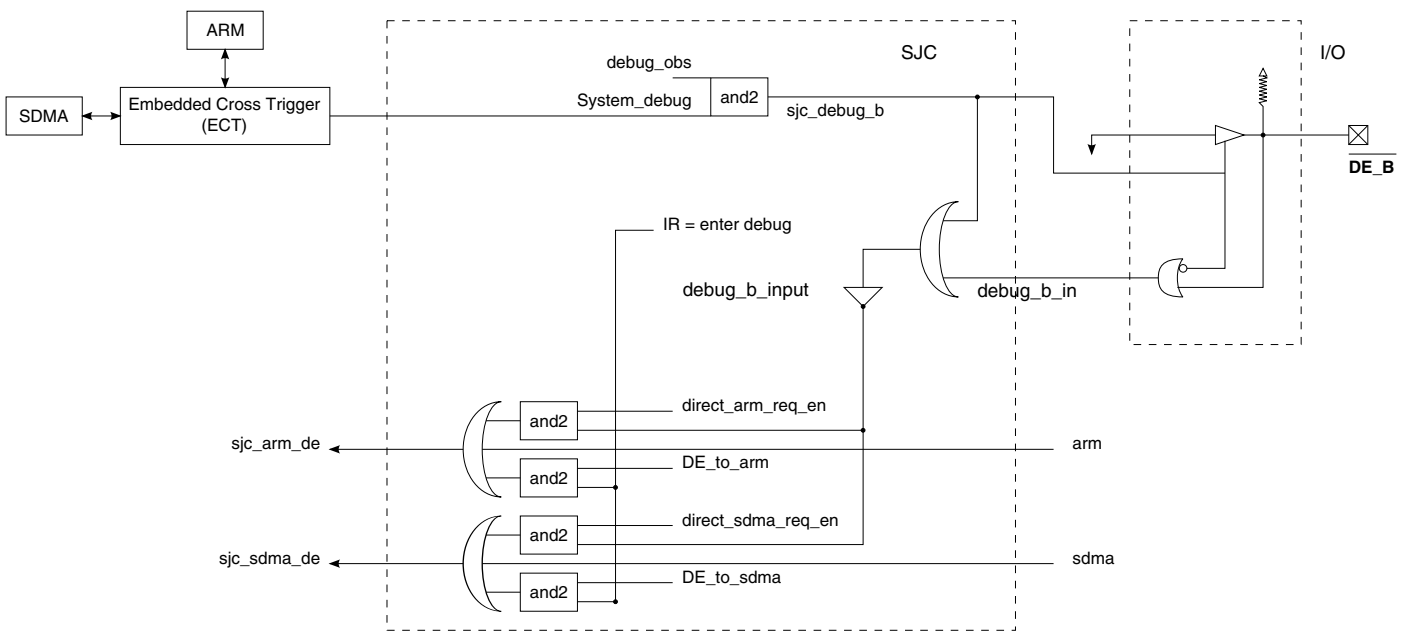


Figure 67-11. $\overline{DE}$ Pin Select Logic

For security reasons, bits for output and input propagation control are at their negated values after reset. A user cannot put the cores in debug mode through $\overline{DE}$ without any Jtag access.

The configuration after reset prevents propagation of debug requests / acknowledges to or from the cores.

67.5 Boundary Scan Register (BSR)

The Boundary Scan Register (BSR) in the JTAG implementation contains bits for all device signal and clock pins and associated control signals. All SoC bidirectional pins have a single register bit in the boundary scan register for pin data, and are controlled by an associated control bit in the boundary scan register.

67.6 SoC JTAG Instruction Register (SJIR)

The SoC JTAG Instruction register is 5 bits wide.

Table 67-3. SoC JTAG Instruction Register (SJIR)

Code					SJC IR
B4	B3	B2	B1	B0	
0	0	0	0	0	IDCODE
0	0	0	0	1	SAMPLE/PRELOAD
0	0	0	1	0	EXTEST
0	0	0	1	1	HI-Z
0	0	1	0	0	ENABLE_ExtraDebug
0	0	1	0	1	ENTER_DEBUG (secured)
0	0	1	1	0	Reserved
0	0	1	1	1	TAP select
0	1	0	0	0	Reserved
0	1	0	0	1	Reserved
0	1	0	1	0	Reserved
0	1	0	1	1	Reserved
0	1	1	0	0	Security Output challenge
0	1	1	0	1	Security Enter response
-					Reserved
1	1	1	1	1	BYPASS

The instruction register is reset to 0b00000 in the test-logic-reset controller state which is equivalent to the IDCODE instruction.

During the capture-IR controller state, the parallel inputs to the instruction register are loaded with the code 01 in the least significant bits as required by the standard; the most significant bits are loaded with the values 00, leading to a capture value of 0b00001

67.6.1 ID_CODE Instruction (IDCODE)

Selects the ID register, and the system logic controls the I/O pins. This instruction is provided as a public instruction to allow the manufacturer, part number and version of a component to be determined through the TAP.

The table below shows the ID register configuration.

Table 67-4. ID Configuration Register (IDCODE)

IDCODE				ID Configuration Register												
	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
	Version Information[3:0]				Part Number (Bits 27-16)											
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x
Note:																
	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	Part Number (Bits 15-12)				Manufacturer Identity											1
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	x	x	x	x	0	0	0	0	0	0	0	1	1	1	0	1
Note:																

Table 67-5. ID Configuration Register Description (IDCODE)

Field	Description
31-28 Version Information	IC/SoC Version information number. Initial value: '0000' This number is subject to changes, for new IC/SoC (System On A Chip) revision releases.
27-12 Part Number	Customer Part Number The 16-bit Part Number value is unique for every Freescale's SoC / IC. See "System Configuration" chapter for exact register value for a specific SoC.
11-1 Manufacturer Identity	Manufacturer Identity Freescale's Manufacturer Identity code. Bits [11:1] - 00000001110
0	Tied to logic 1.

One application of the ID register is to distinguish the manufacturer(s) of components on a board when multiple sourcing is used. As more components emerge which conform to the IEEE 1149.1 standard, it is desirable to allow for a system diagnostic controller unit

to blindly interrogate a board design to determine the type of each component in each location. This information is also available for factory process monitoring and for failure mode analysis of assembled boards.

Once the IDCODE instruction is decoded, it selects the ID register which is a 32 Bit data register. Because the bypass register loads a logic 0 at the start of a scan cycle, whereas the ID register loads a logic 1 into its least significant bit, examination of the first bit of data shifted out of a component during a test data scan sequence immediate following exit from Test-Logic-Reset controller state shows whether such a register is included in the design. When the IDCODE instruction is selected, the operation of the test logic has no effect on the operation of the on-chip system logic as required by the IEEE 1149.1 standard.

67.6.2 SAMPLE/PRELOAD Instruction

Selects the boundary scan register and the system logic controls the I/O pins.

The SAMPLE/PRELOAD instruction provides two separate functions:

- First, it provides a means to obtain a snapshot of system data and control signals. The snapshot occurs on the rising edge of TCK in the capture-DR controller state. The data can be observed by shifting it transparently through the boundary scan register.
- The second function of SAMPLE/PRELOAD is to initialize the boundary scan register output cells prior to selection of EXTEST. This initialization ensures that known data appears on the outputs when entering the EXTEST instruction.

NOTE

Because there is no internal synchronization between the JTAG clock (TCK) and the system clock (CLK), the user must provide some form of external synchronization to achieve meaningful results.

For more details on the function and use of SAMPLE/PRELOAD, refer to the appropriate IEEE 1149.1 document.

67.6.3 EXTEST Instruction

Selects the boundary scan register, and the 1149.1 test logic has control of the I/O pins.

By using the TAP controller, the register is capable of:

- Scanning user-defined values into the output buffers,
- Capturing values presented to input pins

- Controlling the direction of bidirectional pins,
- Controlling the output drive of tri-statable output pins.

For more details on the function and use of EXTEST, refer to the appropriate IEEE 1149.1 document.

The EXTEST instruction also asserts internal reset for the cores (through CCM, refer to Figure 67-14) to force a predictable internal state while performing external boundary scan operations.

67.6.4 HIGHZ Instruction

All output drivers, including the two-state drivers, are turned off (that is, high impedance). The instruction selects the bypass register. In this mode, all internal pullup resistors on all the pins (except for the TMS TDI TCK $\overline{\text{TRST}}$ pins) are disabled. This disabling functionality is not built into SJC, but should be implemented by some logic in the SOC/IO Pads.

For more details on the function and use of HIGHZ, refer to the IEEE 1149.1 document.

The HIGHZ instruction also asserts internal reset for the cores (through CCM, refer to Figure 67-14) to force a predictable internal state while performing external boundary scan operations.

67.6.5 BYPASS Instruction

Selects the single Bit bypass register and the system logic controls the I/O pins. This creates a shift-register path from TDI to the bypass register and, finally, to TDO, circumventing the boundary scan register. This instruction is used to enhance test efficiency when a component other than the SoC Core based device becomes the device under test.

When the bypass register is selected by the current instruction, the shift-register stage is set to a logic zero on the rising edge of TCK in the capture-DR controller state. Therefore, the first bit to be shifted out after selecting the bypass register is always a logic zero.

For more details on the function and use of BYPASS, refer to the appropriate IEEE 1149.1 document.

67.6.6 ENABLE_ExtraDebug Instruction

The TDI and TDO pins are connected directly to the ExtraDebug registers, the SJC TAP controller remaining connected to TDI and TMS.

The ExtraDebug shift register consists of 38 bits (maximum) comprising a 32-bits data field, a 5 bits address field and read/write bit. On a register read, the data field does not need to be filled in. The particular ExtraDebug register connected between TDI and TDO at a given time is selected by the ExtraDebug controller depending on the ExtraDebug Address being currently decoded. All communication with the ExtraDebug controller is done through the Select-DR-Scan path of the JTAG TAP Controller.

67.6.7 ENTER_DEBUG instruction

The ENTER_DEBUG instruction is used to generate a debug request event to SDMA and the ARMCore Platform simultaneously (practically, inherited minimal skew is expected, due to difference in event signal propagation in the different modules).

The TDI and TDO are connected to the Instruction Register (IR). After the acknowledgment of the Debug Mode is received (can be checked by reading the Core Status Register part of the ExtraDebug logic), the user can perform system debug functions on the cores.

NOTE

The ENTER_DEBUG event issue to the cores, can be masked, by bits in DCR register.

It is user's responsibility to shift-in another IR value (like IDCODE) before trying to bring the cores out of debug mode, as the debug request signals to the cores remains asserted as long as ENTER_DEBUG IR is in place.

The user need to check that cores are in debug mode (watching debug acknowledge signal) before leaving ENTER_DEBUG instruction, otherwise debug request might not take affect.

67.6.8 TAP Select Instruction

By means of TAP select instruction a user can access TAP select register and by controlling its only bit SDMA Bypass, control whether SDMA TAP is bypassed or not.

Table 67-6. TAP Select Register (TSR)

	TAP Select Register
	BIT 0
	Connect SDMA
TYPE	rw
RESET	0
Note:	

Table 67-7. TAP Select Register Description

Field	Description
0 SDMA Bypass	Connect SDMA Control whether SDMA TAP is bypassed or not: • 0 - SDMA TAP is bypassed by the alternate TAP inside SJC (emulating 4-bit IR and 1-bit bypass path). • 1 - SDMA TAP is connected to the TDI-TDO chain. **NOTE:** Additional cycle with TMS '0' should be inserted, after writing to this register, to allow the SDMA tap be sync before SDMA get into / out of bypass.

67.7 Security

JTAG manipulation is one of the known hackers' ways of executing unauthorized program code, getting control over the OS and run code in privileged modes.

The SJC provides a debug access to several H/W blocks including the ARM processor and the system bus. This allows for program control and manipulation as well as visibility into system peripherals and memory. The ETM and NEXUS interfaces allow bus transactions to be traced. Together these tools provide the hacker all the access needed to completely comprise the system. Means must be provided to block any malicious JTAG access.

The SJC provides a way of regulating the JTAG access.

The following are the different JTAG security modes:

- Mode #1: No Debug-Maximum Security. All security sensitive JTAG features are permanently blocked.

- Mode #2: Secure JTAG-High security. JTAG use is regulated by secret key based authentication mechanism.
- Mode #3: JTAG Enabled-Low security. JTAG always enabled.

The JTAG security modes are configured using eFUSEs which can burned after packaging by applying electrical signals. The fuse burning is irreversible process, once a fuse is burned (e-fuse or laser fuse) it is impossible to change the fuse back to the un-burned state.

67.7.1 JTAG Security Modes

JTAG can be in one of JTAG security modes which is selected by setting the SJC eFUSE configuration. The physical location of the fuses is not in the SJC.

67.7.1.1 Mode 1: No Debug - Maximum Security

No Debug JTAG security mode provides the highest security level.

In this mode, all JTAG features are disabled except for:

- ScanBoundary Scan
- MBIST, all modes except for debug modes which enable controlled memory contents output
- PLL BIST
- BIST monitor mode, allowing routing to external pins BIST pass/fail/invoke information
- PLL bypass- Bypass ARM or/and USB PLL.
- Visibility of the following status bits: power mode - normal, standby, stop, shutdown, and so on

These features do not reduce the security level of the product, and they allows to perform important tests and board connectivity checks.

67.7.1.2 Mode 2: Secure JTAG - High Security

The Secure JTAG mode limits the JTAG access by using challenge/response based authentication mechanism. Any access to JTAG port is being checked. Only authorized debug devices (that is, devices having the right response) can access the JTAG, unauthorized JTAG access attempts are denied.

The intent of this mode is to allow return field testing. When a secured JTAG device is being returned for debugging, this mode allows authorized re-activation of the JTAG.

67.7.1.2.1 Challenge/Response Mechanism in System JTAG Mode

When SJC is in Sysytem JTAG mode the authentication process is as follows:

1. Shift Output Challenge instruction to IR.
2. Passing through Capture-DR state of the SJC and by performing Shift-DR operations Challenge code can be accessed from TDO.
3. Shift Enter Response instruction to IR. By performing Shift-DR, operations enter Response code value through TDI. As Update-DR state is entered, Response code is compared with the correct one.

In Fixed challenge-response pair mode, each part has its individual challenge - response pair which is determined at manufacturing time, and does not change later on. The SJC compares the user's response to the expected response.

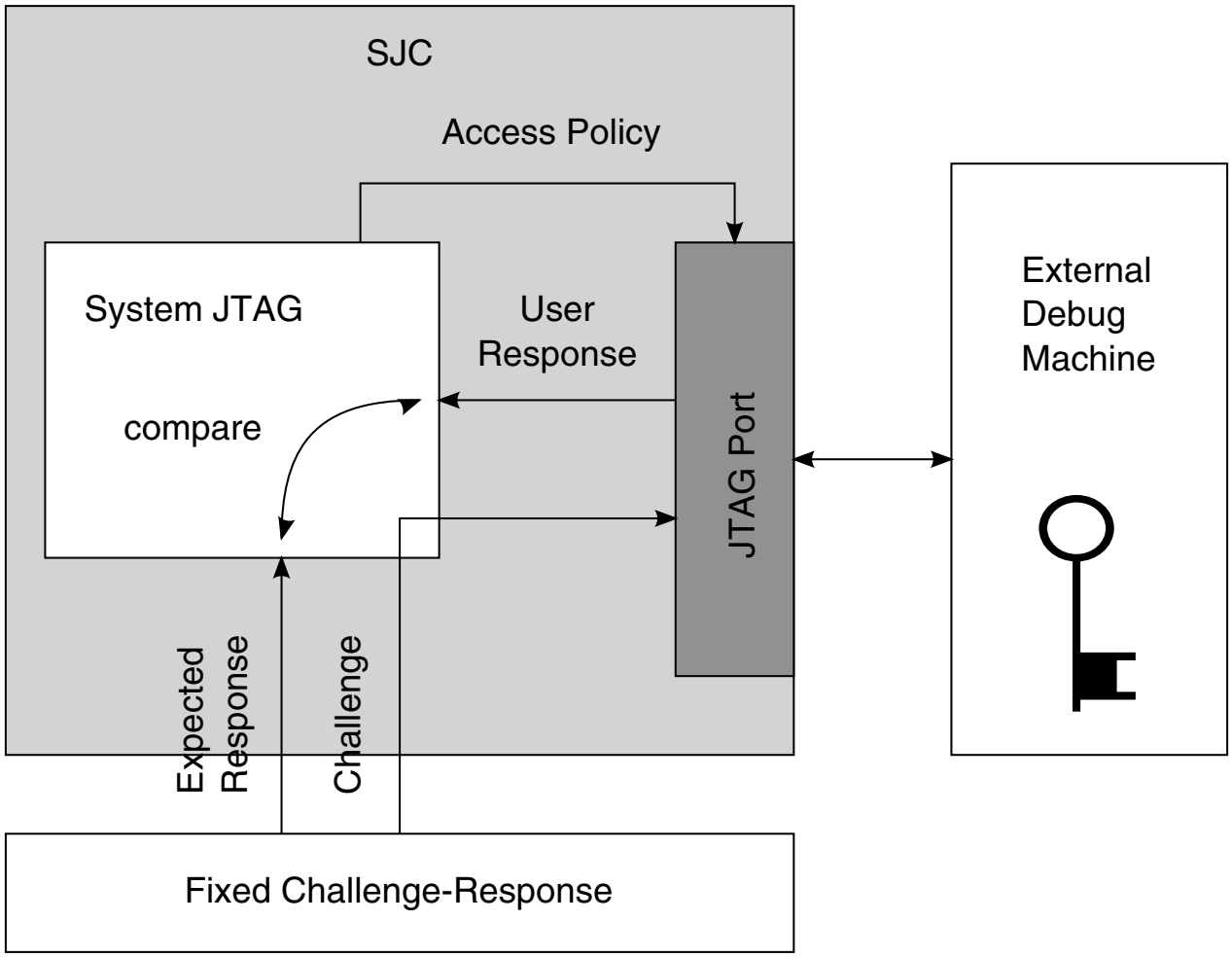


Figure 67-12. Mode #2 - Secure JTAG with Fixed Challenge-response Pair

67.7.1.3 Mode 3: JTAG Enabled - Low Security

In the JTAG Enabled JTAG security mode, all JTAG features are enabled.

67.7.2 Software Enabled JTAG

To increase the flexibility of the SJC, an option to enable the JTAG via software is added and is available only in Secure JTAG mode. By writing '1' to HAB_JDE (HAB JTAG DEBUG ENABLE) bit in the e-fuse controller module (IIM), the JTAG is opened, regardless of its security mode. It is the responsibility of software to assert or negate this bit.

Additionally, a corresponding lock bit is available (in the e-fuse control module) to ensure that only trusted software is able to set the JDE bit. When the LOCK bit is set, no future change of JDE is possible, until the next POR (power-on-reset) cycle.

The platform initialization software should set the LOCK bit for JDE bit before transferring control to the application code.

The S/W JTAG enable allows JTAG enabling without activating the challenge-Response mechanism (which requires JTAG access tool enhancement or special H/W). The JTAG S/W enable does not allow debug in case of boot or memory fault as it requires reset before entering debug.

This feature can be permanently blocked by burning the dedicated e-fuse.

NOTE

The S/W enabled JTAG feature reduces the overall security level of the system as it relies on S/W protections. If this feature is not required, it is strongly recommended to burn the JTAG_HEO e-fuse which disables this feature.

67.7.3 Kill Trace

The kill trace signal disables any output of the ETM block. The ETM can be accessed either via JTAG port and/or by direct software code. Blocking the JTAG port also yields assertion of the kill trace signal. This resulted in blocking of trace port. The intention of this action is to block any attempt to break into the system via software manipulation of the debug modules. The kill trace, when active, prevents trace output even in case where it can be activated via chip pin.

The kill trace feature needs to be activated by burning a dedicated e-fuse. If the fuse is left intact, kill trace is never activated as seen in Figure 67-13.

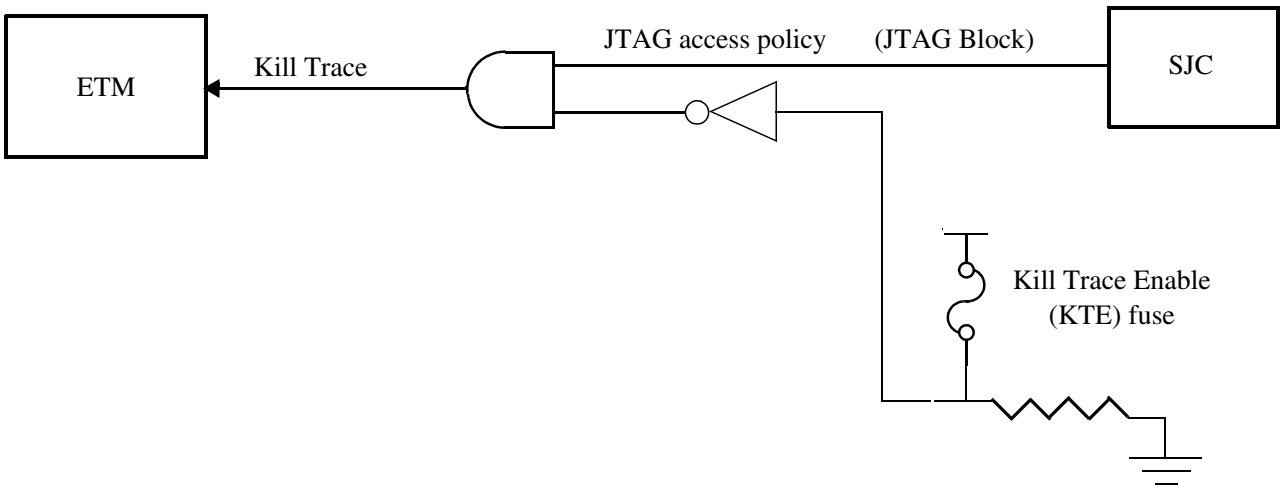


Figure 67-13. Kill Trace eFUSE

The kill trace is asserted when "kill trace enable" fuse is burned and "ipt_secur_block" signal in SJC is asserted, which happens when at least one of the following is true:

- Mode #2 (Secure JTAG) and no code has been entered
- Mode #2 (Secure JTAG) with burned Bypass and Re-enable fuses
- Mode #2 (Secure JTAG) with incorrect response entered
- Mode #1 (No debug)
- TRST_B signal is active
- POR has not ever been asserted

67.7.4 SJC Disable Fuse

In addition to the different JTAG security modes that are implemented internally in the System JTAG Controller (SJC), there is an option to disable the SJC functionality by eFUSE configuration. This creates additional JTAG mode that is, JTAG Disabled with highest level of JTAG protection. In this mode all JTAG features are disabled.

Specifically, the following debug features are disabled in addition to the features that were already disabled in No Debug JTAG mode:

- Memory BIST
- Boundary scan register (SJC_BSR)
- Non-Secure JTAG control registers (PLL configuration, Deterministic Reset, PLL bypass)
- Non-Secure JTAG status registers (Core status)
- Chip Identification Code (IDCODE)

67.8 Functional Description

67.8.1 Static Core Debug

The SJC JTAG TAP controller is fully compatible with the IEEE 1149.1a-2001 Standard Test Access Port and Boundary Scan Architecture specifications.

The ARM platform has an integrated JTAG interface and a TAP controller to manage its own ICE. Also it can access an embedded trace ETM interface, see ARM core and ETM Technical reference guide for more information.

The SDMA has a TAP controller to manage its own OnCE, see SDMA OnCE specifications for more details.

The OnCE and ICE provide a mean of interacting with the cores and their peripherals non-intrusively so that a user may examine registers, memories to facilitate hardware and software development.

67.8.2 Reset Mechanism

The following figure shows the SJC reset logic

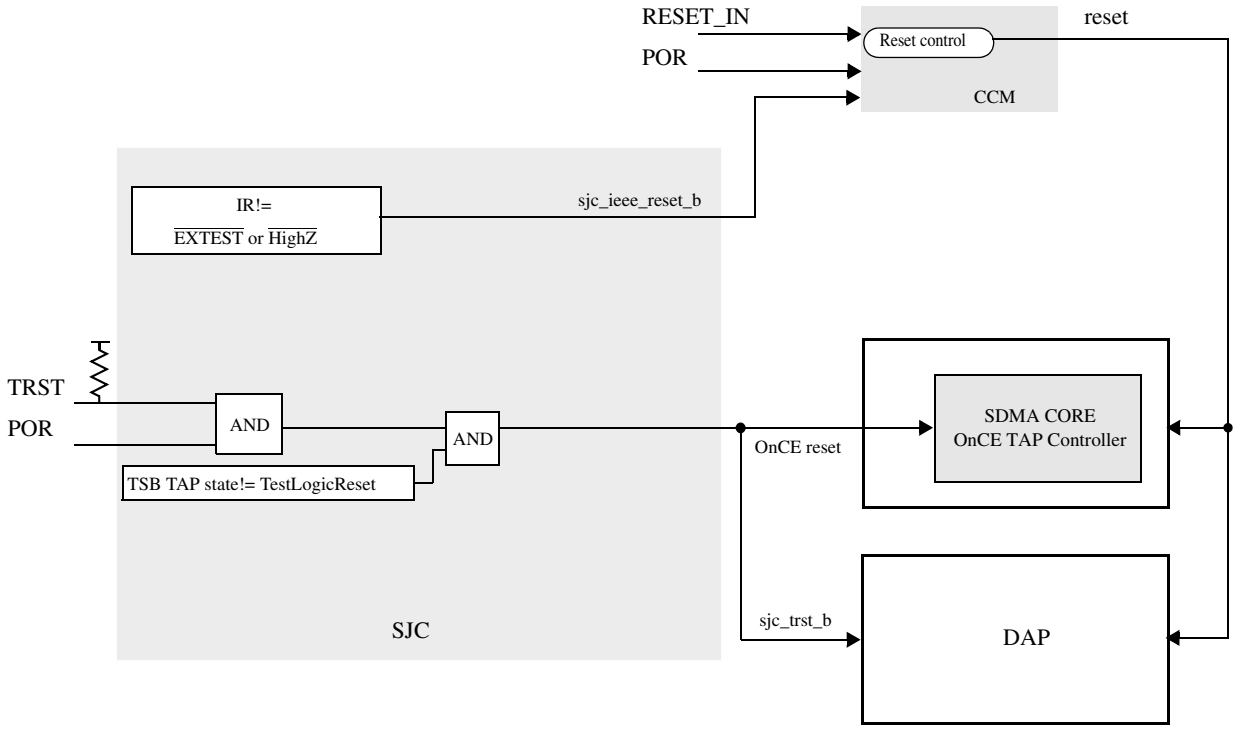


Figure 67-14. SJC Reset Logic

NOTE

- Asserting TRST in any scan mode resets the TCR loosing the testmode configuration and selects default TAP.
- SJC generates an IEEE reset signal to the CCM when in one of the IEEE modes HIGHZ or EXTEST. This signal generates a system reset to the cores until exit from one of these modes.
- The TSB generates Once/ICE reset (either TRST if implemented or other) when its TAP state reaches Test-Logic-Reset (meaning that TAP accessed is also reaching Test-Logic-Reset).
- When the SJC is in scan mode reset lines are controlled from RESET_IN_B for manufacturing test purpose.

67.9 Initialization/Application Information

The control afforded by the output enable signals using the boundary scan register and the EXTEST instruction requires a compatible circuit-board test environment to avoid device-destructive configurations. The user must avoid situations in which the SJC output drivers are enabled into actively driven networks.

There are two constraints related to the JTAG interface:

- Ensure that the JTAG test logic is kept transparent to the system logic by forcing TAP into the Test-Logic-Reset controller state. During power-up, SJC's internal $\overline{\text{TRST}}$ is asserted as IC's $\overline{\text{POR}}$ is asserted which forces the TAP controller into this state. After that, if TMS either remains unconnected or is connected to VCC, then the TAP controller cannot leave the Test-Logic-Reset state, regardless of the state of TCK.
- $\overline{\text{DE\_B}}$ is an IO pin with pullup and care must be taken of the direction when driving this signal.

67.10 Programmable Registers

In addition to the standard accessible JTAG registers (per IEEE1149.1 standard) listed in SoC JTAG Instruction Register (SJIR), the chip contains the following registers accessed using the ExtraDebug mechanism, controlled via "ENABLE_ExtraDebug" IR instruction.

NOTE

SJC registers are only accessible by JTAG interface. They are not memory mapped to processor address space, so the absolute addresses provided by default in the SJC memory map are not valid.

This section assumes the JTAG controller is accessed in standalone mode or daisy chained (defined by TAP Selection Block) using the appropriate TSB configuration.

See "System Debug" chapter for more details about the general purpose register descriptions that are unique to this chip.

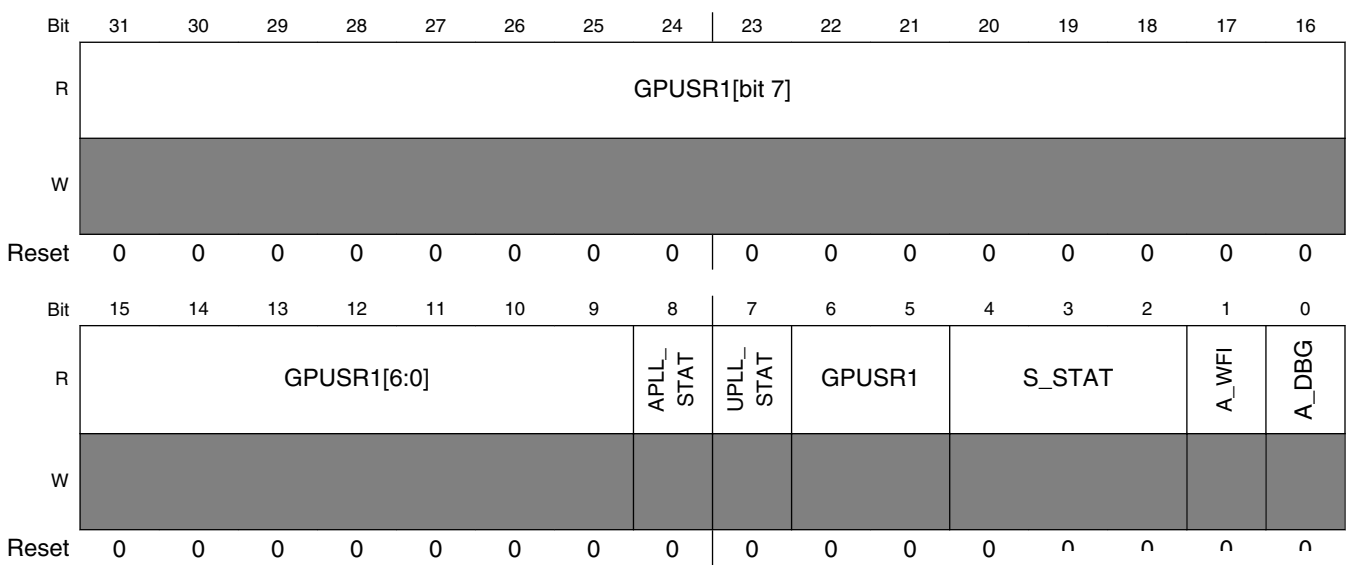
SJC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0000_0000	General Purpose Unsecured Status Register 1 (SJC_GPUSR1)	32	R	0000_0000h	67.10.1/ 4295
0000_0001	General Purpose Unsecured Status Register 2 (SJC_GPUSR2)	32	R	0000_0000h	67.10.2/ 4296
0000_0002	General Purpose Unsecured Status Register 3 (SJC_GPUSR3)	32	R	0000_0000h	67.10.3/ 4297
0000_0003	General Purpose Secured Status Register (SJC_GPSSR)	32	R	0000_0000h	67.10.4/ 4297
0000_0004	Debug Control Register (SJC_DCR)	32	R/W	0000_0000h	67.10.5/ 4298
0000_0005	Security Status Register (SJC_SSR)	32	R	0000_0000h	67.10.6/ 4299
0000_0007	General Purpose Clocks Control Register (SJC_GPCCR)	32	R/W	0000_0000h	67.10.7/ 4301
0000_0008	General Purpose Unsecured Control Register n (SJC_GPUCR)	32	R/W	0000_0000h	67.10.8/ 4302
0000_000B	General Purpose Secured Control Register (SJC_GPSCR)	32	R/W	0000_0000h	67.10.9/ 4302

67.10.1 General Purpose Unsecured Status Register 1 (SJC_GPUSR1)

The General Purpose Unsecured Status Register 1 is a read only registers used to check the status of the different Cores and of the PLL. The rest of its bits are for general purpose use.

Address: SJC_GPUSR1 is 0h base + 0h offset = 0000_0000h



SJC_GPUSR1 field descriptions

Field	Description
31–9 GPUSR1	General Purpose Unsecured Status Register Register is used for testing and debug.
8 APLL_STAT	ARM platform PLL status bit This bits indicates the status of the ARM platform PLL. When this bit is HIGH, the ARM platform PLL is locked.
7 UPLL_STAT	USB PLL status bit This bits indicates the status of the USB PLL. When this bit is HIGH, the USB PLL is locked.
6–5 GPUSR1	General Purpose Unsecured Status Register Register is used for testing and debug.
4–2 S_STAT	3 LSBits of SDMA core statusH.
1 A_WFI	ARM core wait-for interrupt bit Bit 1 is the ARM core standbywfi (stand by wait-for interrupt). When this bit is HIGH, ARM core is in wait for interrupt mode.
0 A_DBG	ARM core debug status bit Bit 0 is the ARM core DBGACK (debug acknowledge) DBGACK can be overwritten in the ARM core DCR to force a particular DBGACK value. Consequently interpretation of the DBGACK value is highly dependent on the debug sequence. When this bit is HIGH, ARM core is in debug.

67.10.2 General Purpose Unsecured Status Register 2 (SJC_GPUSR2)

Address: SJC_GPUSR2 is 0h base + 1h offset = 0000_0001h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPUSR2																												S_STAT			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_GPUSR2 field descriptions

Field	Description
31–4 GPUSR2	General Purpose Unsecured Status Register Register is used for testing and debug.
3–0 S_STAT	S_STAT

67.10.3 General Purpose Unsecured Status Register 3 (SJC_GPUSR3)

Address: SJC_GPUSR3 is 0h base + 2h offset = 0000_0002h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPUSR3																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_GPUSR3 field descriptions

Field	Description
31–0 GPUSR3	General Purpose Unsecured Status Register Register is used for testing and debug.

67.10.4 General Purpose Secured Status Register (SJC_GPSSR)

The General Purpose Secured Status Register is a read-only register used to check the status of the different critical information in the SoC. This register cannot be accessed in secure modes.

Address: SJC_GPSSR is 0h base + 3h offset = 0000_0003h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPSSR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_GPSSR field descriptions

Field	Description
31–0 GPSSR	General Purpose Secured Status Register Register is used for testing and debug.

67.10.5 Debug Control Register (SJC_DCR)

This register is used to control propagation of debug request from DE_B pad to the cores and debug signals from internal logic to the DE_B pad.

Address: SJC_DCR is 0h base + 4h offset = 0000_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DCR[bit 9]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DCR[8:0]									DIRECT_ ARM_REQ_ EN	DIRECT_ SDMA_ REQ_EN	-	DEBUG_ OBS	-	DE_TO_ SDMA	DE_TO_ ARM
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_DCR field descriptions

Field	Description
31–7 DCR	General Purpose Unsecured Status Register Register is used for testing and debug.
6 DIRECT_ARM_ REQ_EN	Pass Debug Enable event from $\overline{\text{DE\_B}}$ pin to ARM platform debug request signal(s). This bit controls the propagation of debug request DE_B to the Arm platform. 0 Disable propagation of system debug to (DE pin) to Arm platform. 1 Enable propagation of system debug to (DE pin) to Arm platform.
5 DIRECT_ SDMA_REQ_EN	Debug enable of the sdma debug request This bit controls the propagation of debug request DE_B to the sdma. 0 Disable propagation of system debug to (DE pin) to sdma. 1 Enable propagation of system debug to (DE pin) to sdma.
4 -	Reserved
3 DEBUG_OBS	Debug observability This bit controls the propagation of the "system debug" input to SJC (coming from ECT logic), to the DE_B pad. (This logic can be used to pass debug acknowledge event from ECT out to the PAD, for example). 0 Disable propagation of system debug to DE pin 1 Enable propagation of system debug to DE pin
2 -	Reserved

Table continues on the next page...

SJC_DCR field descriptions (continued)

Field	Description
1 DE_TO_SDMA	SDMA debug request input propagation This bit controls the propagation of debug request to SDMA, when the JTAG state machine is put in "ENTER_DEBUG" IR instruction.. 0 Disable propagation of debug request to SDMA 1 Enable propagation of debug request to SDMA
0 DE_TO_ARM	ARM platform debug request input propagation This bit controls the propagation of debug request to ARM platform ("dbgreq"), when the JTAG state machine is put in "ENTER_DEBUG" IR instruction. 0 Disable propagation of debug request to ARM platform 1 Enable propagation of debug request to ARM platform

67.10.6 Security Status Register (SJC_SSR)

Address: SJC_SSR is 0h base + 5h offset = 0000_0005h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SSR[bit 3]															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SSR[2:0]			RSSTAT		SJM		FT	BSF	RSF	EBG	EBF	SWE	SWF	KTA	KTF
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_SSR field descriptions

Field	Description
31–13 SSR	Security Status Register Register is used for testing and debug.
12–11 RSSTAT	Response status Response status bits 00 Response wasn't entered 01 Response was entered but not verified 10 Response was entered and is incorrect 11 Response is correct
10–9 SJM	SJC mode Secure JTAG mode, as set by external fuses. These bits do not include the setting of the BSF fuse. 00 No debug (#1)

Table continues on the next page...

SJC_SSR field descriptions (continued)

Field	Description
	01 Secure JTAG (#2) 10 JTAG enabled (#3) 11 Reserved
8 FT	Fuse type Fuse type bit - e-fuse or laser fuse 0 E-fuse technology 1 Laser fuse technology
7 BSF	Bypass secure JTAG fuse Status of the bypass secure JTAG fuse. 0 (intact) - no bypass 1
6 RSF	Re-enable secure JTAG fuse Status of the re-enable secure JTAG fuse 0 (intact) - no re-enable 1 (burned) - secure JTAG is re-enabled
5 EBG	External boot granted External boot enabled, requested and granted 1 granted 0 not granted
4 EBF	External Boot fuse Status of the external boot disable fuse 0 (intact) - external boot is allowed 1 (burned) - external boot is disabled
3 SWE	SW enable SW JTAG enable status 1 enabled 0 disabled
2 SWF	Software JTAG enable fuse Status of the no SW disable JTAG fuse 0 (intact) - SW enable possible 1 (intact) - no SW enable possible
1 KTA	Kill Trace is active 1 active 0 not active
0 KTF	Kill Trace Enable fuse value

Table continues on the next page...

SJC_SSR field descriptions (continued)

Field	Description
	0 (intact) - kill trace is never active 1 (burned) - kill trace functionality enabled

67.10.7 General Purpose Clocks Control Register (SJC_GPCCR)

This register is used to configure clock related modes in SOC, see System Configuration chapter for more information. Those bits are directly connected to JTAG outputs. Bit 0 of GPCCR controls SDMA clocks invocation. When out of reset, the SDMA is in sleep mode with no SDMA clock running. Unlike events, debug requests does not wake SDMA if it is in sleep mode. The debug request is recognized by the SDMA only when it exits sleep mode upon reception of an event. To be able to enter debug mode even if no event is triggered, the SDMA clock on bit needs to be set prior to sending the debug request (clear at reset).

Address: SJC_GPCCR is 0h base + 7h offset = 0000_0007h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPCCR[bit 13]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPCCR[12:0]													SWRES	ACLKOFFDIS	SCLKR
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_GPCCR field descriptions

Field	Description
31–3 GPCCR	General Purpose Clocks Control Register-Register is used for testing and debug.
2 SWRES	SW reset
1 ACLKOFFDIS	Disable/prevent ARM platform clock/power shutdown
0 SCLKR	SDMA Clock ON Register - This bit will force the clock on of the SDMA

67.10.8 General Purpose Unsecured Control Register n (SJC_GPUCR)

Address: SJC_GPUCR is 0h base + 8h offset = 0000_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPUCR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_GPUCR field descriptions

Field	Description
31–0 GPUCR	General Purpose Unsecured Control Register Register is used for testing and debug.

67.10.9 General Purpose Secured Control Register (SJC_GPSCR)

This register is used to configure JTAG for special test or debug modes. This register is secured (accessible in secure jtag mode #3, #4 and #2 with response entered). Those bits are directly connected to SJC outputs.

Address: SJC_GPSCR is 0h base + Bh offset = 0000_000Bh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPSCR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_GPSCR field descriptions

Field	Description
31–0 GPSCR	General Purpose Control Register Register is used for testing and debug.

Chapter 68
Shared Peripheral Bus Arbiter (SPBA)

68.1 Introduction

The Shared Peripheral Bus Arbiter (SPBA) is a three-to-one IP Bus line interface arbiter, with a resource locking mechanism. The masters can access up to thirty one shared peripherals through the SPBA.

The figure below shows the SPBA block diagram

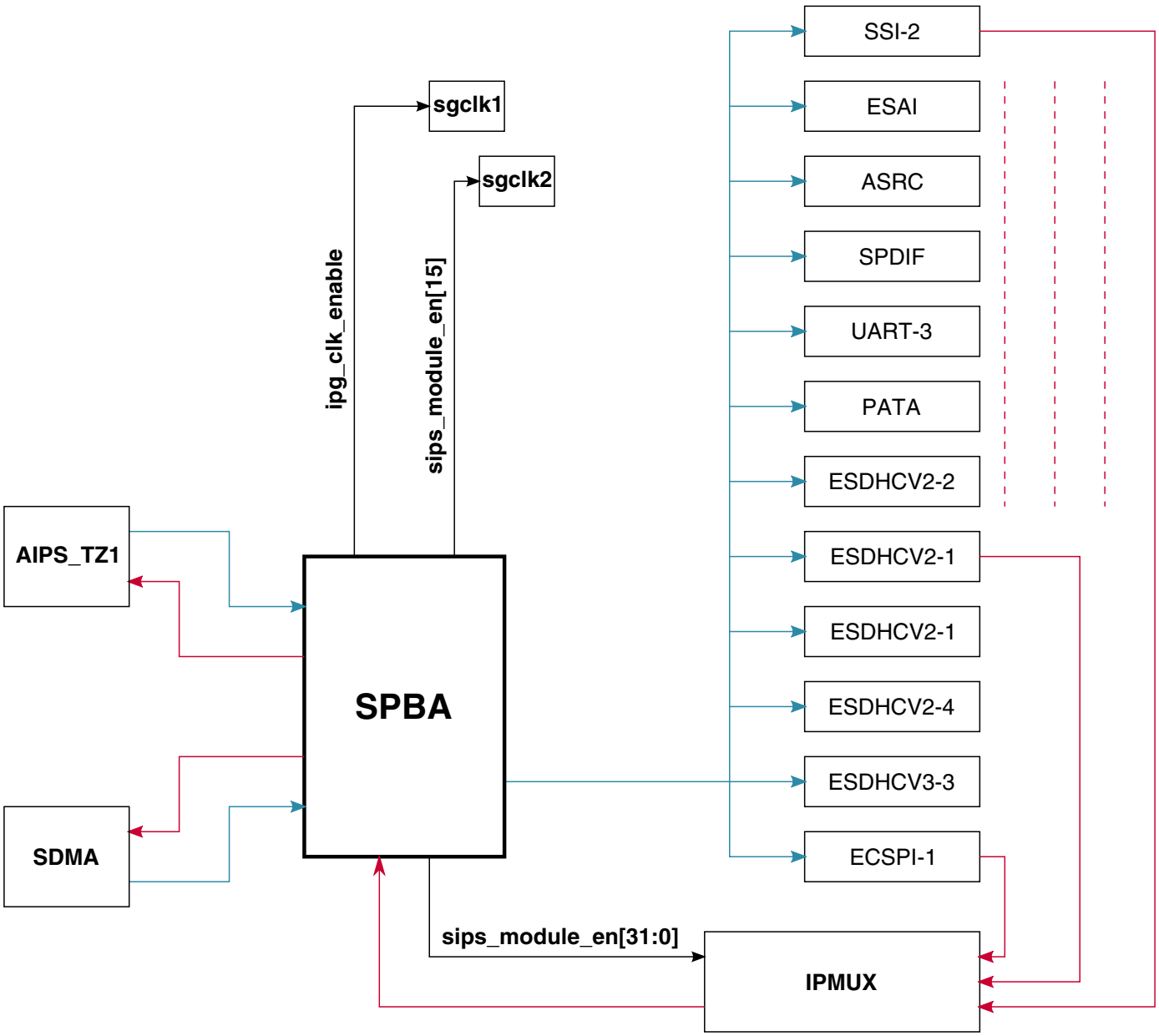


Figure 68-1. i.MX53 SPBA connectivity

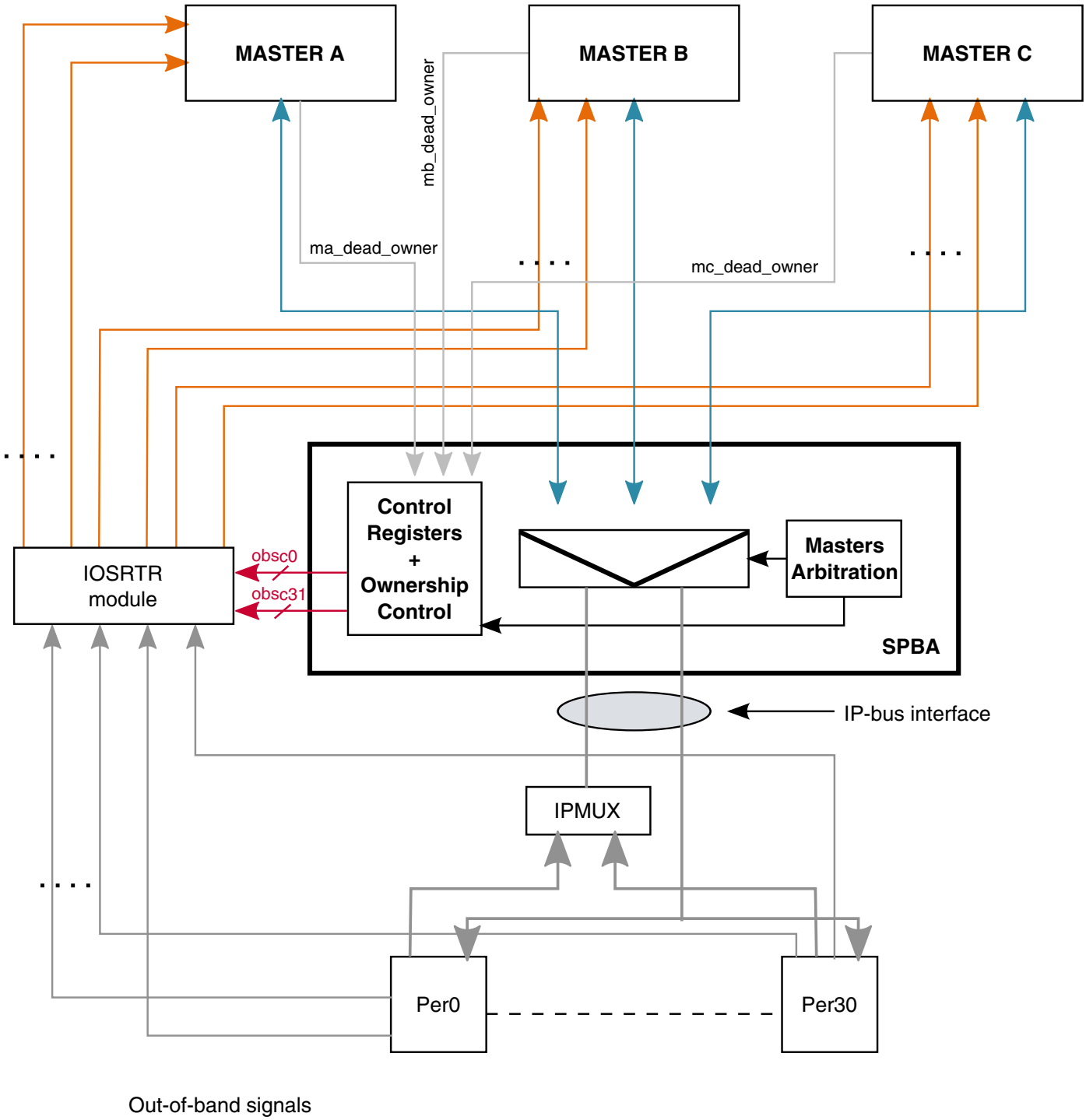


Figure 68-2. SPBA Block Diagram

68.2 General Overview

The Shared Peripheral Bus Arbiter (SPBA) is a three-to-one IP Bus interfaces arbiter. Three masters arbitrate for shared peripherals access through the SPBA.

Indeed three main parts are involved in SPBA.

- The IP Bus Line switches a master to one peripheral.
- The Masters arbiter arbitrates between the three masters to solve concurrent access or restricted access to peripherals.
- The Control Registers and Ownership Control including a set of registers, which are reachable through software, permits the access scheme definition to each peripheral (Resource Ownership and Access Control). It generates signals usable for the external steering logic of interrupts and DMA signals.

68.3 Features

The SPBA includes the following features:

- Three IP Bus masters arbitration, master A, master B and master C
- Support for DMA masters.
- 32 bits data
- Supports up to 31 shared peripherals, each consuming 16 Kilobytes of address space.
- SPBA can be considered as the 32th peripheral, used for resource ownership and access control mechanism to the 31 peripherals,
- Provides 31 sets of Out of Band Steering Control (obsc) signals to the off-block steering logic,
- Operating frequency up to 67 MHz,
- clocks: ipg_clk, ipg_clk_s.

68.4 Modes of Operation

Thus the SPBA behavior is transparent when accessing a peripheral, it can distinguish different modes of operation:

Reset/Abort

The SPBA has a hardware reset which initialize all registers, arbitration and PRR (peripherals rights registers).

Additionally, an abort signal input is provided allowing each master to abort their current access and release their ownership (in case of master reset sequence,...).

Functional

Once a master request is granted, its IP Bus signals are steering to the requested peripheral.

Standby

No clock needed. The SPBA needs clocks only during: access to the PRRs, arbitration and abort phases. It generates two clock enable signals indicating when the clocks must be provided.

Configuration

This is the phase where a master is accessing the SPBA PRRs. Indeed, SPBA memory mapped registers are seen as a shared peripheral.

68.5 Functional Description

The following section intends to describe the main features of the SPBA.

68.5.1 Masters Arbitration

The arbitration mechanism determines which port will control the master port, based on a simple round-robin arbitration scheme.

We can distinguish between different cases:

- Only one master request per different access. So the master is switched to the shared peripheral bus, without arbitration. Figure 68-3 shows the MB request on the global module enable signal, served without wait state.
- If two masters simultaneously access to SPBA, then the last granted master is held-off, using **<master>_ips_xfr_wait** output signal (default value is high). When the master is granted **sips_xfr_wait** from shared IP Bus peripheral is connected to <master>_ips_xfr_wait outputs.
- If three masters simultaneously access to SPBA, then the last two granted masters are held-off, using their <master>_ips_xfr_wait signal. Figure 68-4 shows the case when the last two accesses granted are MA then MB, and how the requests are used even if they are in the same cycle.
- After reset, at the first multiple access, no master has been granted, thus, the priority is static: Master A (MA), Master B (MB) and last Master C (MC) port.
- No master request. No master switch to shared peripherals.

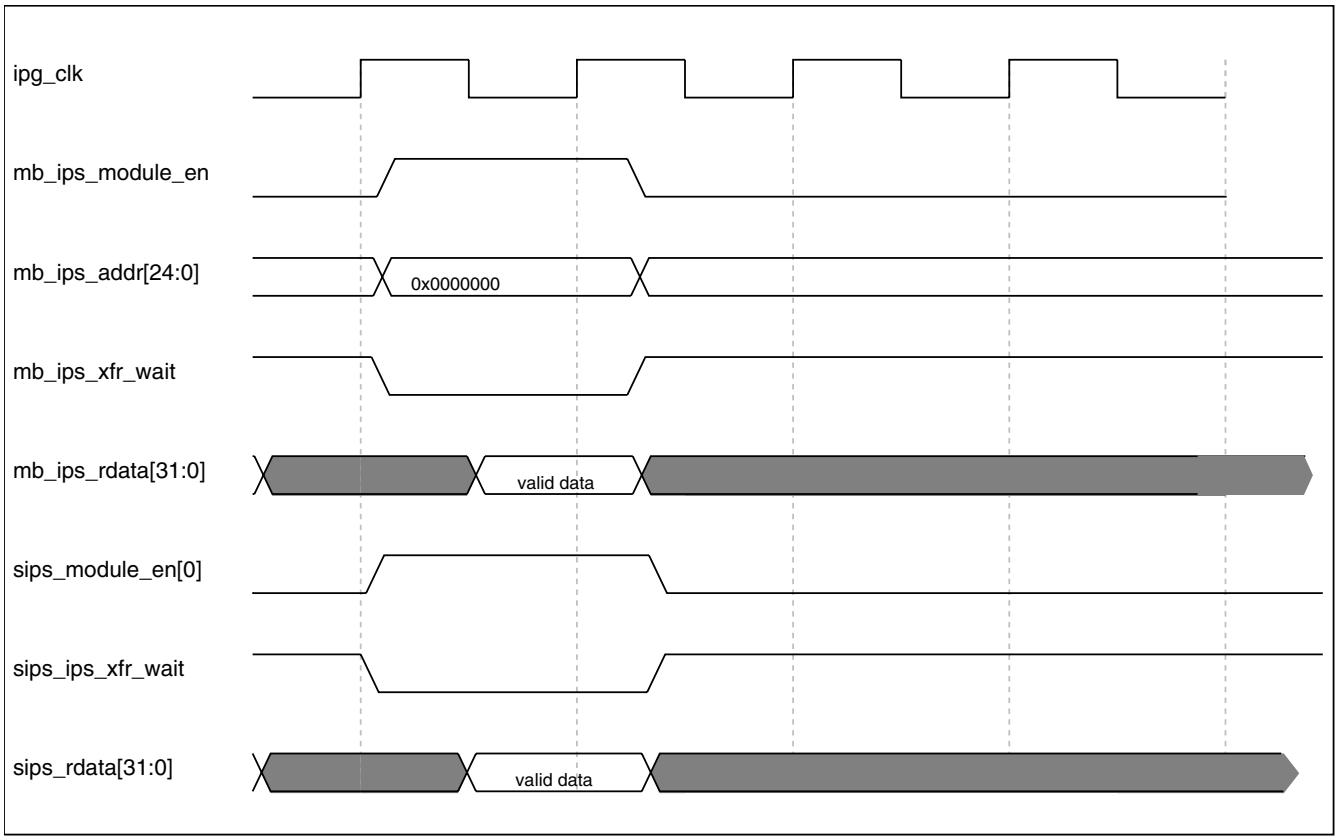


Figure 68-3. Example of one master request: no SPBA Arbitration;

The following figure assumes MA and MB have been the last two masters already granted in the previous transfers (MA then MB).

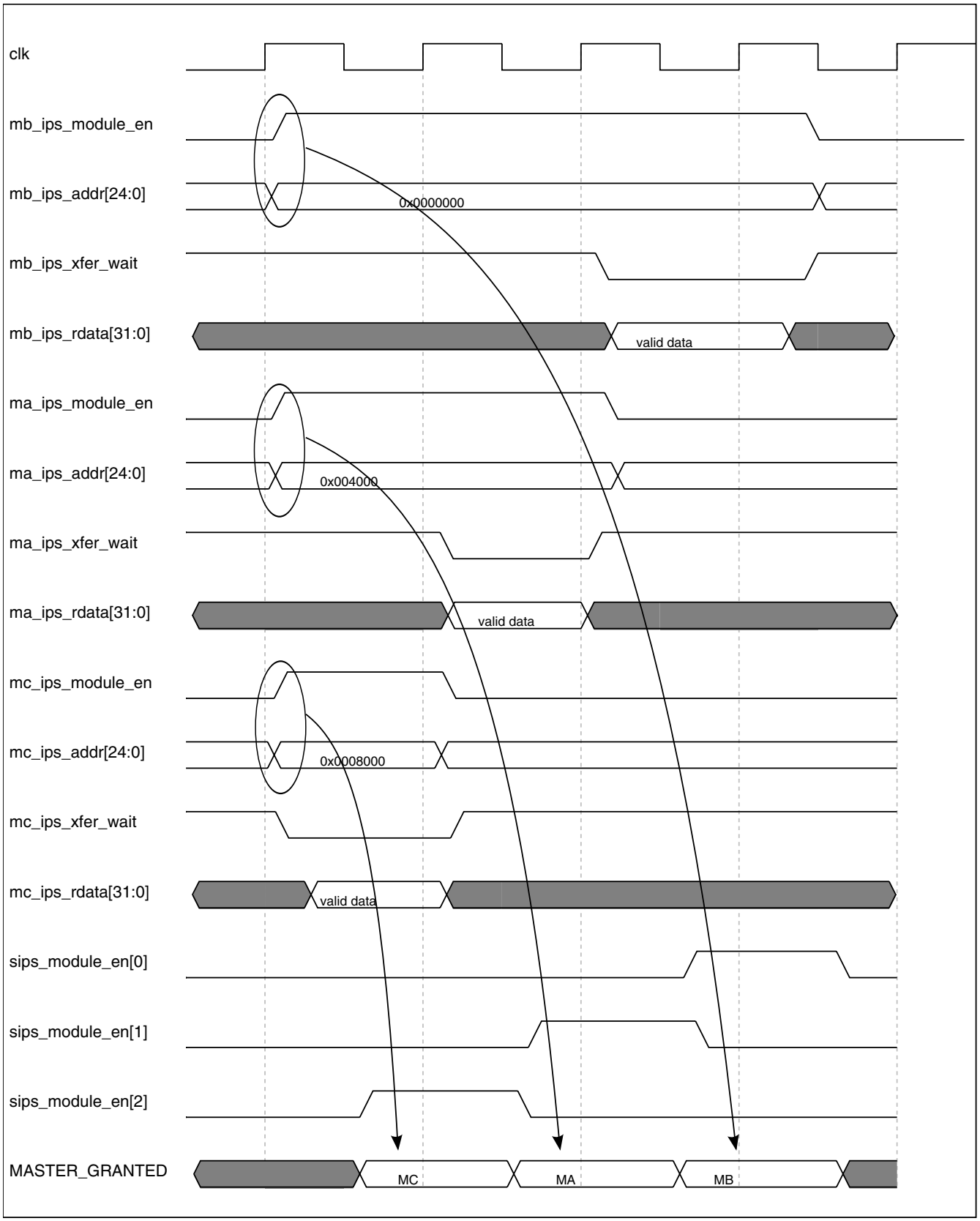


Figure 68-4. Example of three master requests: Masters already granted are "waited";

68.6 Resource Ownership Control

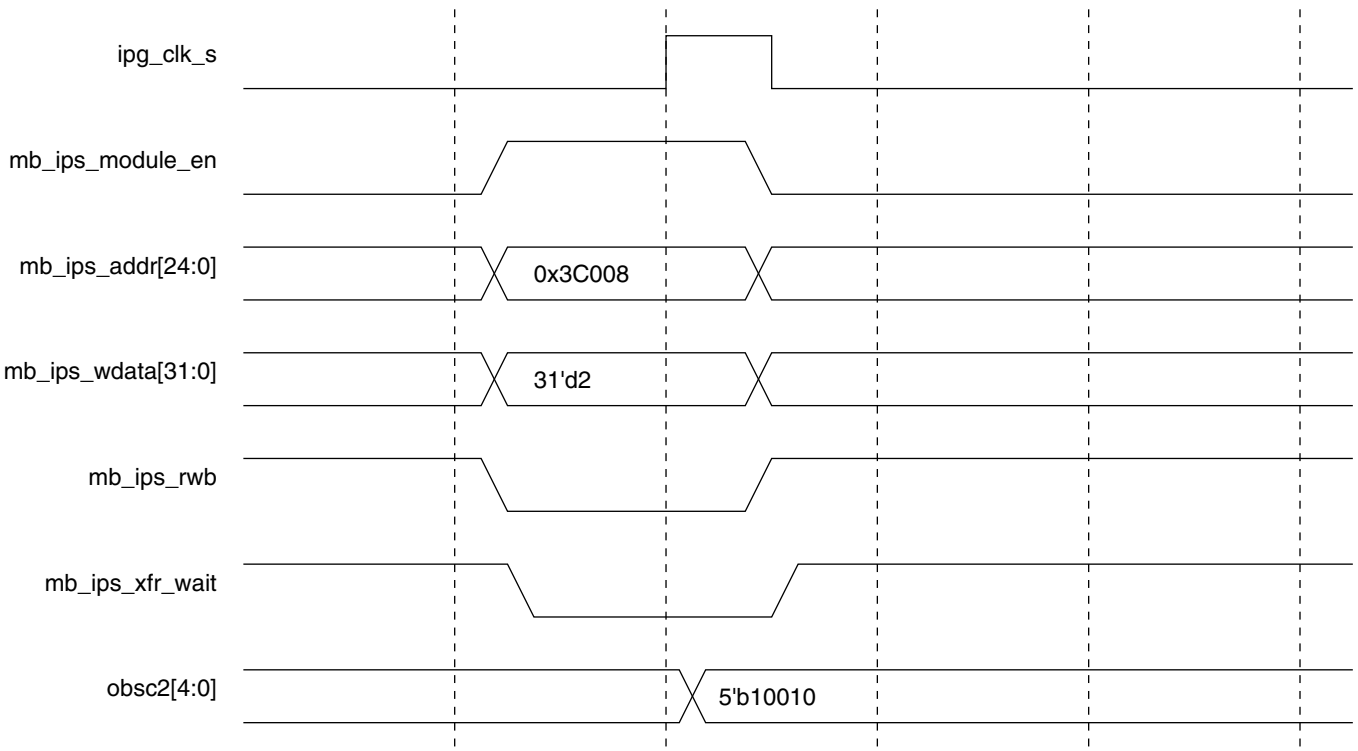
The Resource Ownership Control controls accesses to the shared peripherals and determines steering of out-of-band signals.

68.6.1 Access Control

68.6.1.1 Peripheral Access

The peripheral access (a.k.a resource access) of the requesting master is given by the corresponding RAR bit of the Peripheral Right Register. It determines if the master has access privilege to the resource.

Any attempted access to a resource by a requesting master whose access privilege bit is not set (in the PRR), is terminated with a bus error (**<master>_ips_xfr_err** is asserted by SPBA logic). The master which owns the resource can lock the peripheral for itself and/or grant other masters access to the peripheral by setting the appropriate bit(s) in the RAR field.



Master B is taking ownership of peripheral 2 by writing 3'b010 in the SPBA peripheral 2 right register (rarfield)
This ownership can be checked on obsc2 output as roi2[1:0] = 2'b10 and rar2[2:0] = 3'b010
(obsc[4:0] = {roi2[1], roi2[0], rar2[2], rar2[1], rar2[0]})

Figure 68-5. Example of one master B gaining ownership of peripheral 2

68.6.1.2 Peripheral Right Register Access

The ROI bits of the Peripheral Right Register (PRR) determine which master is allowed to make write access to PRR. The identification of the requesting master is compared to the ROI bits of the Peripheral Right Register to determine if the master has ownership of the corresponding register.

Any attempted write access to a Peripheral Right Register (PRR) already owned by another master will be ignored.

68.6.2 Owner Election

When the peripheral is not owned by any master (ROI="00", such as after coming out of reset), the first master to perform successfully a write to the RAR bits of the PRR is granted ownership of the peripheral and its associated PRR.

After writing to the PRR (RAR bit(s)), the master must read it back to make sure that it was granted ownership. If the RMO field is 2'b11, then the ownership claim is successful. If RMO is 2'b10, then another master claimed ownership before this master was able to complete its write. This resolves the case in which two or more masters attempt to write the PRR at the same time; only the first master will be granted ownership. However all masters must read the PRR to determine if this case occurred, and if so, whether they were the first master which was actually granted ownership.

NOTE

A master which has been granted ownership of the PRR does not automatically have the right access to the peripheral; it must still set its own RAR bits in the PRR to access the peripheral.

68.6.3 Ending Ownership

Ownership may be voluntarily ended by the owning master, or automatically upon assertion of a master-specific dead_owner signal.

The former is appropriate for software-controlled yielding of ownership. The latter is appropriate for automatic yielding of ownership when the owner has gone into reset. Hardware Controlled Ownership Ending

When a master is reset, it clears the ROI bits of the PRRs owned by the corresponding master. When the owner is dead (owner is in reset), all peripherals previously owned by that master must be changed to the un-owned state.

NOTE

It is the programmer's responsibility to make sure the peripherals are placed in an appropriate state before ending ownership.

68.6.3.1 Software Controlled Ownership Ending

The ROI bits will be automatically cleared when the master which owns the PRR access right clears (write) the RAR bits (Table 68-5).

It will then end the ownership of the PRR.

68.6.4 The Un-owned State

During the time when the peripheral is un-owned (i.e the ROI field contains all 0's), all masters have full access to it (RAR bits can then be modified by a master if ROI[1:0] = 2'b0).

In such cases it is necessary for software to ensure any necessary coherency in the resource, there is no hardware protection.

68.7 Memory Map/Register Definition

The following section describes the memory maps and detailed descriptions of all registers which are accessible to the end user within and through SPBA.

From a master side, the absolute address of one peripheral memory map registers is accessible by setting the <master>_ips_addr[24:0]:

- <master>_ips_addr[24:19] to the SPBA master base address (defined by top level parameters SPBA_<MASTER>_BASE_ADDR),
- <master>_ips_addr[18:14] to one of the 32 IP Bus peripherals (including SPBA). Refer to the table below,
- and <master>_ips_addr[13:0] to one of the 16-kilobyte memory mapped registers (16 Kilobytes only if accessing to a shared peripheral, not the same for SPBA PRR access).

The table below describes for each IP Bus peripheral, its absolute memory mapped address range accessible through the SPBA for one master (does not include the <master>_ips_addr[24:19] bits).

Table 68-1. Peripherals Absolute Address

Peripheral	<master>_ips_addr[18:0] start address	<master>_ips_addr[18:0] end address[1]
Peripheral 0	{5'b0_0000, 14'b00_0000_0000_0000}	{5'b0_0000, 14'b11_1111_1111_1111}
Peripheral 1	{5'b0_0001, 14'b00_0000_0000_0000}	{5'b0_0001, 14'b11_1111_1111_1111}

Table continues on the next page...

Table 68-1. Peripherals Absolute Address (continued)

Peripheral	<master>_ips_addr[18:0] start address	<master>_ips_addr[18:0] end address[1]
Peripheral 2	{5'b0_0010, 14'b00_0000_0000_0000}	{5'b0_0010, 14'b11_1111_1111_1111}
Peripheral 3	{5'b0_0011, 14'b00_0000_0000_0000}	{5'b0_0011, 14'b11_1111_1111_1111}
Peripheral 4	{5'b0_0100, 14'b00_0000_0000_0000}	{5'b0_0100, 14'b11_1111_1111_1111}
Peripheral 5	{5'b0_0101, 14'b00_0000_0000_0000}	{5'b0_0101, 14'b11_1111_1111_1111}
Peripheral 6	{5'b0_0110, 14'b00_0000_0000_0000}	{5'b0_0110, 14'b11_1111_1111_1111}
Peripheral 7	{5'b0_0111, 14'b00_0000_0000_0000}	{5'b0_0111, 14'b11_1111_1111_1111}
Peripheral 8	{5'b0_1000, 14'b00_0000_0000_0000}	{5'b0_1000, 14'b11_1111_1111_1111}
Peripheral 9	{5'b0_1001, 14'b00_0000_0000_0000}	{5'b0_1001, 14'b11_1111_1111_1111}
Peripheral 10	{5'b0_1010, 14'b00_0000_0000_0000}	{5'b0_1010, 14'b11_1111_1111_1111}
Peripheral 11	{5'b0_1011, 14'b00_0000_0000_0000}	{5'b0_1011, 14'b11_1111_1111_1111}
Peripheral 12	{5'b0_1100, 14'b00_0000_0000_0000}	{5'b0_1100, 14'b11_1111_1111_1111}
Peripheral 13	{5'b0_1101, 14'b00_0000_0000_0000}	{5'b0_1101, 14'b11_1111_1111_1111}
Peripheral 14	{5'b0_1110, 14'b00_0000_0000_0000}	{5'b0_1110, 14'b11_1111_1111_1111}
Peripheral 15	{5'b0_1111, 14'b00_0000_0000_0000}	{5'b0_1111, 14'b11_1111_1111_1111}
Peripheral 16	{5'b1_0000, 14'b00_0000_0000_0000}	{5'b1_0000, 14'b11_1111_1111_1111}
Peripheral 17	{5'b1_0001, 14'b00_0000_0000_0000}	{5'b1_0001, 14'b11_1111_1111_1111}
Peripheral 18	{5'b1_0010, 14'b00_0000_0000_0000}	{5'b1_0010, 14'b11_1111_1111_1111}
Peripheral 19	{5'b1_0011, 14'b00_0000_0000_0000}	{5'b1_0011, 14'b11_1111_1111_1111}
Peripheral 20	{5'b1_0100, 14'b00_0000_0000_0000}	{5'b1_0100, 14'b11_1111_1111_1111}
Peripheral 21	{5'b1_0101, 14'b00_0000_0000_0000}	{5'b1_0101, 14'b11_1111_1111_1111}
Peripheral 22	{5'b1_0110, 14'b00_0000_0000_0000}	{5'b1_0110, 14'b11_1111_1111_1111}
Peripheral 23	{5'b1_0111, 14'b00_0000_0000_0000}	{5'b1_0111, 14'b11_1111_1111_1111}
Peripheral 24	{5'b1_1000, 14'b00_0000_0000_0000}	{5'b1_1000, 14'b11_1111_1111_1111}
Peripheral 25	{5'b1_1001, 14'b00_0000_0000_0000}	{5'b1_1001, 14'b11_1111_1111_1111}
Peripheral 26	{5'b1_1010, 14'b00_0000_0000_0000}	{5'b1_1010, 14'b11_1111_1111_1111}
Peripheral 27	{5'b1_1011, 14'b00_0000_0000_0000}	{5'b1_1011, 14'b11_1111_1111_1111}
Peripheral 28	{5'b1_1100, 14'b00_0000_0000_0000}	{5'b1_1100, 14'b11_1111_1111_1111}
Peripheral 29	{5'b1_1101, 14'b00_0000_0000_0000}	{5'b1_1101, 14'b11_1111_1111_1111}
Peripheral 30	{5'b1_1110, 14'b00_0000_0000_0000}	{5'b1_1110, 14'b11_1111_1111_1111}
Peripheral 31	{5'b1_1111, 14'b00_0000_0000_0000}	{5'b1_1111, 14'b11_1111_1111_1111}

1. One of these peripherals must be defined as the SPBA. The end address range given is not valid for SPBA internal registers. If <master>_ips_addr[13:0] is greater than the last PRR location, the SPBA will return a transfer error to the master (report to Table 68-2).

68.7.1 SPBA Register Definition

The SPBA control registers (Peripheral Right Registers) are mapped as a virtual shared peripheral using by default the sips_module_en[15] (or another one using verilog parameter during integration phase) which must not be used for shared blocks.

SPBA can support up to 31 shared peripherals. Each of them has its own Peripheral Right Register (PRR) accessible within the SPBA memory mapped registers, and consists of the Requesting Master Owner, the Resource Owner ID and the Resource Access Right fields (using verilog parameters it can be defined which peripheral is present or not, and so if the corresponding PRR exists or not).

Table 68-2. SPBA PRR Register Summary

Name		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRRn[1] address = ($BASE[2] + n*4)	R	RMOn														ROIn	
	W																
	R														RARn		
	W																

1. for n=0 to 31
2. $BASE = {6'bSPBA_<MASTER>_BASE_ADDR, 5'bx_xxxx, 14'b00_0000_0000_0000}, where 5'bx_xxxx are <master>_ips_addr[18:14] reserved for SPBA registers access. The following table details the value of <master>_ips_addr[13:0] for master access any of the shared PRR

Table 68-3. PRR addresses

Peripheral Rights Registers (SPBA control registers)	<master>_ips_addr[13:0][1]	Note
Peripheral 0 Rights Registers	14'b00_0000_0000_0000	if MODULE0_PRESENT = 1'b1 & MODULE0_IS_SPBA = 1'b0
Peripheral 1 Rights Registers	14'b00_0000_0000_0100	if MODULE1_PRESENT = 1'b1 & MODULE1_IS_SPBA = 1'b0
Peripheral 2 Rights Registers	14'b00_0000_0000_1000	if MODULE2_PRESENT = 1'b1 & MODULE2_IS_SPBA = 1'b0
Peripheral 3 Rights Registers	14'b00_0000_0000_1100	if MODULE3_PRESENT = 1'b1 & MODULE3_IS_SPBA = 1'b0
Peripheral 4 Rights Registers	14'b00_0000_0001_0000	if MODULE4_PRESENT = 1'b1 & MODULE4_IS_SPBA = 1'b0
Peripheral 5 Rights Registers	14'b00_0000_0001_0100	if MODULE5_PRESENT = 1'b1 & MODULE5_IS_SPBA = 1'b0
Peripheral 6 Rights Registers	14'b00_0000_0001_1000	if MODULE6_PRESENT = 1'b1 & MODULE6_IS_SPBA = 1'b0
Peripheral 7 Rights Registers	14'b00_0000_0001_1100	if MODULE7_PRESENT = 1'b1 & MODULE7_IS_SPBA = 1'b0

Table continues on the next page...

Table 68-3. PRR addresses (continued)

Peripheral Rights Registers (SPBA control registers)	<master>_ips_addr[13:0][1]	Note
Peripheral 8 Rights Registers	14'b00_0000_0010_0000	if MODULE8_PRESENT = 1'b1 & MODULE8_IS_SPBA = 1'b0
Peripheral 9 Rights Registers	14'b00_0000_0010_0100	if MODULE9_PRESENT = 1'b1 & MODULE9_IS_SPBA = 1'b0
Peripheral 10 Rights Registers	14'b00_0000_0010_1000	if MODULE10_PRESENT = 1'b1 & MODULE10_IS_SPBA = 1'b0
Peripheral 11 Rights Registers	14'b00_0000_0010_1100	if MODULE11_PRESENT = 1'b1 & MODULE11_IS_SPBA = 1'b0
Peripheral 12 Rights Registers	14'b00_0000_0011_0000	if MODULE12_PRESENT = 1'b1 & MODULE12_IS_SPBA = 1'b0
Peripheral 13 Rights Registers	14'b00_0000_0011_0100	if MODULE13_PRESENT = 1'b1 & MODULE13_IS_SPBA = 1'b0
Peripheral 14 Rights Registers	14'b00_0000_0011_1000	if MODULE14_PRESENT = 1'b1 & MODULE14_IS_SPBA = 1'b0
Peripheral 15 Rights Registers	14'b00_0000_0011_1100	if MODULE15_PRESENT = 1'b1 & MODULE15_IS_SPBA = 1'b0
Peripheral 16 Rights Registers	14'b00_0000_0100_0000	if MODULE16_PRESENT = 1'b1 & MODULE16_IS_SPBA = 1'b0
Peripheral 17 Rights Registers	14'b00_0000_0100_0100	if MODULE17_PRESENT = 1'b1 & MODULE17_IS_SPBA = 1'b0
Peripheral 18 Rights Registers	14'b00_0000_0100_1000	if MODULE18_PRESENT = 1'b1 & MODULE18_IS_SPBA = 1'b0
Peripheral 19 Rights Registers	14'b00_0000_0100_1100	if MODULE19_PRESENT = 1'b1 & MODULE19_IS_SPBA = 1'b0
Peripheral 20 Rights Registers	14'b00_0000_0101_0000	if MODULE20_PRESENT = 1'b1 & MODULE20_IS_SPBA = 1'b0
Peripheral 21 Rights Registers	14'b00_0000_0101_0100	if MODULE21_PRESENT = 1'b1 & MODULE21_IS_SPBA = 1'b0
Peripheral 22 Rights Registers	14'b00_0000_0101_1000	if MODULE22_PRESENT = 1'b1 & MODULE22_IS_SPBA = 1'b0
Peripheral 23 Rights Registers	14'b00_0000_0101_1100	if MODULE23_PRESENT = 1'b1 & MODULE23_IS_SPBA = 1'b0
Peripheral 24 Rights Registers	14'b00_0000_0110_0000	if MODULE24_PRESENT = 1'b1 & MODULE23_IS_SPBA = 1'b0
Peripheral 25 Rights Registers	14'b00_0000_0110_0100	if MODULE25_PRESENT = 1'b1 & MODULE24_IS_SPBA = 1'b0
Peripheral 26 Rights Registers	14'b00_0000_0110_1000	if MODULE26_PRESENT = 1'b1 & MODULE26_IS_SPBA = 1'b0
Peripheral 27 Rights Registers	14'b00_0000_0110_1100	if MODULE27_PRESENT = 1'b1 & MODULE27_IS_SPBA = 1'b0
Peripheral 28 Rights Registers	14'b00_0000_0111_0000	if MODULE28_PRESENT = 1'b1 & MODULE28_IS_SPBA = 1'b0

Table continues on the next page...

Table 68-3. PRR addresses (continued)

Peripheral Rights Registers (SPBA control registers)	<master>_ips_addr[13:0][1]	Note
Peripheral 29 Rights Registers	14'b00_0000_0111_0100	if MODULE29_PRESENT = 1'b1 & MODULE29_IS_SPBA = 1'b0
Peripheral 30 Rights Registers	14'b00_0000_0111_1000	if MODULE30_PRESENT = 1'b1 & MODULE30_IS_SPBA = 1'b0
Peripheral 31 Rights Registers	14'b00_0000_0111_1100	if MODULE31_PRESENT = 1'b1 & MODULE31_IS_SPBA = 1'b0

1. Any accesses to SPBA inexistent location will generate a <master>_ips_xfr_err

68.7.1.1 Peripheral Right Register (SPBA_PRRn)

Peripheral Right Register Diagram (SPBA_PRRn)

SPBA_PRRn[1]				Peripheral Right Register n									Addr $BASE[2] + n*4 Wait State: 0 Access: -			
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMOn														ROIn	
TYPE:	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Note:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
														RARn		
TYPE:	r	r	r	r	r	r	r	r	r	r	r	r	r	rw	rw	rw
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
Note:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

1. for n=0 to 31
2. $BASE = {6'bSPBA_MX_BASE_ADDR, 5'bx_xxxx, 14'b00_0000_0000_0000}, where 5'bx_xxxx are the bit [18:14] of master address bus reserved for SPBA access

RMOn[1:0] - Requesting Master Owner

This 2-bit register field indicates if the corresponding resource is owned by the requesting master or not. This register is reset to 2'b0 if ROI[1:0] = 2'b0.

Table 68-4. RMOn Values

Value	Meaning
00	resource un-owned
10	resource owned by another master
11	resource owned by requesting master

ROIn[1:0] - Resource Owner ID

This 2-bits register field indicates which master (one at a time) can access to the PRR for rights modification. This is a Read-Only register.

Table 68-5. ROIn Values

Value	Meaning
00	un-owned resource
01	resource owned by master A port
10	resource owned by master B port
11	resource owned by master C port

After reset, ROI bits are cleared ("00" -> un-owned resource).

A master performing a write access to the an un-owned PRR will get its ID automatically written into ROI, while modifying RAR bits. It can then read back the RMO, RAR, ROI bits to make sure RMO returns the right value, ROI bits contain its ID and RAR bits are correctly asserted. Then no other master (whom ID is different from the one stored in ROI) will be able to modify RAR fields.

Owner master of a peripheral can assert its dead_owner signal, or write 3'b0 in the RAR to release the ownership (ROI[1:0] reset to 2'b0).

RARn[2:0] - Resource Access Right

This 3-bit register field indicates which master can access to the peripheral. From 0 up to 3 masters can have permission to access a resource (all the master can be granted on a peripheral, but only one access at a time will be granted by SPBA).

RARn[0] - Control and Status bit for master A which is connected to port MA,

access to peripheral is granted.

access to peripheral is not allowed.

RARn[1] - Control and Status bit for master B which is connected to port MB,

access to peripheral is granted.

access to peripheral is not allowed.

RARn[2] - Control and Status bit for master C which is connected to port MC,

access to peripheral is granted.

access to peripheral is not allowed.

After reset all RAR registers are set to value 1. So by default all masters have access granted to all the shared peripherals. This is verified until one master modify this default value.

Chapter 69
Sony/Philips Digital Interface (SPDIF)

69.1 Introduction

The Sony/Philips Digital Interface (SPDIF) audio block is a stereo transceiver that allows the processor to receive and transmit digital audio. The SPDIF transceiver allows the handling of both SPDIF channel status (CS) and User (U) data and includes a frequency measurement block that allows the precise measurement of an incoming sampling frequency.

A recovered clock is provided to drive both internal and external components in the system such as ESAI ports, as well as external A/Ds or D/As, with clocking control provided via related registers.

As the SPDIF internal data width is 24-bit, the eight most-significant bits of all registers return zeros.

The figure below shows a block diagram of the SPDIF transceiver data paths (receiver and transmitter) and its interface.

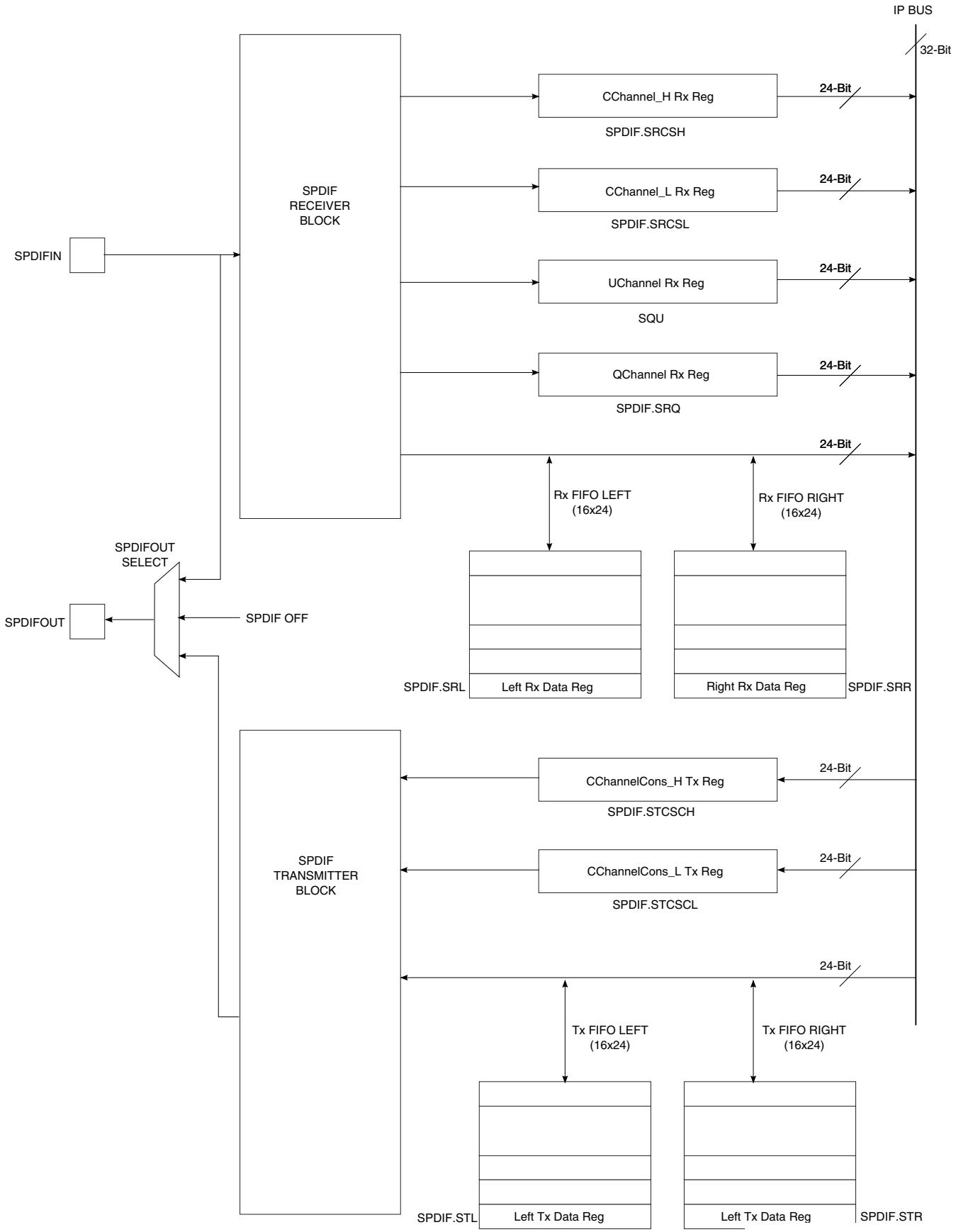


Figure 69-1. SPDIF Transceiver Data Interface Block Diagram

69.1.1 Overview

The SPDIF is composed of two parts: SPDIF Receiver and SPDIF Transmitter.

The SPDIF receiver extracts the audio data from each SPDIF frame and places the data in the SPDIF Rx left and right FIFOs. The Channel Status and User Bits are also extracted from each frame and placed in the corresponding registers. The SPDIF receiver also provides a bypass option for direct transfer of the SPDIF input signal to the SPDIF transmitter.

For the SPDIF transmitter, the audio data is provided by the processor via the SPDIFTxLeft and SPDIFTxRight registers. The Channel Status bits are also provided via the corresponding registers. The SPDIF transmitter generates a SPDIF output bitstream in the biphase mark format (IEC60958), which consists of audio data, channel status and user bits.

In the SPDIF transmitter, the IEC60958 biphase bit stream is generated on both edges of the SPDIF Transmit clock. The SPDIF Transmit clock is generated by the SPDIF internal clock generate block and the sources are from outside of the SPDIF block. For the SPDIF receiver, it can recover the SPDIF Rx clock. Both the Rx clock and Tx clock are be sent to the ASRC. shows the clock structure of the SPDIF transceiver.

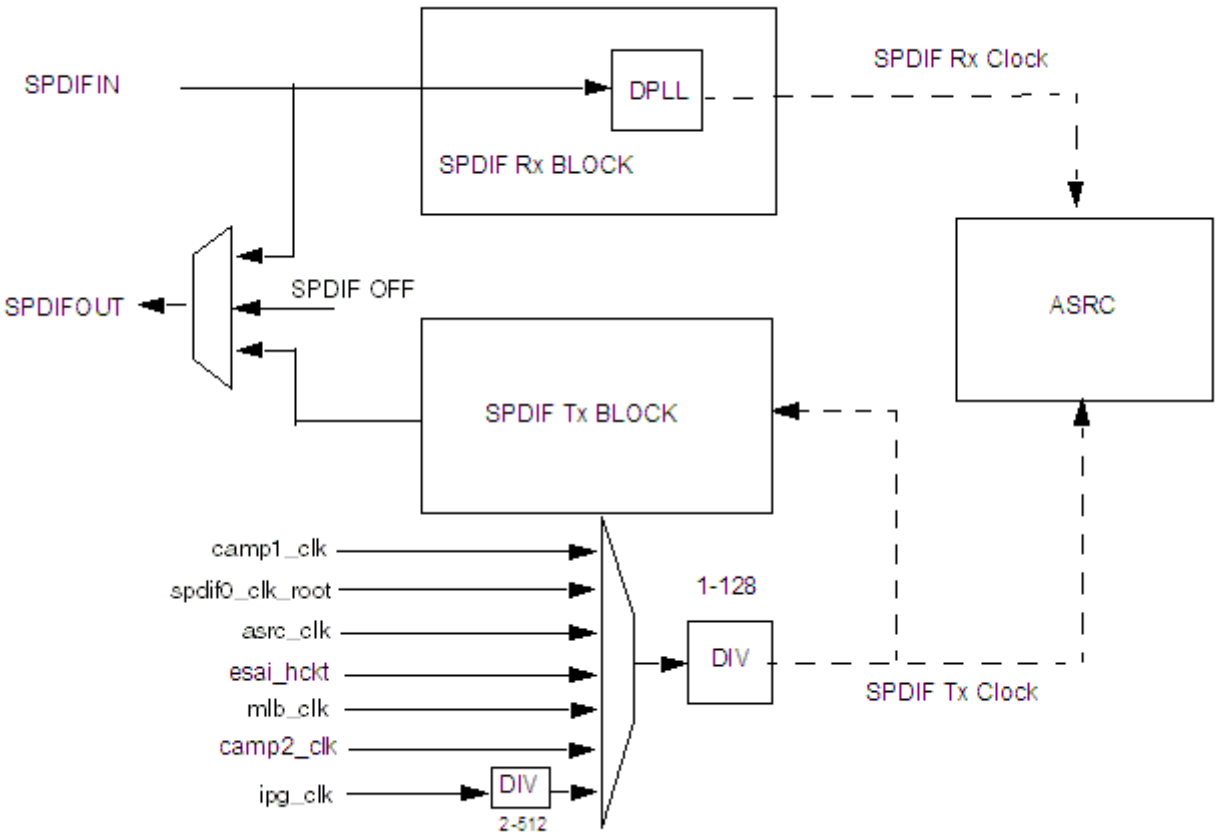


Figure 69-2. SPDIF Transceiver Clock Diagram

69.2 External Signal Description

Table 69-1. Signal Properties

Signal Name	Signal Type	Description
SPDIFIN	input	SPDIF Input Line IEC60958 data in biphase mark format.
SPDIFOUT	output	SPDIF Output Line IEC60958 data in biphase mark format. (Consumer C channel).

69.3 Functional Description

69.3.1 SPDIF Receiver

The SPDIF receiver extracts the audio data from each SPDIF frame and places the data in Rx left and right FIFOs.

The Tx left and right FIFOs are 16-deep and 24-wide (equal to the audio data width). The Channel Status and User Bits are also extracted from each frame and placed in corresponding registers. The SPDIF receiver also provides a bypass option for direct transfer of the SPDIF input signal to the SPDIF transmitter.

The SPDIF receiver handles the main data audio stream and recovers the bit clock from the SPDIF input signal. The sample rate can be determined from the frequency measuring block. Additionally, the receiver supports the SPDIF C and U channels. The SPDIF C and U channel data is interfaced directly to memory-mapped registers.

All the data registers are controlled by the Interrupt Control Block and transferred to the memory-mapped IP bus.

The following functions are performed by the SPDIF receiver:

- Audio Data Reception see Audio Data Reception
- Channel Status bits Reception see Channel Status Reception
- U Channel bits Reception see User Bit Reception
- Validity Flag Reception see Validity Flag Reception
- SPDIF Receiver Exception support see SPDIF Receiver
- SPDIF Lock Detection

69.3.1.1 Audio Data Reception

The SPDIF Receiver block extracts the audio data from the IEC60958 stream, and outputs this via Rx left and right FIFOs to the memory-mapped registers SPDIFRxLeft and SPDIFRxRight. Data from the SPDIF receiver is buffered in receive FIFO, and can be read by the processor from the memory-mapped registers.

- **SPDIF receiver data registers - Behavior on overrun, underrun**

The SPDIF Data Receive registers (SPDIFRxLeft and SPDIFRxRight) have individual FIFOs for left and right channel. As a result, there is always the possibility that left and right FIFOs may go out of sync due to FIFO underruns and FIFO overruns that affect only one part (left or right) of any FIFOs. To prevent this from happening, hardware has been added to the device. Two mechanisms to prevent mismatch between the FIFOs are available.

If a SPDIF Data Rx FIFO overrun occurs on e.g. the right half of the FIFO, the sample that caused the overrun is not written to the right half (due to overrun). Special hardware will make sure the next sample is not written to the left half of the FIFO. If the overrun occurs on the left half of the FIFO, the next sample is not written to the right half of the FIFO.

- **SPDIF receiver data registers - Automatic resynchronization of FIFOs**

An automatic FIFO resynchronization feature is available. It can be enabled and disabled separately for every FIFO. If it is enabled, the hardware will check to see if the left and right FIFOs are in sync. If that is not the case, it will set the filling pointer of the right FIFO to be equal to the filling pointer of the left FIFO.

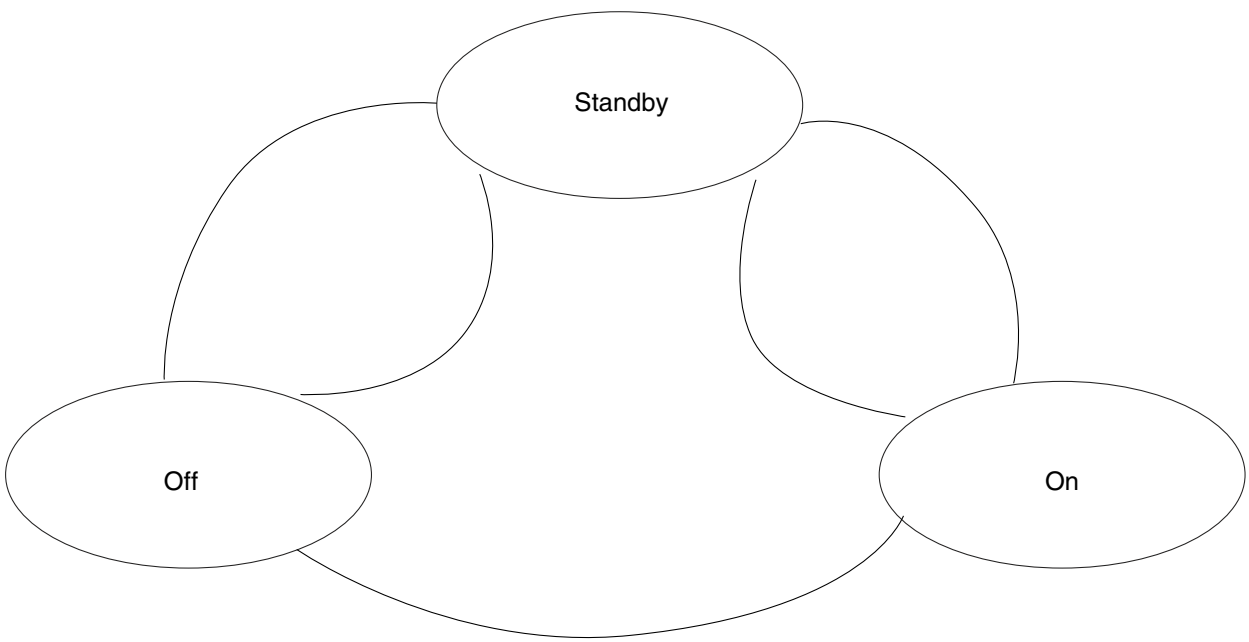


Figure 69-3. FIFO Auto-resync Controller State Machine

The operation is explained from the state diagram shown above. Every FIFO auto-resync controller has a state machine with 3 states: Off, StandBy and On. In the On state, the filling of the left FIFO is compared with the filling of right, and if they are not equal, right is made equal to left, and an interrupt is generated.

The controller will stay in Off state when the feature is disabled. When not disabled, the state machine will go to Off state on any processor read or write to the FIFO. It will go from On or Off to Standby on any left sample read from SPDIF Tx FIFOs, or on any left sample write to SPDIF Rx FIFOs. The controller will go from Standby to On on any right

sample read from SPDIF Tx FIFO, or on any right sample write to SPDIF Rx FIFO. There is a control bit in the SPDIFConfig register to enable/disable the feature for the SPDIF Rx FIFO and SPDIF Tx FIFO.

69.3.1.1.1 Application Note

The automatic FIFO resynchronization can be switched on, and will avoid all mismatches between left and right FIFOs, if the software obeys the following rules: 1. When the left data is read or written to the left FIFO, in the same place of the program, data must be read or written to the right FIFO. Maximum time difference between left and right is 1/2 sample clock. (E.g. if sample frequency is 44 Khz, approximately 10 micro-seconds. For 88 Khz, approximately 5 micro-seconds.) 2. Write/read data to FIFO s at least 2 samples at the time. If there is a mismatch Left-Right, the resync logic may go on only 1 sample clock after last data is read/written to the FIFO. Also acceptable is polling the FIFO, if at least part of the time 2 samples will be read/written to it.

- **SPDIF receiver - Additional features**

There are three exceptions associated with the SPDIF Receivers FIFOs

- full
- under/overrun
- resync

When the "full" condition is set for processor data input registers, the processor should read data from the FIFO, before overrun occurs. When "full" is set, and the FIFO contains e.g. 6 samples, it is acceptable for the software to read first 6 samples from the LEFT address, followed by 6 samples from the RIGHT address, or 6 samples from the RIGHT address, followed by 6 samples from the LEFT address, or 1 sample LEFT, followed by 1 sample RIGHT repeated 6 times. There is no order specified.

The implementation for SPDIF Rx is a double FIFO, one for left and one for right. "full" is set when both FIFOs are full. "underrun, overrun" are set when one of the FIFOs do underrun or do overrun. The resync interrupt means hardware took special action to resynchronize left and right FIFOs.

The FIFO level at which the "full" interrupt is generated, is programmable via the Full Select field in the SPDIFConfigReg register.

Rx FIFO on and Rx FIFO reset.

Two additional control fields of the SPDIF Rx FIFO are the on/off select and FIFO reset fields.

If on/off select is set to off, all-zero will be read from the FIFO, irrespective of the data received over the SPDIF interface.

If FIFO reset is set, the FIFO is blocked at "1 sample in FIFO". In this, the full interrupt will be on if FullSelect is set to "00". If FullSelect is set to any other value, interrupt will be off. The other interrupts are always off.

69.3.1.2 Channel Status Reception

A total of 48 channel status bits are received in two registers. No interpretation is performed by the SPDIF receiver block.

Channel Status Bits are ordered first bit left. CS-channel MSB bit "0" is located in bit position 23 in the memory-mapped register SPDIFRxCChannel_h. CS-channel bit "23" is considered the LSB bit 0 in the register. C-channel bit 24 to 47 is seen as [23:0] bits of register SPDIFRxCChannel_l.

69.3.1.2.1 Channel Status Interrupt

When the value of a new SPDIF "CS" channel status frame is loaded in the register, an interrupt is generated. The interrupt is cleared when the processor writes the corresponding bit in the InterruptStat register.

69.3.1.3 User Bit Reception

There are two modes for U Channel reception, CD and non-CD. As is decided by USyncMode (bit 1 of CDText_Control register).

- **Behavior of U Channel receive interface on incoming CD U Channel Sub-code in SPDIF receiver**.

This mode is selected if UsyncMode, bit 1 in register CD Text control is set "1".

The CD sub-code stream embedded into the SPDIF U channel consists of a sequence of packets. Every packet is made up 98 "symbols". The first two symbols of every packet are "sync symbols", the other 96 symbols are "data symbols".

Any sequence found in the SPDIF U channel stream starting with a leading one, followed by 7 information bits, is recognized as a "data symbol". Subsequent data symbols are separated by "pauses". During the "pause", "zero bits" are seen on the SPDIF U channel.

Data symbols are coming in MSB first. The MSB is the leading one.

When a "long pause" is seen between 2 subsequent "data symbols", the SPDIF receiver will assume the reception of one or more "sync symbols". Table below gives details.

Table 69-2. Sync Control Bits

Number of U Channel zero bits	Corresponding number of sync symbols
0-1	Unpredictable, not allowed
2-10	0
11-22	1
23-34	2
35-46	3
>45	Unpredictable, not allowed

The recognition of the number of sync symbols derives from the fact that the U channel transmitter in the CD channel decoder will transmit one symbol on average every 12 SPDIF channel bits. On this average rate, there is a maximum tolerance of 5%.

The SPDIF receiver is tolerant of symbol errors. Due to the physical nature of the transmission of the data over the CD disc, not more than 1 out of any 5 consecutive user channel symbols may be in error. The error may cause a change in data value, which is not detected by this interface, or it may cause a data symbol to be seen as a sync symbol, or a sync symbol to be seen as a data symbol. However, not more than 1 out of any 5 consecutive user channel symbols should be affected in this way.

The SPDIF U channel circuitry recognizes the 98-symbol packet structure, and sends the 96 symbol payload to the processor application. The 96 symbol payload is transmitted to the processor via 2 registers:

- The SPDIFRxUChannel register. In this register, data is presented 3 symbols at the time to the processor. Every time 3 new valid symbols, received on the SPDIF U Channel are present, the UChannelRxFull interrupt is asserted. For one 98-symbol packet, 96 symbols are carried across SPDIFRxUChannel. To transfer all this data, 32 UChannelRxFull interrupts are generated.
- The QChannelReceive register. In this register, only the Q bit of the packet is accumulated.Operation is similar to UChannelReceive. Because only Q-bit is transferred, only 96 Q-bits are transferred for any 98-symbol packet. To transfer this data, 4 QChannelRxFull interrupts are generated. When QChannelRxFull occurs, it is coincident with UChannelRxFull. There is only one QChannelRxFull for every 8 UChannelRxFull. The convention is that most significant data is transmitted first, and is left-aligned in the registers.
- Timing regarding packet boundary is extracted by hardware. The last UChannelRxFull corresponding to a given packet should be coincident with the last QChannelRxFull. In this last U, Q channel interrupt, symbols 95-98 are received, Q

channel bits 67-98. The interrupts are coincident with UQSyncFound, flagging last symbols of the current frame.
- When the start of the new packet is found before the current packet is complete (less than 98 symbols in the packet), the UQFrameError interrupt is set. The application software should read out UChannelReceive and QchannelReceive registers, discard the value, and assume the start of a new packet.
- As already said, packet sync extraction is tolerant for single-symbol errors. Packet sync detection is based on the recognition of the sequence data-sync-sync-data in the symbol stream, because this is the only syncing sequence that is not affected by single errors. If the sync symbols are not found 98 symbols after the previous occurrence, it is assumed to be destroyed by channel error, and a new sync symbols is interpolated.
- Normally, only data bytes are passed to the application software. Every databyte will have its most significant bit set. If sync symbols are passed to the application software, they are seen as all-zero symbols. Sync symbols can only end up in the data stream due to channel error.

- **Behavior of U Channel receive interface on incoming non-CD data**.

This mode is selected if UsyncMode, bit 1 in register CD Text control is set'0'.

In non-CD mode, the SPDIF U channel stream is recognized as a sequence of "data symbols". No packet recognition is done.

Any sequence found in the SPDIF U channel stream starting with a leading one, followed by 7 information bits, is recognized as a "data symbol". Subsequent data symbols are separated by "pauses". During the "pause", "zero bits" are seen on the SPDIF U channel.

3 consecutive data symbols seen in the SPDIF U Channel stream are grouped together into the SPDIFRxUChannel register. First symbol is left, last symbol is right aligned. When SPDIFRxUChannel contains 3 new data symbols, UChannelRxFull is asserted.

In this mode, the operation of QchannelRx and associated interrupt QchannelRxFull is reserved, undefined. And the operation of UQFrameError and UQSyncFound is also reserved, undefined.

The U channel is extracted, and output by the SPDIF Rx on SPDIFRxUChannel-Stream.

When incoming SPDIF data parity error or bit error is detected, and if the next SPDIF word for that channel is error-free, the SPDIF word in error is replaced with the average of the previous word and next word. When incoming SPDIF data parity error or bit error is detected, and the next SPDIF word is in error, the previous SPDIF word is repeated.

69.3.1.4 Validity Flag Reception

An interrupt is associated with the Validity flag. (interrupt 16 - SPDIFValNoGood). This interrupt is set every time a frame is seen on the SPDIF interface with the validity bit set to "invalid".

69.3.1.5 SPDIF Receiver Interrupt Exception Definition

Several SPDIF exceptions can trigger an interrupt.

They are:

- Control Status channel change. Set when SPDIFRxCChannel1 register is updated. The register is updated for every new C-Channel received. The exception is reset on write to InterruptClear register.
- SPDIF Illegal Symbol. Set on reception of illegal symbol during SPDIF receive. Reset by writing register InterruptClear.[1]
- SPDIF bit error. Set on reception of bit error. (Parity bit does not match). Reset on write to InterruptClear register.
- Receive data FIFO full. Set when SPDIF receive data FIFO is full.
- Receive data FIFO underrun/overrun. Set when there is a underrun/overrun on the SPDIF receive data FIFO.
- Receive data FIFO resynchronization. Set when a resynchronization event occurs on the SPDIF receive data FIFO.
- Receive U Channel buffer full. Set when next 24 bits of U channel code are available.
- Receive Q Channel buffer overrun. Set when Q channel buffer overrun.
- Receive U Channel buffer overrun. Set on U channel buffer overrun.
- Receive Q Channel buffer full. Set when next 24 bits of Q channel code are available.
- Receive UQ sync found. Set when UQ channel sync found.
- Receive UQ frame error. Set when UQ frame error found.

1. The SPDIF input is a biphase/mark modulated signal. The time between any two successive transitions of the SPDIF signal is always 1, 2 or 3 SPDIF symbol periods long. The SPDIF receiver will parse the stream, and split it in so-called symbols. It recognizes s1, s2 and s3 symbols, depending on the length of the symbols. Not all sequences of these symbols are allowed. To give an example, a sequence s2-s1-s1-s1-s2 cannot occur in a no-error SPDIF signal. If the receiver finds such an illegal sequence, the illegal symbol interrupt is set. No corrective action is undertaken. When the interrupt occurs, this means that(a) The SPDIF signal is destroyed by noise (b) The SPDIF frequency changed.

69.3.1.6 Standards Compliance

The SPDIF interface is compatible with the Tech 3250-E standard of the European Broadcasting Union, except clause 6.3.3 and the IEC60958-3 Ed2 for relevant topics.

Supported input frequency range is 12 Khz up to 96 Khz. (fully compliant) and 96 Khz up to 176 Khz (Can interface with compliant SPDIF transmitter within same cabinet, making reasonable assumptions on jitter added due to interconnecting wire.)

Tolerated jitter on SPDIF input signals are 0.25 bit peak-peak for high frequencies. There is no jitter limit for low frequencies. The user channel extraction in CD mode is capable of coping with single-symbol errors, and still retrieve U channel frames on correct boundaries. This capability is required for reliable reception of CD-Text from some Philips CD channel decoders. This capability was deemed more important than compliance with the IEC60958 annex A.3 standard, and for this reason user channel reception is not compliant with IEC60958 annex A.3. However, the interface is capable to receive U channel inserted by a typical CD channel decoder. Also, in this case, it is more robust and tolerant for channel error than what is required by IEC60958 annex A.3.

69.3.1.7 SPDIF PLOCK Detection and Rxclk Output

Using the high speed system clock, the internal DPLL can extract the bit clock (advanced pulse) from the input bitstream. When this internal DPLL is locked, the LOCK bit of PhaseConfig Register will be set, and the SPDIF Lock output pin PLOCK will be asserted.

After DPLL has locked, the pulses are generated, and the average pulse rate is 128 x the sampling frequency.(For a 44.1 Khz input sampling frequency, the average pulse rate = 128 x 44.1 Khz.) The pulse signal is used in the FreqMeas circuit to generate the frequency measurement result.

69.3.1.8 Measuring Frequency of SPDIF_RxClk

The internal DPLL can extract the bit clock (advanced plus) from the input bitstream. To do that, it is necessary to measure the frequency of the incoming signal in relationship with the system clock (BUS_CLK).

Associated with it are two registers, PhaseConfig and FreqMeas. The circuit will measure the frequency of the incoming clock as a function of the BUS_CLK. The circuit is a second-order filter. The output is a value represented by an unsigned number stored in the 24-bit FreqMeas register, giving the frequency of the source as a function of the BUS_CLK.

FreqMeas[23:0] = FreqMeas_CLK / BUS_CLK * 2^{10} * GAIN.

For example, if the GAIN is selected as 8*210 (PhaseConfig[5:3] = 3'b011), the actual result

FreqMeas_CLK / BUS_CLK is equal to FreqMeas[23:0] / 2^{23}.

69.3.2 SPDIF Transmitter

Audio data for the SPDIF transmitter is provided by processor via the SPDIFTxLeft and SPDIFTxRight registers.

Clocking for SPDIF transmitter is from either the osc_audio, ipg_baud_spdif_clk, hckt, or spdif_extclk, mlbclk or frequency divided ipg_clk.A multiplexer is used to choose the clock source. The SPDIF transmitter clock source can be divided down as needed using Txclk_DF. The SPDIF transmitter output can be chosen from either the SPDIF transmitter block, directly from the SPDIF receiver (via the output multiplexer), or disabled.

The SPDIF transmitter generates a SPDIF output bitstream in IEC60958 biphase mark format, consisting of audio data, channel status.

69.3.2.1 Audio Data Transmission

Audio data for the SPDIF transmitter is provided by the processor via SPDIFTxLeft and SPDIFTxRight registers. They send audio data to Tx left and right FIFOs. The Tx left and right FIFOs are also 16-deep and 24-width (equal to the audio data width).

- **SPDIF transmitter data registers - Behavior on overrun, underrun**

The SPDIF Data Transmit registers (SPDIFTxLeft and SPDIFTxRight) have individual FIFOs for left and right channel. As a result, there is always the possibility that left and right FIFOs may go out of sync due to FIFO underruns and FIFO overruns that affect only one part (left or right) of any FIFO. To prevent this from happening, hardware has been added on the device. Two mechanisms to prevent mismatch between the FIFOs are available.

If SPDIF Tx FIFO underruns on the right half of the FIFO, no sample leaves that FIFO (because it was already empty). Special hardware will make sure that the next sample read from the left FIFO will not leave the FIFO (no read strobe is generated). If the underrun occurs on the left half of the FIFO, next read strobe to the right FIFO is blocked.

- **SPDIF transmitter data registers - Automatic resynchronization of FIFOs**

See Audio Data Reception.

- **SPDIFTxLeft, SPDIFTxRight details**

With SPDIF Tx FIFOs three exceptions are associated.

- empty
- under/overrun
- resync

When the empty condition is set for processor data output registers, the processor should write data to the FIFO, before underrun occurs. When empty is set and, for instance, 6 samples need to be written, it is acceptable for the software to write first 6 samples from the LEFT address, followed by 6 samples from the RIGHT address, or 1 sample LEFT, followed by 1 sample RIGHT repeated 6 times. Left should be written before right. The implementation of all data out FIFOs is a double FIFO, one for left and one for right. Empty is set when both FIFOs are empty. Underrun, overrun are set when one of the FIFOs do underrun or do overrun. Resync is set when the hardware resynchronizes left and right FIFOs.

On receiving underrun, overrun interrupt, synchronization between Left and Right words in the FIFOs may be lost. Synchronization will not be lost when the underrun or overrun comes from the IEC60958 side of the FIFO. If the processor reads or writes more data from, for example, left than from right, synchronization will be lost. If automatic resynchronization is enabled, and if the software obeys the rules to let this work, resynchronization will be automatic.

69.3.2.2 Channel Status Transmission

A total of 48 Consumer channel status bits are transmitted from two registers. Channel Status Bits are ordered first bit left.

CS-channel MSB bit "0" is located in bit position 23 in the memory-mapped register SPDIFTxCChannelCons_h. CS-channel bit "23" is considered bit 0 in the register. C-channel bits 24-47 are seen as MSB-LSB bits of register SPDIFTxCChannelCons_l.

69.3.2.3 Validity Flag Transmission

The validity bit setting is performed via bit 5 of the SPDIF_SCR register.

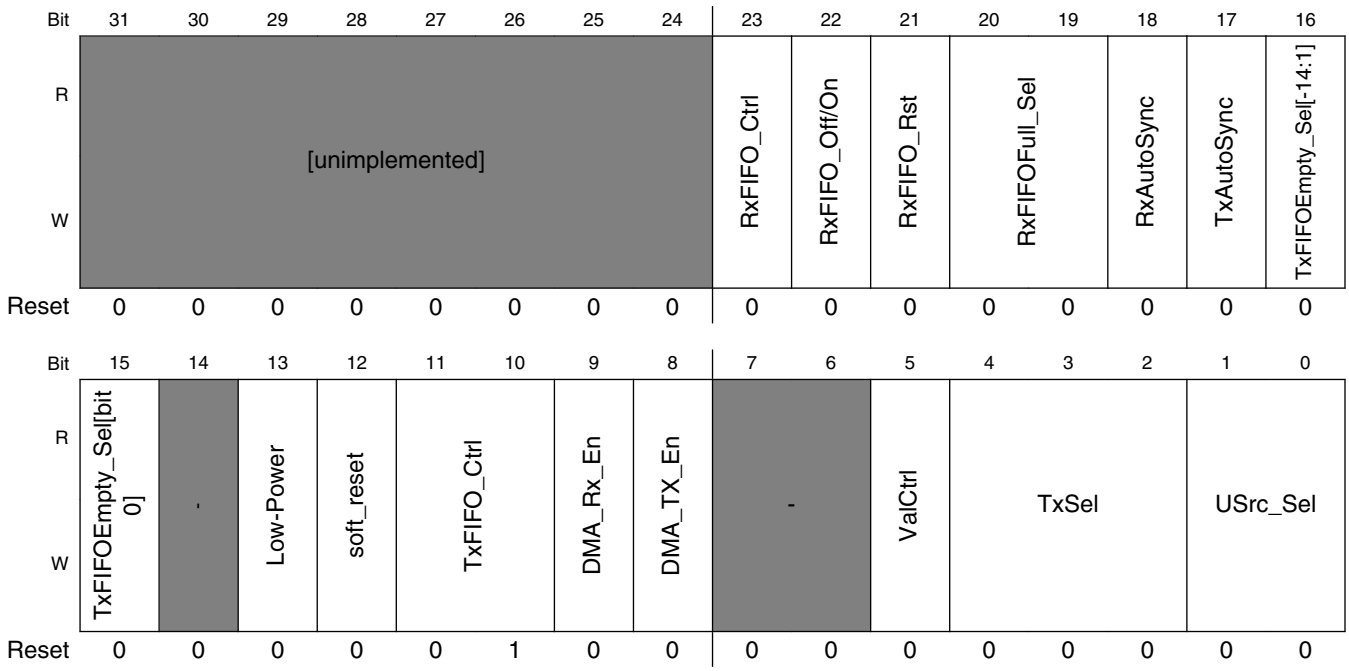
69.4 Programmable Registers

SPDIF memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5002_8000	SPDIF Configuration Register (SPDIF_SCR)	32	R/W	0000_0400h	69.4.1/ 4336
5002_8004	CDText Control Register (SPDIF_SRCD)	32	R/W	0000_0000h	69.4.2/ 4338
5002_8008	PhaseConfig Register (SPDIF_SRPC)	32	R/W	0000_0000h	69.4.3/ 4339
5002_800C	InterruptEn Register (SPDIF_SIE)	32	R/W	0000_0000h	69.4.4/ 4340
5002_8010	InterruptStat Register (SPDIF_SIS)	32	R	0000_0002h	69.4.5/ 4342
5002_8010	InterruptClear Register (SPDIF_SIC)	32	W	0000_0000h	69.4.6/ 4344
5002_8014	SPDIFRxLeft Register (SPDIF_SRL)	32	R	0000_0000h	69.4.7/ 4345
5002_8018	SPDIFRxRight Register (SPDIF_SRR)	32	R	0000_0000h	69.4.8/ 4346
5002_801C	SPDIFRxCChannel_h Register (SPDIF_SRCSH)	32	R	0000_0000h	69.4.9/ 4346
5002_8020	SPDIFRxCChannel_l Register (SPDIF_SRCSL)	32	R	0000_0000h	69.4.10/ 4347
5002_8024	UchannelRx Register (SPDIF_SRU)	32	R	0000_0000h	69.4.11/ 4347
5002_8028	QchannelRx Register (SPDIF_SRQ)	32	R	0000_0000h	69.4.12/ 4348
5002_802C	SPDIFTxLeft Register (SPDIF_STL)	32	W	0000_0000h	69.4.13/ 4348
5002_8030	SPDIFTxRight Register (SPDIF_STR)	32	W	0000_0000h	69.4.14/ 4349
5002_8034	SPDIFTxCChannelCons_h Register (SPDIF_STCSCH)	32	R/W	0000_0000h	69.4.15/ 4349
5002_8038	SPDIFTxCChannelCons_l Register (SPDIF_STCSCL)	32	R/W	0000_0000h	69.4.16/ 4350
5002_8044	FreqMeas Register (SPDIF_SRFM)	32	R	0000_0000h	69.4.17/ 4350
5002_8050	SPDIFTxClk Register (SPDIF_STC)	32	R/W	0002_0F00h	69.4.18/ 4351

69.4.1 SPDIF Configuration Register (SPDIF_SCR)

Address: SPDIF_SCR is 5002_8000h base + 0h offset = 5002_8000h



SPDIF_SCR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23 RxFIFO_Ctrl	0 Normal operation 1 Always read zero from Rx data register
22 RxFIFO_Off/On	0 SPDIF Rx FIFO is on 1 SPDIF Rx FIFO is off. Does not accept data from interface
21 RxFIFO_Rst	0 Normal operation 1 Reset register to 1 sample remaining
20–19 RxFIFOFull_Sel	00 Full interrupt if at least 1 sample in Rx left and right FIFOs 01 Full interrupt if at least 4 sample in Rx left and right FIFOs 10 Full interrupt if at least 8 sample in Rx left and right FIFOs 11 Full interrupt if at least 16 sample in Rx left and right FIFO
18 RxAutoSync	0 Rx FIFO auto sync off 1 RxFIFO auto sync on
17 TxAutoSync	0 Tx FIFO auto sync off 1 Tx FIFO auto sync on
16–15 TxFIFOEmpty_Sel	00 Empty interrupt if 0 sample in Tx left and right FIFOs 01 Empty interrupt if at most 4 sample in Tx left and right FIFOs

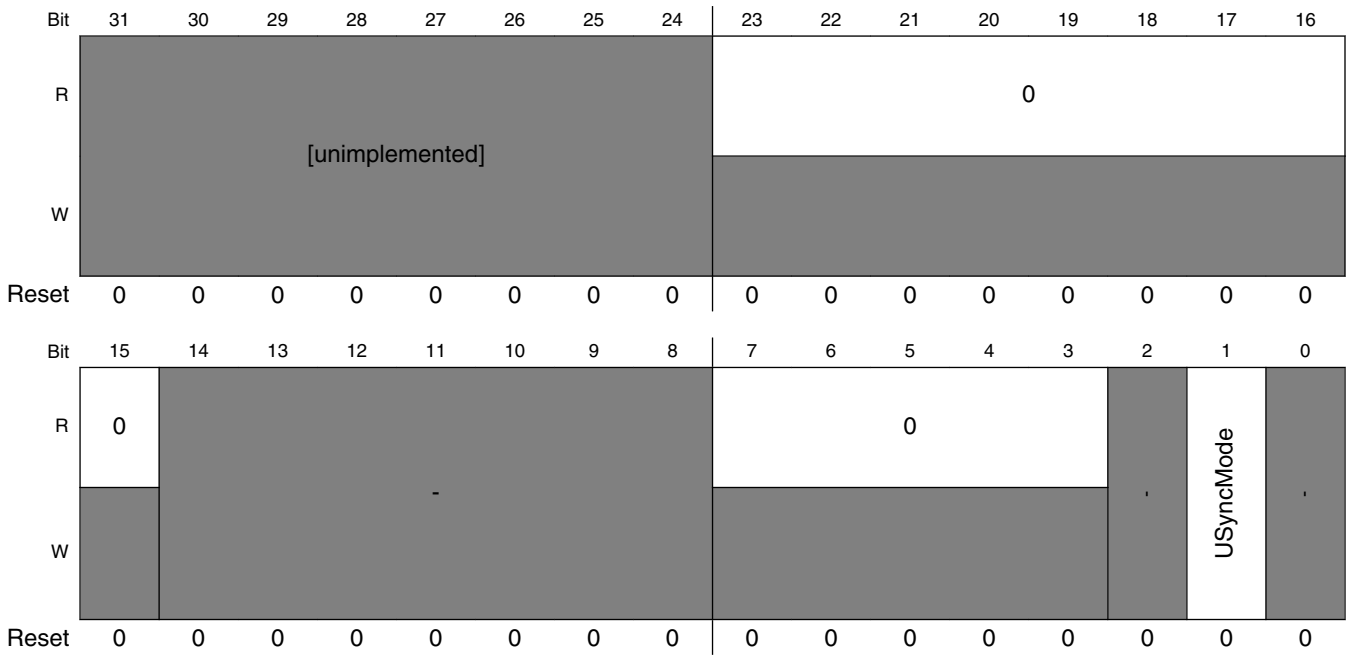
Table continues on the next page...

SPDIF_SCR field descriptions (continued)

Field	Description
	10 Empty interrupt if at most 8 sample in Tx left and right FIFOs 11 Empty interrupt if at most 12 sample in Tx left and right FIFOs
14 -	Reserved
13 Low-Power	When write 1 to this bit, it will cause SPDIF enter low-power mode. return 1 when SPDIF in Low-Power mode.
12 soft_reset	When write 1 to this bit, it will cause SPDIF software reset. The software reset will last 8 cycles. When in the reset process, return 1 when read. else return 0 when read.
11–10 TxFIFO_Ctrl	00 Send out digital zero on SPDIF Tx 01 Tx Normal operation 10 Reset to 1 sample remaining 11 Reserved
9 DMA_Rx_En	DMA Receive Request Enable (RX FIFO full)
8 DMA_TX_En	DMA Transmit Request Enable (Tx FIFO empty)
7–6 -	Reserved
5 ValCtrl	0 Outgoing Validity always set 1 Outgoing Validity always clear
4–2 TxSel	000 Off and output 0 001 Feed-through SPDIFIN 101 Tx Normal operation
1–0 USrc_Sel	00 No embedded U channel 01 U channel from SPDIF receive block (CD mode) 10 Reserved 11 U channel from on chip transmitter

69.4.2 CDText Control Register (SPDIF_SRCD)

Address: SPDIF_SRCD is 5002_8000h base + 4h offset = 5002_8004h

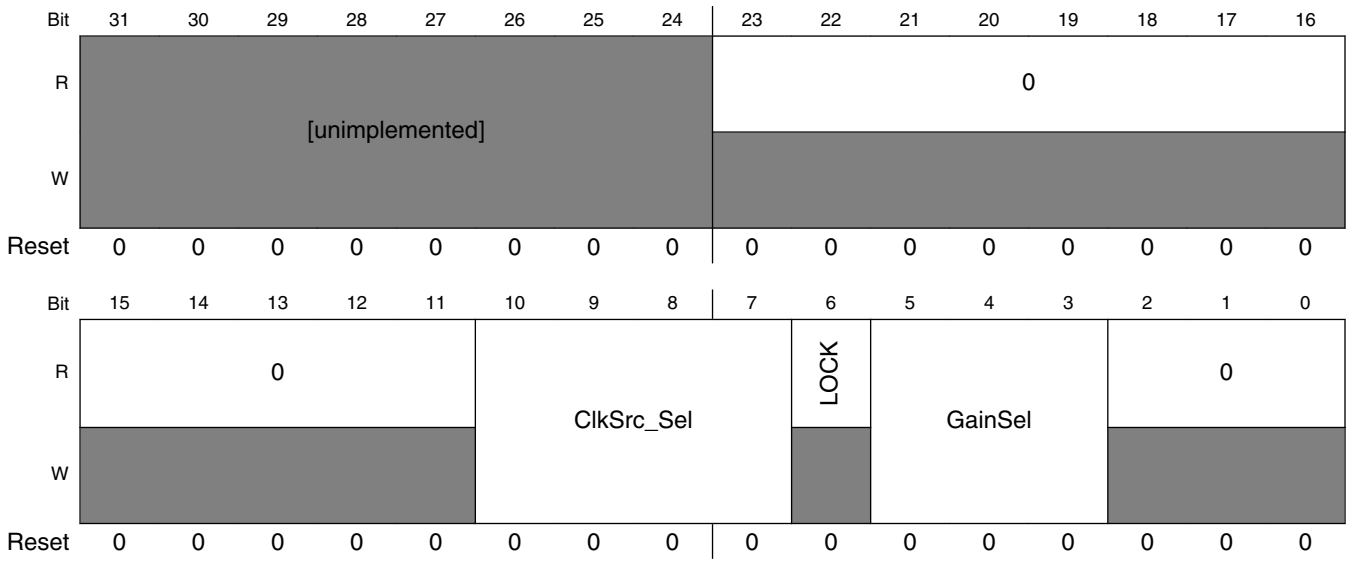


SPDIF_SRCD field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–15 Reserved	This read-only field is reserved and always has the value zero. Return zeros when read
14–8 -	Reserved
7–3 Reserved	This read-only field is reserved and always has the value zero. Return zeros when read
2 -	Reserved
1 USyncMode	0 Non-CD data 1 CD user channel subcode
0 -	Reserved

69.4.3 PhaseConfig Register (SPDIF_SRPC)

Address: SPDIF_SRPC is 5002_8000h base + 8h offset = 5002_8008h



SPDIF_SRPC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–11 Reserved	This read-only field is reserved and always has the value zero. Reserved, return zeros when read
10–7 ClkSrc_Sel	Clock source selection, all other settings not shown are reserved: 0000 if (DPLL Locked) SPDIF_RxClk else extal 0001 if (DPLL Locked) SPDIF_RxClk else spdif_clk 0011 if (DPLL Locked) SPDIF_RxClk else asrc_clk 0100 if (DPLL Locked) SPDIF_Rxclk else esai_hckt 0101 extal_clk 0110 spdif_clk 1000 asrc_clk 1001 esai_hckt 1010 if (DPLL Locked) SPDIF_RxClk else mlb_clk 1011 if (DPLL Locked) SPDIF_RxClk else camp_clk 1100 mkb_clk 1101 camp_clk
6 LOCK	LOCK bit to show that the internal DPLL is locked, read only
5–3 GainSel	Gain selection: 000 24*2**10 001 16*2**10

Table continues on the next page...

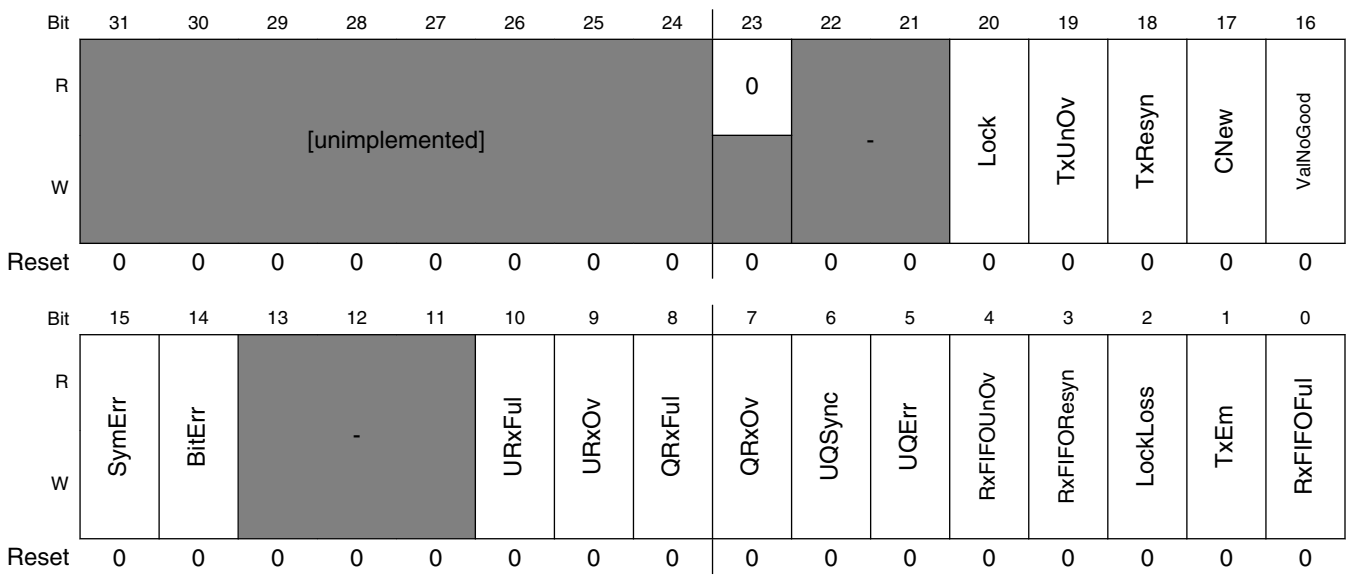
SPDIF_SRPC field descriptions (continued)

Field	Description
	010 12*2**10 011 8*2**10 100 6*2**10 101 4*2**10 110 3*2**10
2–0 Reserved	This read-only field is reserved and always has the value zero. Reserved, return zeros when read

69.4.4 InterruptEn Register (SPDIF_SIE)

The InterruptEn register (SPDIF_SIE) provides control over the enabling of interrupts.

Address: SPDIF_SIE is 5002_8000h base + Ch offset = 5002_800Ch



SPDIF_SIE field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23 Reserved	This read-only field is reserved and always has the value zero. Reserved, for InterruptStat/Clear return zeros when read, for InterruptEn, bit 23 also read zero
22–21 -	Reserved, for InterruptStat/Clear return zeros when read, for InterruptEn, bit 23 also read zero
20 Lock	SPDIF receiver's DPLL is locked
19 TxUnOv	SPDIF Tx FIFO under/overrun

Table continues on the next page...

SPDIF_SIE field descriptions (continued)

Field	Description
18 TxResyn	SPDIF Tx FIFO resync
17 CNew	SPDIF receive change in value of control channel
16 ValNoGood	SPDIF validity flag no good
15 SymErr	SPDIF receiver found illegal symbol
14 BitErr	SPDIF receiver found parity bit error
13–11 -	Reserved. Return zeros when read
10 URxFul	U Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from U Rx reg.
9 URxOv	U Channel receive register overrun
8 QRxFul	Q Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from Q Rx reg.
7 QRxOv	Q Channel receive register overrun
6 UQSync	U/Q Channel sync found
5 UQErr	U/Q Channel framing error
4 RxFIFOUnOv	Rx FIFO underrun/overrun
3 RxFIFOResyn	Rx FIFO resync
2 LockLoss	SPDIF receiver loss of lock
1 TxEm	SPDIF Tx FIFO empty, can't be cleared with reg. IntClear. To clear it, write toTx FIFO.
0 RxFIFOFul	SPDIF Rx FIFO full, can't be cleared with reg. IntClear. To clear it, read from Rx FIFO.

69.4.5 InterruptStat Register (SPDIF_SIS)

The InterruptStat (SPDIF_SIS) register is a read only register that provides the status on interrupt operations.

Address: SPDIF_SIS is 5002_8000h base + 10h offset = 5002_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	[unimplemented]								0			Lock	TxUnOv	TxResyn	CNew	ValNoGood
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SymErr	BitErr	0			URxFul	URxOv	QRxFul	QRxOv	UQSync	UQErr	RxFIFOUnOv	RxFIFOResyn	LockLoss	TxEm	RxFIFOFul
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

SPDIF_SIS field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–21 Reserved	This read-only field is reserved and always has the value zero. Reserved, for InterruptStat/Clear return zeros when read.
20 Lock	SPDIF receiver's DPLL is locked
19 TxUnOv	SPDIF Tx FIFO under/overrun
18 TxResyn	SPDIF Tx FIFO resync
17 CNew	SPDIF receive change in value of control channel

Table continues on the next page...

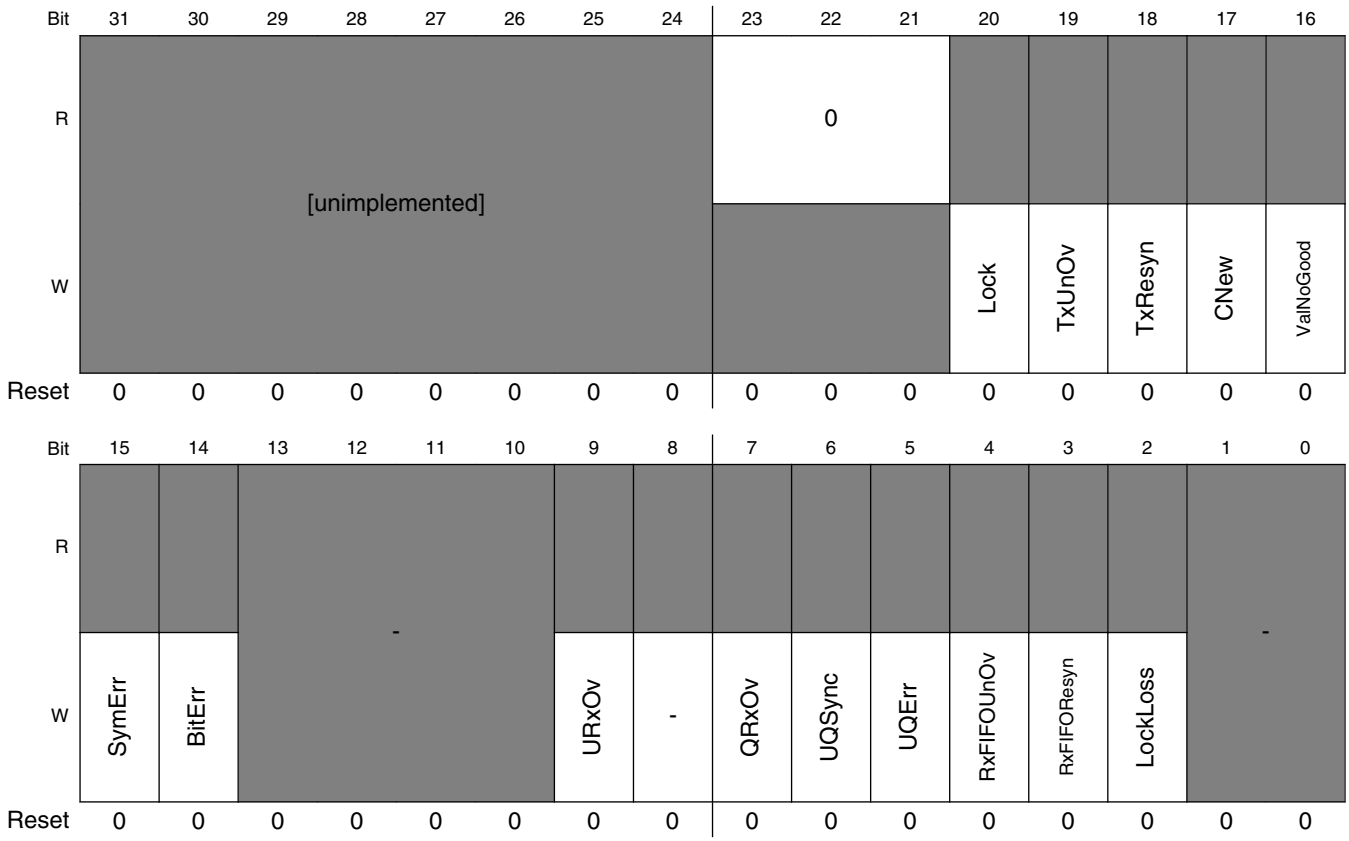
SPDIF_SIS field descriptions (continued)

Field	Description
16 ValNoGood	SPDIF validity flag no good
15 SymErr	SPDIF receiver found illegal symbol
14 BitErr	SPDIF receiver found parity bit error
13–11 Reserved	This read-only field is reserved and always has the value zero. Reserved. Return zeros when read
10 URxFul	U Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from U Rx reg.
9 URxOv	U Channel receive register overrun
8 QRxFul	Q Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from Q Rx reg.
7 QRxOv	Q Channel receive register overrun
6 UQSync	U/Q Channel sync found
5 UQErr	U/Q Channel framing error
4 RxFIFOUnOv	Rx FIFO underrun/overrun
3 RxFIFOResyn	Rx FIFO resync
2 LockLoss	SPDIF receiver loss of lock
1 TxEm	SPDIF Tx FIFO empty, can't be cleared with reg. IntClear. To clear it, write toTx FIFO.
0 RxFIFOFul	SPDIF Rx FIFO full, can't be cleared with reg. IntClear. To clear it, read from Rx FIFO.

69.4.6 InterruptClear Register (SPDIF_SIC)

The InterruptClear (SPDIF_SIC) register is a write only register and is used to clear interrupts.

Address: SPDIF_SIC is 5002_8000h base + 10h offset = 5002_8010h



SPDIF_SIC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–21 Reserved	This read-only field is reserved and always has the value zero. Reserved, for InterruptStat/Clear return zeros when read.
20 Lock	SPDIF receiver's DPLL is locked
19 TxUnOv	SPDIF Tx FIFO under/overrun
18 TxResyn	SPDIF Tx FIFO resync
17 CNew	SPDIF receive change in value of control channel

Table continues on the next page...

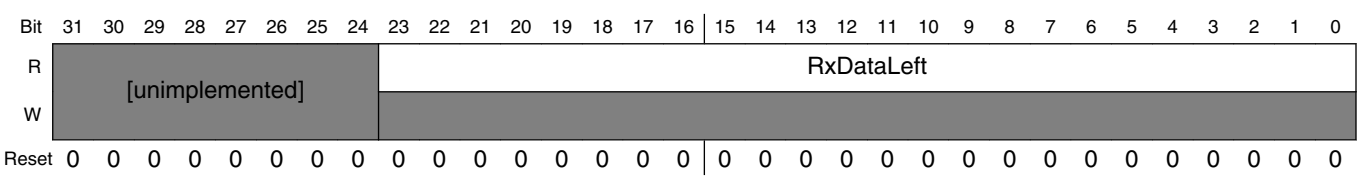
SPDIF_SIC field descriptions (continued)

Field	Description
16 ValNoGood	SPDIF validity flag no good
15 SymErr	SPDIF receiver found illegal symbol
14 BitErr	SPDIF receiver found parity bit error
13–10 -	Reserved.
9 URxOv	U Channel receive register overrun
8 -	Reserved
7 QRxOv	Q Channel receive register overrun
6 UQSync	U/Q Channel sync found
5 UQErr	U/Q Channel framing error
4 RxFIFOUnOv	Rx FIFO underrun/overrun
3 RxFIFOResyn	Rx FIFO resync
2 LockLoss	SPDIF receiver loss of lock
1–0 -	Reserved.

69.4.7 SPDIFRxLeft Register (SPDIF_SRL)

SPDIFRxLeft register is an audio data reception register.

Address: SPDIF_SRL is 5002_8000h base + 14h offset = 5002_8014h



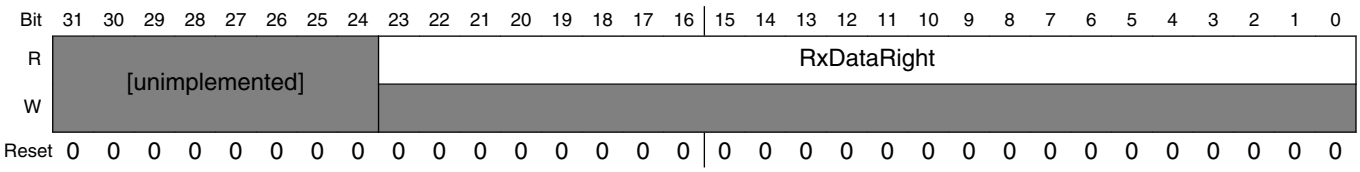
SPDIF_SRL field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–0 RxDataLeft	Processor receive SPDIF data left

69.4.8 SPDIFRxRight Register (SPDIF_SRR)

SPDIFRxRight register is an audio data reception register.

Address: SPDIF_SRR is 5002_8000h base + 18h offset = 5002_8018h



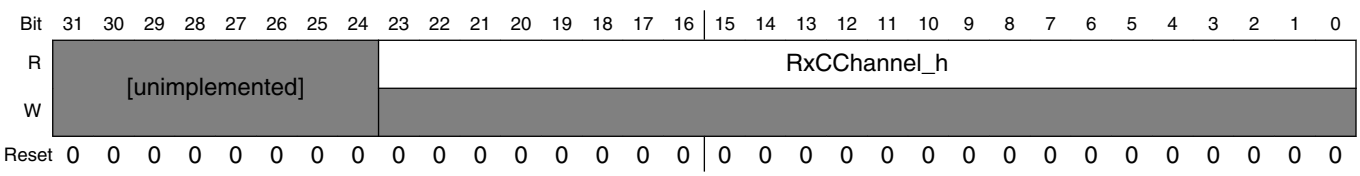
SPDIF_SRR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–0 RxDataRight	Processor receive SPDIF data right

69.4.9 SPDIFRxCChannel_h Register (SPDIF_SRCSH)

SPDIFRxCChannel_h register is a channel status reception register.

Address: SPDIF_SRCSH is 5002_8000h base + 1Ch offset = 5002_801Ch



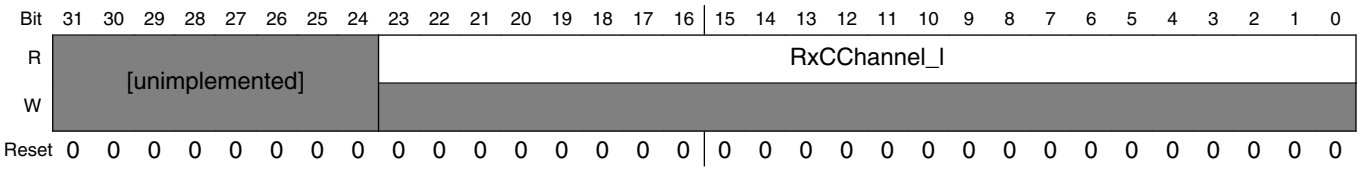
SPDIF_SRCSH field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–0 RxCChannel_h	SPDIF receive C channel register, contains first 24 bits of C channel without interpretation

69.4.10 SPDIFRxCChannel_l Register (SPDIF_SRCSL)

SPDIFRxCChannel_l register is a channel status reception register.

Address: SPDIF_SRCSL is 5002_8000h base + 20h offset = 5002_8020h



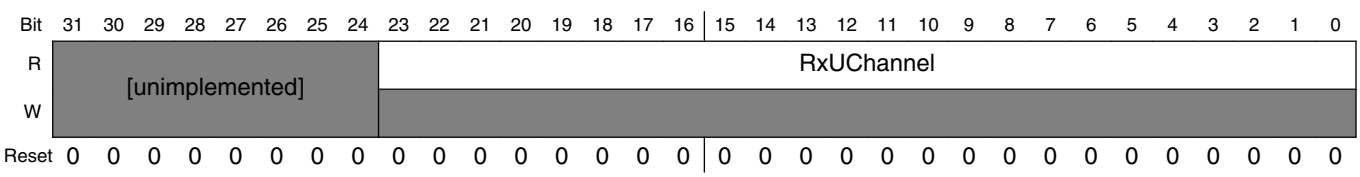
SPDIF_SRCSL field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–0 RxCChannel_l	SPDIF receive C channel register, contains next 24 bits of C channel without interpretation

69.4.11 UchannelRx Register (SPDIF_SRU)

UChannelRx register is a user bits reception register.

Address: SPDIF_SRU is 5002_8000h base + 24h offset = 5002_8024h



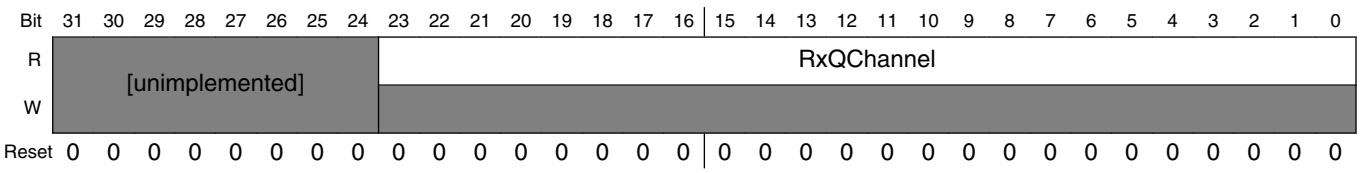
SPDIF_SRU field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–0 RxUChannel	SPDIF receive U channel register, contains next 3 U channel bytes

69.4.12 QchannelRx Register (SPDIF_SRQ)

QChannelRx register is a user bits reception register.

Address: SPDIF_SRQ is 5002_8000h base + 28h offset = 5002_8028h



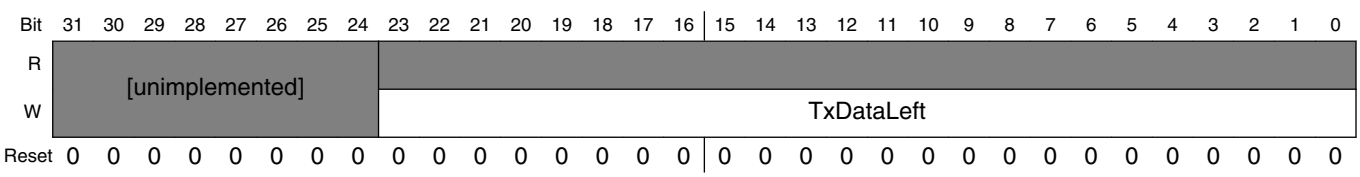
SPDIF_SRQ field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–0 RxQChannel	SPDIF receive Q channel register, contains next 3 Q channel bytes

69.4.13 SPDIFTxLeft Register (SPDIF_STL)

SPDIFTxLeft register is an audio data transmission register.

Address: SPDIF_STL is 5002_8000h base + 2Ch offset = 5002_802Ch



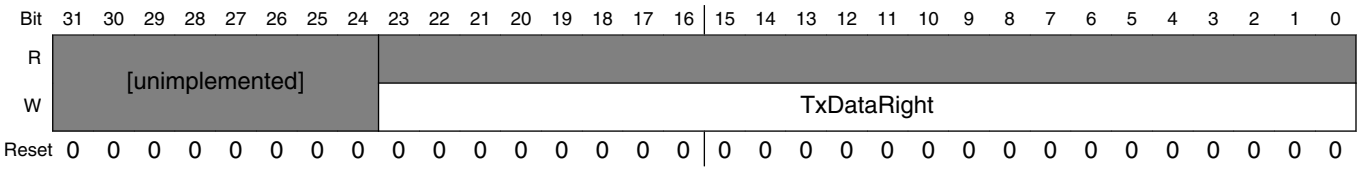
SPDIF_STL field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–0 TxDataLeft	SPDIF transmit left channel data. It is write-only, and always returns zeros when read

69.4.14 SPDIFTxRight Register (SPDIF_STR)

SPDIFTxRight register is an audio data transmission register.

Address: SPDIF_STR is 5002_8000h base + 30h offset = 5002_8030h



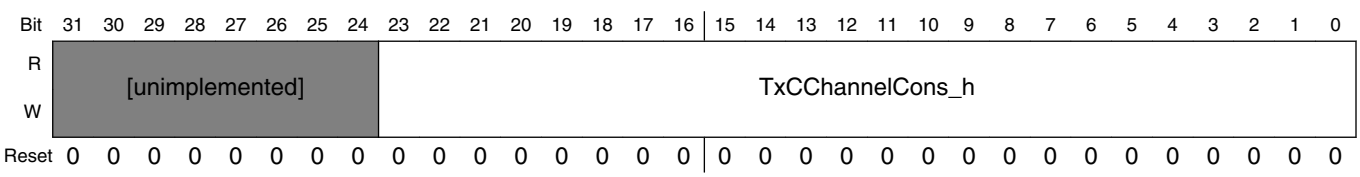
SPDIF_STR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–0 TxDataRight	SPDIF transmit right channel data. It is write-only, and always returns zeros when read

69.4.15 SPDIFTxCChannelCons_h Register (SPDIF_STCSCH)

SPDIFTxCChannelCons_h register is a channel status transmission register.

Address: SPDIF_STCSCH is 5002_8000h base + 34h offset = 5002_8034h



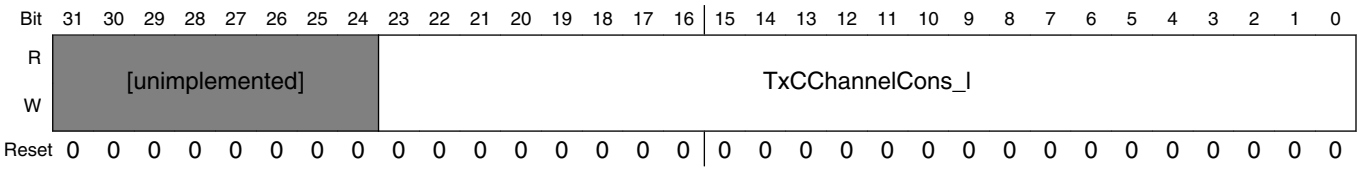
SPDIF_STCSCH field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–0 TxCChannelCons_h	SPDIF transmit Cons. C channel data, contains first 24 bits without interpretation. When read, it returns the latest data written by the processor

69.4.16 SPDIFTxCChannelCons_l Register (SPDIF_STCSCL)

SPDIFTxCChannelCons_l register is a channel status transmission register.

Address: SPDIF_STCSCL is 5002_8000h base + 38h offset = 5002_8038h

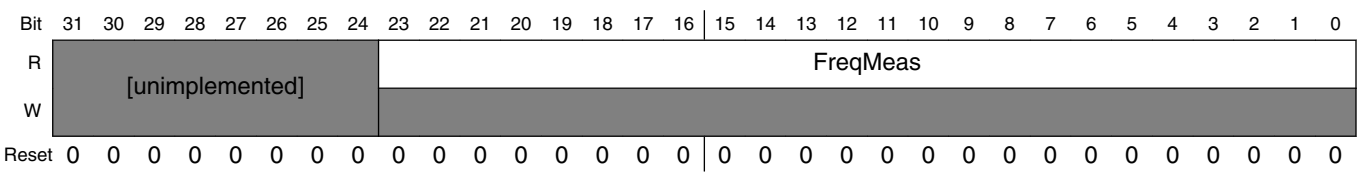


SPDIF_STCSCL field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–0 TxCChannelCons_l	SPDIF transmit Cons. C channel data, contains next 24 bits without interpretation. When read, it returns the latest data written by the processor

69.4.17 FreqMeas Register (SPDIF_SRFM)

Address: SPDIF_SRFM is 5002_8000h base + 44h offset = 5002_8044h



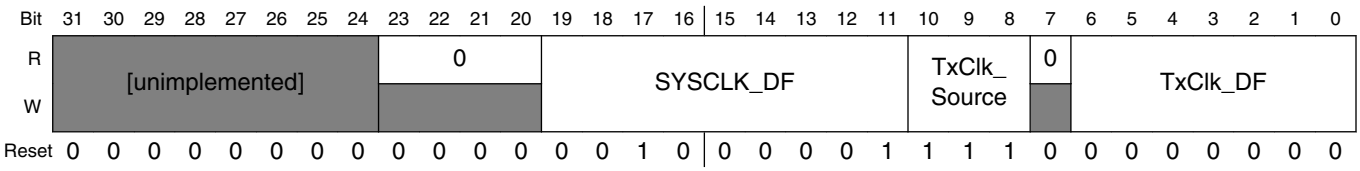
SPDIF_SRFM field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–0 FreqMeas	Frequency measurement data

69.4.18 SPDIFTxClk Register (SPDIF_STC)

The SPDIFTxClk Control register includes the means to select the transmit clock and frequency division.

Address: SPDIF_STC is 5002_8000h base + 50h offset = 5002_8050h



SPDIF_STC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.
23–20 Reserved	This read-only field is reserved and always has the value zero. Reserved, return zeros when read
19–11 SYSCLK_DF	system clock divider factor, 2~512. 0 no clock signal 1 divider factor is 2 511 divider factor is 512
10–8 TxClk_Source	000 camp1 clk input 001 CCM spdif0_clk_root input 011 asrc_clk input 100 esai_hckt input 101 frequency divided ipg_clk input 110 mlb_clk input 111 camp2 clk input
7 Reserved	This read-only field is reserved and always has the value zero. Reserved, return zero when read
6–0 TxClk_DF	Divider factor (1-128) 0 divider factor is 1 1 divider factor is 2 127 divider factor is 128

Chapter 70
System Reset Controller (SRC)

70.1 Introduction

SRC generates all reset signals for each block within i.MX53.

70.1.1 Overview

The System Reset Controller (SRC) generates all functional block reset signals. It also determines the boot image device by decoding the BOOT_MODE[1:0] value and the values of several BOOT CFG fields in the register. The reset control determines the source and the type of reset, such as POR, WARM, COLD, and performs the necessary reset qualification and stretching sequences. Based on the type of reset, the reset logic generates the reset sequence for the entire SoC. On initial power on, RESET_B pin signal is asserted (low) and the entire chip is reset.

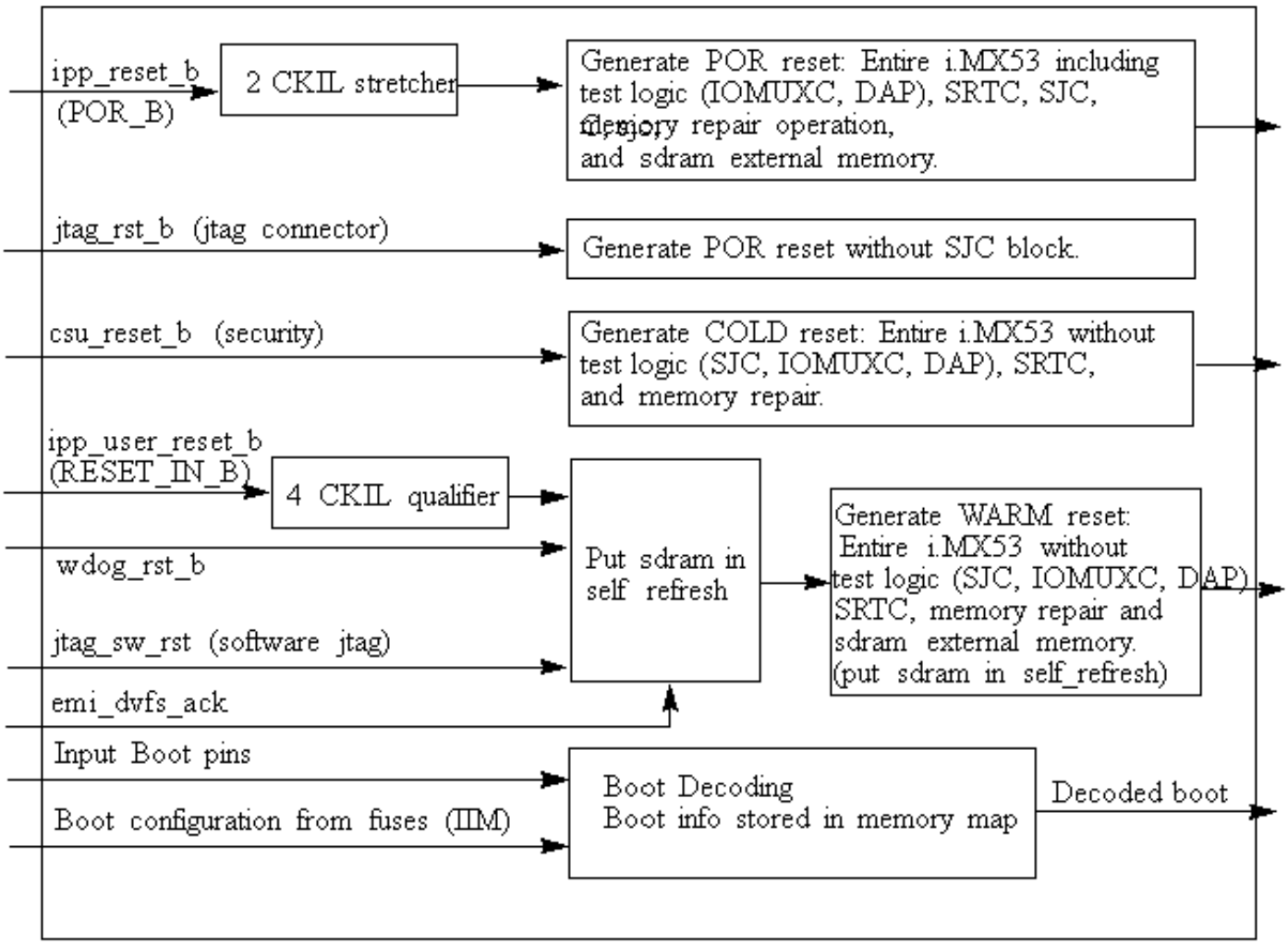


Figure 70-1. SRC High Level diagram

70.1.2 Features

The SRC includes the following features.

- Receives and handles the resets from all the reset sources
- Reset the appropriate domains based upon the resets sources and the nature of the reset.
- Latches the BOOT_MODE pins and common configuration signals from the internal fuse
- The SRC has 32-bit IP Bus interface

70.2 Programmable Registers

SRC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FD_0000	SRC Control Register (SRC_SCR)	32	R/W	0000_0D21h	70.2.1/ 4355
53FD_0004	SRC Boot Mode Register (SRC_SBMR)	32	R	0000_0000h	70.2.2/ 4357
53FD_0008	SRC Reset Status Register (SRC_SRSR)	32	R/W	0000_0001h	70.2.3/ 4358
53FD_0014	SRC Interrupt Status Register (SRC_SISR)	32	R	0000_0000h	70.2.4/ 4360
53FD_0018	SRC Interrupt Mask Register (SRC_SIMR)	32	R/W	0000_000Fh	70.2.5/ 4361

70.2.1 SRC Control Register (SRC_SCR)

Address: SRC_SCR is 53FD_0000h base + 0h offset = 53FD_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				weim_rst	mask_wdog_rst				warm_rst_ bypass_ count		sw_open_ vg_rst	sw_ipu_rst	sw_vpu_rst	sw_gpu_rst	warm_reset_ enable
W																
Reset	0	0	0	0	1	1	0	1	0	0	1	0	0	0	0	1

SRC_SCR field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11 weim_rst	EIM reset is needed in order to reconfigure the EIM chip select. The software reset bit must deasserted.

Table continues on the next page...

SRC_SCR field descriptions (continued)

Field	Description
	The EIM chip select configuration should be updated. The software bit must be re-asserted since this is not self-refresh.
10–7 mask_wdog_rst	Mask wdog_rst_b source. If these 4 bits are coded from A to 5 then, wdog_rst_b input to SRC will be masked and wdog_rst_b will not create reset to the IC. **NOTE:** any other code will be coded to 1010 i.e. wdog_rst_b is not masked During the time the WDOG event is masked using SRC logic, it is likely that WDOG Reset Status Register (WRSR) bit 1 (which indicates WDOG timeout event) will get asserted. SW / OS developer must prepare for this case. Re-enable WDOG is possible, by un-mask it in SRC, though it must be preceded by servicing the WDOG. However, for the case that the event has been asserted, the status bit (WRSR bit-1) will remain asserted, regardless of servicing the WDOG. (HW reset is the only mean to cause deassertion of that bit). 0101 wdog_rst_b is masked 1010 wdog_rst_b is not masked (default)
6–5 warm_rst_ bypass_count	Defines the CKIL cycles to count before bypassing the EXTMC acknowledge for warm reset. If the EXTMC acknowledge will not be asserted before this counter has elapsed, then a cold reset will be initiated. 00 Counter not to be used - system will wait until EXTMC acknowledge until it is asserted. 01 Wait 16 CKIL cycles before changing warm reset to a cold reset. 10 Wait 32 CKIL cycles before changing warm reset to a cold reset. 11 Wait 64 CKIL cycles before changing warm reset to a cold reset
4 sw_open_vg_rst	Software reset for open_vg **NOTE:** this is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. Software can determine that the reset has finished once this bit is cleared. Software can also configure SRC to generate interrupt once the software has finished. Please refer to SISR register for details. **NOTE:** the reset process will involve 8 open_vg cycles before negating the open_vg reset, to allow reset assertion to propagate into open_vg. 0 do not assert open_vg reset 1 assert open_vg reset
3 sw_ipu_rst	Software reset for IPU **NOTE:** this is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. Software can determine that the reset has finished once this bit is cleared. Software can also configure SRC to generate interrupt once the software has finished. Please refer to SISR register for details. 0 do not assert IPU reset 1 assert IPU reset
2 sw_vpu_rst	Software reset for VPU **NOTE:** this is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. Software can determine that the reset has finished once this bit is cleared. Software can also configure SRC to generate interrupt once the software has finished. Please refer to SISR register for details.

Table continues on the next page...

SRC_SCR field descriptions (continued)

Field	Description
	NOTE: the reset process will involve 8 VPU cycles before negating the VPU reset, to allow reset assertion to propagate into VPU. 0 do not assert VPU reset 1 assert VPU reset
1 sw_gpu_rst	Software reset for GPU **NOTE:** this is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. Software can determine that the reset has finished once this bit is cleared. Software can also configure SRC to generate interrupt once the software has finished. Please referto SISR register for details. **NOTE:** the reset process will involve 8 GPU cycles before negating the GPU reset, to allow reset assertion to propagate into GPU. 0 do not assert GPU reset 1 assert GPU reset
0 warm_reset_ enable	Warm reset enable bit. Warm reset will be enabled only if warm_reset_enable bit is set. Otherwise all warm reset sources will generate cold reset. 0 Warm reset disabled 1 Warm reset enabled

70.2.2 SRC Boot Mode Register (SRC_SBMR)

The figure below presents the Boot Mode register which contains bits that reflect the status of boot Mode Pins of the chip. The default values for those bits depends on the values of pins/fuses during reset sequence .

Address: SRC_SBMR is 53FD_0000h base + 4h offset = 53FD_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-		TEST_MODE[2:0]			BT_FUSE_SEL	BMOD[1:0]		BOOT_CFG3[7:0]							
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BOOT_CFG2[7:0]								BOOT_CFG1[7:0]							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_SBMR field descriptions

Field	Description
31–30 -	
29–27 TEST_ MODE[2:0]	Reserved for test purposes.
26 BT_FUSE_SEL	Please refer to i.MX53 fuse map.
25–24 BMOD[1:0]	BMOD[1:0] reflects the settings of the BOOT_MODE pins at exit from reset. BMOD[1:0] = {BOOT_MODE1, BOOT_MODE0}
23–16 BOOT_ CFG3[7:0]	Please refer to i.MX53 fuse map.
15–8 BOOT_ CFG2[7:0]	Please refer to i.MX53 fuse map.
7–0 BOOT_ CFG1[7:0]	Please refer to i.MX53 fuse map.

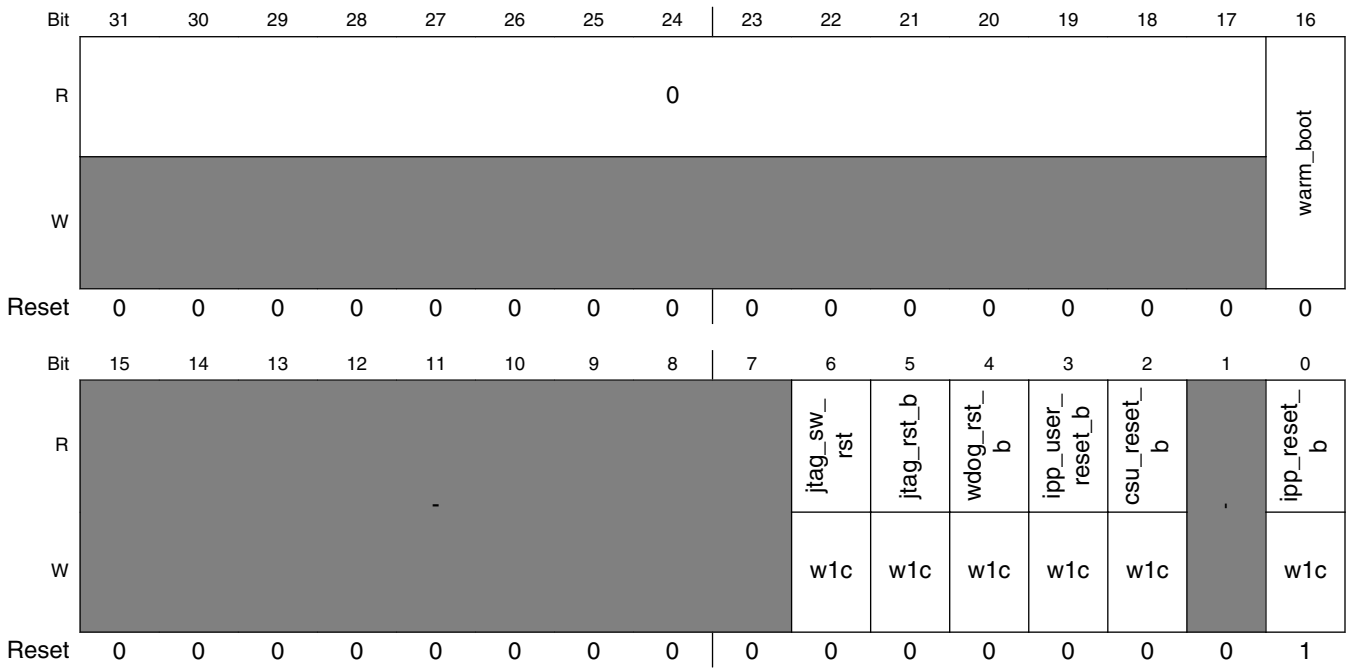
70.2.3 SRC Reset Status Register (SRC_SRSR)

The SRC_SRSR is a write to one clear register which records the source of the reset events for the chip. The SRC reset status register will capture all the reset sources that have occurred.

This register is reset on the rising edge of ipp_reset_b. This is a read-write register,

For bit[6-0] - writing zero does not have any effect. Writing one will clear the corresponding bit. The individual bits can be cleared by writing one to that bit. When the system comes out of reset, this register will have bits set corresponding to all the reset sources that occurred during system reset. Software has to take care to clear this register by writing one after every reset that occurs so that the register will contain the information of recently occurred reset.

Address: SRC_SRSR is 53FD_0000h base + 8h offset = 53FD_0008h



SRC_SRSR field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value zero. Reserved
16 warm_boot	Warm boot indication gives software capability to notify that warm boot was initiated by software. This indicates to the software that it saved the needed info in the memory before initiating the warm reset. In this case, software will set this bit to '1', before initiating the warm reset. The warm_boot bit should be used as indication only after a warm_reset sequence. Software should clear this bit after warm_reset to indicate that next warm_reset is not done with warm_boot. Please refer to Reset Sequence and Deassertion for details on warm_reset. 0 warm boot process not initiated by software. 1 warm boot initiated by software.
15–7 -	Reserved
6 jtag_sw_rst	JTAG SW reset. Indicates whether the reset was the result of software reset from JTAG. Connections at chip-level: jtag_sw_rst -> sjc_gpccr_reg_2_b 0 Reset is not a result of software reset from JTAG. 1 Reset is a result of software reset from JTAG.
5 jtag_rst_b	HIGH - Z JTAG reset. Indicates whether the reset was the result of HIGH-Z reset or reset from JTAG. Connections at chip-level: jtag_rst_b -> sjc_ieee_reset_b 0 Reset is not a result of HIGH-Z reset from JTAG. 1 Reset is a result of HIGH-Z reset from JTAG.

Table continues on the next page...

SRC_SRSR field descriptions (continued)

Field	Description
4 wdog_rst_b	IC Watchdog Time-out reset. Indicates whether the reset was the result of the watchdog time-out event. 0 Reset is not a result of the watchdog time-out event. 1 Reset is a result of the watchdog time-out event.
3 ipp_user_reset_b	Indicates whether the reset was the result of the ipp_user_reset_b qulified reset. 0 Reset is not a result of the ipp_user_reset_b qulified as COLD event. 1 Reset is a result of the ipp_user_reset_b qulified as COLD event.
2 csu_reset_b	Indicates whether the reset was the result of the csu_reset_b input. Note: If case the csu_reset_b occurred during a warm reset process, during the phase that ipg_clk is not available yet, then the occurrence of csu reset will not be reflected in this bit. 0 Reset is not a result of the csu_reset_b event. 1 Reset is a result of the csu_reset_b event.
1 -	Reserved.
0 ipp_reset_b	Indicates whether reset was the result of ipp_reset_b pin (Power-up sequence) 0 Reset is not a result of ipp_reset_b pin. 1 Reset is a result of ipp_reset_b pin.

70.2.4 SRC Interrupt Status Register (SRC_SISR)

Address: SRC_SISR is 53FD_0000h base + 14h offset = 53FD_0014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-[bit 12]															
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-[11:0]												open_vg_passed_reset	ipu_passed_reset	vpu_passed_reset	gpu_passed_reset
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_SISR field descriptions

Field	Description
31–4 -	Reserved
3 open_vg_ passed_reset	Interrupt generated to indicate that open_vg passed software reset and is ready to be used 0 - interrupt generation not due to open_vg passed reset 1 - interrupt generation due to open_vg passed reset
2 ipu_passed_ reset	Interrupt generated to indicate that IPU passed software reset and is ready to be used 0 - interrupt generation not due to IPU passed reset 1 - interrupt generation due to IPU passed reset
1 vpu_passed_ reset	Interrupt generated to indicate that VPU passed software reset and is ready to be used 0 - interrupt generation not due to VPU passed reset 1 - interrupt generation due to VPU passed reset
0 gpu_passed_ reset	Interrupt generated to indicate that GPU passed software reset and is ready to be used 0 - interrupt generation not due to GPU passed reset 1 - interrupt generation due to GPU passed reset

70.2.5 SRC Interrupt Mask Register (SRC_SIMR)

Address: SRC_SIMR is 53FD_0000h base + 18h offset = 53FD_0018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-[bit 12]															
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-[11:0]												mask_open_vg_passed_reset	mask_ipu_passed_reset	mask_vpu_passed_reset	mask_gpu_passed_reset
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

SRC_SIMR field descriptions

Field	Description
31–4 -	Reserved
3 mask_open_vg_ passed_reset	mask interrupt generation due to open_vg passed reset

Table continues on the next page...

SRC_SIMR field descriptions (continued)

Field	Description
	0 - don't mask interrupt due to open_vg passed reset - interrupt will be created 1 - mask interrupt due to open_vg passed reset
2 mask_ipu_ passed_reset	mask interrupt generation due to IPU passed reset 0 - don't mask interrupt due to IPU passed reset - interrupt will be created 1 - mask interrupt due to IPU passed reset
1 mask_vpu_ passed_reset	mask interrupt generation due to VPU passed reset 0 - don't mask interrupt due to VPU passed reset - interrupt will be created 1 - mask interrupt due to VPU passed reset
0 mask_gpu_ passed_reset	mask interrupt generation due to GPU passed reset 0 - don't mask interrupt due to GPU passed reset - interrupt will be created 1 - mask interrupt due to GPU passed reset

70.3 Functional Description

70.3.1 Reset Control

This section details the reset control of this SoC device.

70.3.1.1 Reset Inputs and Outputs

The SRC receives reset requests from all the potential reset sources. All the immediate sources of reset are directly passed to the reset stretching block whereas the resets which require qualification are passed on to the reset qualification logic before they are sent to the reset stretching block.

All reset inputs and outputs are described in the following figure:

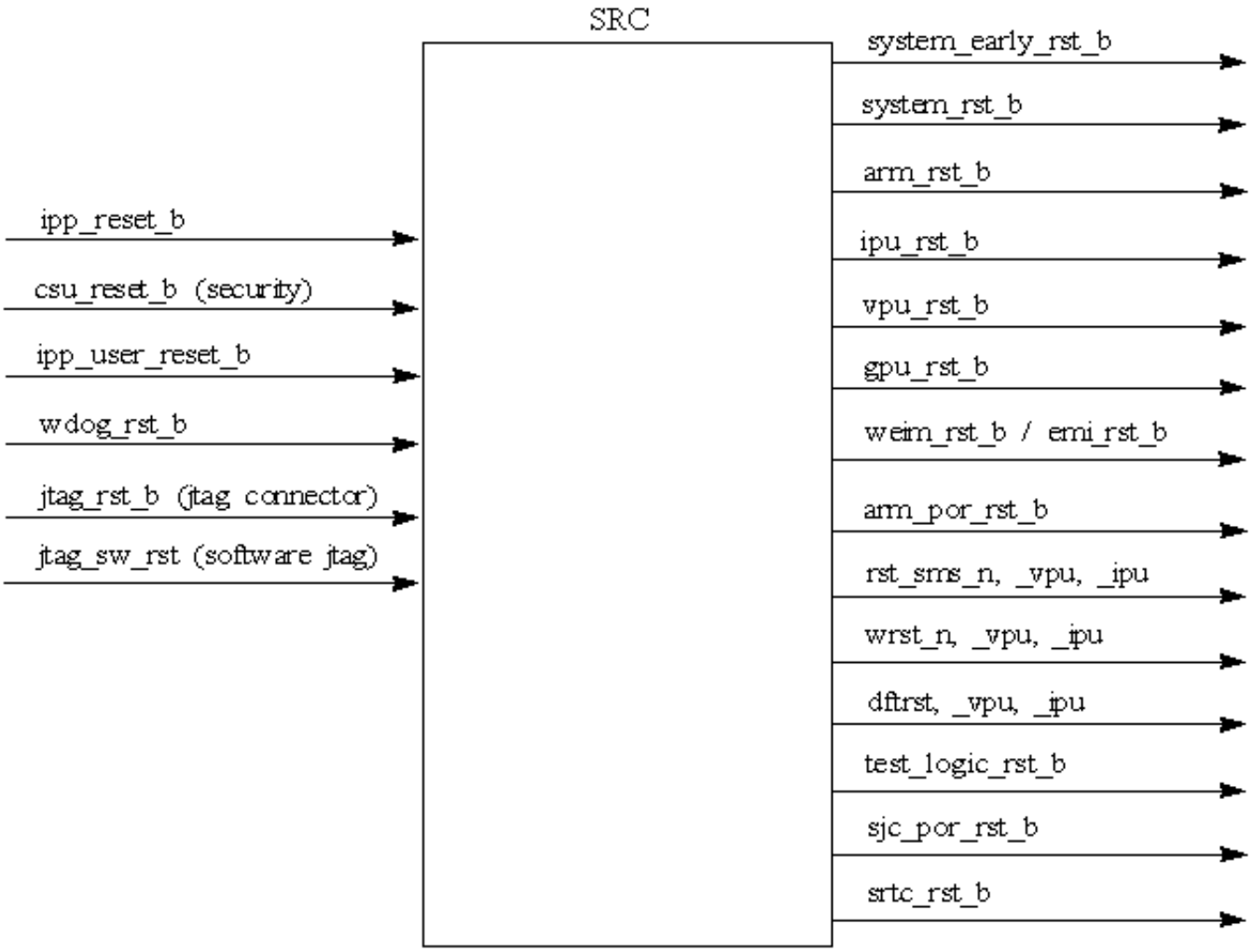


Figure 70-7. SRC Inputs and Outputs

The reset sources, type and behavior are shown in the Table 70-7. As there is no chip POR the $\overline{\text{ipp\_reset\_b}}$ is used to reset the entire chip including test logic and JTAG blocks.

NOTE

All resets are expected to be active low except jtag_sw_rst.

Table 70-7. Reset Priorities, Sources, Types, and Behavior

Priority	Sources	Type	Behavior
4.	$\overline{\text{ipp\_reset\_b}}$	POR	Reset IC = $\overline{\text{system\_early\_rst\_b}}$ + $\overline{\text{system\_rst\_b}}$ + arm_por_rst+ $\overline{\text{arm\_soc\_rst\_b}}$ + $\overline{\text{arm\_dbg\_rst\_b}}$ + $\overline{\text{ipu\_rst\_b}}$ + $\overline{\text{vpu\_rst\_b}}$ + $\overline{\text{gpu\_rst\_b}}$ + $\overline{\text{open\_vg\_rst\_b}}$ + $\overline{\text{emi\_rst\_b}}$ + $\overline{\text{rst\_sms\_n}}$ + $\overline{\text{wrst}}$ + $\overline{\text{dftrst}}$ + $\overline{\text{test\_logic\_rst\_b}}$ + $\overline{\text{srtc\_rst\_b}}$ + $\overline{\text{sjc\_por\_rst\_b}}$ Note: ipp_reset_b is connected to POR_B pin.

Table continues on the next page...

Table 70-7. Reset Priorities, Sources, Types, and Behavior (continued)

Priority	Sources	Type	Behavior
5.	csu_reset_b	Cold	Reset IC - rst_sms_n - wrst - dftrst - test_logic_rst_b - srtc_rst_b - sjc_por_rst_b - arm_por_rst - arm_dbg_rst_b = system_early_rst_b + system_rst_b + arm_soc_rst_b + ipu_rst_b + vpu_rst_b + gpu_rst_b + open_vg_rst_b + emi_rst_b
6.	ipp_user_reset_b (qualified 4 CKIL's)	Warm	Reset IC - rst_sms_n - wrst - dftrst - test_logic_rst_b - srtc_rst_b- arm_por_rst - arm_dbg_rst_b - sjc_por_rst_b + warm_reset_signal = system_early_rst_b + system_rst_b + arm_soc_rst_b + ipu_rst_b + vpu_rst_b + gpu_rst_b + open_vg_rst_b + emi_rst_b+ warm_reset_signal Note: ipp_user_reset_b is connected to RESET_IN_B pin.
7.	wdog_rst_b This reset will not be generated if mask_wdog_rst bits are coded to 5.	Warm	Reset IC - rst_sms_n - wrst - dftrst - test_logic_rst_b - srtc_rst_b - arm_por_rst - arm_dbg_rst_b - sjc_por_rst_b + warm_reset_signal = system_early_rst_b + system_rst_b + arm_soc_rst_b + ipu_rst_b + vpu_rst_b + gpu_rst_b + open_vg_rst_b + emi_rst_b+ warm_reset_signal
8.	jtag_rst_b (jtag connector)	POR - sjc_por_rst_b	Reset IC - sjc_por_rst_b = system_early_rst_b + system_rst_b + arm_por_rst+ arm_soc_rst_b + arm_dbg_rst_b + ipu_rst_b + vpu_rst_b + gpu_rst_b + open_vg_rst_b + emi_rst_b + rst_sms_n + wrst + dftrst + test_logic_rst_b + srtc_rst_b
9.	jtag_sw_rst (software jtag)	Warm	Reset IC - rst_sms_n - wrst - dftrst - test_logic_rst_b - srtc_rst_b - arm_por_rst - arm_dbg_rst_b + warm_reset_signal - sjc_por_rst_b = system_early_rst_b + system_rst_b + arm_rst_b + ipu_rst_b + vpu_rst_b + gpu_rst_b+ open_vg_rst_b + emi_rst_b+ warm_reset_signal

The reset priorities are POR (strongest), COLD and WARM (weakest). If a stronger reset is asserted during the sequence of a wicker reset, then the wicker sequence will be interfered and the stronger reset sequence will commence. There is no priority inside a reset type group. If a reset is asserted during reset sequence of reset from the same type group, then the reset sequence will be interfered and a new reset sequence (from the same type) will commence.

The following lists the functionality of each of these reset outputs:

- system_early_rst_b - Resets the system blocks that need to start first as ccm, pll_ip's, iim, fuseboxes, EXTMC etc.
- system_rst_b - Resets functional blocks.
- arm_rst_b - Resets ARM core (on regular system reset)
- ipu_rst_b - Resets IPU (on regular system reset)
- vpu_rst_b - Resets VPU (on regular system reset)
- gpu_rst_b - Resets GPU (on regular system reset)

- open_vg_rst_b - Resets Open_VG (on regular system reset)
- emi_rst_b - Resets SDRAM controller.
- arm_por_rst - Resets ARM por input.
- arm_soc_rst_b - Reset for arm SoC.
- arm_dbg_rst_b - Reset debug logic of ARM.
- test_logic_rst_b - Reset test logic. (IOMUXC, DAP)
- sjc_por_rst_b - reset to SJC.
- srtc_rst_b - Resets SRTC.

The following are related to SMS(STAR memory system):

- rst_sms_n - Resets SMS
- dftrst - Resets SMS DFT
- wrst - Resets JPC wrapper.

SRC will also generate rst_sms_n dftrst and wrst_n for specific accelerators. Since those blocks are not involved in the memory repair process, then they will be similar to the functional resets of those blocks. Note that dftrst is active high, hence its inverted version equals the functional reset.

- dftrst_ipu = wrst_n_ipu = rst_sms_n_ipu = ipu_rst_b
- dftrst_vpu = wrst_n_vpu = rst_sms_n_vpu = vpu_rst_b

NOTE

It is assumed that each reset source will deassert after its assertion, either due to reset generated to the system from SRC, or either by negation of the reset source (in case it came from an external source to the IC). In the later case, the reset source is assumed to be held for atleast 2 CKIL cycles so that SRC can sample it.

70.3.1.2 Reset Handling

70.3.1.2.1 Reset Qualification

The reset qualification logic qualifies the reset source before sending it out to the chip as a valid reset. Warm resets are in this category. All remaining reset sources are immediate resets and are acknowledged by the reset circuitry the moment they are asserted.

Warm resets are not immediate resets. Warm resets do reset the clock control module (CCM), the source of SDRAMC clock. So, if a Warm reset were to immediately reset the CCM, then the SDRAMC clock would be shut off and this may cause the SDRAM data to be lost. During normal mode of operation of the chip, the protocol that is followed

before shutting off the SDRAMC clock is that an emi_dvfs_req signal is sent to SDRAMC and only after the SDRAMC sends an acknowledge signal, emi_dvfs_ack, are the clock to the SDRAMC gated off.

However, the implication here is that a valid warm reset source condition will not be able to cause a chip reset until the SDRAMC sends the acknowledge signal (emi_dvfs_ack). For example, a JTAG reset event has occurred but the JTAG reset will not be serviced until the emi_dvfs_ack signal is received. So, essentially all Warm reset sources depend on the SDRAMC providing the emi_dvfs_ack acknowledge signal before the reset is performed. When the SDRAM controller is not used, emi_dvfs_ack is defaulted high.

The occurrence of warm reset results in the assertion of warm_reset signal before the system resets are asserted to indicate EXTMC that the reset occurred is a warm reset.

A reset source is updated in the Reset status register when it becomes valid, provided it is asserted for the minimum amount of time after asserting. So, all immediate resets would be immediately updated in the Status register. Warm resets would be updated when the emi_dvfs_ack signal is received from the SDRAMC.

Once the reset is qualified, depending on the source of the reset, internal resets would be asserted appropriately.

70.3.1.2.2 Reset Sequence and Deassertion

The system_early_rst_b and the system_rst_b will assert immediately after any reset source is recognized (except for the case of warm reset when EXTMC needs to answer emi_dvfs_ack first). After all of the reset sources are released, the SRC will start the following set of events depending on the type of reset that occurred.

$\overline{\text{IPP\_RESET\_B}}$ (POR)

This is a external signal from the pads and whenever the chip is powered up the reset signal is passed through this pin indicating power-up sequence and the SRC resets the entire chip including test logic and JTAG. All SRC registers will be reset on POR sequence.

As soon as $\overline{\text{IPP\_RESET\_B}}$ occurs all resets are asserted and the entire chip is reset by SRC. The $\overline{\text{IPP\_RESET\_B}}$ is stretched for 2 CKIL cycles and the stretching sequence takes place after 2 CKIL clocks of $\overline{\text{IPP\_RESET\_B}}$ pin deassertion.

test_logic_rst_b, sjc_por_rst_b and srtc_rst_b signals are deasserted together with IPP_RESET_B signal. Those outputs are also deasserted after the stretching of IPP_RESET_B has deasserted.

After the above resets deasserts, system_early_rst_b reset is deasserted after 2 CKIL clock. The system_early_rst_b is used for the CCM and digital phase-locked loop's (DPLL) to start generating DPLL clock ouputs and the system root clocks.

Once system root clocks are ready, CCM will assert system_clk_ready signal. This signal is generated during the ignition sequence in CCM and it involves the preparation of CKIH/DPLL's to generate clock roots for functional operation. Please refer to Ignition Sequence in CCM chapter.

Once this signal arrives to SRC, it will initiate smart sfp operation. This operation will be finished once the SRC receives the assertion on signal "sfp_ready".

After SFP operation finish, SRC will enable IIM and FUSEBOX clocks, so that fuses can be loaded to IIM. IIM will notify with iim_ready_flag on the finish of the FUSEBOX loading.

SRC will prepare the boot information and then deassert emi_rst_b.

SRC will wait 8 ipg clock cycles, and then SRC will enable EXTMC clocks.

SRC will wait 8 CKIL clocks to allow EXTMC to generate fixed external clock to external memory SDRAM.

SRC will enable VPU and GPU clocks so that reset will penetrate this block.

After 8 ipg cycles, SRC will disable VPU, open_vg and GPU clocks.

After 8 ipg cycles, resets to all blocks will be deasserted (system_rst_b, vpu_rst_b, gpu_rst_b, open_vg_rst_b, ipu_rst_b).

If en_tester_control signal is asserted, SRC will wait for assertion of tester_ack signal to continue. This signal allows tester to receive determinism on the reset sequence.

After 8 ipg cycles, system clocks will be enabled (en_system_clk).

NOTE

The memory repair operation is bypassed when the smart_bypass signal is asserted.

COLD RESET

The sequence is similar to $\overline{\text{IPP\_RESET\_B}}$ except the memory repair operation is not performed and test_logic_reset is not fired.

Once the reset source deasserts, system_early_rst_b reset is deasserted after at least 2 CKIL clock. The system_early_rst_b is used for the CCM and DPLLCs to start generating DPLL clock ouputs and the system root clocks.

Once system root clocks are ready, CCM will assert system_clk_ready signal. This signal is generated during the ignition sequence in CCM and it involves the preparation of CKIH/DPLLs to generate clock roots for functional operation. Please refer to Ignition Sequence in CCM chapter.

Once this signal arrives to SRC, SRC will enable IIM and FUSEBOX clocks, so that fuses can be loaded to IIM. IIM will notify with iim_ready_flag on the finish of the FUSEBOX loading.

SRC will prepare the boot information and then deassert emi_rst_b.

SRC will wait 8 ipg clock cycles, and then SRC will enable EXTMC clocks.

SRC will wait 8 CKIL clocks to allow EXTMC to generate fixed external clock to external memory SDRAM.

SRC will enable VPU and GPU clocks so that reset will penetrate this block.

After 8 ipg cycles, src will disable VPU, open_vg and GPU clocks.

After 8 ipg cycles resets to all blocks will be deasserted (system_rst_b, vpu_rst_b, gpu_rst_b, open_vg_rst_b, ipu_rst_b).

If en_tester_control signal is asserted, SRC will wait for assertion of tester_ack signal to continue. This signal allows tester to receive determinism on the reset sequence.

After 8 ipg cycles, system clocks will be enabled (en_system_clk).

WARM RESET

WARM reset will be enabled only if warm_reset_enable bit is programed. Otherwise all warm reset sources will generate cold reset. This bit will be reset only by por reset.

A WARM reset is similar to a cold reset except that before the reset is fired a signal to EXTMC is asserted emi_dvfs_req (generates DVFS assertion to EXTMC) to request to prepare EXTMC to a WARM reset, i.e. to finish the transactions and to put the sdram in self refresh. Another signal will be asserted to EXTMC (warm_reset) that will wrap the warm reset sequence and will notify EXTMC that a warm reset is in process.

One of the sources of the WARM reset is ipp_user_reset_b. If this is the case, it is qualified for 4 CKIL edges. The WARM reset is initiated immediately after 4 CKIL edges. The system does not come out of reset until the ipp_user_reset_b is released.

During WARM reset, request to put SDRAM in self refresh is sent to SDRAMC. The request is sent threw generation of DVFS signal to EXTMC. The request is passed threw CCM to be combined with CCM's capability to request DVFS. Only after the EXTMC sends an acknowledge signal, emi_dvfs_ack, warm reset is generated. The implication here is that a valid WARM reset source condition will not be able to cause a chip reset

until the EXTMC sends the acknowledge signal (emi_dvfs_ack). For example, a WDOG reset event has occurred but the WDOG reset will not be serviced until the emi_dvfs_ack signal is received. So, essentially all WARM reset sources depend on the EXTMC providing the emi_dvfs_ack acknowledge signal before the reset is performed. When the EXTMC is not used, emi_dvfs_ack is defaulted high.

After the acknowledge signal, emi_dvfs_ack, is received the warm reset is generated. The warm_reset input to SDRAMC reflects the value of the warm_reset_enable bit. This signal indicates to SDRAMC that the upcoming reset is WARM.

In case the handshake mechanism with SDRAMC is stuck meaning no emi_dvfs_ack is received, cold reset will be generated after a number of CKIL clocks. The number of CKIL clocks is defined in register SCR bit warm_rst_bypass_count. the default of this bit is 16 CKIL count.

The following is a basic description of the warm reset sequence:

ARM sets warm_reset_enable bit to enable the WARM Reset functionality. If this bit is not set, all WARM reset sources will result in Cold Reset.

Assertion of one of the WARM reset sources.

The reset source is qualified in the SRC.

If emi_dvfs_ack signal is low, then SRC triggers the SDRAM Ctrl. to switch to self refresh mode using emi_dvfs_req signal. This is done threw CCM to combine with the DVFS sent from CCM in case of frequency change of EXTMC.

Wait for emi_dvfs_ack signal from the EXTMC. If no ack is received during warm_rst_bypass_count number of CKIL clocks, cold reset will be generated.

Assert warm_reset signal to EXTMC.

SRC asserts system resets

The deassertion sequence is exactly the same as in the Cold Reset except waiting for 8 CKIL's for EXTMC to generate fixed external clock to external memory SDRAM. This stage is not needed in WARM reset since SDRAM is held in self refresh in WARM reset and there is no need to reconfigure it when going out of WARM reset.

WARM BOOT

Software can save any needed information in the memory before initiating a warm reset. In this case, software will set warm_boot bit before initiating warm reset. After the system returns to run mode, the warm_boot bit will still be set, indicating the software that data was saved in memory and can be reused.

Figure 70-8, Figure 70-9, Figure 70-10, Figure 70-11 shows the behavior of different resets for POR, Cold and Warm reset cases respectively.

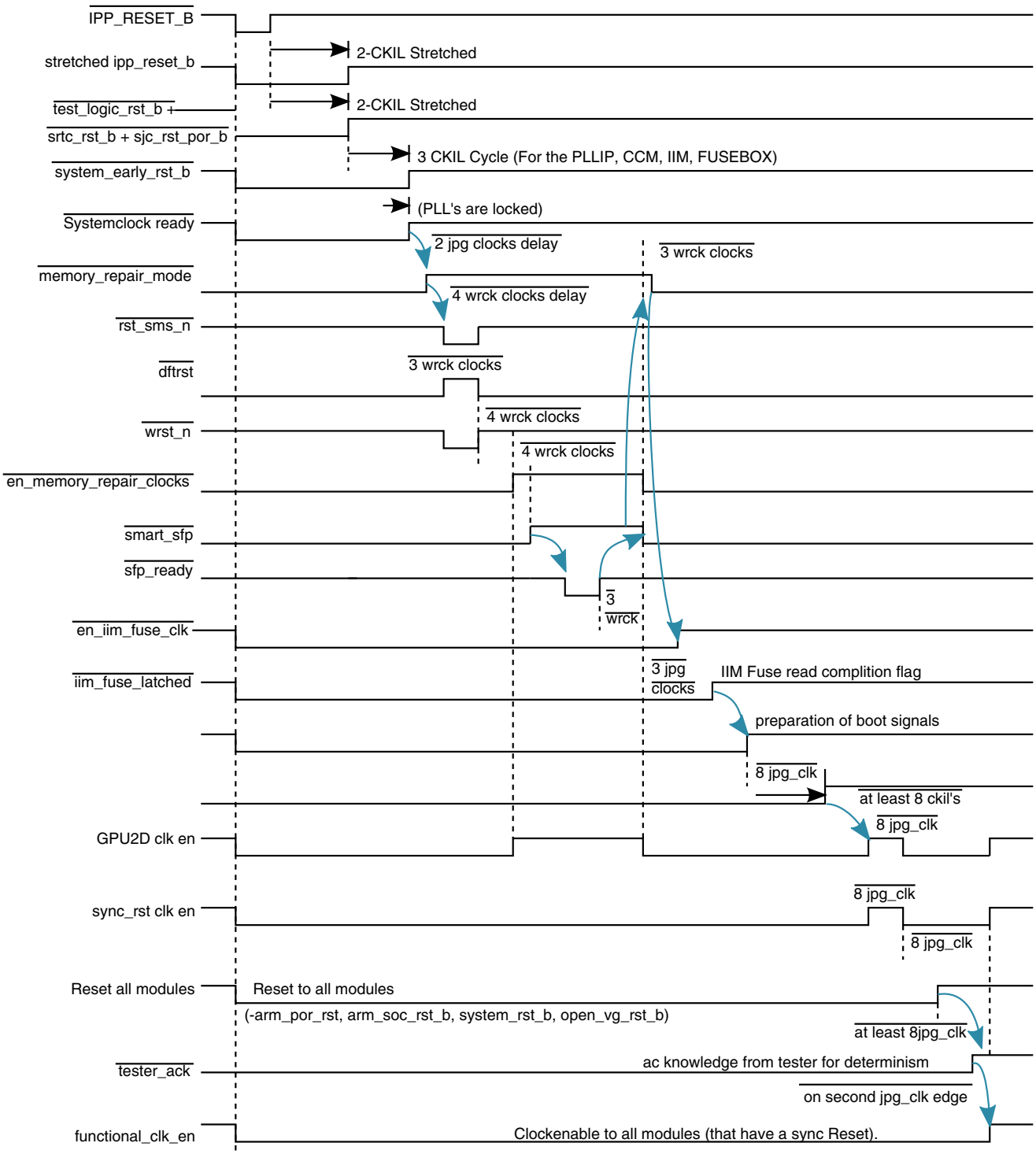


Figure 70-8. Behavior of IPP_RESET_B(POR)

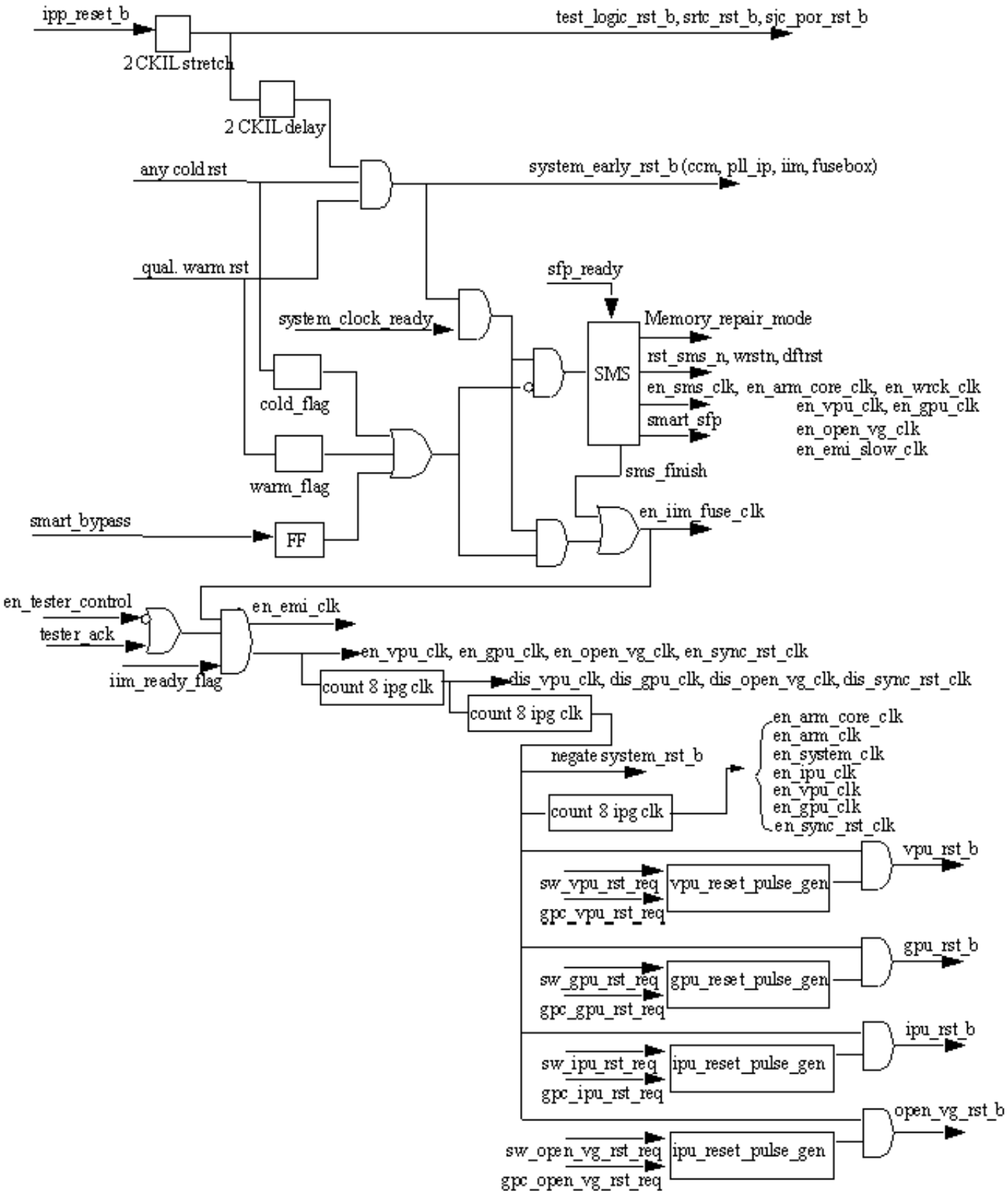


Figure 70-9. Reset Diagram

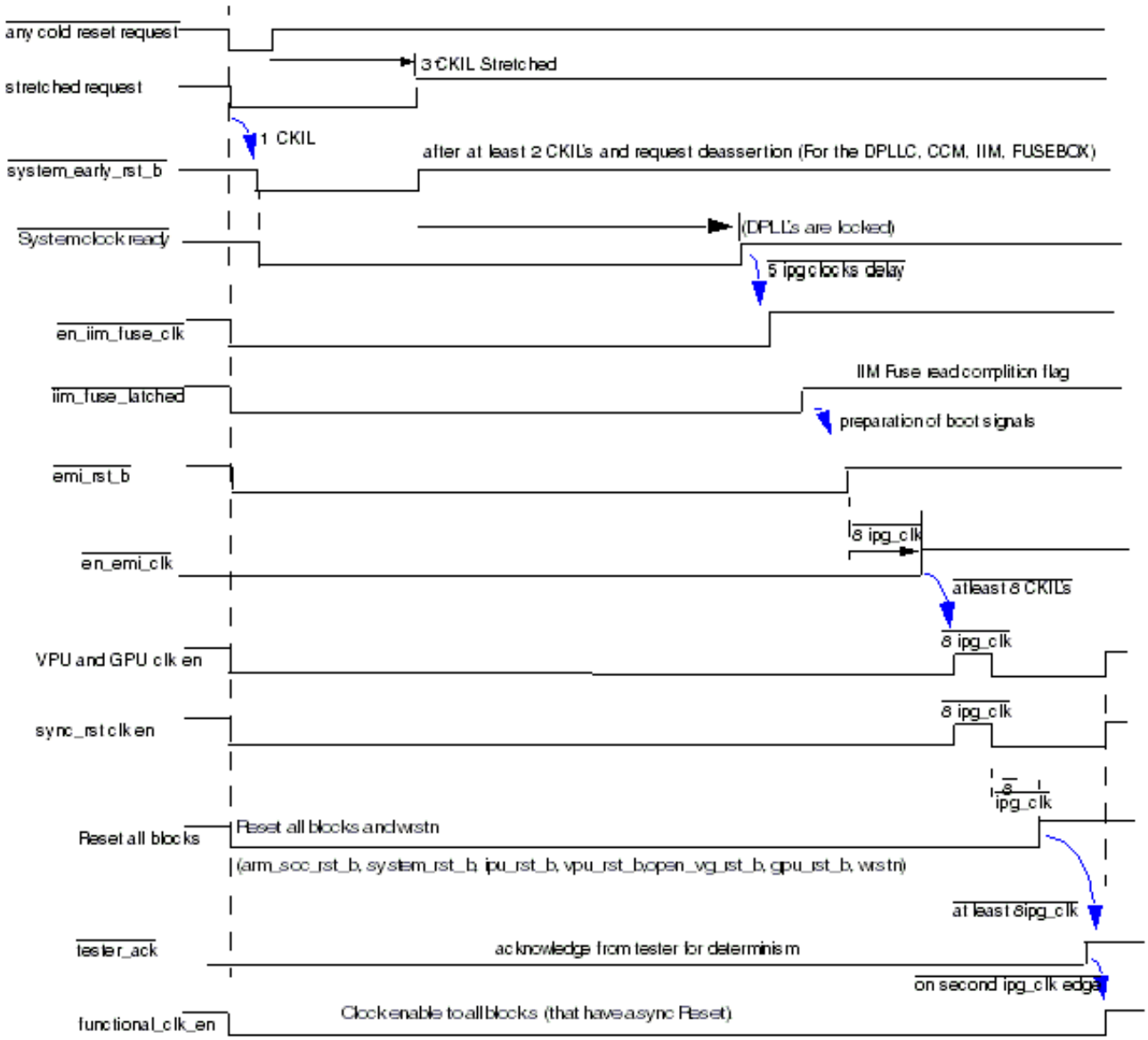


Figure 70-10. Cold reset sequence

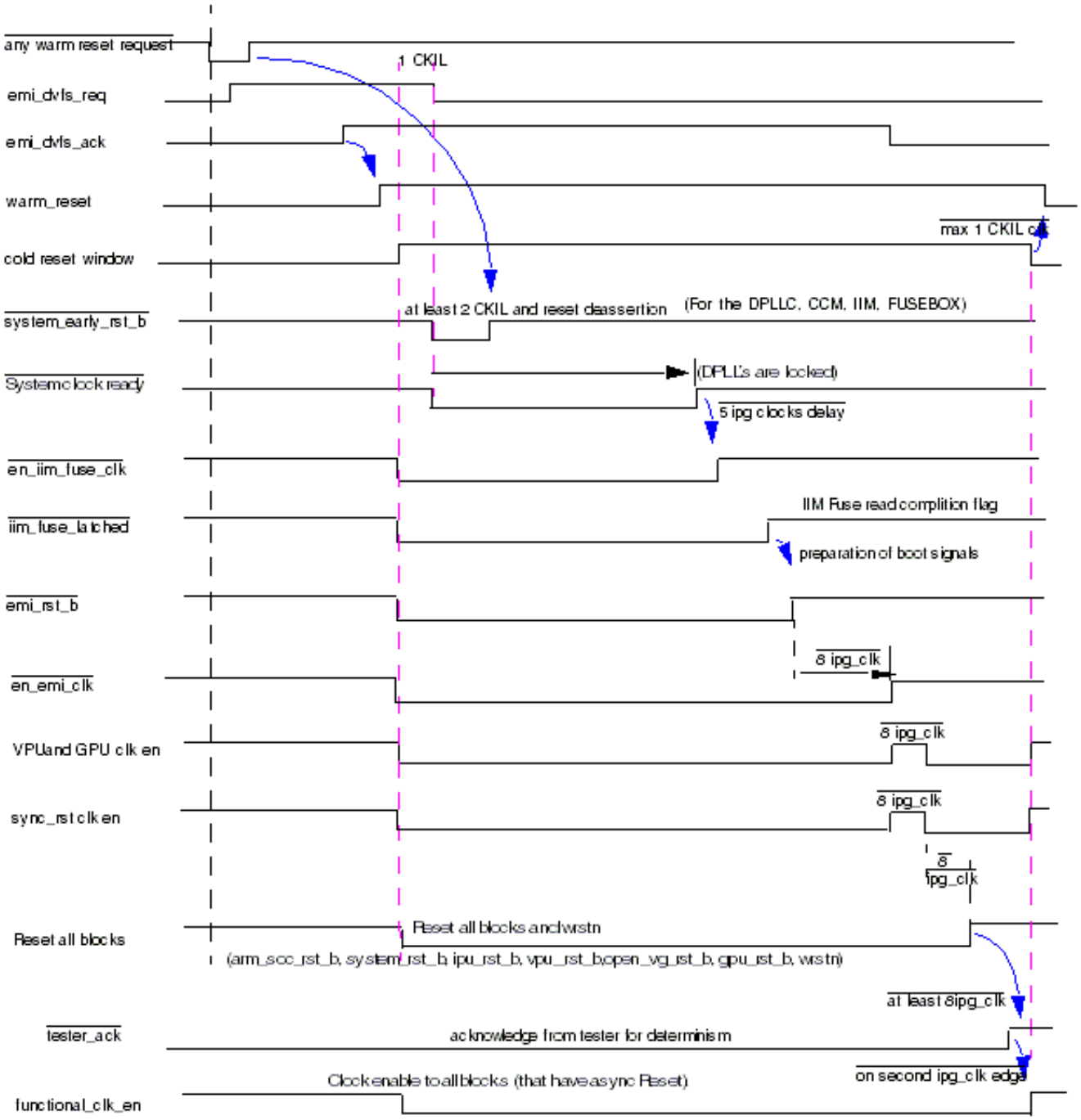


Figure 70-11. Warm reset sequence

70.3.1.2.3 SMS sub-block

The SMS sub-block manages the memory repair signals during por. The SMS sub-block will start its operation after system_clock_ready notification from CCM and only if the reset sequence is due to por sequence (ipp_reset_b). The SMS sub-block operation can be bypassed by smart_bypass assertion signal from the pad. If the smart_bypass is asserted, only the assertion part of SMS reset signals will take place, but smart_sfp - sfp_ready sequence will not be performed.

Following is the sms sequence:

1. Upon assertion of system_clock_ready signal, count 2 ipg clocks and assert the memory_repair_mode signal.
2. Wait 4 wrck clocks.
3. Assert sms reset signals on wrck clock edge:

   rst_sms_n = 0

   wrst_n = 0

   dftrst = 1

4. Wait 3 wrck clocks
5. Deassert the reset signals on wrck clock edge:

   rst_sms_n = 1

   wrst_n = 1

   dftrst = 0

6. Wait 4 wrck clocks
7. Enable memory repair clocks, i.e. en_arm_core_clk=1, en_emi_slow_clk=1, en_vpu_clk=1, en_gpu_clk=1, en_open_vg_clk=1, en_sms_clk=1 and en_wrck_clk=1 on wrck edge.
8. Wait 4 wrck clocks
9. Assert smart_sfp signal on wrck edge.
10. Monitor deassertion and assertion of sfp_ready signal.
11. Once assertion of sfp_ready signal, deassert smart_sfp, deassert clock enables en_arm_core_clk=0, en_emi_slow_clk=0, en_vpu_clk=0, en_gpu_clk=0, en_open_vg_clk=0, en_sms_clk=0 and en_wrck_clk=0 on wrck edge, deassert memory_repair_mode signal. (all on wrck edges).
12. Wait 4 wrck clocks
13. Assert sms_finish to notify that SRC can continue with enable IIM clocks.

Next figure describes the sms flow.

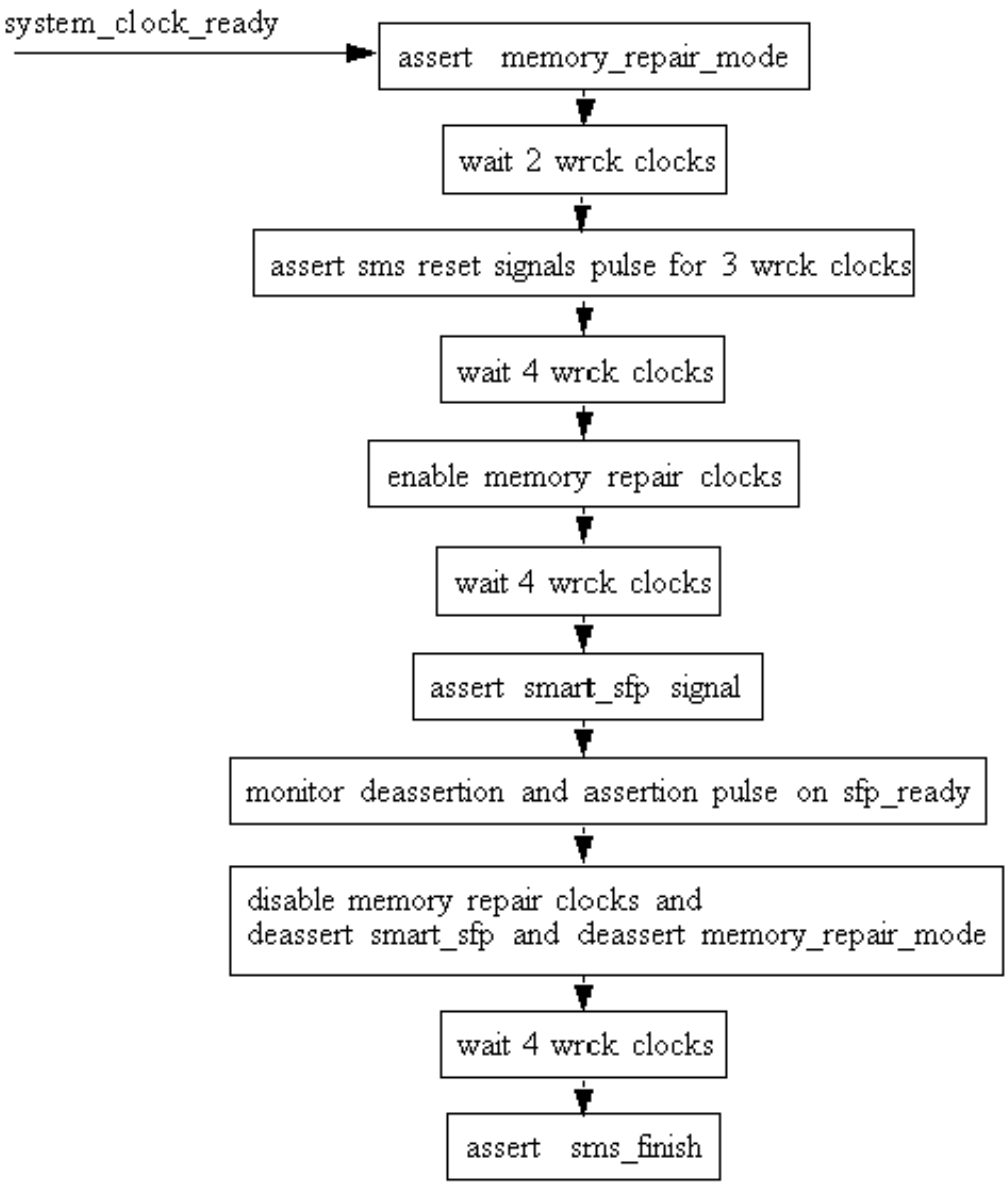


Figure 70-12. sms flow diagram

70.3.1.2.4 SRTC_RST_B generation

SRTC_RST_B will be generated during POR sequence together with test_logic_rst_b.

On the srtc_rst_b path there will be a mux that enables a generation of reset on srtc_rst_b pad from TCU, in case of test_mode. The enable of this mux is test_mode input. When the test_mode input equals '0', the generated srtc_rst_b will be the functional srtc reset (generated during POR scenario). When the test_mode input equals '1' the generated

srtc_rst_b will be connected to the tcu_reset input. In this case, tcu will have complete controlability of srtc_rst_b, and will be able to generate a reset for srtc. Below diagram describes this mux.

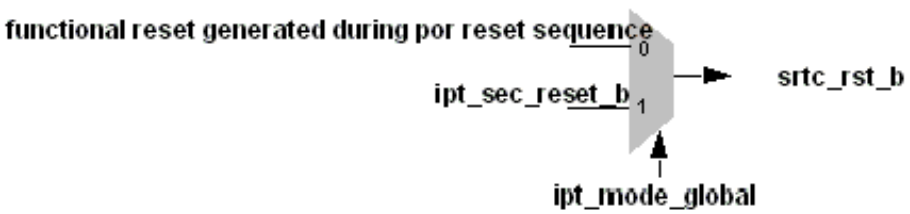


Figure 70-13. srtc_rst_b generation

70.3.2 SJC_POR_RST_B Generation

SJC will be reset by sjc_por_rst_b output of SRC. This output will be toggled only on POR sequence together with test_logic_rst_b. It will not be toggled on IEEE reset (jtag_rst_b).

70.3.3 Parallel Reset Requests

This chapter describes the behavior when parallel reset requests are received. In the parallel reset requests case SRC will follow the following rules:

1. The order of strength of resets is POR - strongest, cold - medium, warm - weakest.
2. If a stronger reset is asserted during weaker reset sequence, then the stronger reset will take over and the stronger reset process will commence. the following cases fall into this category:

   POR reset request in the middle of cold or warm reset process - the cold or warm reset process will be stopped and the POR sequence will start.

   COLD reset request in the middle of warm reset process - the warm reset process will be stopped and the cold sequence will start.

3. If a weaker reset is asserted during stronger reset sequence, then the stronger reset sequence will continue without interference. If at the end of the stronger reset process the weaker request is still asserted then the weaker sequence will commence. The following cases fall into this category:

COLD or WARM reset requests in the middle of POR reset process - the POR process will continue without interference.

WARM reset request in the middle of COLD reset process - the COLD process will continue without interference.

4. If a similar reset request is asserted during the process of reset handling, then the process of reset handling will start over (with the same process). The following cases fall into this category:

   POR reset request in the middle of POR reset process - the POR process will start over.

   COLD reset request in the middle of COLD reset process - the COLD process will start over.

There is one exception to this category - WARM reset request in the middle of WARM reset process - in this case the new WARM reset process cant restart because EXTMC is reset and there can't be a handshake with EXTMC if EXTMC is reset. For this case the first WARM reset will continue, and only if the second WARM reset is still asserted after the first one has finished, then the WARM sequence will start again.

70.3.4 DFT Mux on Reset Outputs

All functional reset output described above are muxed before exiting SRC. The mux is controlled by "ipt_mode_global "signal. The mux chooses between the functional reset and the DFT reset.

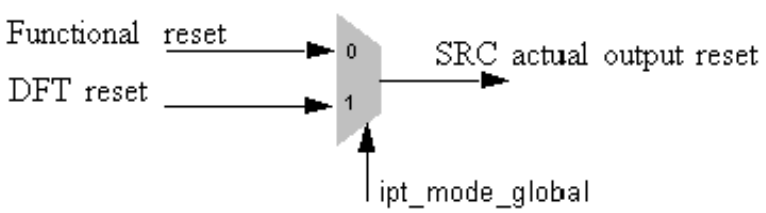


Figure 70-14. MUX between functional reset and DFT reset

Next table describes the DFT reset assigned to each of the reset outputs.

Table 70-8. DFT Reset Assignment

SRC reset output	DFT reset generation
arm_por_rst	~(ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b)
arm_soc_rst_b	ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b
dftrst	~(ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b)
dftrst_vpu	~(ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b)
dftrst_ipu	~(ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b)
emi_rst_b	ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b
ipu_rst_b	ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b
rst_sms_n	ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b
rst_sms_n_vpu	ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b
rst_sms_n_ipu	ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b
system_early_rst_b	ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b
system_rst_b	ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b
test_logic_rst_b	ipp_reset_b
vpu_rst_b	ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b
wrst_n	ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b
wrst_n_vpu	ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b
wrst_n_open_vg	ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b
wrst_n_gpu	ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b
wrst_n_ipu	ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b
srtc_rst_b	**ipt_sec_reset_b**
src_por_b	ipp_reset_b
weim_rst_b	ipp_reset_b & ipp_user_reset_b & ipt_sec_reset_b

Note: src_por_b is an internal reset generated for SRC only. This reset is asserted on por only (ipp_reset_b). It is equal to the functional test_logic_reset_b.

70.3.5 Boot Mode Control

70.3.5.1 BOOT_MODE Pin Latching

The exact boot sequence for i.MX53 is controlled by the values of the BOOT_MODE0 and BOOT_MODE1 pins on this device.

Both BOOT_MODE pins will be sampled at deassertion of system_early_rst_b.

The value of the BOOT_MODE pins will be latched after the IIM asserts the fuse read completion flag. After latching, the values of the BOOT_MODE pins are used to determine the booting options of the core as given in the SBMR register.

The Boot mode general purpose bits can be provided to the SRC from either e-fuses or GPIO signals. The gpio_bt_sel e-fuse defines the source to be used to derive the boot information. When gpio_bt_sel is set, e-fuses are used. When cleared, GPIO signals are used.

When FSL test mode is chosen (BMOD[1:0] = 0b01) then the Fuses signals are used.

When Internal Production Boot Mode is chosen (10 for BMOD[1:0]) then e-fuses signals are used, independent of the gpio_bt_sel e-fuse value.

The Boot information is provided in SBMR register. Figure 70-15 shows the selection of Boot mode information.

NOTE

BOOT_MODE[1:0] inputs of SRC are connected to the BOOT_MODE pins. BOOT_MODE[1:0] = {BOOT_MODE1, BOOT_MODE0}

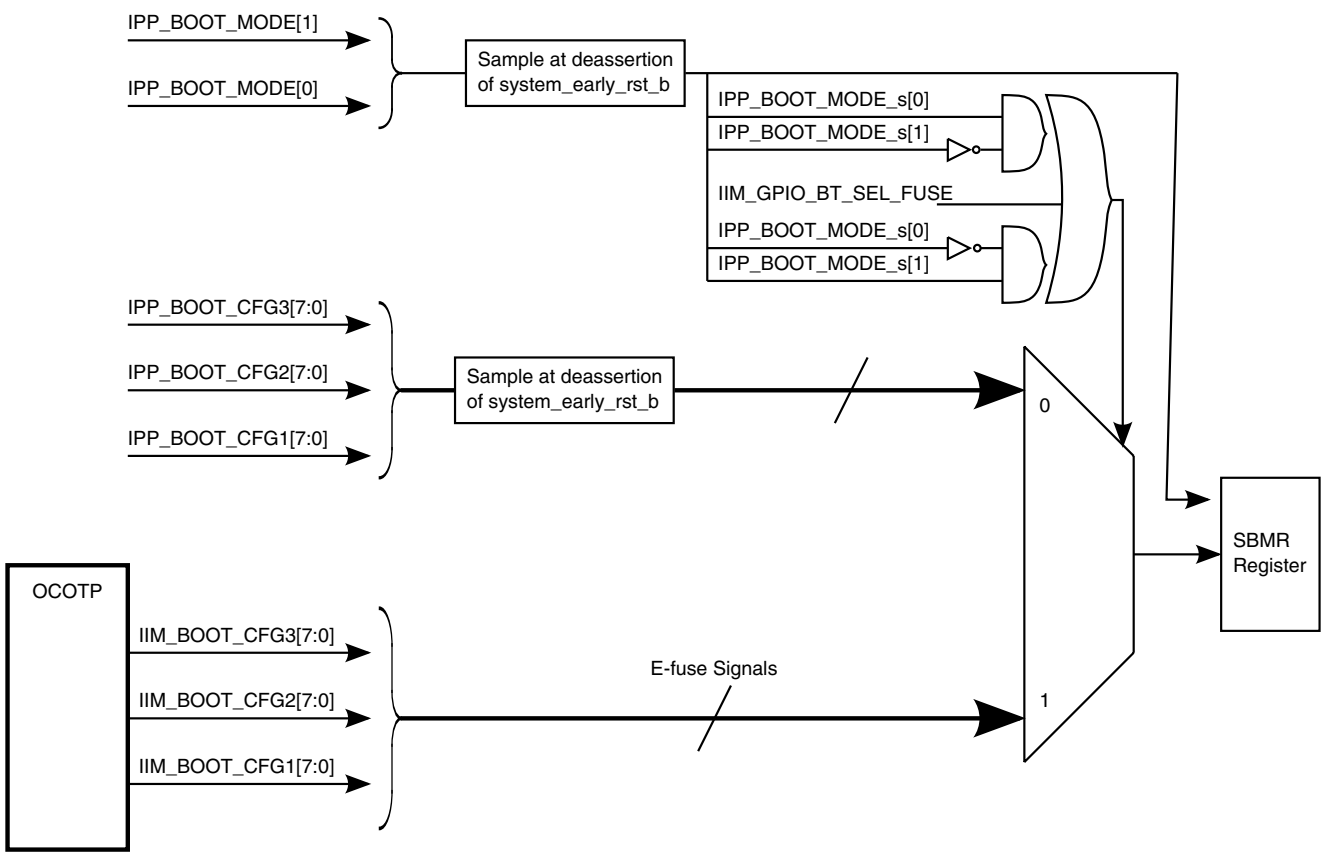


Figure 70-15. Boot Mode Information

NOTE

IPP signals that need to be latched by SRC will be set to the corresponding mode based on system_rst_b signal. When system_rst_b signal equals '0', the IOMUX will shift to the needed mode for transferring boot values on the IO pins. When system_rst_b signal equals '1', the IOMUX will shift to the default mode.

70.3.5.2 Correspondence Table between SRC SBMR bit/signal Names and Fuse Names

The SBMR bit names and boot signals corresponding to those bits were changed in i.MX53 fuse map. Below table describes the matching between the new fuse names and the ones represented above in SBMR register.

70.3.5.3 Boot Output Signals

1. External BOOT is set when the chip is in FSL TEST MODE and BOOT FROM EIM is selected.

   Otherwise, internal boot is selected.

2. EIM_BOOT_CFG bits:

   Used to determine if the chip uses muxed EIM 16 bits low half or unmuxed EIM 16 bits high half data.

   Look at i.MX53 Fuse Map for more details.

3. src_boot_add[31:0] - boot address - output to ARM Platform

   IF src_int_boot=0 then src_boot_add[31:0]=0x0800_0000

   Otherwise src_boot_add[31:0]=0x0000_0000

4. ipp_do_boot_mode_inv[1:0]=NOTBOOT_MODE[1:0]. Once the boot pins (BOOT_MODE[1:0]) are sampled in SRC, SRC will drive ipp_do_boot_mode_inv[1:0] back out on the BOOT_MODE pins. The I/O ring will use SYSTEM_RESET_B to determine when to drive the value ipp_do_boot_mode_inv[1:0] on the BOOT_MODE pins.

When system_rst_b=0, the BOOT_MODE pins will be configured as inputs, allowing boot mode information to propagate to SRC's BOOT_MODE[1:0].

When system_rst_b=1, the boot pads will be configured as outputs, allowing ipp_do_boot_mode_inv[1:0] to be driven out on the BOOT_MODE pins.

Board-level logic can detect the inversion of the BOOT_MODE pin state in order to determine that the system boot sequence has changed. This allows the subsequent use of the boot mode pins for some alternate purpose.

Chapter 71
State Retention Power Gating Controller (SRPGC)

71.1 Introduction

The State Retention Power Gating Controller (SRPGC) is a part of GPC. It controls the save, restore & power gating sequence of a platform implemented using SRPG cells. Power supply to all the logic except the retention latch of the SRPG flop, can be switched off in low power modes. For designs using SRPG flops, a specific sequence of various control signals must be followed in order to save and restore the state of flops correctly. SRPGC provides this sequence based on the various control registers.

71.1.1 Features

The SRPGC features include:

- Provides options to switch power off to a platform.
- Controls the operation of state-retaining/state-restoring circuitry, and stages the retain/restore signals in order to control the IR drop on the power grid during these events.
- Generates Power-up and Power-down control sequences
- Programmable registers to stage the various SRPG and Power control signals
- 32-bit IP Bus interface
- All registers are byte-accessible

71.1.2 Modes of Operation

71.1.2.1 Functional Mode

The SRPGC provides an option to switch OFF power to a platforms. The various power sequence control registers should be programmed to the desired value before powering down a platform.

71.1.2.2 Debug Mode

If the debug bit in the Debug control register is set to one, the SRPGC does not asserts the various outputs in the power down sequence, based on the other control bits in the Debug control register.

71.2 Power Gating Cells

71.2.1 Power Supply Switch Cells

In order to satisfy the power supply requirements of a platform, power switches are placed between the continuous supplies and the gated supplies as shown below. The power switches are composed of parallel power switch cells placed external to the platform.

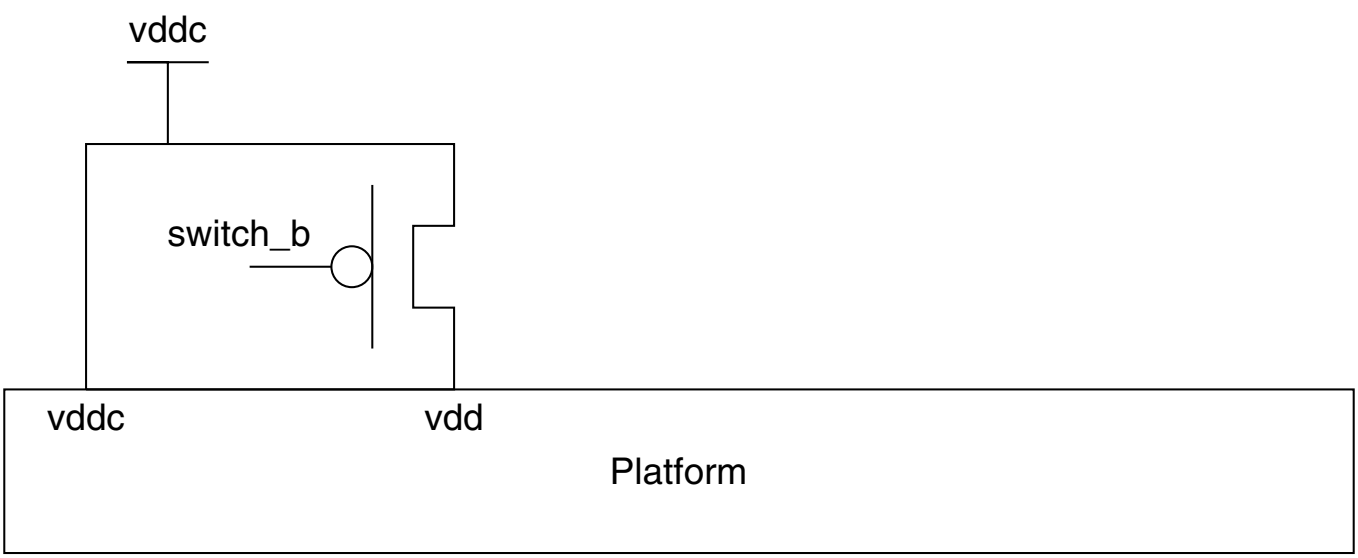


Figure 71-1. Power Switches External to the Platform

The power switch cells are positioned in strategic locations that satisfy the active current requirements of a platform. Table 71-1 describes the pins of the power switch cell

Table 71-1. Power Switch Cell Pins

Name	Type	Description
vddc	SUPPLY	Continuous supply1 connection
vdd	GATED_SUPPLY	Gated supply1 connection that will be placed into a resistive state when power_en_b is negated.
switch_b	INPUT	Active low power control signal that allows high conductivity between vddc and vdd when asserted. Invokes low conductivity between vddc and vdd when negated.

71.2.2 Power Supply Shorting Cells

Internal to a Platform, there are a number of power supply shorting cells. The power supply shorting cells provide local clamping between the continuous and the gated vdd nodes. Internal clamping ensures that the supply1 reference correlates for both the continuous logic (the state retaining circuitry) and the gated logic (the combinational logic). During periods of high switching activity, IR drops on the continuous supply will be transferred to the gated supply and vice versa. A primary purpose of the shorting cells is to give the Vdd supply access to the nwell-psubstrate diode decoupling capacitance on VDDC.

During power up states, the power supply shorting cells will be placed in a high conductivity configuration by assertions on the SHORT_B signal. Consequently, the power supply shorting cells will be held in a high impedance configuration during power down states.

Figure 71-2 shows a block diagram with a platform power supply shorting cells. The power supply shorting cells are instantiated as a parallel array of cells.

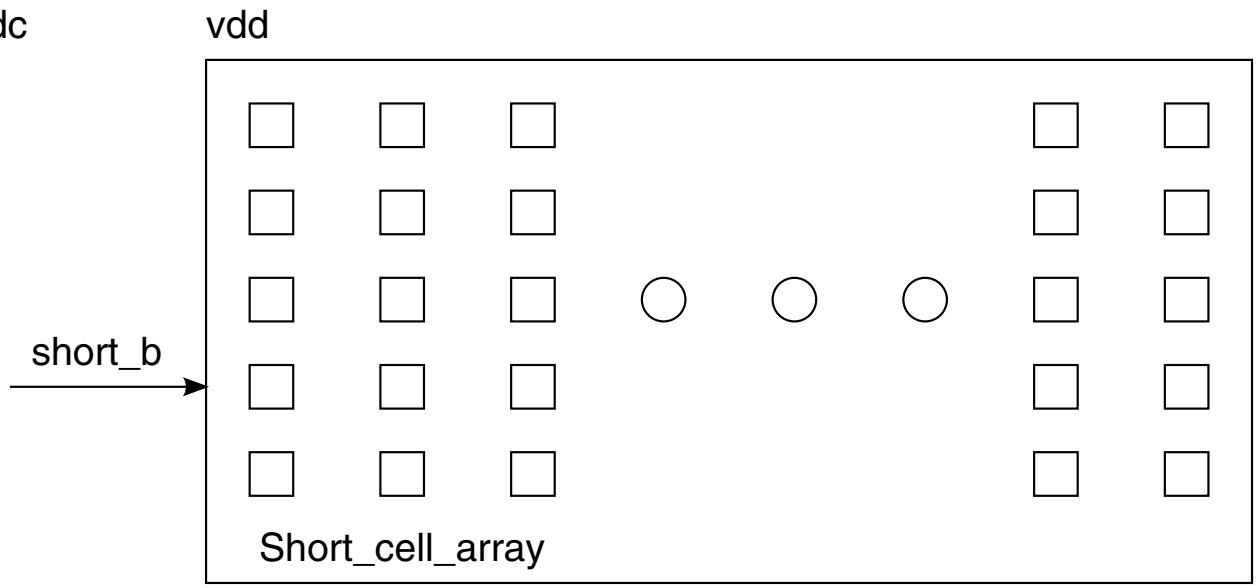


Figure 71-2. Power Supply Shorting Cells Internal to a Platform

Figure 71-3 shows the schematic diagram of power supply shorting cell. This may be combined with a well tie cell.

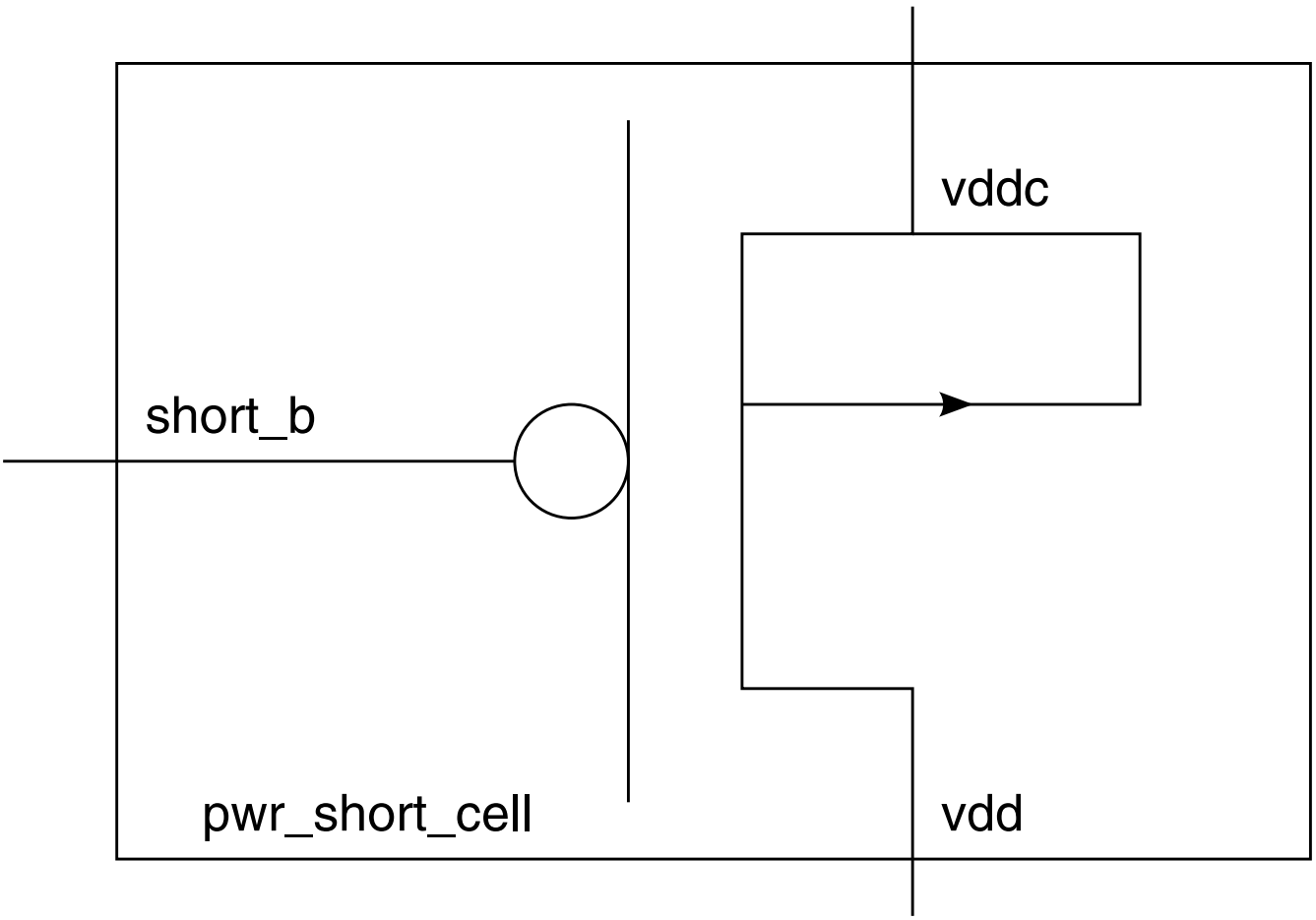


Figure 71-3. Power Supply Shorting Cell Schematic Diagram

Table 71-2 describes the pins of the power supply shorting cell.

Table 71-2. Power Supply Shorting Cell Pins

Name	Type	Description
vddc	SUPPLY	Continuous supply1 connection
vdd	GATED_SUPPLY	Gated supply1 connection that will be placed into a resistive state when short_b is negated.
short_b	INPUT	Active low power control signal that allows high conductivity between vddc and vdd_gtd when asserted. Invokes low conductivity between vddc and vdd_gtd when negated.

71.2.3 State Retention Power Gating (SRPG) Cell

The different platforms have the ability to keep the state of every register element preserved during power gating by using special SRPG flip flop elements. The flip flop consists of a master latch and a slave latch. During active modes, both the master and slave latches are powered up. During power gating, the master latch is completely powered off, but the slave latch remains powered up. This allows the state to be preserved with low leakage. Figure 71-4 shows a simplified version of a normal SRPG flop.

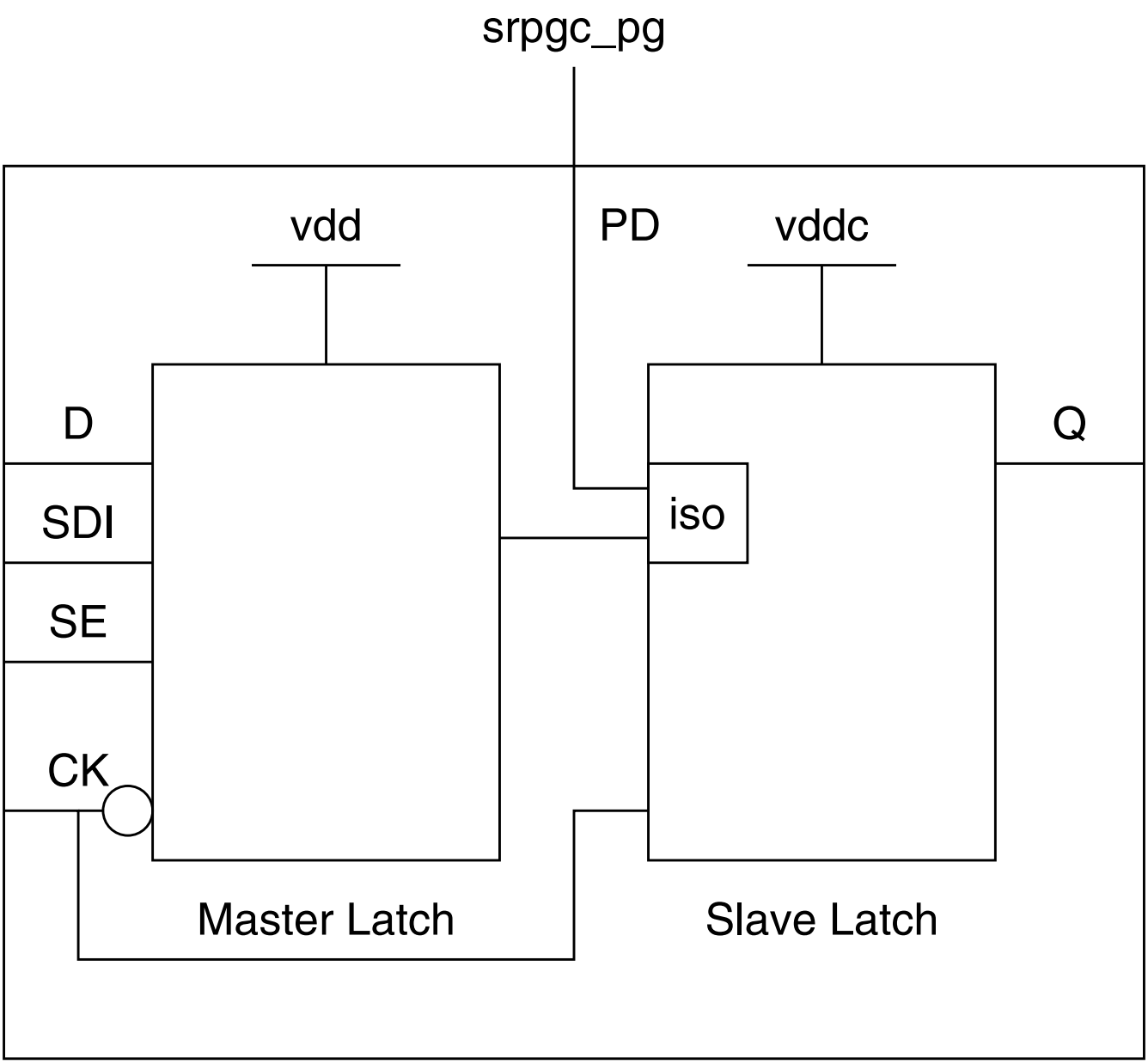


Figure 71-4. Simplified SRPG Cell Diagram

The SRPG flop provides a single extra control signal, PD, that is used to safely isolate the master and the slave latch so that the master latch can be powered down. The PD control signal must remain powered up and asserted during power gating. All other input signals can float including set, reset, clock, etc. The Q output shown in Figure 71-4 and the QN output in an inverted output flop (not shown) will be forced to a value when PD is asserted and vdd is not shut off. This is a result of the internal circuit design and not due to any special requirement. Upon exit from power gating, the PD signal will be deasserted to restore the flip flop to normal operation. For posedge flip-flops, the clock must be low before PD is asserted and held low until PD is deasserted. This also allows for needing only the slave latch state to be saved.

Because there may be many flip flops in a platform, the process of isolating all the flip flops for power gating, and forcing the Q and QN outputs will cause a moment of high instantaneous current, particularly if the Q or QN was previously a different state. Likewise the process of restoring all Q and QN outputs will cause a moment of high instantaneous current. For this reason, the platform must provide multiple PD signals that must be individually asserted/deasserted staged. Staging the negations and assertions on the PD signals provides guard banding against the unacceptable IR drop which might have caused flops to lose state. By transitioning these PD chains at different times during the SRPG_ENTRY and SRPG_EXIT sequences, an acceptable IR drop can be achieved in all cases.

There are several other very important design details which should be adhered to. First, reset synchronizing flops PD signals must be the last PD signals asserted and the first to be deasserted. This ensures proper operation of the reset synchronizing flops which must not have an incorrect Q or QN output prematurely. Second, the PD signals should be asserted/deasserted after the clocks have stopped low for posedge flops. This ensures that the master to slave pass gates are not opened, and is a requirement of the flop design.

The above text describes the SRPG flop architecture called SRPG7. Some variations of the SRPG flop exist. For example, the SRPG2 flop, which is available for LP transistor designs only, does not change Q output when PD is asserted. This lessens the need to stagger PD assertions and eliminates the need to give reset synchronizers their own PD signal. Another flop, applicable only to GP transistor designs, is the SRPG7RS flop. This flop is designed for reset synchronizers and, at the expense of area, does not change the Q output when PD is asserted.

71.3 SRPGC Interface

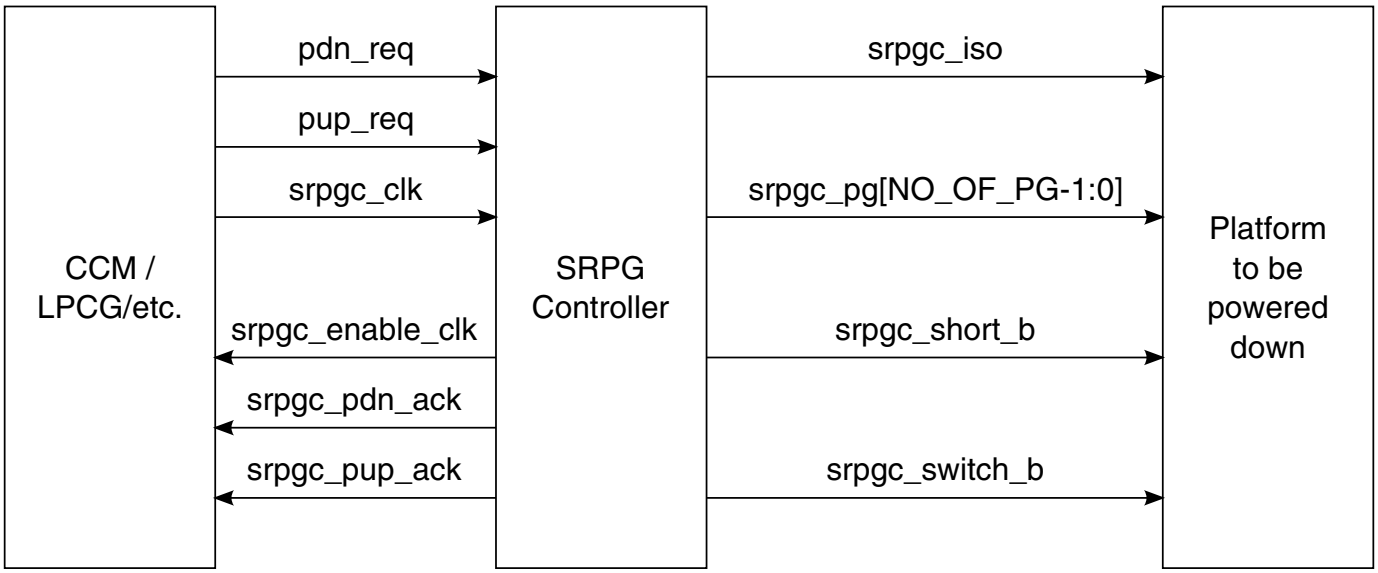


Figure 71-5. SRPGC Interface

71.3.1 Power-Down Sequence (SRPG Entry Sequence)

The power-down sequence control register (SRPGC_PDNSCR) and power-up sequence control register (PUPSCR) must be programmed to the desired value before initiating the "pdn_req" and must not be changed until power up is completed. The SRPG entry sequence is outlined in the following list of steps:

1. Software program PCR bit in the SRPG control register (SRPGC_SRPGCR).
2. CCM stops the clocks to the powering down platform.
3. pdn_req is asserted to SRPGC.
4. Upon detecting the pdn_req assertion, SRPGC asserts "srpgc_enable_clk".
5. SRPGC asserts "srpgc_iso" to isolate the outputs, based on the programming of PDNSCR[5:0].
6. SRPGC asserts "srpgc_pg[0]" to save the state of SRPG flops, based on the programming of PDNSCR[13:8]
7. SRPGC asserts "srpgc_pg[NO_OF_PG-1:1]" to save the state of SRPG flops. "srpgc_pg[1]" is asserted after one "srpgc_clk" of srpgc_pg[0] assertion, "srpgc_pg[2]" is asserted after two "srpgc_clk" of srpgc_pg[0] assertion, "srpgc_pg[NO_OF_PG-1]" is asserted after (NO_OF_PG-1) "srpgc_clk" of srpgc_pg[0] assertion.

8. SRPGC deasserts "srpgc_short_b", based on the programming of PDNSCR[17:16].
9. SRPGC deasserts "srpgc_switch_b", based on the programming of PDNSCR[29:24].
10. SRPGC asserts "srpgc_pdn_ack", one "srpgc_clk" cycle wide pulse.
11. SRPGC deasserts "srpgc_enable_clk".

NOTE

If the "pup_req" is asserted before completion of power down then the SRPGC stops the power down sequence immediately and starts the power-up sequence. "srpgc_pdn_ack" is asserted as soon as the power down sequence stops. If "pup_req" is asserted before or along with "pdn_req" then power down sequence does not start and "srpgc_pdn_ack" is not asserted by SRPGC.

71.3.1.1 Power-Down Sequence Timing Waveforms

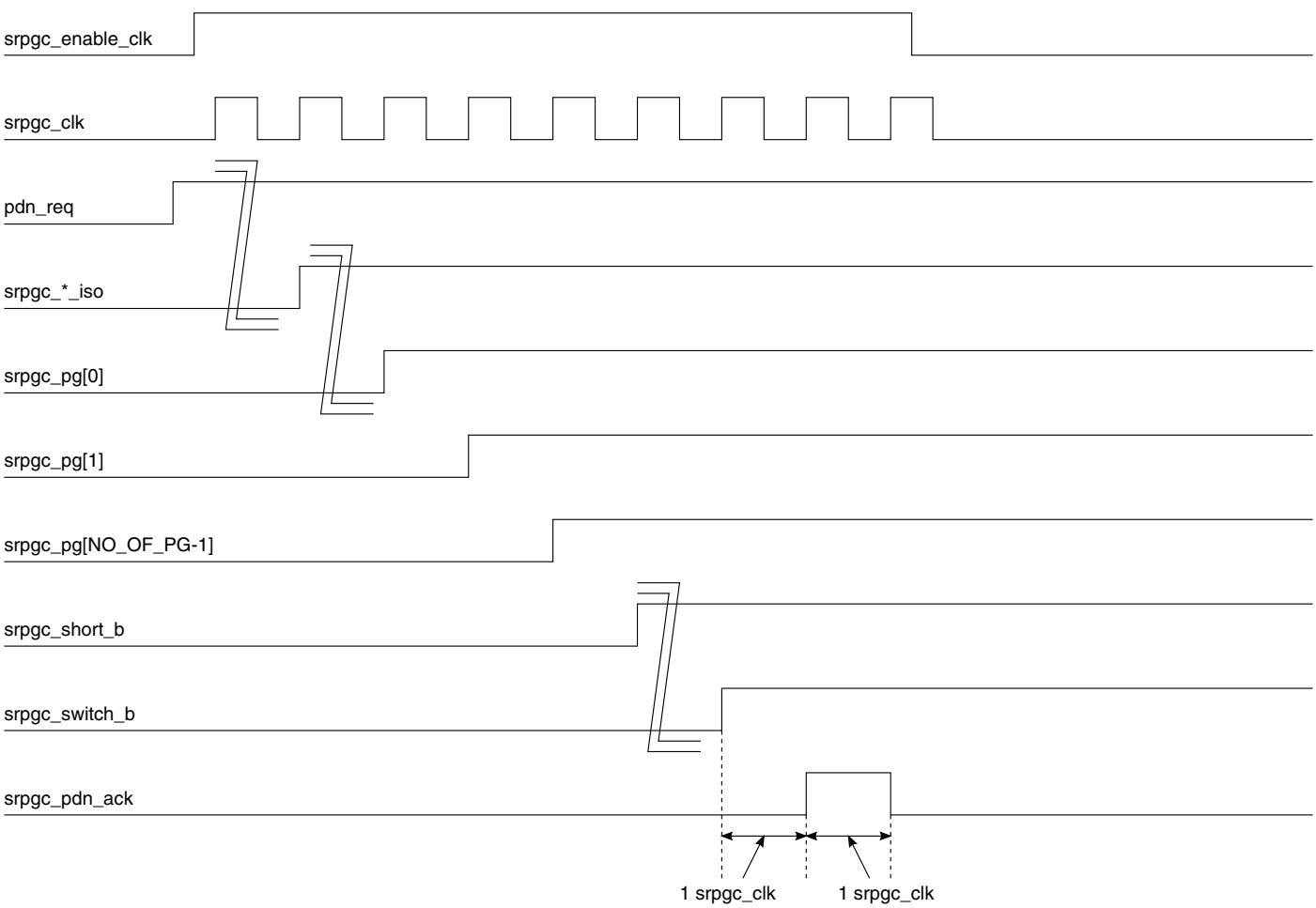


Figure 71-6. Power-down Sequence Timing Waveforms

71.3.2 Power-Up Sequence (SRPG Exit Sequence)

SRPG exit sequence is outlined in the following list of steps:

1. pup_req is asserted to SRPGC.
2. SRPGC asserts "srpgc_enable_clk" to enable the clock.
3. SRPGC asserts "srpgc_switch_b", based on the programming of PUPSCR[5:0].
4. SRPGC asserts "srpgc_short_b", based on the programming of PUPSCR[13:8].
5. SRPGC deasserts "srpgc_pg[NO_OF_PG-1]" to restore the state of SRPG flops, based on the programming of PUPSCR[21:16].
6. SRPGC deasserts "srpgc_pg[NO_OF_PG-2:0]" to restore the state of SRPG flops. "srpgc_pg[NO_OF_PG-2]" is deasserted after one "srpgc_clk" of srpgc_pg[NO_OF_PG-1] deassertion, "srpgc_pg[NO_OF_PG-3]" is deasserted after two "srpgc_clk" of srpgc_pg[NO_OF_PG-1] deassertion, "srpgc_pg[0]" is deasserted after (NO_OF_PG-1) "srpgc_clk" of srpgc_pg[NO_OF_PG-1] deassertion.
7. SRPGC deasserts "srpgc_iso" to remove the isolation from the output of the platform, based on the programming of PUPSCR[25:24].
8. SRPGC asserts "srpgc_pup_ack" after one clock cycles of "srpgc_iso" deassertion. This is a one "srpgc_clk" cycle wide pulse.
9. "pup_req" is deasserted to SRPGC.
10. SRPGC deasserts "srpgc_enable_clk".

NOTE

If the "pup_req" is asserted before completion of power down then the SRPGC stops the power down sequence immediately and starts the power-up sequence. "srpgc_pup_ack" is asserted after completion of power up. If "pup_req" is asserted before "pdn_req" then SRPGC does not initiate the power down sequence and asserts the "srpgc_pup_ack".

71.3.2.1 Power-up Sequence Timing Waveforms

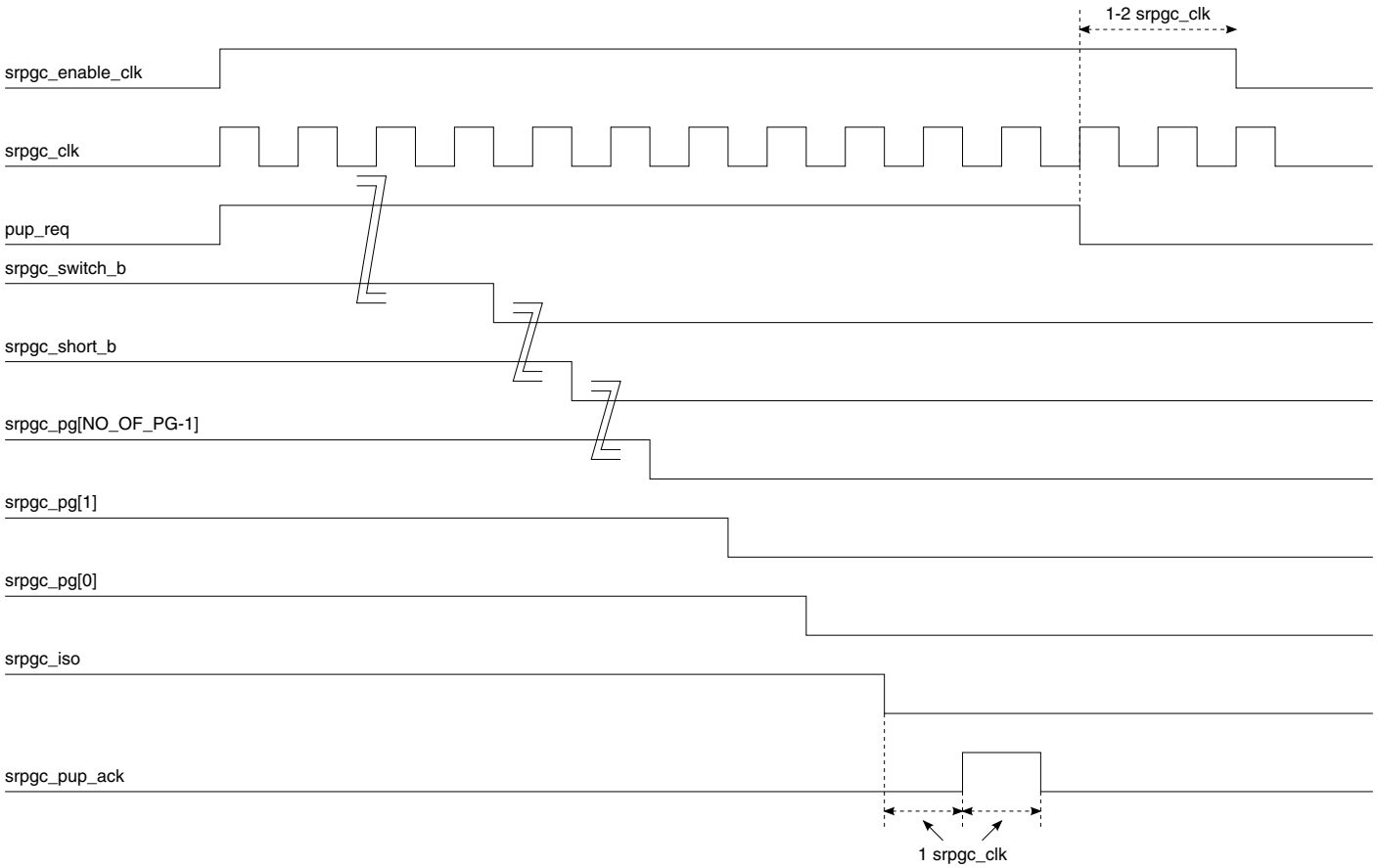


Figure 71-7. Power-up Sequence Timing Waveforms

71.4 Programmable Registers

The SRPGC registers can only be accessed in supervisor mode. Any access in normal mode, or an access to an unimplemented address location can assert transfer error signal "ips_xfr_err", if "resp_sel" is low.

SRPGC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FD_8280	SPRGC Control Register (SRPGC-1_SRPGCR)	32	R/W	0000_0000h	71.4.1/ 4394

Table continues on the next page...

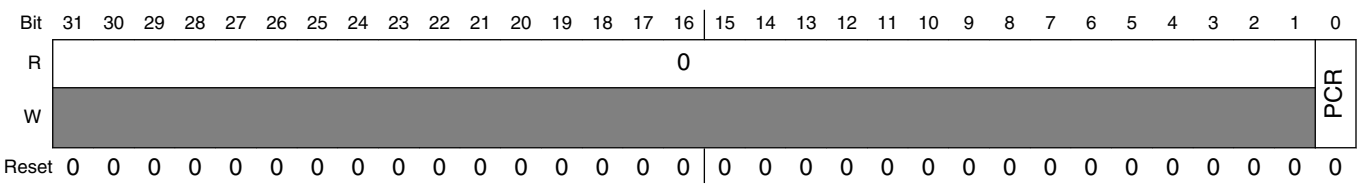
SRPGC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FD_8284	Power-up Sequence Control Register (SRPGC-1_PUPSCR)	32	R/W	0102_0F01h	71.4.2/ 4395
53FD_8288	Power-down Sequence Control Register (SRPGC-1_PDNSCR)	32	R/W	0101_0101h	71.4.3/ 4395
53FD_828C	SPRGC Status Register (SRPGC-1_SRPGSR)	32	w1c	0000_0000h	71.4.4/ 4396
53FD_8290	SPRGC Debug Register (SRPGC-1_SRPGDR)	32	R/W	0000_0000h	71.4.5/ 4397
53FD_82A0	SPRGC Control Register (SRPGC-2_SRPGCR)	32	R/W	0000_0000h	71.4.1/ 4394
53FD_82A4	Power-up Sequence Control Register (SRPGC-2_PUPSCR)	32	R/W	0102_0F01h	71.4.2/ 4395
53FD_82A8	Power-down Sequence Control Register (SRPGC-2_PDNSCR)	32	R/W	0101_0101h	71.4.3/ 4395
53FD_82AC	SPRGC Status Register (SRPGC-2_SRPGSR)	32	w1c	0000_0000h	71.4.4/ 4396
53FD_82B0	SPRGC Debug Register (SRPGC-2_SRPGDR)	32	R/W	0000_0000h	71.4.5/ 4397

71.4.1 SPRGC Control Register (SRPGCx_SRPGCR)

Addresses: SRPGC-1_SRPGCR is 53FD_8280h base + 0h offset = 53FD_8280h

SRPGC-2_SRPGCR is 53FD_82A0h base + 0h offset = 53FD_82A0h



SRPGCx_SRPGCR field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 PCR	Power control WARNING: This bit should not change from "pdn_req" assertion until platform is completely powered up. 0 Do Not Switch OFF power even if the pdn_req is asserted 1 Switch OFF Power when pdn_req is asserted

71.4.2 Power-up Sequence Control Register (SRPGCx_PUPSCR)

Addresses: SRPGC-1_PUPSCR is 53FD_8280h base + 4h offset = 53FD_8284h

SRPGC-2_PUPSCR is 53FD_82A0h base + 4h offset = 53FD_82A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						PG2ISO		0		SH2PG						0		SW2SH						0		SW					
W																																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	1

SRPGCx_PUPSCR field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value zero. Reserved
25–24 PG2ISO	Number of clocks from Last PG (srpgc_pg[0]) deassertion to ISO deassertion. **NOTE:** It is not recommended to program these bits to zero.
23–22 Reserved	This read-only field is reserved and always has the value zero. Reserved
21–16 SH2PG	Number of clocks from SHORT assertion to First PG (srpgc_pg[NO_OF_PG-1]) deassertion. **NOTE:** It is not recommended to program these bits to zero.
15–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13–8 SW2SH	Number of clocks from SWITCH assertion to SHORT assertion. **NOTE:** It is not recommended to program these bits to zero.
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5–0 SW	Number of clocks to assert SWITCH from Power-up request. WARNING: This should not be programmed to zero. A zero value could cause a glitch on the srpgc_switch_b if the power-up request (pup_req) is asserted immediately after deassertion of srpgc_switch_b.

71.4.3 Power-down Sequence Control Register (SRPGCx_PDNSCR)

Addresses: SRPGC-1_PDNSCR is 53FD_8280h base + 8h offset = 53FD_8288h

SRPGC-2_PDNSCR is 53FD_82A0h base + 8h offset = 53FD_82A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0						PG2SH		0								0							
W			SH2SW																ISO2PG								ISO					
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

SRPGCx_PDNSCR field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved
29–24 SH2SW	Number of clocks from SHORT deassertion to SWITCH deassertion. **NOTE:** It is not recommended to program these bits to zero.
23–18 Reserved	This read-only field is reserved and always has the value zero. Reserved
17–16 PG2SH	Number of clocks from last PG (srpgc_pg[NO_OF_PG-1]) assertion to SHORT deassertion. **NOTE:** It is not recommended to program these bits to zero.
15–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13–8 ISO2PG	Number of clocks from ISO assertion to First PG (srpgc_pg[0]) assertion. **NOTE:** It is not recommended to program these bits to zero.
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved
5–0 ISO	Number of clocks from Power down request to ISO assertion. WARNING: It is not recommended to Program ISO to zero, if it is programmed to zero, ISO may be deasserted before the power is completely turned on

71.4.4 SPRGC Status Register (SRPGCx_SRPGSR)

Addresses: SRPGC-1_SRPGSR is 53FD_8280h base + Ch offset = 53FD_828Ch

SRPGC-2_SRPGSR is 53FD_82A0h base + Ch offset = 53FD_82ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															PSR
W																w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRPGCx_SRPGSR field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 PSR	Power status. Write 1 to clear this bit. User should clear this bit after power-up, otherwise this bit will continue to reflect the power status of the very first power down.

Table continues on the next page...

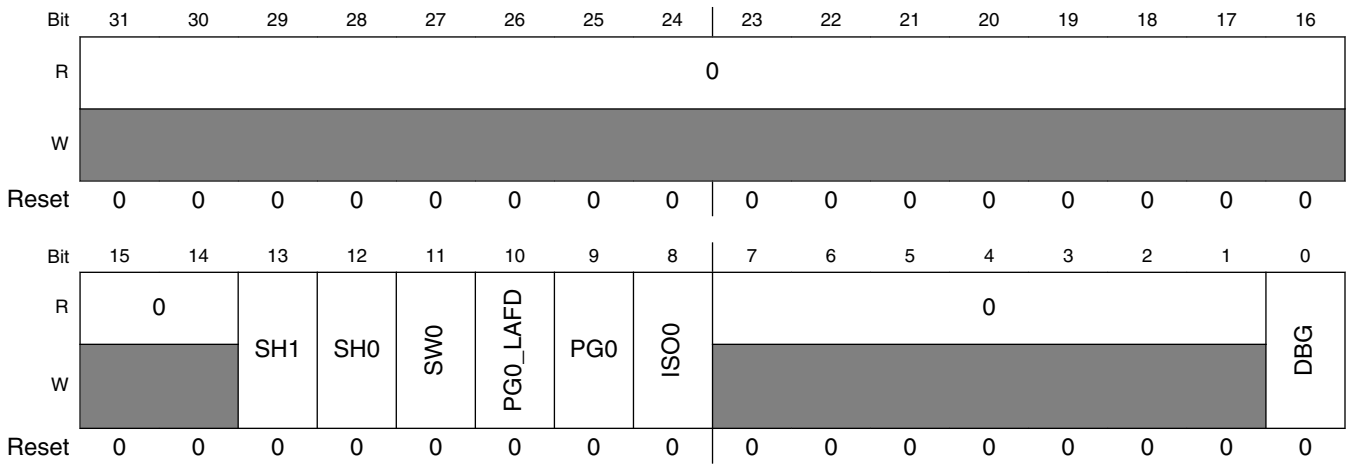
SRPGCx_SRPGSR field descriptions (continued)

Field	Description
	0 was NOT powered down in previous power down request 1 was powered down in previous power down request.

71.4.5 SPRGC Debug Register (SRPGCx_SRPGDR)

Addresses: SRPGC-1_SRPGDR is 53FD_8280h base + 10h offset = 53FD_8290h

SRPGC-2_SRPGDR is 53FD_82A0h base + 10h offset = 53FD_82B0h



SRPGCx_SRPGDR field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 SH1	srpgc_short_b control in debug mode. This bit has no effect if DBG bit is zero. **NOTE:** srpgc_short_b is asserted high as soon as SH1 is written to one if DBG bit is one and SH0 bit is zero. 0 srpgc_short_b is not forced to one. 1 srpgc_short_b is forced to one.
12 SH0	srpgc_short_b control in debug mode. This bit has no effect if DBG bit is zero. **NOTE:** If SH0 is also set when SH1 is set then srpgc_short_b is driven to Zero. 0 srpgc_short_b is not forced to zero. 1 srpgc_short_b is forced to zero.
11 SW0	srpgc_switch_b control in debug mode. This bit has no effect if DBG bit is zero. 0 srpgc_switch_b is not forced to zero. 1 srpgc_switch_b is forced to zero.
10 PG0_LAFD	srpgc_pg[NO_OF_PG-1](Last asserted and First deasserted) control in debug mode. This bit has no effect if DBG bit is zero.

Table continues on the next page...

SRPGCx_SRPGDR field descriptions (continued)

Field	Description
	NOTE: When NO_OF_PG is one then this bit forces srpgc_pg[0] to zero. 0 srpgc_pg[NO_OF_PG-1] is not forced to zero. 1 srpgc_pg[NO_OF_PG-1] is forced to zero.
9 PG0	srpgc_pg[NO_OF_PG-2: 0] control in debug mode. This bit has no effect if DBG bit is zero. **NOTE:** When NO_OF_PG is one then this bit has no meaning. 0 srpgc_pg[NO_OF_PG-2:0] is not forced to zero. 1 srpgc_pg[NO_OF_PG-2:0] is forced to zero.
8 ISO0	srpgc_iso control in debug mode. This bit has no effect if DBG bit is zero. 0 srpgc_iso is not forced to zero. 1 srpgc_iso is forced to zero.
7–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DBG	Debug control. 0 Debug mode is not selected. 1 Debug mode is selected.

Chapter 72
Secure Real Time Clock (SRTC)

72.1 Overview

This section briefly introduces the block. The full description of the block is in Functional Description.

The SRTC is comprised of two separated sub-blocks:

- Low power domain SRTC (SRTC LP)
- High power domain SRTC (SRTC HP)

SRTC LP is in an always powered up domain and is isolated from the rest of the logic by means of an isolation cell. Isolation cells are library instantiated cells, which insure that the powered up logic does not get corrupted when the power goes down in the rest of the SoC. It is powered by the Coin Cell battery.

SRTC HP is in the normal supply domain. It is powered by the main digital supply powering the SoC.

72.1.1 Low Power SRTC (SRTC LP) Overview

The SRTC LP is a real time clock with enhanced security capabilities. Its purpose is to provide an accurate time read at all times, regardless of the main system power state and without the need to use an external on-board time source such as external RTC. The SRTC LP can wake up the system when preset time alarm is asserted.

It is divided into following units:

- Counters and registers
  - Non-rollover 47-bit time counter
  - 32-bit alarm register

- General purpose registers
- Secured monotonic counter

- SRTC Monitor
  - SRTC state machine
  - SRTC power and clock detectors

Figure 72-1 shows the SRTC partitioning.

72.1.2 High Power SRTC (SRTC HP) Overview

The SRTC HP contains all the interfaces to the core (IP Bus Interface). It contains a counter, an alarm and has the capability of generating a number of user-defined periodic interrupts. Access to SRTC LP registers can only be done through the SRTC HP and when SRTC HP is powered up.

SRTC HP is partitioned into following units.

- Address Decode
- Counters and registers
  - Nonsecure 47-bit time counter
  - Programmable nonsecure 47-bit alarm with interrupt
  - Periodic alarm with interrupt

The nonsecure time counter can be synchronized with the secured time counter by writing to a SRTC HP bit (time_sync-Time Synchronization). When written, the content of the secure counter is automatically copied to the nonsecure counter.

The following figure shows the SRTC partitioning.

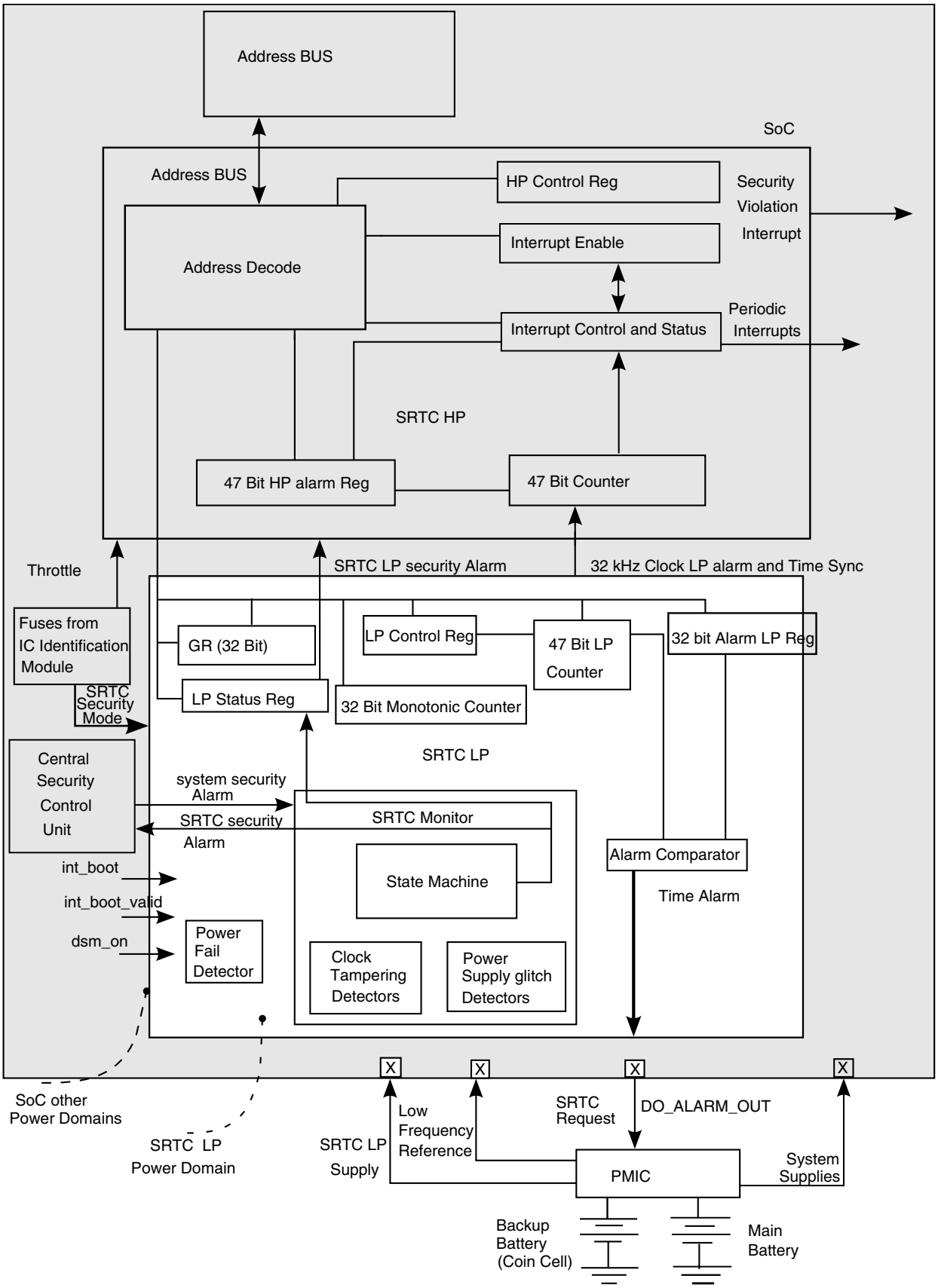


Figure 72-1. Block Diagram

72.1.3 Features

The SRTC includes the following features:

- Secure 47-bit time counter running on 32.768 kHz clock
- Nonsecure 47-bit time counter running on 32.768 kHz clock
- User mode protection and/or ARM's TrustZone mode protection- SRTC secured registers cannot be configured by nonsecured SW (see the Programmable Registers section). A dedicated control bit can allow nonsecured SW to update the secured registers. This bit can be set only by secured SW. In any case, SRTC secured registers cannot be configured when SRTC is locked.
- Re-programming protection:
  - SRTC time cannot be altered after SRTC is locked (when SRTC set to the appropriate security mode)
  - SRTC cannot be disabled after SRTC is locked (when SRTC set to the appropriate security mode)
- Power loss protection:
  - Separated power supply
  - Accurate continuous time is kept during power down system states and main power source losses
  - In the event of a total power loss (main power source and backup power source), SRTC time read is invalidated
  - In the event of unstable voltage power source, SRTC time read is invalidated
  - In the event of low voltage power source that disrupts normal time counting, SRTC time read is invalidated
- Clock source protection:
  - Directly connects to a low frequency crystal (32.768 kHz) or to a PMIC 32.768 kHz clock source
  - In the event that SRTC frequency drops, SRTC time read is invalidated
- SRTC time can be invalidated by a system security alarm
- SRTC can assert a system security alarm and fast interrupt request to alert the processor that time read was invalidated
- Programmable secure 32-bit alarm with interrupt (alarm of the secure- LP section). In case of system power up mode, this alarm generates interrupt to alert the processor. In case of system power down mode, this alarm alerts the power management IC (PMIC) via an external pin to instruct the PMIC to power up the system. Note that this feature does not require any PMIC special support. This signal can be connected to an on-board power on button circuit.
- Programmable nonsecure 47-bit alarm with interrupt (alarm of the nonsecure, HP, section)

- Periodic nonsecure alarm with interrupt
- Secured monotonic counter
- General purpose always powered registers
- Ultra low power consumption. SRTC LP power consumption during system power down state is less than 3 µA.(TYP)
- Separate calibration logic for 47 bit HP and LP counters

72.1.4 Modes of Operations

The SRTC has three functional modes, defined by IC Identification Module dedicated fuses, and can be found in one of four states.

The SRTC modes are as follows:

- Mode #1 - High Security
- Mode #2 - Medium Security
- Mode #3 - Low Security

SRTC states are defined as follows:

- Initialization- SRTC is powered up for the first time or after system POR.
- Nonvalid- SRTC time is not set, time read is not authenticated
- Valid- Valid time was set, time read is authentic
- Failure- SRTC logic detected a failure. Time is invalidated.

See SRTC State Machine, for more details.

NOTE

Detailed information about the SRTC security modes is provided in the Security Reference Manual. Contact your Freescale representative for information about obtaining this document.

72.2 External Signal Description

Table 72-1. Signal Properties[1]

Name	PORT	Function	I/O	Reset	Pull Up
IND_CKIL_IN	CKIL	Input 32.768 kHz Low Frequency Reference clock for the LP Section.	I	0	-
DO_ALARM_OUT	SRTC_ALARM	Output wake up alarm to PMIC on time out. Polarity is active low.	O	0	-

1. This table indicates external signal connected to PAD

72.3 Functional Description

The SRTC is a real time clock with enhanced security capabilities. Its purpose is to provide an accurate time read at all times, regardless of the main system power states and without the need to use an external time source such as external RTC. The following sections describe in detail the theory and basic design principals of the SRTC.

72.3.1 Power and Clock Source

The SRTC is divided into two main sub-blocks based on the power supply.

The SRTC LP is the always powered section of the SRTC. SRTC LP is a separate power domain with its own power supply. No other logical sub-blocks within the SoC boundary share the power supply with this section.

SRTC LP power is supplied by a power management IC (PMIC), such as Freescale MC13783 (Atlas). It is assumed that the power management IC is responsible for supplying a stable voltage level at all times from a main power source or from a back-up power source (for example, a coin-cell battery). The SRTC LP is not connected directly to a backup power source and is not responsible for switching between power sources in case of main power source losses: main battery removals or power cuts (momentary lost of power). It is assumed that the PMIC is responsible for switching between main power source and backup power source as required.

The SRTC LP section is separated from the rest of the SRTC and SoC by means of isolation cells. SRTC receives a signal which indicates a power loss to the SoC, and enables the isolation cells. This signal is deactivated once stable power is restored to the SoC.

Figure 72-2 shows low power domain connectivity. SRTC HP section is in the normal power supply domain and receive power along with the rest of the SoC.

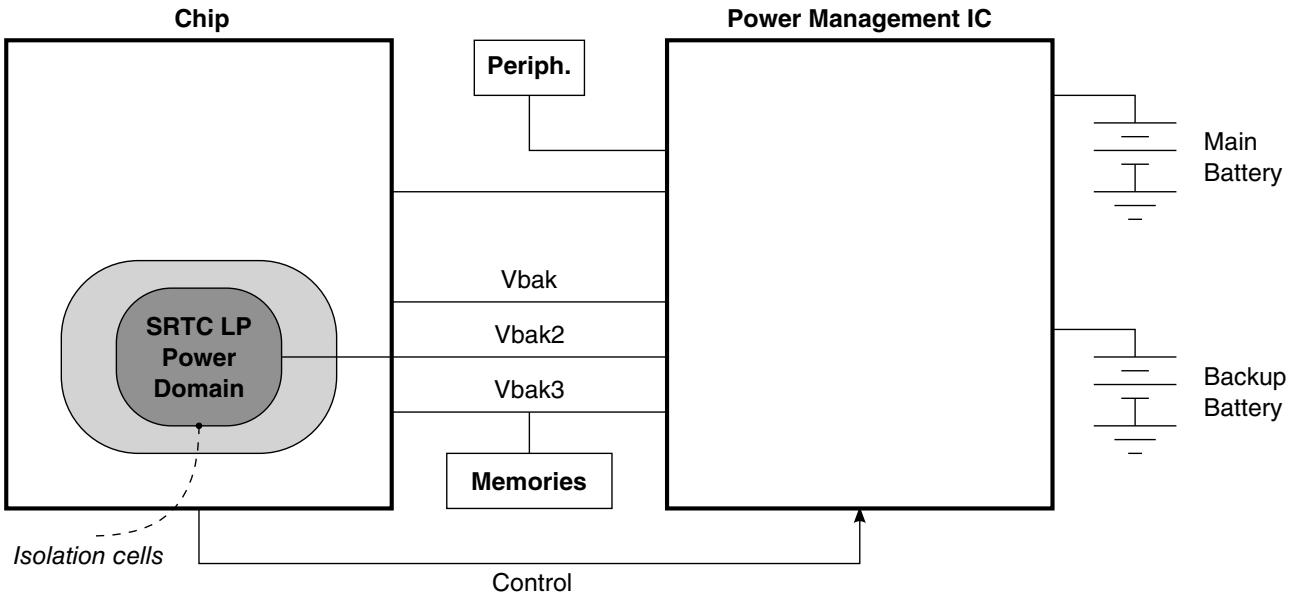


Figure 72-2. SRTC Power Domains

72.3.1.1 Clocks

SRTC runs on two clock sources, high frequency peripherals clock and low frequency timers clock. The high frequency peripheral IP clock is used by the SRTC peripheral bus interface, control and status registers. The low frequency 32.768 kHz clock is the always-active clock used by the SRTC timers and by the PTD and CTD logics. This can be driven directly from an oscillator or from a PMIC clock source.

72.3.2 High Power SRTC (SRTC HP) Description

The SRTC HP contains all the interfaces to the core (IPBus interface). Access to SRTC LP registers can only be done through the SRTC HP and when SRTC HP is powered up. It also contains the following sub- blocks.

72.3.2.1 SRTC HP nonsecured Counter

The SRTC HP incorporates autonomous 47-bit nonsecured time counter. The counter is not active and is reset when system is powered down, however, it remains active during system low-power modes (for example, wait, doze and stop). The nonsecure counter can be used by any application as it has no access restrictions. The SRTC HP counter can be

synchronized with the SRTC LP secure time counter by writing to a specific bit in the SRTC HP control register. After synchronization, time integrity of the nonsecured timer cannot be guaranteed as it has no access limitations

72.3.2.1.1 SRTC HP Counter Calibration

A clock calibration system is provided to adjust the 32,768 cycle counter that generates the 1 Hz timer for SRTC HP timing registers. The general implementation relies on the system processor to measure the 32.768 kHz crystal oscillator against a higher frequency and more accurate system clock such as a TCXO. If the SRTC timer needs a correction, a 5-bit 2-second complement calibration word can be sent via IP to compensate the SRTC for inaccuracy in its reference oscillator as defined in the Table 72-2. The available correction range should be sufficient to ensure that drift accuracy is in compliance with standards for secure time applications like DRM time keeping. Note that the 32.768 kHz oscillator is not affected by SCAL_HP settings; calibration is only applied to the SRTC time base counter. Therefore, the frequency at the clock output Low Frequency Reference Clock is not affected.

The SRTC system calibration is enabled by programming the SCALM_HP for desired behavior by operational mode.

Table 72-2. SRTC HP Calibration Setting

Calibration Settings Code in SCAL_HP[4:0	Correction in Counts per 32768	Relative Correction in ppm
01111	+15	+458
00011	+3	+92
00001	+1	+31
00000	0	0
11111	-1	-31
11101	-3	-92
10001	-15	-458
10000	-16	-488

However, the calibration is only as good as the SCAL_HP data that has been provided, so occasional refreshing is recommended to ensure that any drift-influencing environmental factors have not skewed the clock beyond desired tolerances.

See HP Control Register (SRTC_HPCR), for bit definitions of SCAL_HP and SCALM_HP.

72.3.2.2 SRTC HP nonsecured Counter Alarm

The SRTC HP nonsecured counter has its own 47-bit time alarm register. An alarm is when the 47 bits of HP counter matches with the value of the 47-bit HP alarm register. If X is the value at which the alarm is desired, then this HP Alarm register should be set with value X-1. Note that this register can be updated by any application. The SRTC HP time alarm can generate an interrupt to alert the ARM core to wake it up from one of its low-power modes (for example, wait, doze, stop). Note that this alarm cannot wake up the entire system if it is powered off.

A nonsecure alarm has been provided with mask bit AHP in the SRTC_HPCR.

72.3.2.3 SRTC HP Periodic Alarm

The SRTC HP nonsecure counter can support a generation of a millisecond resolution periodic alarm. The periodic interrupt generation can be set in a range of 1Hz to 32.768 kHz. This interrupt is enabled and set from the SRTC Interrupt enable register. The interrupt source of the periodic alarm is chosen by comparing one of the16 LSB Bits of the 47-bit nonsecured counter (this interrupt is generated when a zero-to-one and one-to-zero transition occurs on the selected bit).

HP SRTC nonsecure counter and its alarms are shown in Figure 72-3.

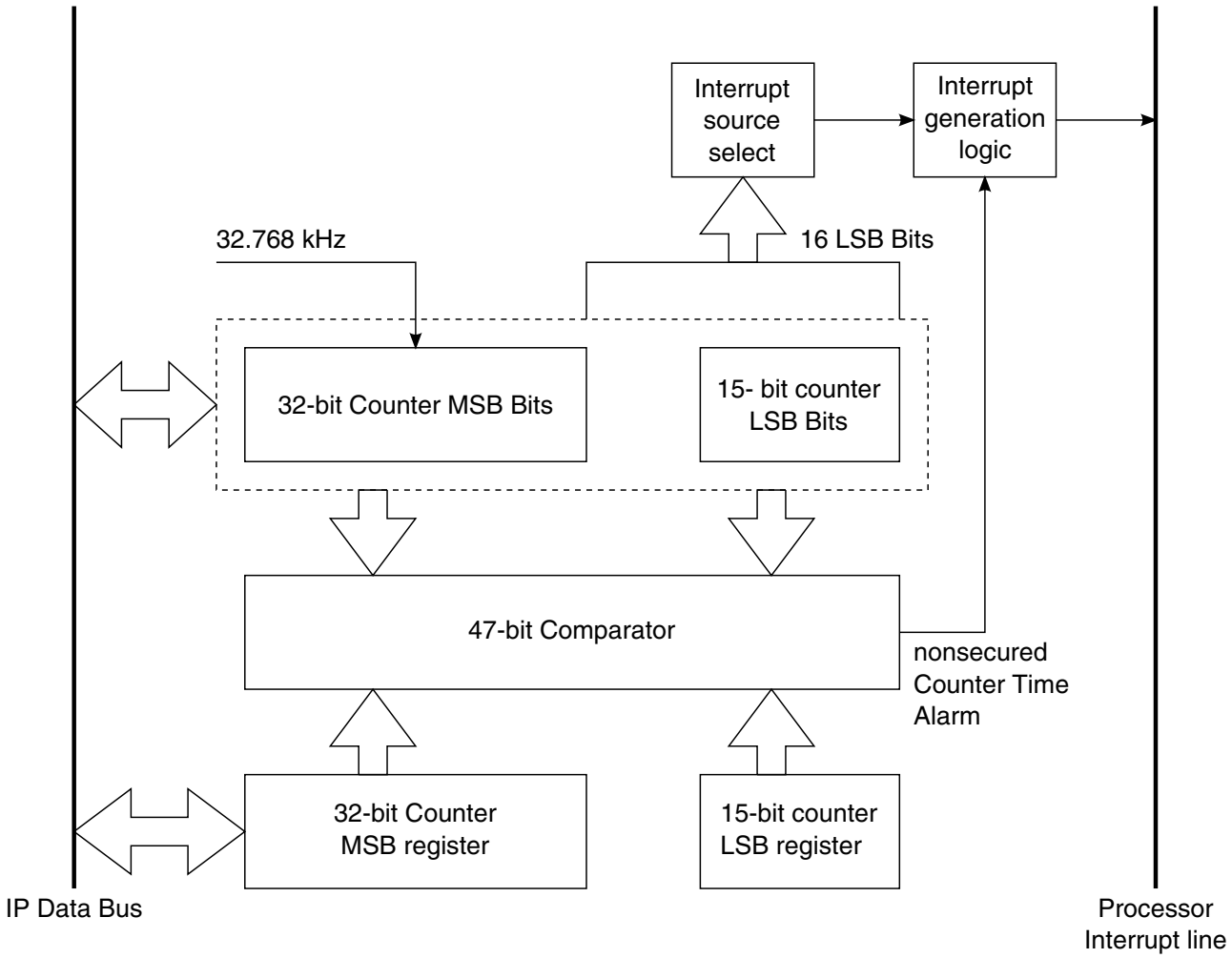


Figure 72-3. SRTC HP nonsecured Counter

72.3.3 Low Power SRTC (SRTC LP) Description

The SRTC LP is a real time clock with enhanced security capabilities. Its purpose is to provide an accurate time read at all times, regardless of the main system power state and without the need to use external on-board time source such as external RTC. The SRTC LP can wake up the system when preset time alarm is asserted. The SRTC LP also contains a monotonic counter and a general purpose register that can be used to store important information while the system is powered down. The following sections describe in detail the theory and basic design principals of the SRTC LP sub-block.

72.3.3.1 SRTC LP Behavior during System Power Down and POR

On system power down the SRTC LP continues to operate normally. It is isolated from the rest of the SoC by means of isolation cells. It keeps the time and retains all parameters stored in its registers. On system power down, the SRTC ignores all system signals (SRTC LP inputs) such as IC Identification Module, SRTC HP, and other input signals. On PoR SRTC LP ignores all system input signals (SRTC LP inputs) until the CCM indicates that the PoR routine ended. This is done to eliminate false signal indications caused by an unstable power state of the system.

72.3.3.2 SRTC LP Secured Counter

The SRTC incorporates autonomous 47-bit secured time counter. Time counter is a non-rollover counter. This means that if the time counter reaches the value of 0xffff_ffff_ffff, it does not rollover. However, the TR bit is set in the SRTC_LPSR and the SRTC enters failure state. This has been done to prevent an attacker from virtually running the time backwards using a higher frequency clock source.

72.3.3.2.1 SRTC LP Counter Calibration

A clock calibration system is provided to adjust the 32,768 cycle counter that generates the 1 Hz timer for SRTC LP timing registers. The general implementation relies on the system processor to measure the 32.768-kHz crystal oscillator against a higher frequency and more accurate system clock such as a TCXO. If the SRTC timer needs a correction, a 5-bit 2-second complement calibration word can be set to compensate the SRTC for inaccuracy in its reference oscillator as defined in the following table.The available correction range should be sufficient to ensure drift accuracy in compliance with standards for secure time applications like DRM time keeping. Note that the 32.768 kHz oscillator is not affected by SCAL_LP settings; calibration is only applied to the SRTC time base counter. Therefore, the frequency at the clock output Low Frequency Reference Clock is not affected.

The SRTC system calibration is enabled by programming the SCALM_LP[1:0] for desired behavior by operational mode.A slight increase in consumption is seen when that calibration circuitry is activated. To minimize consumption and maximize lifetime when the SRTC system is maintained by the coin cell, the SRTC Calibration circuitry can be automatically disabled when main battery contact is lost or if it is so deeply discharged that SRTC power draw is switched to the coin cell (configured with SCALM_LP=01). Because of the low SRTC consumption, SRTC accuracy can be maintained through long periods with the application being shut down, even after the main battery has discharged.

However, calibration is only as good as the SCAL_LP data that has been provided, so occasional refreshing is recommended to ensure that any drift influencing environmental factors have not skewed the clock beyond desired tolerances.

Table 72-3. SRTC LP Calibration Setting

Calibration Settings Code in SCAL_LP[4:0	Correction in Counts per 32768	Relative Correction in ppm
01111	+15	+458
00011	+3	+92
00001	+1	+31
00000	0	0
11111	-1	-31
11101	-3	-92
10001	-15	-458
10000	-16	-488

See LP Control Register (SRTC_LPCR), for bit definition of SCAL_LP and SCALM_LP.

72.3.3.3 SRTC LP Secured Counter Alarm

The SRTC LP section has its own 32-bit time alarm register. An alarm is generated when a 32-bit MSB (bits 46 to 15) value of the LP secure counter (SRTC_LPSCMR) matches with the 32-bit secure alarm register (SRTC_LPSAR). If X is the value at which the alarm is desired, then this SRTC_LPSAR should be set with the value X-1. The alarm register can be updated only by secure application. The SRTC LP time alarm can generate interrupt to alert the core and can wake-up the core from one of its low-power modes (for example, wait, doze, stop). This alarm can also wake-up the entire system by asserting the DO_ALARM_OUT signal to the PMIC. DO_ALARM_OUT is asserted only if the main system is powered off as indicated by power fail circuity or external signal (dsm_on) input to DO_ALARM_OUT is active-low .

The secured alarm has been provided with a separate mask bit, ALP and WAE, in the SRTC_LPCR .

72.3.3.4 Monotonic Counter

Security applications require a monotonic counter that cannot be exhausted or return to any previous value during the life-time of the product.

The monotonic counter is a counter that counts in one direction only. The monotonic counter can never repeat a number implying that it cannot be reset or cycled back to its starting count. The monotonic counter incorporates an innovative mechanism that prevents exhaustive usage of the counter while still allowing short rapid counter increments.

The monotonic counter features are listed below:

- 32-bits counter that can only go up
- Counter increment can only be done by secured SW (by writing any value to the counter's register)
- Average number of updates per predefined period is limited

72.3.3.4.1 Monotonic Counter Roll-Over Protection Mechanism

To prevent a case where a malicious SW rolls over the monotonic counter, a counter update limitation mechanism is implemented. This mechanism assures that the number of updates within a period of 512 seconds is limited. Updates beyond the specified number are ignored and the counter is not incremented.

The number of updates per 512 seconds is set by the IC Identification Module (e-fuse programmed) as follows: (number of updates (average updates per second))

- 8192 (16)
- 4096 (8)
- 2048 (4)
- 1024 (2)
- 512 (1)
- Unlimited

The mechanism gates the increment request signal with output overflow signal of a special 13-bit counter, as shown in the figure below.

Table 72-4 shows the mapping between the IC Identification Module throttle values and number of updates.

Table 72-4. Throttle and Monotonic Counter Updates Mapping

Value of IIM Throttle	Number of Updates per 512 seconds (Average Updates per second)
000	512(1)
001	1024 (2)
010	2048 (4)
011	4096 (8)
100	8192 (16)

Table continues on the next page...

Table 72-4. Throttle and Monotonic Counter Updates Mapping (continued)

Value of IIM Throttle	Number of Updates per 512 seconds (Average Updates per second)
101	8192 (16)
110	8192 (16)
110	8192 (16)
111	Unlimited

On reaching the value of 0xffff_ffff, the monotonic counter does not roll over. The MR bit in SRTC_LPSR is set and SRTC enters failure state.

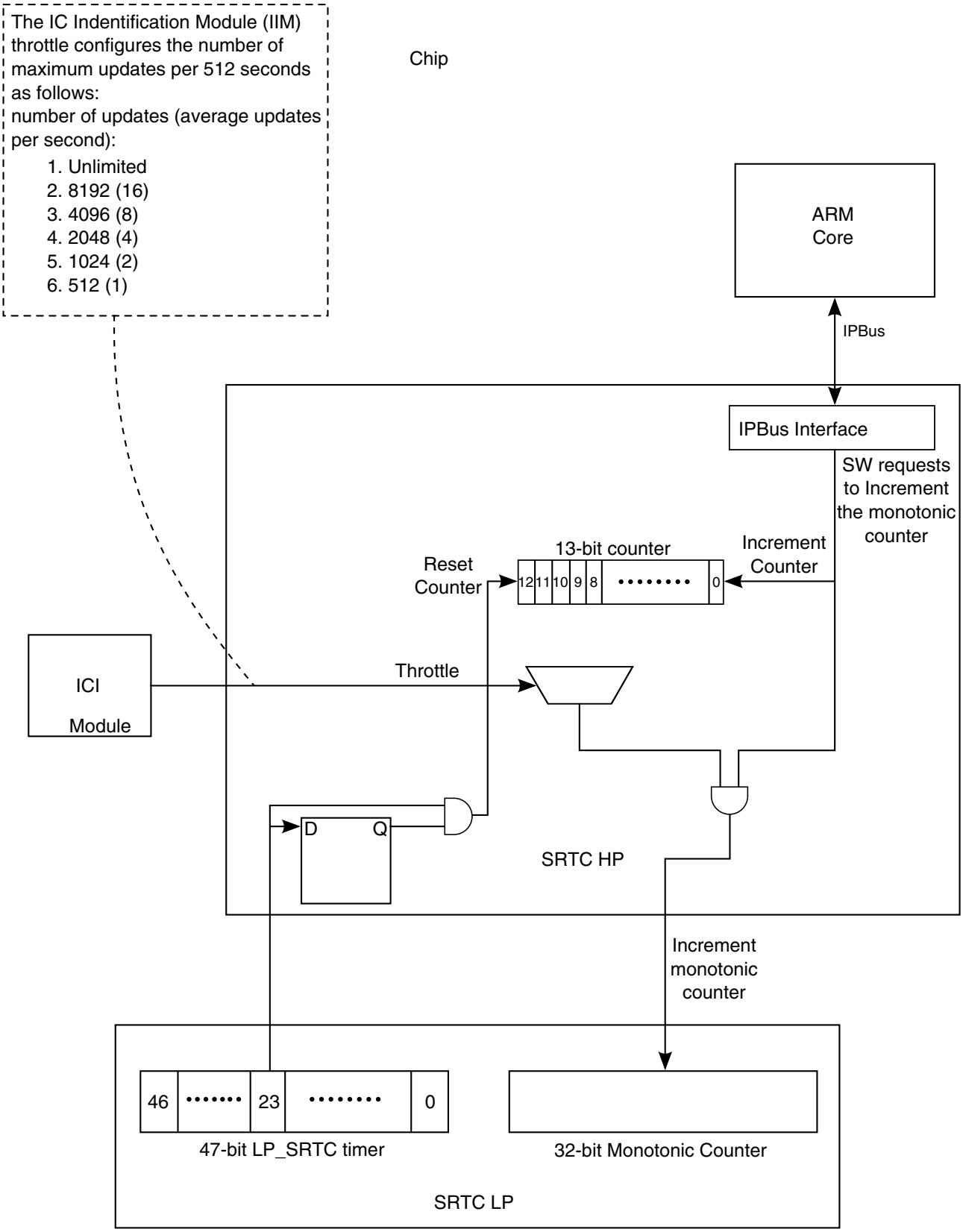


Figure 72-4. Monotonic Counter Roll-Over Protection Mechanism

72.3.3.5 General Purpose Always-Powered Registers

One 32-bits general purpose register that is a part of the SRTC LP and is always powered.

72.3.3.6 SRTC LP Monitor

The SRTC monitor watches for security alarms (SRTC LP and security block alarms) and then shifts the SRTC to failure state, in case of a security violation. The SRTC is comprised of three sub-blocks, SRTC State Machine (SSM), Power Supply Glitch Detector (PGD) and Clock Tampering Detector (CTD).

72.3.3.6.1 SRTC State Machine

The SRTC LP State Machine (SSM) can be found in one of four states:

- Initialization- SRTC failure indication can be cleared. Time read and monotonic counter read are reset and kept reset (except for nonsecure mode). Monotonic counter and secured time cannot be programmed. During this state the power glitch register can be loaded with the 41736166 value. This mode is exited by writing to a IE SRTC control bit. Failure monitors are not active in this state.
- Nonvalid- Time and monotonic counters are not valid and need to be set. Time is not running however time value and monotonic counter can be updated (in nonsecure mode, timer and monotonic counter are not reset and remained active). Failure detectors are active. This mode is exited by writing to a NVE SRTC control bit.
- Valid- Time and monotonic counter read are valid and operational. Re-programing is allowed only in appropriate security mode. Failure detectors are active.
- Failure- SRTC detected a security violation. Time and monotonic read are invalidated. Time and monotonic read operation returns zero (except for nonsecure mode where timer is operational, can be read and reprogrammed).

SRTC states are shown in Figure 72-5.

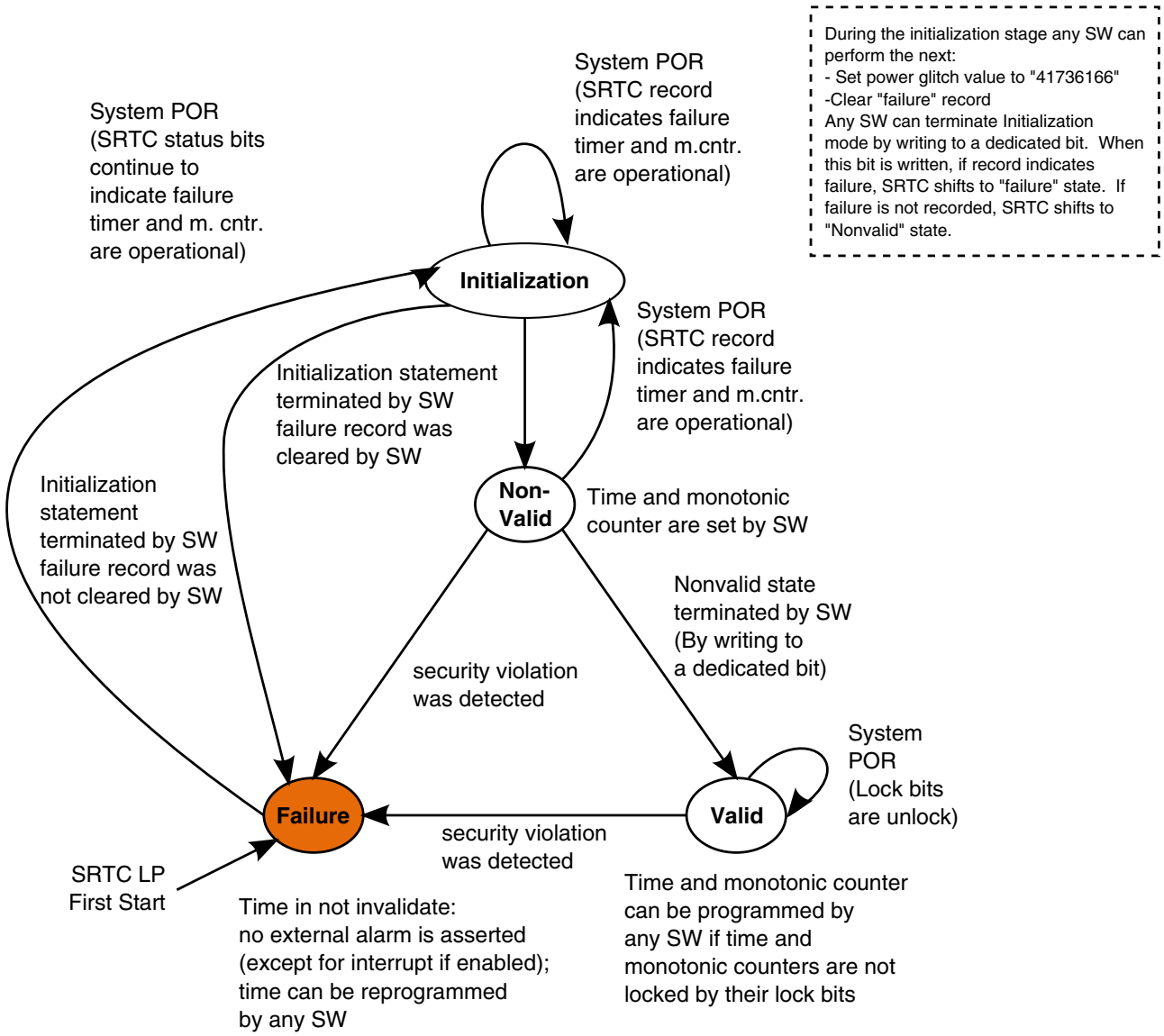


Figure 72-5. SRTC States - Low Security Mode

72.3.3.6.2 Power Supply Glitch Detector (PGD)

SRTC LP power supply can be power gated for short periods to cause state machine or secure counter register value to change. To detect such attack, SRTC incorporates an innovative method to detect and alert the system whenever power supply glitches endanger SRTC registers value. The SRTC glitch detector comprised of a simple register and few logic gates. This register is loaded to a known specific value (41736166) as part of the SRTC initialization process. This register have no reset input and its power-up value is pretty random. Comparing of these register value to a hardened value allow detection of power gating. The power supply glitch detector mechanism is shown in Figure 72-6.

The PGD and the TRI bit are set in the SRTC_LPSR and security alarm is raised to security block on SRTC security alarm (active high) pin and security interrupt (active low) to the core depending upon the SAE bit in the SRTC_LPCR and SI bit HPIENR respectively.

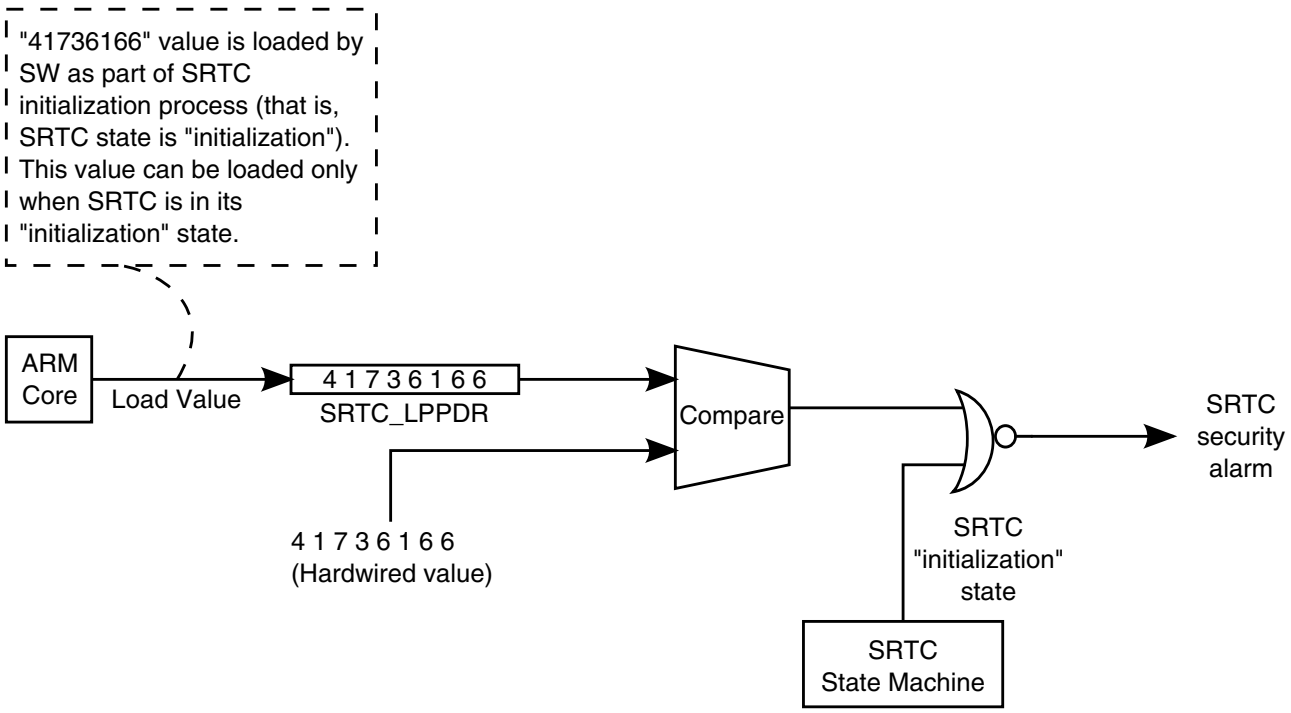


Figure 72-6. Power Supply Glitch Detector

72.3.3.6.3 Clock Tampering Detector (CTD)

External clock removal or frequency reduction can stop or slow down the secure counter time count. Because the clock signal is not generated internally in the SoC, SRTC cannot assure the reliability of the clock source and as consequence, cannot provide absolute assurance of the time read. However, SRTC provides means to detect significant reduction in its clock source frequency. SRTC clock tampering detector can detect frequency drop of 30%-100%. In current SRTC version, only clock frequency drop of 90% or more is detected.

If an attempt to tamper with the SRTC clock is detected, the SRTC enters failure state.

Storing fail indication of the clock monitor is done by SR latch with low, as possible, operating voltage. In this way, even if this register is not functional due to low voltage supply, on system power up, this register recalls the fail indication. The SRTC clock monitor mechanism is shown in Figure 72-7.

The CTD and the TRI bits are set in the SRTC_LPSR, and the security alarm is raised to the security block on the SRTC security alarm (active high) pin and the security interrupt (active low) to the core on the SRTC security Interrupt (Active low) pin SRTC security Interrupt (Active low) depending upon the SAE bit in the SRTC_LPCR and the SI bit in the SRTC_HPIENR, respectively.

72.3.3.6.4 Voltage Level Tampering Detector

Supply voltage of SRTC can be reduced to a value that causes the SRTC FF elements to be stuck at a constant value. Although clock frequency is valid, counter values can be stuck. Such attack has the same effect as clock input removal. Moreover, state machine registers can be also stuck, so even if violation is detected, it is not recorded. To make sure that voltage level is sufficient for proper functionality, the clock monitor does not monitor the raw clock input, but monitors a divided version of the actual real time counter. In this way, the same clock monitor can detect low voltage attack, where time counter LSB FF is stuck. LSB Bit of real time counter should be built from a high threshold transistors (this FF is more sensitive to low voltage than other SRTC FFs).

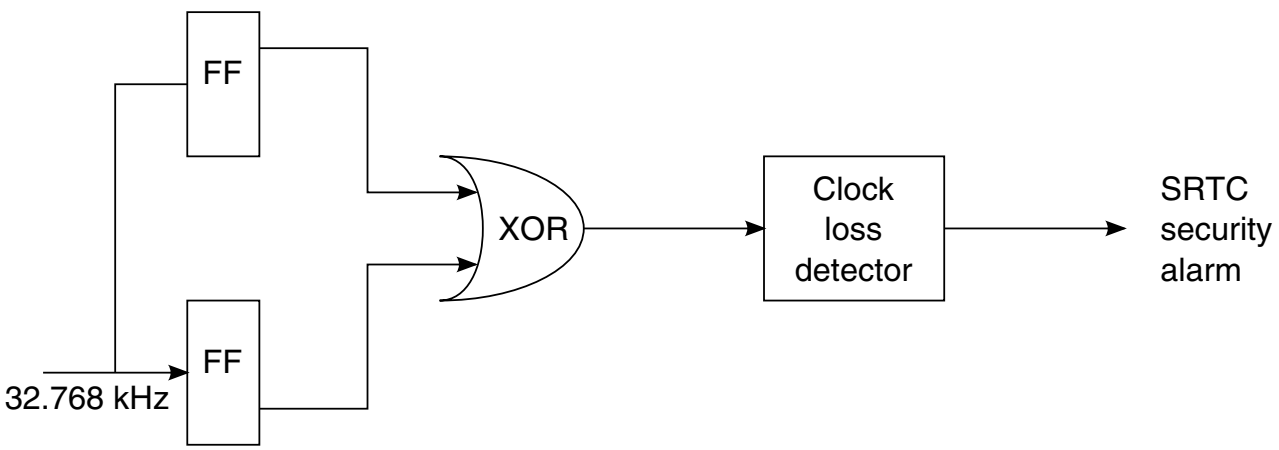


Figure 72-7. Clock Tampering Detector

72.3.3.6.5 Power Fail Detector (PFD)

Power fail detector is provided to detect the power failure in the SoG and isolate the SRTC LP from the rest of the SoG. It is provided with dsm_on input from the SoG. PFD is triggered when the dsm_on is asserted or when the there is power failure in the SoC.

72.4 SRTC Reset and System Power-Up

SRTC LP (always powered up section) is provided with its own POR, which provides power-on-reset as the power is switched on. This POR clears all registers inside SRTC.

SRTC LP section is in the Initialization state of the SRTC state machine upon its POR. Upon System POR the security mode of the SRTC is determined. If the security mode is Low (no security mode) then the SRTC LP counter and monotonic counter can be operational in the Initialization state.

Both the HP and LP section of the SRTC are provided with the their own software reset bit SWR in their respective control registers. The SWR bit of HP section can be set at any time and it resets all the HP section. This is a self clearing bit and always read as zero.

The SWR bit of LP section resets all the LP section. This bit can be set only in the Initialization state of SRTC state machine. This is a self clearing bit and always read as zero.

72.5 SRTC Interrupts and Alarms

The SRTC provides following interrupt and alarm lines

- $\overline{\text{SRTC Interrupt (Active low)}}$: Consolidated (ORed) interrupt pin for the following SRTC interrupts.
  - Periodic interrupt from 1 Hz to 32768 Hz frequency
  - Alarm interrupt of the HP section, see SRTC HP nonsecured Counter Alarm, for more details.
  - Alarm interrupt of the LP section, see SRTC LP Secured Counter Alarm, for more details.
  - Initialization Exit and Non Valid Exit Interrupt. Indicates the exit from the Initialization and Nonvalid state of the SRTC LP state machine.
  - Write done interrupt of the LP and HP section. To take care of the asynchronous Low Frequency Reference Clock, all the write from IP domain is synchronized to the Low Frequency Reference Clock domain. To indicate this delay the write done interrupt are provided for both the LP and HP section. Write done indicates that the consolidated status of the various IP writes and is triggered when all the registers have been updated. A separate interrupt is provided for the IP write to LP and HP section.
- SRTC security interrupt (Active low)- Security interrupt line to the core upon setting of the TRI bit in the LP section.

- SRTC security alarm- Security alarm to the security block upon setting of the TRI bit in the LP section.
- DO_ALARM_OUT- Time alarm of the LP section to the external pad, see SRTC LP Secured Counter Alarm, for more details. This alarm is generated when the system is powered off as indicated by the power fail circuitry or by external signal dsm_on input to the block. DO_ALARM_OUT is active low .

72.6 Initialization Information/Application Information

72.6.1 Flow Chart of SRTC LP Operation

See Figure 72-8 for the illustration of the flow chart of a typical SRTC LP operation.

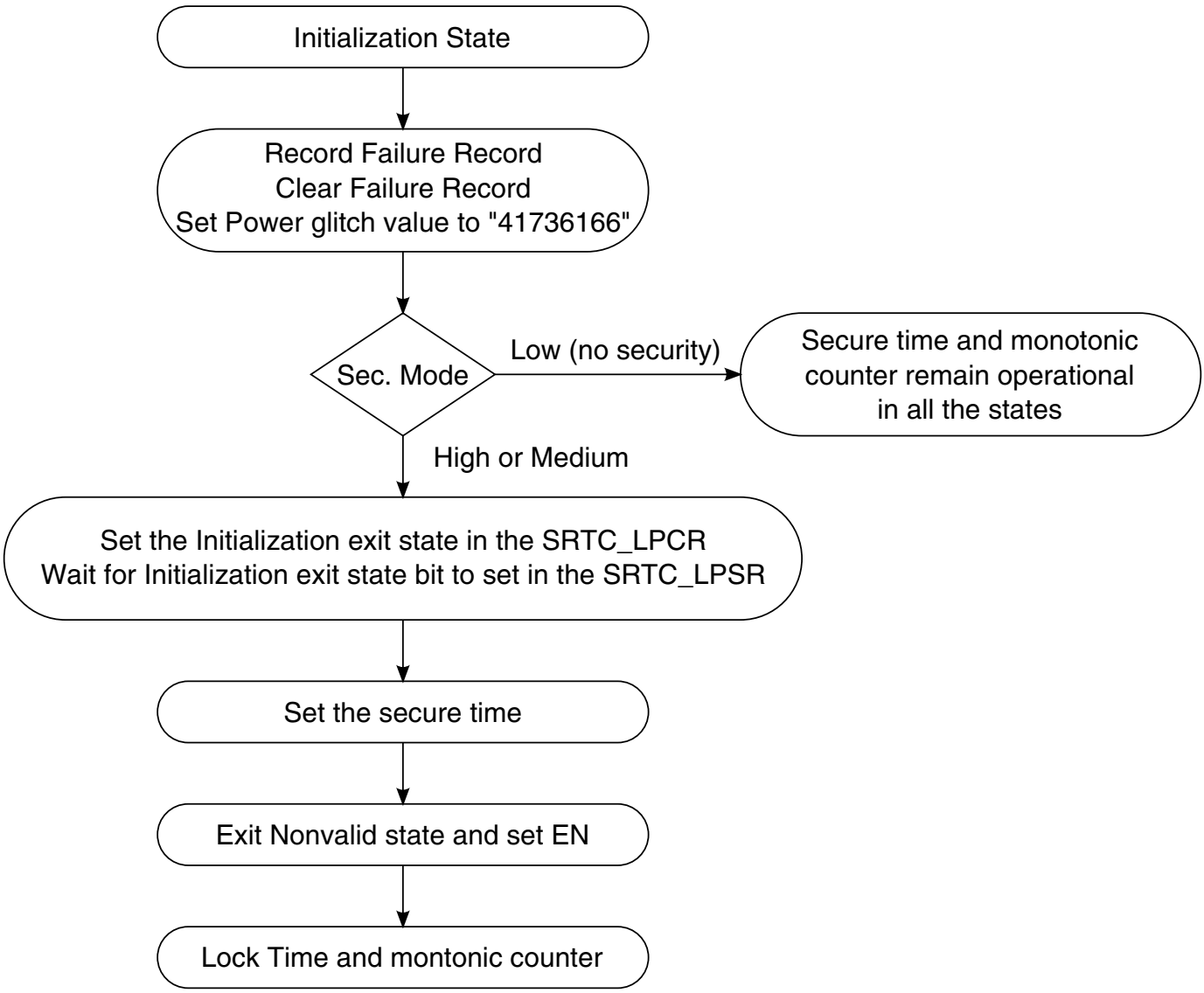


Figure 72-8. Flow Chart of SRTC LP Operations

72.6.2 Flow Chart of SRTC HP Operation

See Figure 72-9 for the illustration of the flow chart of a typical SRTC HP operation.

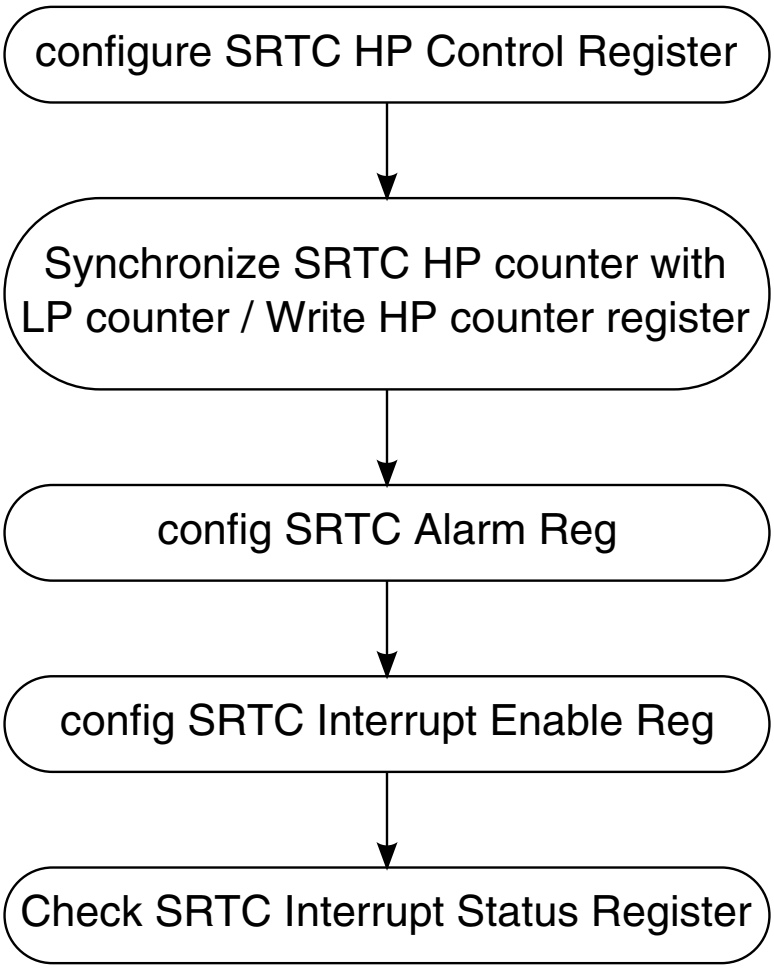


Figure 72-9. Flow Chart of SRTC HP Operations

72.7 Software Restrictions

The SRTC has the following software restrictions.

- The 32-bit MSB part and 15-bit LSB part of all the 47-bit registers should be updated together.
- The registers are not byte-writeable. All Valid byte enable should be asserted high for updating the register.
- LTC bit in the SRTC_LPCR. After it is set, this bit prevents write access to 47 bit SRTC LP secure counter by all application. Once set this bit is reset only upon system POR.

- LMC bit in the SRTC_LPCR. Once set this bit prevents write access to 32 bit SRTC LP secure monotonic counter by all application.Once set this bit is reset only upon system POR.
- SWR bit in the SRTC_LPCR. This bit can set by the secure application only in the Initialization state of the SRTC state machine.
- CTD, PTD and TRI bits in the SRTC_LPSR are unaffected by the System POR and are cleared only in Initialization state of SRTC state machine.
- SRTC LP power supply glitch detector register (SRTC_LPPDR). This register is write accessible only in the Initialization state of the SRTC state machine.
- Security mode. Security mode are retained upon system POR and can only be updated to higher security mode
- HPCR_TS that is, time synchronize operation and write on SRTC_HPCLR and SRTC_HPCMR shall not be performed at the same time. There is one count difference between LP and HP count value after TS has been performed
- There are two Low Frequency Reference Clock delay between the IP write operation on 47 bit counter, and value is updated on these counter.
- For write operation on monotonic counter the SRTC EN shall be enabled by setting the EN_LP. The software then shall wait for write pending LP bit (WPLP in SRTC_HPISR) to be zero before any write operation on Monotonic counter.
- IE and NVE in SRTC_LPCR are used to end the Initialization and Nonvalid states, respectively. They are used in the same sequence, that is, Initialization ends first, followed by Nonvalid exit.

When reading either SRTC_LPSCLR or SRTC_LPSCMR register, the software will read twice. The value of the register is valid when both the read data matches. In case there is mismatch the software will again read it twice.

72.8 Programmable Registers

The SRTC has fifteen 32-bit registers.

SRTC registers are marked with User and Secure access permissions. The distinction between User and Secure accesses does not apply when SRTC is configured for Low Security or the NSA bit is set in the SRTC_LPCR register.

- TrustZone Supervisor mode for TrustZone based systems (ARM1176 or further versions); or
- Supervisor mode for non-TrustZone based systems.

SRTC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FA_4000	LP Secure Counter MSB Register (SRTC_LPSCMR)	32	R/W	Unaffected	72.8.1/ 4424
53FA_4004	LP Secure Counter LSB Register (SRTC_LPSCLR)	32	R/W	See section	72.8.2/ 4424
53FA_4008	LP Secure Alarm Register (SRTC_LPSAR)	32	R/W	Unaffected	72.8.3/ 4425
53FA_400C	LP Secure Monotonic Counter Register (SRTC_LPSMCR)	32	R/W	Unaffected	72.8.4/ 4425
53FA_4010	LP Control Register (SRTC_LPCR)	32	R/W	0000_0000h	72.8.5/ 4426
53FA_4014	LP Status Register (SRTC_LPSR)	32	w1c	0000_0000h	72.8.6/ 4429
53FA_4018	LP Power Supply Glitch Detector Register (SRTC_LPPDR)	32	R/W	Unaffected	72.8.7/ 4432
53FA_401C	LP General Purpose Register (SRTC_LPGR)	32	R/W	Unaffected	72.8.8/ 4432
53FA_4020	HP Counter MSB Register (SRTC_HPCMR)	32	R/W	0000_0000h	72.8.9/ 4433
53FA_4024	HP Counter LSB Register (SRTC_HPCLR)	32	R/W	0000_0000h	72.8.10/ 4434
53FA_4028	HP Alarm MSB Register (SRTC_HPAMR)	32	R/W	0000_0000h	72.8.11/ 4434
53FA_402C	HP Alarm LSB Register (SRTC_HPALR)	32	R/W	0000_0000h	72.8.12/ 4435
53FA_4030	HP Control Register (SRTC_HPCR)	32	R/W	0000_0008h	72.8.13/ 4435
53FA_4034	HP Interrupt Status Register (SRTC_HPISR)	32	w1c	0000_0000h	72.8.14/ 4437
53FA_4038	HP Interrupt Enable Register (SRTC_HPIENR)	32	R/W	0000_0000h	72.8.15/ 4440

72.8.1 LP Secure Counter MSB Register (SRTC_LPSCMR)

The LP secure counter MSB register contains the 32 most significant bits (MSB bits 46 to 15) of the 47-bit LP secure counter.

Address: SRTC_LPSCMR is 53FA_4000h base + 0h offset = 53FA_4000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MLPSC																															
Reset	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*

* Notes:
- u = Unaffected by reset.

SRTC_LPSCMR field descriptions

Field	Description
31–0 MLPSC	MSB value of the LP secure counter Contains MSB bits 46 to 15 of the 47 Bit LP secure counter.

72.8.2 LP Secure Counter LSB Register (SRTC_LPSCLR)

The LP secure counter LSB register contains the 15 least significant bits LSB (bits 14 to 0) of the 47-bit LP secure counter.

Address: SRTC_LPSCLR is 53FA_4000h base + 4h offset = 53FA_4004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LLPSC																0															
W																																
Reset	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

* Notes:
- u = Unaffected by reset.

SRTC_LPSCLR field descriptions

Field	Description
31–17 LLPSC	LSB value of the LP secure counter Contains LSB bits 14 to 0 of 47 Bit LP secure counter.
16–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

72.8.3 LP Secure Alarm Register (SRTC_LPSAR)

The LP secure alarm register contains the 32-bit value of LP section secure alarm register.

Address: SRTC_LPSAR is 53FA_4000h base + 8h offset = 53FA_4008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LPSALM																															
Reset	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*

* Notes:
- u = Unaffected by reset.

SRTC_LPSAR field descriptions

Field	Description
31–0 LPSALM	LP Secure alarm Contains 32 bit value of LP secure alarm register, see SRTC LP Secured Counter Alarm, for more details.

72.8.4 LP Secure Monotonic Counter Register (SRTC_LPSMCR)

The LP secure monotonic counter register contains the 32-bit value of LP section monotonic counter. The counter is increment by writing any value to the counters register.

Address: SRTC_LPSMCR is 53FA_4000h base + Ch offset = 53FA_400Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LPSMC																															
Reset	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*

* Notes:
- u = Unaffected by reset.

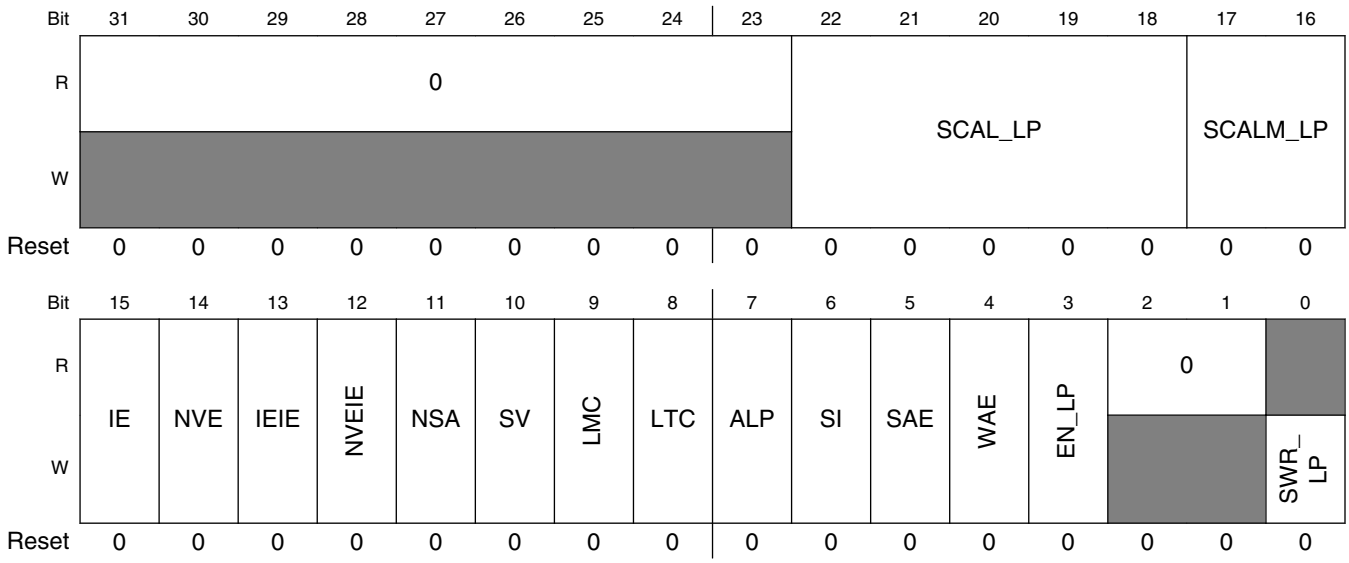
SRTC_LPSMCR field descriptions

Field	Description
31–0 LPSMC	LP Secure Monotonic counter Contains 32 bit value of LP secure monotonic counter.

72.8.5 LP Control Register (SRTC_LPCR)

The LP control register (SRTC_LPCR) contains various control bits of the LP section of SRTC.

Address: SRTC_LPCR is 53FA_4000h base + 10h offset = 53FA_4010h



SRTC_LPCR field descriptions

Field	Description
31–23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–18 SCAL_LP	SRTC Calibration Value These five bits indicates signed calibration value for SRTC LP timer. In this, SCAL_LP[21:18] represent 4 bit value and SCAL_LP[22] represents the sign. Hence, SCAL_LP[22:18] allow calibration +15 to -16 in SRTC LP timer.
17–16 SCALM_LP	SRTCCALMODEL_LP Decides the SRTC LP calibration mode. 00 SRTC LP Timer calibration disabled. 01 SRTC LP Timer calibration enabled in all modes except coin cell mode 10 SRTC LP calibration disabled (Reserved). 11 SRTC LP Calibration enabled in all modes.
15 IE	Initialization state exit bit When set, this bit ends Initialization state of the SRTC state machine. **NOTE:** IE bit is a control bit to end the Initialization state. For the current state of SRTC LP see the state_lp field in the SRTC_LPSR. **NOTE:** This bit is unaffected by reset.

Table continues on the next page...

SRTC_LPCR field descriptions (continued)

Field	Description
	0 SRTC is in Initialization state. 1 End of Initialization state
14 NVE	Nonvalid state exit bit When set this bit ends the Nonvalid state of the SRTC state machine. **NOTE:** NVE bit is a control bit to end the Non Valid state. For the current state of SRTC LP see the state_lp field in the SRTC_LPSR. **NOTE:** This bit is unaffected by reset. 0 SRTC is in Nonvalid state. 1 End of Nonvalid state.
13 IEIE	Initialization state exit interrupt enable bit This bit enables the interrupt on exit from the Initialization State of the SRTC state machine. **NOTE:** This bit is unaffected by reset. 0 An interrupt is disabled. 1 An interrupt is enabled.
12 NVEIE	Nonvalid state exit interrupt enable bit This bit enables the interrupt on exit from the Nonvalid state of the SRTC state machine. **NOTE:** This bit is unaffected by reset. 0 An interrupt is disabled. 1 An interrupt is enabled.
11 NSA	Nonsecure access When set this bit allows nonsecure application to access secure register. This bit is reset by the system por. 0 Nonsecure access of secure register is not allowed 1 Nonsecure access is allowed
10 SV	Security Violation When set this bit move the SRTC LP to failure state upon termination of Initialization state. Also if the current state is Nonvalid/Valid the SRTC LP is moved to failure state. This bit is reset by the system por. **NOTE:** Setting the SV invalidates the SRTC_LPSCR and SRTC_LPSMCR contents. 0 No security violation 1 Security Violation
9 LMC	Lock monotonic counter When set this bit prevents write access to 32-bit secure LP monotonic counter register (SRTC_LPSMCR) by all application.Once set this bit cannot be reset except by system por. 0 Write access is allowed. 1 Write access is not allowed
8 LTC	Lock time counter

Table continues on the next page...

SRTC_LPCR field descriptions (continued)

Field	Description
	When set this bit prevents write access to secure 47-bit LP counter (SRTC_LPSCMR and SRTC_LPSCLR) by all application.Once set this bit cannot be reset except by system por. 0 Write access is allowed. 1 Write access is not allowed
7 ALP	Alarm Flag of LP Section Enables the alarm interrupt to the core. See SRTC LP Secured Counter Alarm, for more details. **NOTE:** This bit is unaffected by reset. 0 Alarm interrupt disabled. 1 Alarm interrupt enabled.
6 SI	Security Interrupt Enable bit This bit enables the security interrupt to be raised to core on SRTC security Interrupt (Active low). Security interrupt is raised upon setting of the of time read invalidate status bit **NOTE:** This bit is unaffected by reset. 0 Security interrupt disabled. 1 Security interrupt enabled
5 SAE	Security Alarm Enable bit This bit enables the security alarm to be raised to security block on SRTC security alarm (Active high) pin. Security alarm is raised upon setting of the of time read invalidate status bit **NOTE:** This bit is unaffected by reset. 0 Security alarm disabled. 1 Security alarm enabled
4 WAE	Wakeup Alarm Enable bit This bit enables the time alarm to be raised on DO_ALARM_OUT pin. See SRTC LP Secured Counter Alarm, for more details.DO_ALARM_OUT is asserted only if the main system is powered off as indicated by power fail circuity or external signal (dsm_on) input to block. DO_ALARM_OUT's polarity is active low **NOTE:** This bit is unaffected by reset. 0 Time alarm disabled. 1 Time alarm enabled.
3 EN_LP	Enables/Disables the LP secure time and secure monotonic counter When set the secure LP counter and monotonic counter becomes operational. This bit cannot be reset once either of LMC or LTC is set. In high and Medium this bit can only be set in Valid state **NOTE:** This bit is unaffected by reset. 0 Disable the real-time clock and monotonic counter 1 Enable the real-time clock and monotonic counter
2–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 SWR_LP	Software reset

Table continues on the next page...

SRTC_LPCR field descriptions (continued)

Field	Description
	Resets the LP section of the SRTC to its default state.This bit can be set only in the Initialization state of the SRTC state machine. It can be set by a secure application only. This is self clearing bit is always read as zero. 0 No effect 1 Reset the block to its default state

72.8.6 LP Status Register (SRTC_LPSR)

The LP status register (LPSR) register is used by the core to read the status of the various bits.

Address: SRTC_LPSR is 53FA_4000h base + 14h offset = 53FA_4014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IES	NVES	STATE_LP		SM		IT		EAD	TR	MR	ALP	CTD	PGD	TRI	
W	w1c	w1c							w1c	w1c	w1c	w1c	w1c	w1c	w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRTC_LPSR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15 IES	Initialization state exit status bit This bit indicates the end of Initialization state of the SRTC state machine. **NOTE:** IES and NVES are w1c status bits corresponding to the Initialization Exit and Nonvalid exit interrupts. **NOTE:** This bit is unaffected by reset. 0 SRTC is in Initialization state. 1 SRTC has exit out of Initialization state
14 NVES	Nonvalid state exit status bit This bit indicate the end of Nonvalid state of the SRTC state machine. **NOTE:** This bit is unaffected by reset. 0 SRTC is in Nonvalid state. 1 SRTC has exit out of Nonvalid state.

Table continues on the next page...

SRTC_LPSR field descriptions (continued)

Field	Description
13–12 STATE_LP	STATE_LP Specify the present state of SRTC LP state machine. These are read-only bits. **NOTE:** This bit is unaffected by reset. 00 Initialize. 01 Nonvalid 10 Valid 11 Failure
11–10 SM	Security Mode Specify the present security mode of the SRTC as determined from the IC Identification Module fuses. These are read-only bits. **NOTE:** This bit is unaffected by reset. 00 Security Mode: Low (no security). Other values: see Security Reference Manual .
9–7 IT	IIM Throttle Specify the 3-bit value of the throttle bits for monotonic counter from the IC Identification Module. These are read-only bits. SeeTable 72-4. **NOTE:** This bit is unaffected by reset.
6 EAD	External alarm detected Indicates that Security alarm generated by the external security block has been detected or external boot is performed in High and Medium security mode. **NOTE:** This bit is unaffected by reset. 0 No security alarm detected. 1 Security alarm detected
5 TR	Time Rollover Indicates that the 47 Bit LP secure counter has reached the value of 0xffff_ffff_ffff. When counter reaches this maximum value, TRI bit is also set and SRTC enters failure state. Once set this bit is not reset by system por, and can be cleared only by secure application in Initialization state of SRTC state machine. **NOTE:** This bit is unaffected by reset. 0 Counter has not reached its maximum value. 1 Counter has reached its maximum value.
4 MR	Monotonic counter Rollover Indicates that the 32 Bit LP secure monotonic counter has reached the value of 0xffff_ffff. When counter reaches this maximum value, TRI bit is also set and SRTC enters failure state. Once set this bit is not reset by system por, and can be cleared only by secure application in Initialization state of SRTC state machine. **NOTE:** This bit is unaffected by reset.

Table continues on the next page...

SRTC_LPSR field descriptions (continued)

Field	Description
	0 Counter has not reached its maximum value. 1 Counter has reached its maximum value
3 ALP	Alarm Flag of LP section Indicates that the SRTC LP secured counter alarm has occured. See SRTC LP Secured Counter Alarm, for more details. **NOTE:** This bit is unaffected by reset. 0 No alarm occurred. 1 An alarm has occurred.
2 CTD	Clock Tampering Detected This bit is set when clock tampering is detected. Once set this bit is not reset by system por, and can be cleared only by secure application in Initialization state of SRTC state machine **NOTE:** This bit is unaffected by reset. 0 No Clock Tampering Detected. 1 Clock Tampering Detected.
1 PGD	Power supply glitch Detected This bit is set when glitch in the power supply is detected. Once set this bit is not reset by system por, and can be cleared only by secure application in Initialization state of SRTC state machine. **NOTE:** This bit is unaffected by reset. 0 No Power supply glitch Detected. 1 Power supply glitch Detected.
0 TRI	Time Read Invalidate This bit is set when SRTC LP state machine is moved to failure by any of the security violation. It indicates the time maintained by the SRTC has been invalidated.Once set this bit is not reset by system por, and can be cleared only by secure application in Initialization state of SRTC state machine. **NOTE:** This bit is unaffected by reset. 0 Time read is validated. 1 Time read is invalidated.

72.8.7 LP Power Supply Glitch Detector Register (SRTC_LPPDR)

The LP power supply glitch detector register (SRTC_LPPDR) provides a 32-bit read write register, which is used for storing power glitch value logic as described in Power Supply Glitch Detector (PGD). This register may be written only in the Initialization state of the SRTC state machine.

Address: SRTC_LPPDR is 53FA_4000h base + 18h offset = 53FA_4018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LPPDR																															
Reset	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*

* Notes:
- u = Unaffected by reset.

SRTC_LPPDR field descriptions

Field	Description
31–0 LPPDR	LP power supply glitch detector register Contains 32 bit value of LP power supply glitch detector register.

72.8.8 LP General Purpose Register (SRTC_LPGR)

The LP general purpose register can be used to retain data while the system power is down. Some bits of this register are used by the boot ROM.

NOTE

SW_ISO will not be set by any software except by the power fail routine. Once set it can only be reset by negedge of power fail detector, that is, by removing power from the SoC. This is provided as an additional safeguard over the Power fail detector's late triggering upon power failure in SoC.

NOTE

An alternate way to reset this bit is to assert dsm_on input.

Address: SRTC_LPGR is 53FA_4000h base + 1Ch offset = 53FA_401Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ISO	SW_ROM		LPGR[13:16]												
Reset	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LPGR[15:0]															
Reset	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*	u*

* Notes:
- u = Unaffected by reset.

SRTC_LPGR field descriptions

Field	Description
31 SW_ISO	Software isolation bit When set it isolates the SRTC LP portion from the rest of the SoC. It is reset by the negedge of the power fail output of internal power fail detector. 0 SRTC is not isolated. 1 SRTC is isolated from the rest of the SoC.
30–29 SW_ROM	Reserved for use by boot ROM.
28–0 LPGR	LP general purpose register Contains user-specified data.

72.8.9 HP Counter MSB Register (SRTC_HPCMR)

The HP counter MSB register contains the 32 most significant bits (MSB bits 46 to 15) of the 47-bit HP counter.

Address: SRTC_HPCMR is 53FA_4000h base + 20h offset = 53FA_4020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MHPC																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

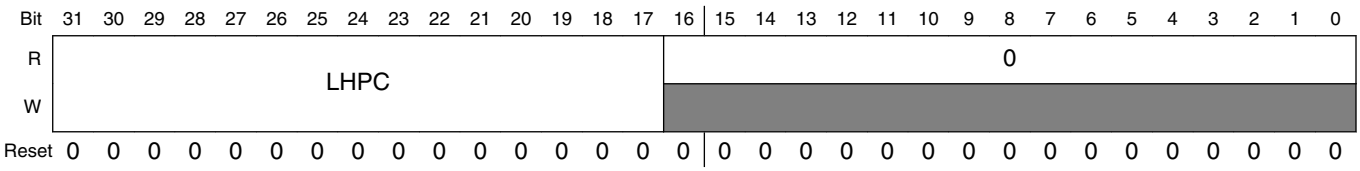
SRTC_HPCMR field descriptions

Field	Description
31–0 MHPC	MSB value of the HP counter Contains MSB bits 46 to 15 of the 47 Bit SRTC HP counter.

72.8.10 HP Counter LSB Register (SRTC_HPCLR)

The HP counter LSB register contains the 15 least significant bits LSB (bits 14 to 0) of the 47-bit SRTC HP counter.

Address: SRTC_HPCLR is 53FA_4000h base + 24h offset = 53FA_4024h



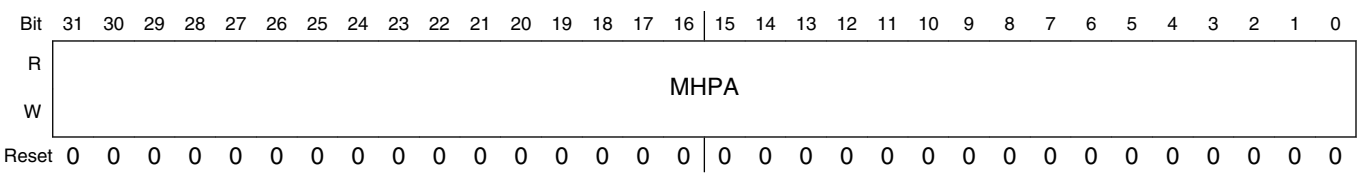
SRTC_HPCLR field descriptions

Field	Description
31–17 LHPC	LSB value of the HP counter Contains LSB bits 14 to 0 of 47 Bit SRTC HP counter.
16–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

72.8.11 HP Alarm MSB Register (SRTC_HPAMR)

The HP alarm MSB register contains the 32 most significant bits (MSB bits 46 to 15) of the 47-bit HP alarm.

Address: SRTC_HPAMR is 53FA_4000h base + 28h offset = 53FA_4028h



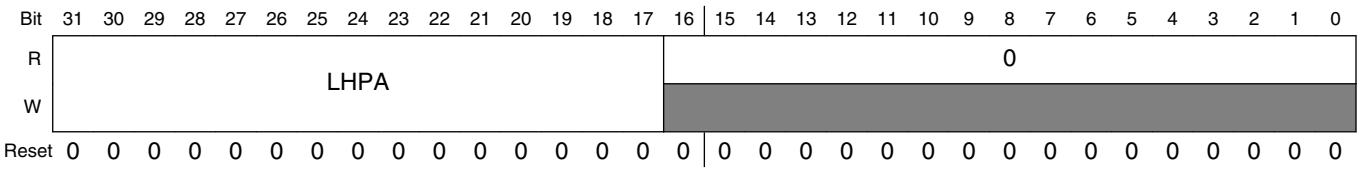
SRTC_HPAMR field descriptions

Field	Description
31–0 MHPA	MSB value of the HP alarm Contains MSB bits 46 to 15 of the 47 Bit SRTC HP alarm. See SRTC HP nonsecured Counter Alarm, for more details.

72.8.12 HP Alarm LSB Register (SRTC_HPALR)

The HP alarm LSB register contains the 15 least significant bits LSB (bits 14 to 0) of the 47-bit SRTC HP alarm.

Address: SRTC_HPALR is 53FA_4000h base + 2Ch offset = 53FA_402Ch



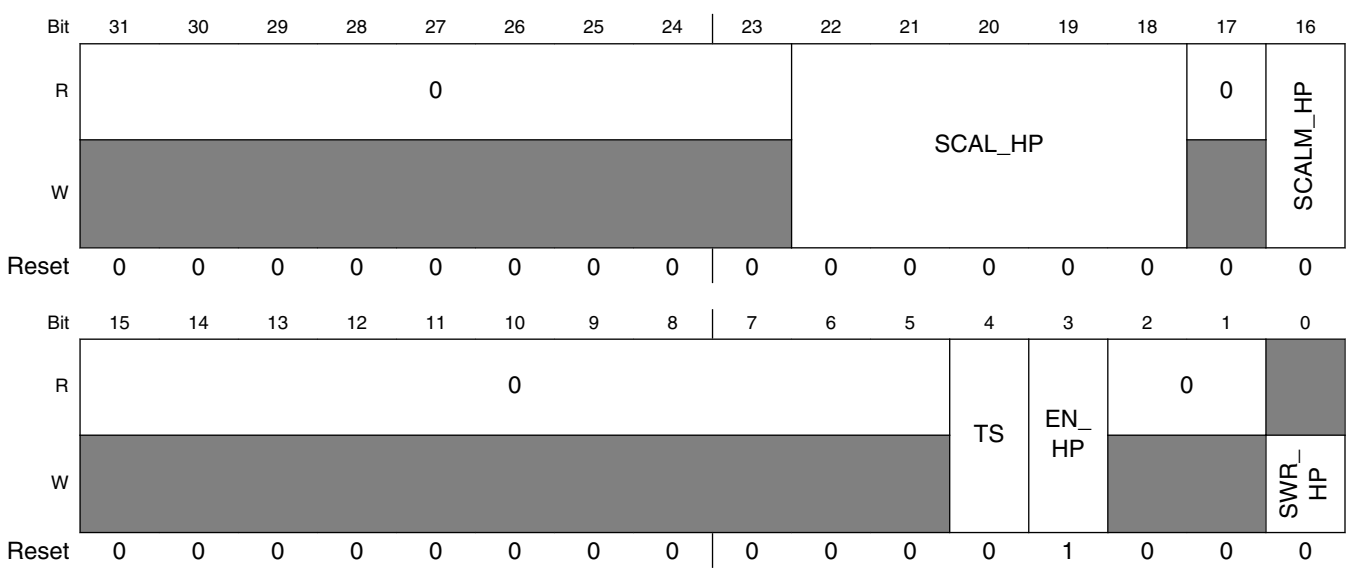
SRTC_HPALR field descriptions

Field	Description
31–17 LHPA	LSB value of the HP alarm Contains LSB bits 14 to 0 of 47 Bit SRTC HP alarm, see SRTC HP nonsecured Counter Alarm, for more details.
16–0 Reserved	This read-only field is reserved and always has the value zero. Reserved

72.8.13 HP Control Register (SRTC_HPCR)

The HP control register (SRTC_HPCR) contains various control bits of the HP section of SRTC.

Address: SRTC_HPCR is 53FA_4000h base + 30h offset = 53FA_4030h



SRTC_HPCR field descriptions

Field	Description
31–23 Reserved	This read-only field is reserved and always has the value zero. Reserved
22–18 SCAL_HP	SRTC Calibration Value These five bits indicates signed calibration value for SRTC HP timer. In this, SCAL_HP[21:18] represent 4 bit value and SCAL_HP[22] represents the sign. Hence, SCAL_HP[22:18] allow calibration +15 to -16 in SRTC HP timer.
17 Reserved	This read-only field is reserved and always has the value zero. Reserved
16 SCALM_HP	SRTCCALMODE_HP Decides the SRTC HP calibration mode. 0 SRTC HP Timer calibration disabled (default). 1 SRTC HP Timer calibration enabled
15–5 Reserved	This read-only field is reserved and always has the value zero. Reserved
4 TS	Time synchronize When set this updates the 47 bit HP counter with the 47 bit LP secure counter. This bit remains set, while the counter is being synchronized, after which it self clears. 0 No Action. 1 Time is synchronized
3 EN_HP	Enables/Disables the HP Time counter 0 Disable the real-time clock 1 Enable the real-time clock
2–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 SWR_HP	Software reset Resets the HP Section of block to its default state. SWR self clears to zero when 0 No effect 1 Reset the block to its default state

72.8.14 HP Interrupt Status Register (SRTC_HPISR)

The HP interrupt status register (SRTC_HPISR) indicates the status of the various periodic, alarm and status interrupts. When an event of the types included in this register occurs, then the bit is set in this register regardless of its corresponding interrupt enable bit. Interrupts may occur while the system clock is idle or in sleep mode. Every interrupt status bit is independent of each other. See SRTC Interrupts and Alarms, for more information on different interrupts.

Address: SRTC_HPISR is 53FA_4000h base + 34h offset = 53FA_4034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0										WPLP	WPHP	WDLP	WDHP	0	AHP
W													w1c	w1c		w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PI15	PI14	PI13	PI12	PI11	PI10	PI9	PI8	PI7	PI6	PI5	PI4	PI3	PI2	PI1	PI0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRTC_HPISR field descriptions

Field	Description
31–22 Reserved	This read-only field is reserved and always has the value zero. Reserved
21 WPLP	Write pending LP Indicates that the write request are pending in the LP section. This is read-only bit. 0 No write requests are pending. 1 Write requests are pending.
20 WPHP	Write pending HP Indicates that the write request are pending in the HP section bit. This is read-only bit. 0 No write requests are pending 1 Write requests are pending
19 WDLP	Write Done LP Indicates that the write request are completed in the LP section. This is w1c bit.

Table continues on the next page...

SRTC_HPISR field descriptions (continued)

Field	Description
	NOTE: The information conveyed by the WPLP and WDLP is about delayed write operation on LP section due to synhronization. WPLP is active high read only bit which indica.tes a pending write operation on LP. WDLP is w1c status bit which is set when the write operation on the LP is completed 0 Write requests are pending. 1 Write requests are completed.
18 WDHP	Write Done HP Indicates that the write request are completed in the HP section. This is w1c bit. **NOTE:** The information conveyed by the WPHP and WDHP is about delayed write operation on HP section due to synhronization. WPHP is active high read only bit which indicates a pending write operation on HP. WDHP is w1c status bit which is set when the write operation on the HP is completed. 0 Write requests are pending. 1 Write requests are completed.
17 Reserved	This read-only field is reserved and always has the value zero. Reserved
16 AHP	Alarm Flag of HP section Indicates that the HP Nonsecured counter alarm has occurred, see SRTC HP nonsecured Counter Alarm, for more details. 0 No alarm interrupt occurred. 1 An alarm interrupt has occurred.
15 PI15	Periodic Interrupt Flag at 32768 Hz Frequency Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 32768 Hz. 0 No interrupt occurred. 1 An interrupt occurred.
14 PI14	Periodic Interrupt Flag at 16384 Hz Frequency Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 16384 Hz. 0 No interrupt occurred. 1 An interrupt occurred.
13 PI13	Periodic Interrupt Flag at 8192 Hz Frequency Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 8192 Hz. 0 No interrupt occurred. 1 An interrupt occurred.
12 PI12	Periodic Interrupt Flag at 4096 Hz Frequency Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 4096 Hz. 0 No interrupt occurred. 1 An interrupt occurred.
11 PI11	Periodic Interrupt Flag at 2048 Hz Frequency Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 2048 Hz.

Table continues on the next page...

SRTC_HPISR field descriptions (continued)

Field	Description
	0 No interrupt occurred. 1 An interrupt occurred.
10 PI10	Periodic Interrupt Flag at 1024 Hz Frequency Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 1024 Hz. 0 No interrupt occurred. 1 An interrupt occurred.
9 PI9	Periodic Interrupt Flag at 512 Hz Frequency Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 512 Hz. 0 No interrupt occurred. 1 An interrupt occurred.
8 PI8	Periodic Interrupt Flag at 256 Hz Frequency Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 256 Hz. 0 No interrupt occurred. 1 An interrupt occurred.
7 PI7	Periodic Interrupt Flag at 128 Hz Frequency Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 128 Hz. 0 No interrupt occurred. 1 An interrupt occurred.
6 PI6	Periodic Interrupt Flag at 64 Hz Frequency Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 64 Hz. 0 No interrupt occurred. 1 An interrupt occurred.
5 PI5	Periodic Interrupt Flag at 32 Hz Frequency Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 32 Hz. 0 No interrupt occurred. 1 An interrupt occurred.
4 PI4	Periodic Interrupt Flag at 16 Hz Frequency Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 16 Hz. 0 No interrupt occurred. 1 An interrupt occurred.
3 PI3	Periodic Interrupt Flag at 8 Hz Frequency Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 8 Hz. 0 No interrupt occurred. 1 An interrupt occurred.
2 PI2	Periodic Interrupt Flag at 4 Hz Frequency Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 4 Hz.

Table continues on the next page...

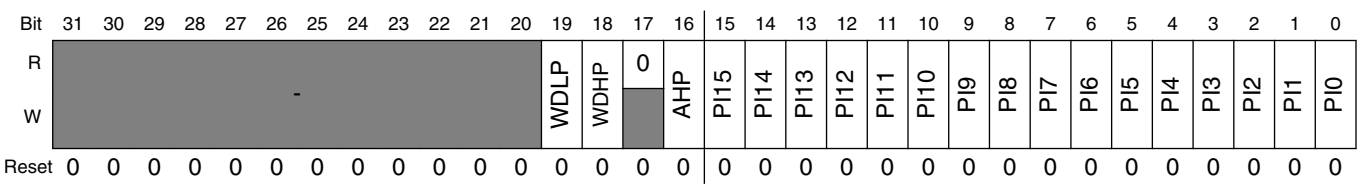
SRTC_HPISR field descriptions (continued)

Field	Description
	0 No interrupt occurred. 1 An interrupt occurred.
1 PI1	Periodic Interrupt Flag at 2 Hz Frequency Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 2 Hz. 0 No interrupt occurred. 1 An interrupt occurred.
0 PI0	Periodic Interrupt Flag at 1 Hz Frequency Indicates that an interrupt has occurred. If enabled, this bit is periodically set at a rate of 1 Hz. 0 No interrupt occurred. 1 An interrupt occurred.

72.8.15 HP Interrupt Enable Register (SRTC_HPIENR)

The HP interrupt enable register (SRTC_HPIENR) is used to enable/disable the various periodic, alarm and security interrupts. Masking an interrupt bit has no effect on its corresponding status bit. Every interrupt enable bit is independent of the other.

Address: SRTC_HPIENR is 53FA_4000h base + 38h offset = 53FA_4038h



SRTC_HPIENR field descriptions

Field	Description
31–20 -	Reserved
19 WDLP	Write Done LP Enable/Disable the Write done interrupt of LP section. 0 An interrupt is disabled. 1 An interrupt is enabled.
18 WDHP	Write Done HP Enable/Disable the Write done interrupt of HP section. 0 An interrupt is disabled. 1 An interrupt is enabled.
17 Reserved	This read-only field is reserved and always has the value zero. Reserved

Table continues on the next page...

SRTC_HPIENR field descriptions (continued)

Field	Description
16 AHP	Alarm Flag of HP section Enable/Disable the alarm interrupt of HP section, see SRTC HP nonsecured Counter Alarm, for more details. 0 An interrupt is disabled. 1 An interrupt is enabled.
15 PI15	Periodic Interrupt Flag at 32768 Hz Frequency Enable/Disable the Periodic Interrupt Flag at 32768 Hz Frequency Interrupt. 0 An interrupt is disabled. 1 An interrupt is enabled.
14 PI14	Periodic Interrupt Flag at 16384 Hz Frequency Enable/Disable the Periodic Interrupt Flag at 16384 Hz Frequency Interrupt. 0 An interrupt is disabled. 1 An interrupt is enabled.
13 PI13	Periodic Interrupt Flag at 8192 Hz Frequency Enable/Disable the Periodic Interrupt Flag at 8192 Hz Frequency Interrupt. 0 An interrupt is disabled. 1 An interrupt is enabled.
12 PI12	Periodic Interrupt Flag at 4096 Hz Frequency Enable/Disable the Periodic Interrupt Flag at 4096 Hz Frequency Interrupt. 0 An interrupt is disabled. 1 An interrupt is enabled.
11 PI11	Periodic Interrupt Flag at 2048 Hz Frequency Enable/Disable the Periodic Interrupt Flag at 2048 Hz Frequency Interrupt. 0 An interrupt is disabled. 1 An interrupt is enabled.
10 PI10	Periodic Interrupt Flag at 1024 Hz Frequency Enable/Disable the Periodic Interrupt Flag at 1024 Hz Frequency Interrupt. 0 An interrupt is disabled. 1 An interrupt is enabled.
9 PI9	Periodic Interrupt Flag at 512 Hz Frequency Enable/Disable the Periodic Interrupt Flag at 512 Hz Frequency Interrupt. 0 An interrupt is disabled. 1 An interrupt is enabled.
8 PI8	Periodic Interrupt Flag at 256 Hz Frequency Enable/Disable the Periodic Interrupt Flag at 256 Hz Frequency Interrupt.

Table continues on the next page...

SRTC_HPIENR field descriptions (continued)

Field	Description
	0 An interrupt is disabled. 1 An interrupt is enabled.
7 PI7	Periodic Interrupt Flag at 128 Hz Frequency Enable/Disable the Periodic Interrupt Flag at 128 Hz Frequency Interrupt. 0 An interrupt is disabled. 1 An interrupt is enabled.
6 PI6	Periodic Interrupt Flag at 64 Hz Frequency Enable/Disable the Periodic Interrupt Flag at 64 Hz Frequency Interrupt. 0 An interrupt is disabled. 1 An interrupt is enabled.
5 PI5	Periodic Interrupt Flag at 32 Hz Frequency Enable/Disable the Periodic Interrupt Flag at 32 Hz Frequency Interrupt. 0 An interrupt is disabled. 1 An interrupt is enabled.
4 PI4	Periodic Interrupt Flag at 16 Hz Frequency Enable/Disable the Periodic Interrupt Flag at 16 Hz Frequency Interrupt. 0 An interrupt is disabled. 1 An interrupt is enabled.
3 PI3	Periodic Interrupt Flag at 8 Hz Frequency Enable/Disable the Periodic Interrupt Flag at 8 Hz Frequency Interrupt. 0 An interrupt is disabled. 1 An interrupt is enabled.
2 PI2	Periodic Interrupt Flag at 4Hz Frequency Enable/Disable the Periodic Interrupt Flag at 4 Hz Frequency Interrupt. 0 An interrupt is disabled. 1 An interrupt is enabled.
1 PI1	Periodic Interrupt Flag at 2 Hz Frequency Enable/Disable the Periodic Interrupt Flag at 2 Hz Frequency Interrupt. 0 An interrupt is disabled. 1 An interrupt is enabled.
0 PI0	Periodic Interrupt Flag at 1 Hz Frequency Enable/Disable the Periodic Interrupt Flag at 1 Hz Frequency Interrupt 0 An interrupt is disabled. 1 An interrupt is enabled.

Chapter 73
Synchronous Serial Interface (SSI)

73.1 Overview

This block guide presents the Synchronous Serial Interface (SSI), and discusses the architecture, the programming model, the operating modes, and initialization of SSI.

The SSI is a full-duplex, serial port that allows the chip to communicate with a variety of serial devices. These serial devices can be standard CODer-DECoder (CODECs), Digital Signal Processors (DSPs), microprocessors, peripherals, and popular industry audio codecs that implement the inter-IC sound bus standard (I2S) and Intel AC97 standard.

SSI is typically used to transfer samples in a periodic manner. The SSI consists of independent transmitter and receiver sections with independent clock generation and frame synchronization.

The figure below illustrates the organization of the SSI. It consists of control registers to set up the port, status register, separate transmit and receive circuits with FIFO registers, and separate serial clock and frame sync generation for the transmit and receive sections. The second set of Tx and Rx FIFOs, replicates the logic used for the first set of FIFOs.

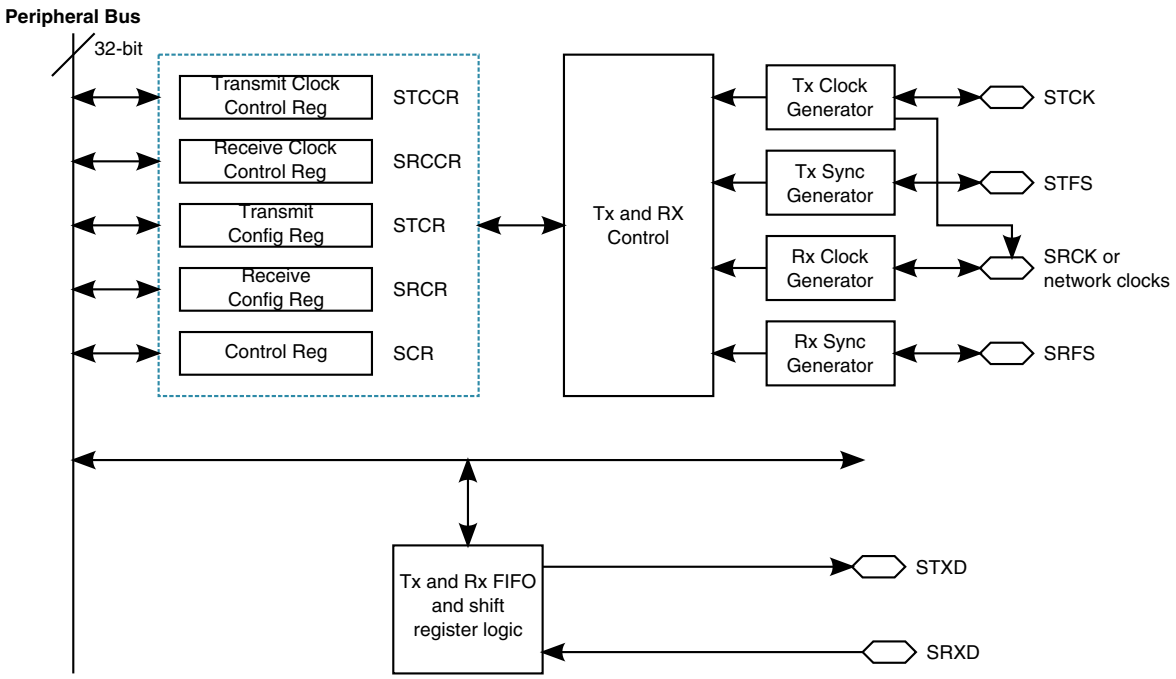


Figure 73-1. SSI Block Diagram

73.1.1 Features

The SSI includes the following features:

- Independent (asynchronous) or shared (synchronous) transmit and receive sections with separate or shared internal/external clocks and frame syncs, operating in Master or Slave mode.
- Normal mode operation using frame sync
- Network mode operation allowing multiple devices to share the port with as many as thirty-two time slots
- Gated Clock mode operation requiring no frame sync
- 2 sets of Transmit and Receive FIFOs. Each of the four FIFOs is 15x32 bits. The two sets of Tx/Rx FIFOs can be used in Network mode to provide 2 independent channels for transmission and reception
- Programmable data interface modes such as I2S, LSB, MSB aligned
- Programmable word length (8, 10, 12, 16, 18, 20, 22 or 24 bits)
- Program options for frame sync and clock generation
- Programmable I2S modes (Master, Slave or Normal). Max audio sampling rate is 196kHz. Min audio sampling rate is 8kHz. Network clock (as an oversampling clock to external device) available as output from SRCK in I2S Master mode
- AC97 support. Max frame rate is 48kHz. Min frame rate is 8kHz.

- Completely separate clock and frame sync selections for the receive and transmit sections. In AC97 standard, the clock is taken from an external source and frame sync is generated internally.
- External network clock input for use in I2S Master mode. Programmable network clock of the sampling frequency available as output in master mode at SRCK, when operated in sync mode.
- Programmable internal clock divider
- Time Slot Mask Registers for reduced ARM platform overhead (for both Tx and Rx)
- SSI power-down feature
- Programmable wait states for ARM platform accesses

73.1.2 Modes of Operation

SSI has the following basic operating modes.

- Normal Mode : Asynchronous protocol, Synchronous protocol
  - Normal Mode Transmit
  - Normal Mode Receive
- Network Mode : Asynchronous protocol, Synchronous protocol
  - Network Mode Transmit
  - Network Mode Receive
- Gated Clock Mode : Synchronous protocol only
- I2S Mode
- AC97 Mode
  - AC97 Fixed Mode (SSI.SACNT[1]=0)
  - AC97 Variable Mode (SSI.SACNT[1]=1)

73.2 External Signal Description

73.2.1 Signals Overview

The Synchronous Serial Interface (SSI) can be connected directly to the external pins or through the Digital Audio Multiplexer (AUDMUX). Refer to the AUDMUX chapter for programming details of the various multiplexing options.

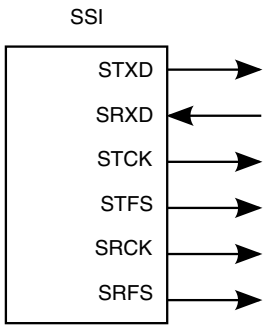
Table 73-1. Off-Chip Block Signals

Name	I/O	Function	Reset State	Pull up
SRCK	I/O	Serial Receive Clock. SRCK can be used as either an input or an output. This clock signal is used by the receiver in asynchronous mode and is always continuos. During synchronous mode, the STCK port is used instead for clocking in data. In SSI synchronous modes, this port can be used as an output for the network clock (oversampling clock) . In I2S master mode, this signal can be used to output the network clock to an external CODEC.	0	Passive
SRFS	I/O	Serial Receive Frame Sync. The SRFS port can be used as either an input or an output. The frame sync is used by the receiver to synchronize the transfer of data. The frame sync signal can be one bit or one word in length and can occur one bit before the transfer of data or right at the transfer of data. If SRFS is configured as an input, the external device should drive SRFS during the rising edge of STCK or SRCK.	0	Passive
SRXD	I	Serial Receive Data. The SRXD port is an input and is used to bring serial data into the Receive Data Shift Register.	-	-
STCK	I/O	Serial Transmit Clock. The STCK port can be used as either an input or an output. This clock signal is used by the transmitter and can be either continuous or gated. During Gated Clock mode, data on the STCK port is valid only during the transmission of data, otherwise it is pulled to the inactive state. In Synchronous mode, this port is used by both the transmit and receive sections.	0	Passive
STFS	I/O	Serial Transmit Frame Sync. The STFS port can be used as either an input or an output. The frame sync is used by the transmitter to synchronize the transfer of data. The frame sync signal can be one bit or one word in length and can occur one bit before the transfer of data or right at the transfer of data. In Synchronous mode, this port is used by both the transmit and receive sections. In Gated Clock mode, frame sync signals are not used. If STFS is configured as an input, the external device should drive STFS during the rising edge of STCK if TSCKP is +ve edge triggered. The external device should drive STFS during the falling edge of STCK if TSCKP is -ve edge triggered.	0	Passive
STXD	O	Serial Transmit Data. The STXD port is an output and transmits data from the Serial Transmit Shift Register. The STXD port is an output port when data is being transmitted and is disabled between data word transmissions and on the trailing edge of the bit clock after the last bit of a word is transmitted.	0	Passive

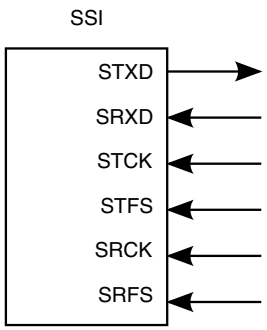
The following figure shows the main SSI configurations. These ports support all transmit and receive functions with continuous or gated clock as shown.

NOTE

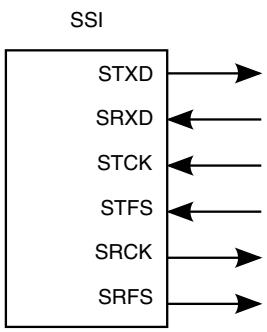
Gated clock implementations do not require the use of the frame sync ports (STFS and SRFS).



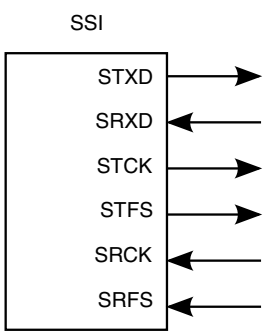
SSI Internal Continuous Clock for TX/RX (RXDIR = 1, TXDIR = 1, RFDIR = 1, TFDIR = 1, SYN = 0)



SSI External Continuous Clock for TX/RX (RXDIR = 0, TXDIR = 0, RFDIR = 0, TFDIR = 0, SYN = 0)



SSI Internal Continuous Clock for RX (RXDIR = 1, TXDIR = 0, RFDIR = 1, TFDIR = 0, SYN = 0)
SSI External Continuous Clock for TX



SSI Internal Continuous Clock for TX (RXDIR = 0, TXDIR = 1, RFDIR = 0, TFDIR = 1, SYN = 0)
SSI External Continuous Clock for RX

Figure 73-2. Asynchronous (SYN=0) SSI Configurations-Continuous Clock

See the following figure for an example of the port signals for an 8-bit data transfer. Continuous and gated clock signals are shown, as well as the bit-length frame sync signal and the word-length frame sync signal.

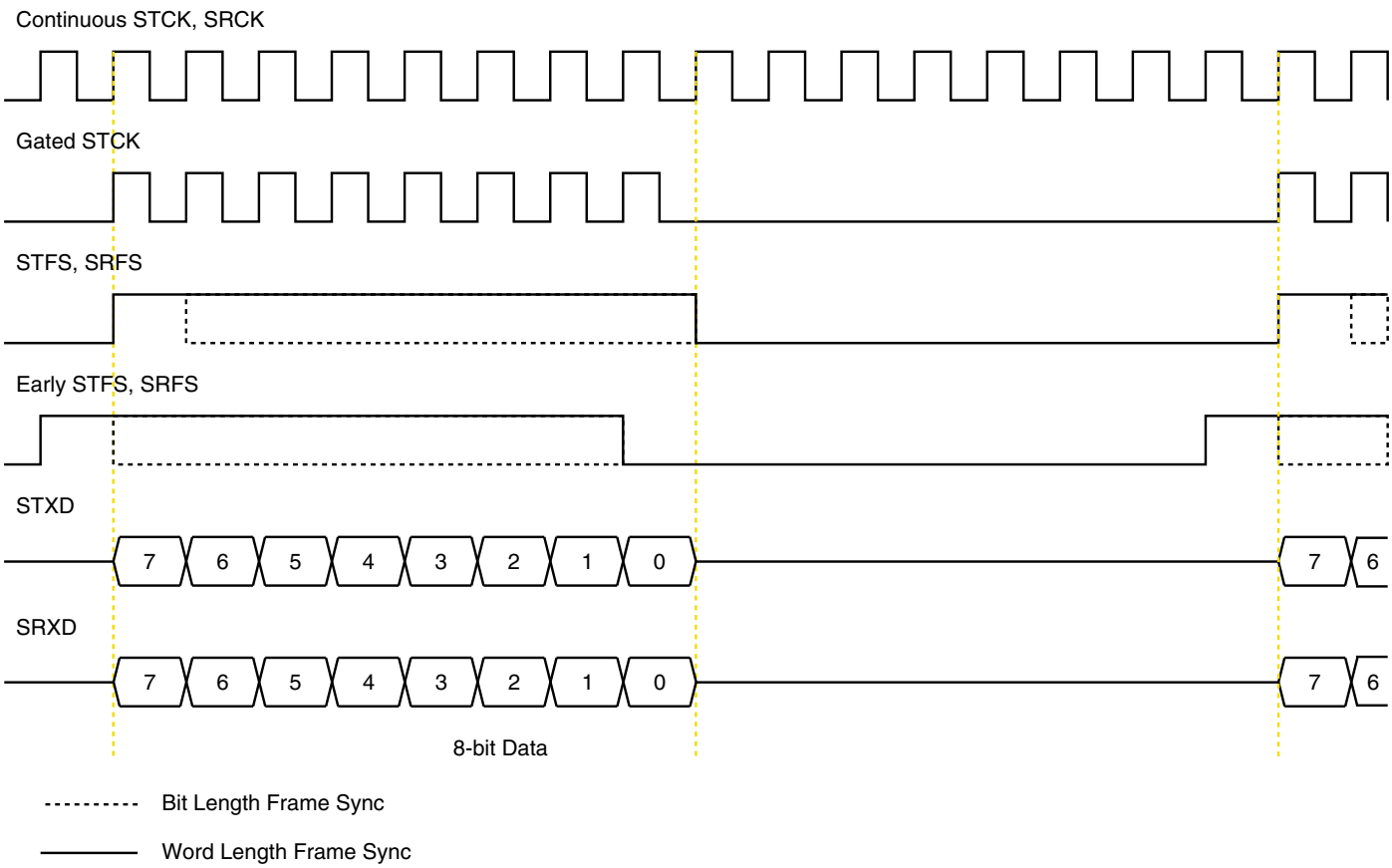


Figure 73-3. Serial Clock and Frame Sync Timing

See the table below for list of clock pin configurations.

Table 73-2. Clock Pin Configurations

SYN	RXDIR	TXDIR	RFDIR	TFDIR	SRCK	STCK	SRFS	STFS
Asynchronous Mode								
0	0	0	0	0	RCK in	TCK in	RFS in	TFS in
0	0	0	0	1	RCK in	TCK in	RFS in	TFS out
0	0	0	1	0	RCK in	TCK in	RFS out	TFS in
0	0	0	1	1	RCK in	TCK in	RFS out	TFS out
0	0	1	0	0	RCK in	TCK out	RFS in	TFS in
0	0	1	0	1	RCK in	TCK out	RFS in	TFS out
0	0	1	1	0	RCK in	TCK out	RFS out	TFS in
0	0	1	1	1	RCK in	TCK out	RFS out	TFS out

Table continues on the next page...

Table 73-2. Clock Pin Configurations (continued)

SYN	RXDIR	TXDIR	RFDIR	TFDIR	SRCK	STCK	SRFS	STFS
0	1	0	0	0	RCK out	TCK in	RFS in	TFS in
0	1	0	0	1	RCK out	TCK in	RFS in	TFS out
0	1	0	1	0	RCK out	TCK in	RFS out	TFS in
0	1	0	1	1	RCK out	TCK in	RFS out	TFS out
0	1	1	0	0	RCK out	TCK out	RFS in	TFS in
0	1	1	0	1	RCK out	TCK out	RFS in	TFS out
0	1	1	1	0	RCK out	TCK out	RFS out	TFS in
0	1	1	1	1	RCK out	TCK out	RFS out	TFS out
Synchronous Mode								
1	0	0	x	0	-	CK in	-	FS in
1	0	0	x	1	-	CK in	-	FS out
1	0	1	x	0	-	CK out	-	FS in
1	0	1	x	1	-	CK out	-	FS out
1	1	0	x	x	-	Gated in	-	-
1	1	1	x	x	-	Gated out	-	-

73.3 SSI Transmit FIFO 0 & 1 Registers

The SSI Transmit FIFO registers are 15x32-bit registers. These registers are not directly accessible by the end user. Transmit Shift Register (TXSR) receives its values from these FIFO registers. Transmitted data is first-in-first-out.

When the Transmit Interrupt Enable (TIE) bit in the SIER and either of the Transmit FIFO Empty Enable (TFE0 or 1) bits in the SIER are set, an interrupt is asserted whenever the number of empty slots exceed or are equal to the selected threshold value of corresponding Tx-FIFO. The threshold value is contained in the corresponding Transmit FIFO Watermark (TFWM0 or 1) field in the SSI FIFO Control/Status Register (SFCSR).

73.4 SSI Transmit Shift Register (TXSR)

The SSI Transmit Shift Register (TXSR) is a 24-bit shift register that contains the data being transmitted. This register is not directly accessible by the end user. When a continuous clock is used, data is shifted out to the Serial Transmit Data (STXD) port by

the selected (internal/external) bit clock when the associated (internal/external) frame sync is asserted. When a gated clock is used, data is shifted out to the STXD port by the selected (internal/external) gated clock.

The Word Length control bits (WL[3:0]) in the SSI Transmit and Receive Clock Control Register (STCCR) determine the number of bits to be shifted out of the TXSR before it is considered empty and can be written to again. This word length can be 8, 10, 12, 16, 18, 20, 22 or 24 bits. The data to be transmitted occupies the most significant portion of the shift register if TXBIT0 is '0', otherwise it occupies the least significant portion. The unused portion of the register is ignored. Data is always shifted out of this register with the Most Significant Bit (MSB) first when the SHFD bit of the STCR is cleared. If this bit is set, the Least Significant Bit (LSB) is shifted out first. The figures below show the transmitter loading and shifting operation. The figures show the working for some WL values, the same can be extended for other values.

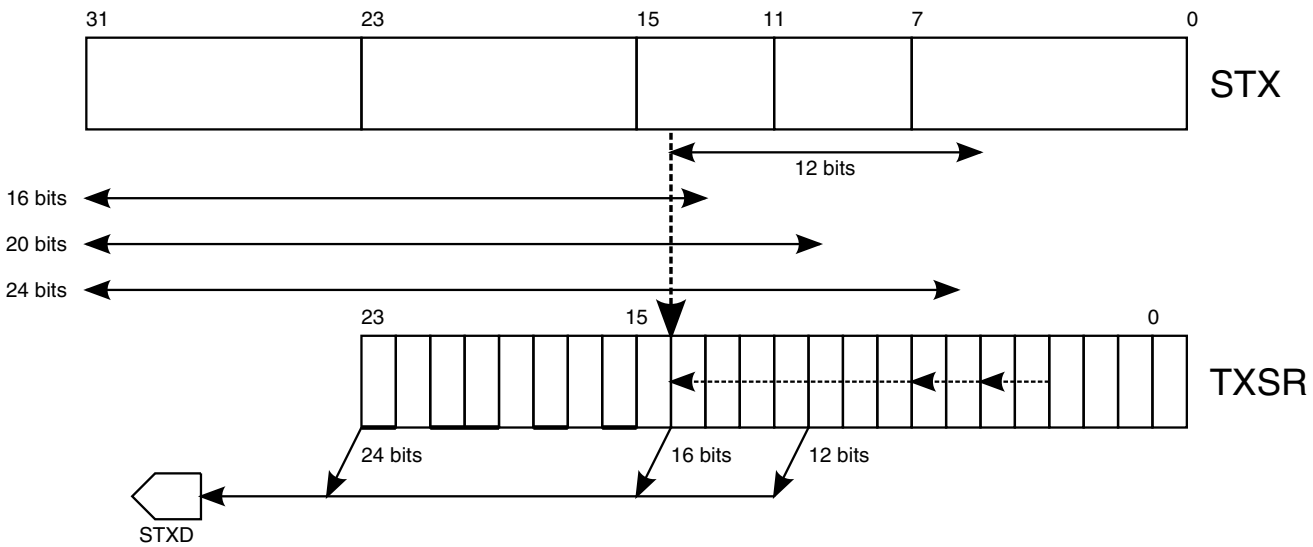


Figure 73-4. Transmit Data Path (TXBIT0=0, TSHFD=0) (MSB Alignment)

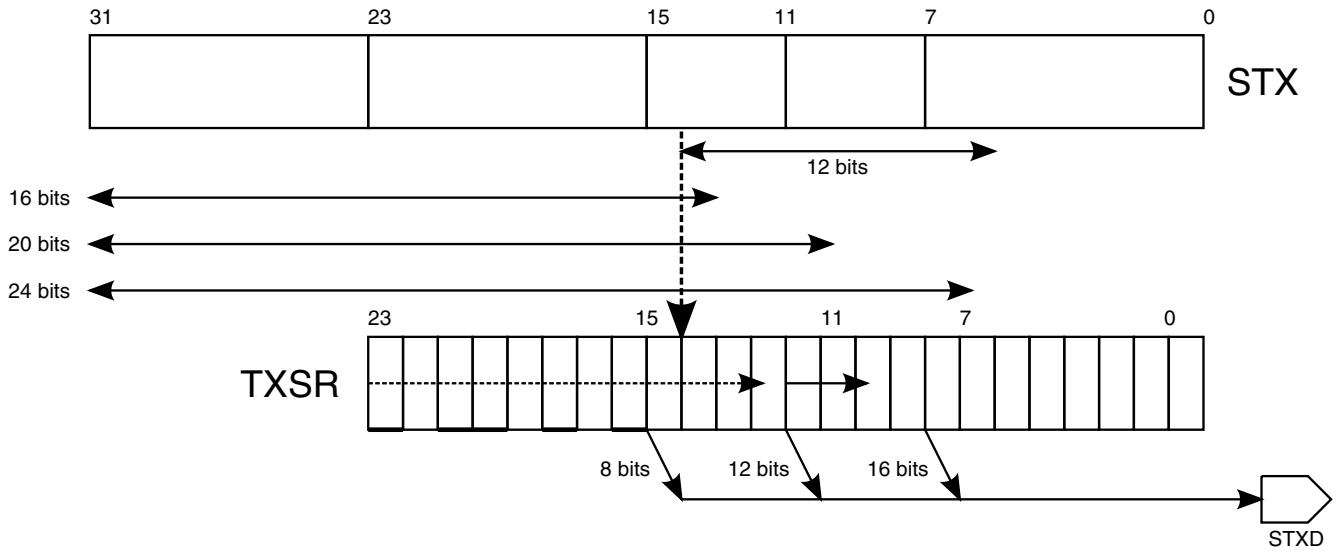


Figure 73-5. Transmit Data Path (TXBIT0=0, TSHFD=1) (MSB Alignment)

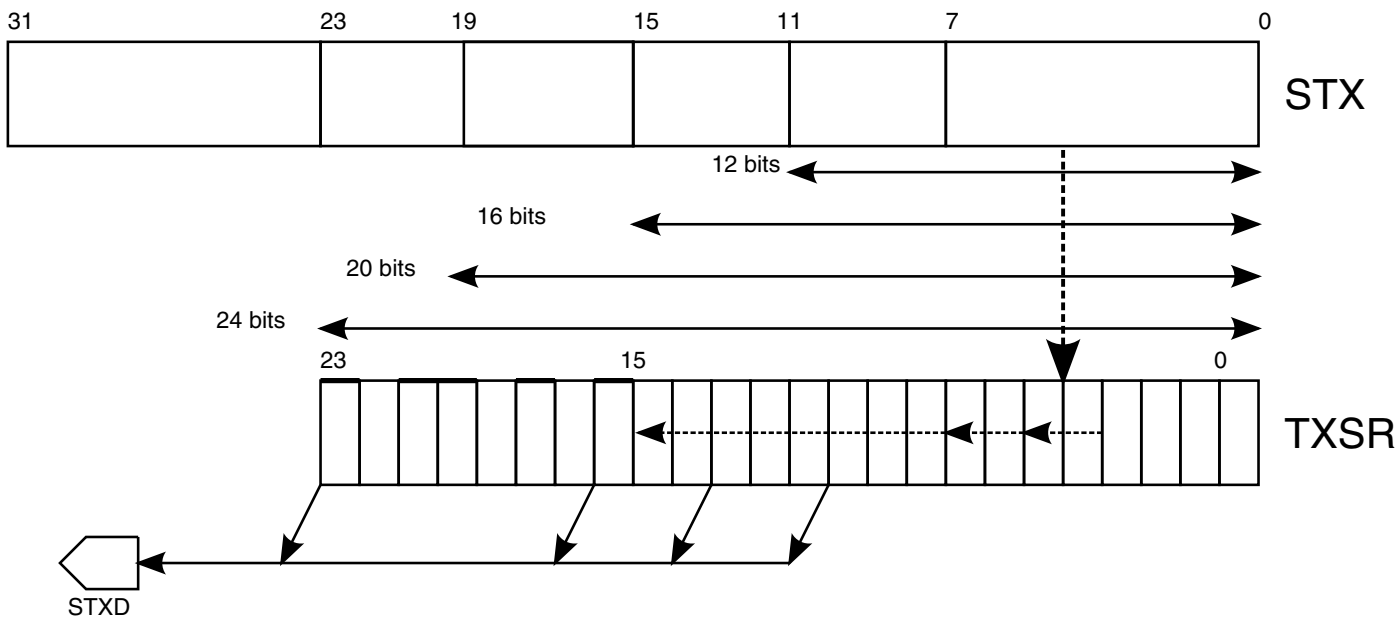


Figure 73-6. Transmit Data Path (TXBIT0=1, TSHFD=0) (LSB Alignment)

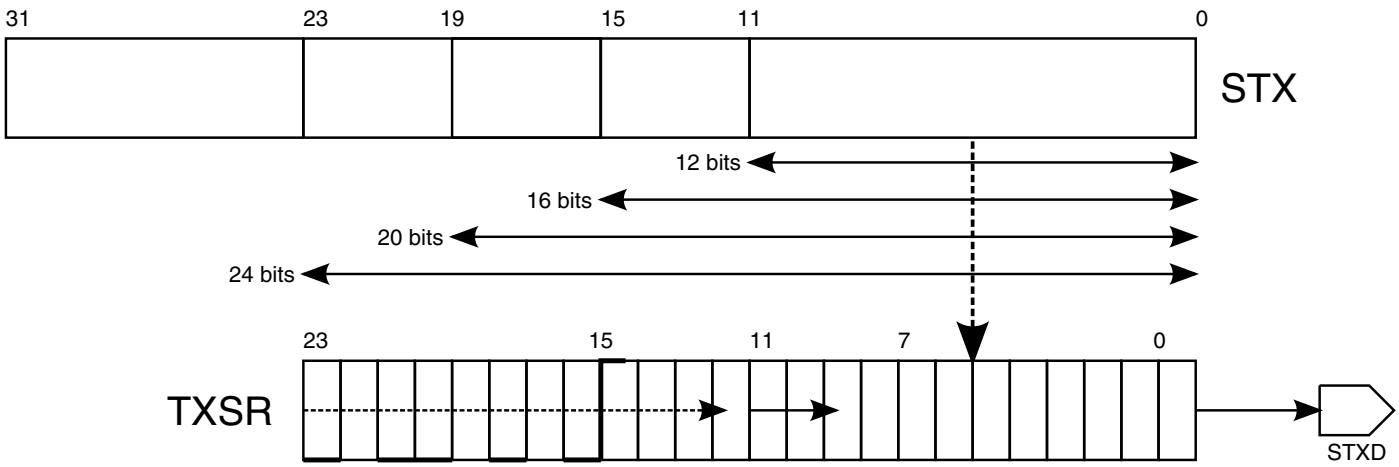


Figure 73-7. Transmit Data Path (TXBIT0=1, TSHFD=1) (LSB Alignment)

73.5 SSI Receive FIFO 0 and 1 Registers

The SSI Receive FIFO registers are 15x32-bit registers. These registers are not directly accessible by the end user. These FIFO registers receive data from the Receive Shift Register (RXSR). Received data is first-in-first-out.

When the Receive Interrupt Enable (RIE) bit in the SIER and either of the Receive FIFO Full Enable (RFF0_EN or RFF1_EN) bits in the SIER are set, an interrupt is asserted whenever the number of full slots exceeds or is equal to the selected threshold value of corresponding Rx-FIFO. The threshold value is contained in the corresponding Receive FIFO Watermark (RFWM0 or 1) field in the SSI FIFO Control/Status Register (SFCSR).

73.6 SSI Receive Shift Register (RXSR)

The SSI Receive Shift Register (RXSR) is a 24-bit, shift register that receives incoming data from the serial receive data SRXD port. This register is not directly accessible by the end user. When a continuous clock is used, data is shifted in by the selected (internal/external) bit clock when the associated (internal/external) frame sync is asserted. When a gated clock is used, data is shifted in by the selected (internal/external) gated clock.

Data is assumed to be received MSB first if the SHFD bit of the SSI.SRCR is cleared. If this bit is set, the data is received LSB first. Data is transferred to the appropriate SSI Receive Data Register (SRX0/1) or Receive FIFOs (if the receive FIFO is enabled and the corresponding SRX is full) after 8, 10, 12, 16, 18, 20, 22 or 24 bits have been shifted in depending on the WL[3:0] control bits. For receiving less than 24 bits of data, LSB bits

are appended with zero. The figures below show the receiver loading and shifting operation. These figures show the operation for several values of WL and the same can be extended for other values.

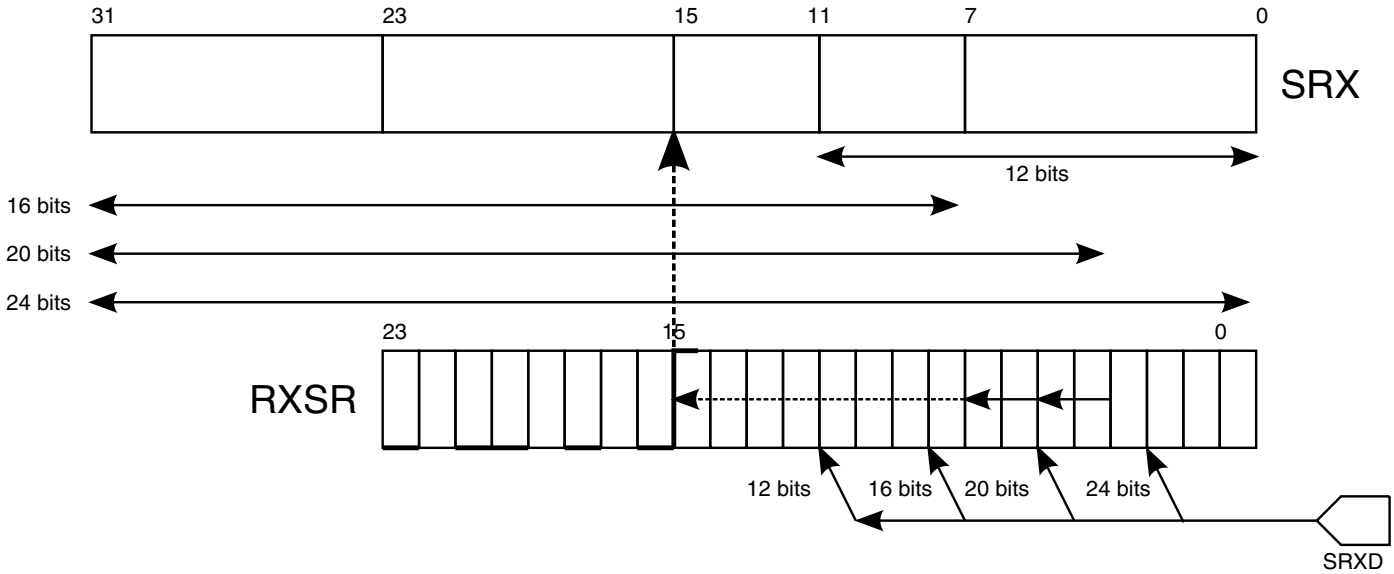


Figure 73-8. Receive Data Path (RXBIT0=0, RSHFD=0) (MSB Alignment)

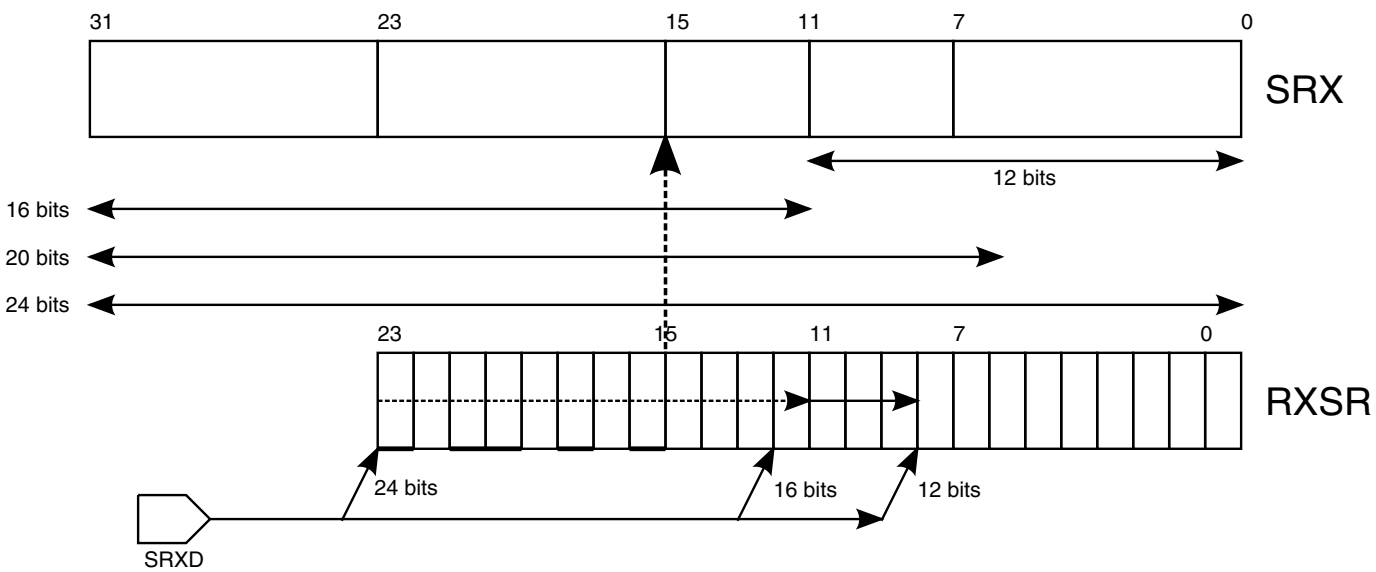


Figure 73-9. Receive Data Path (RXBIT0=0, RSHFD=1) (MSB Alignment)

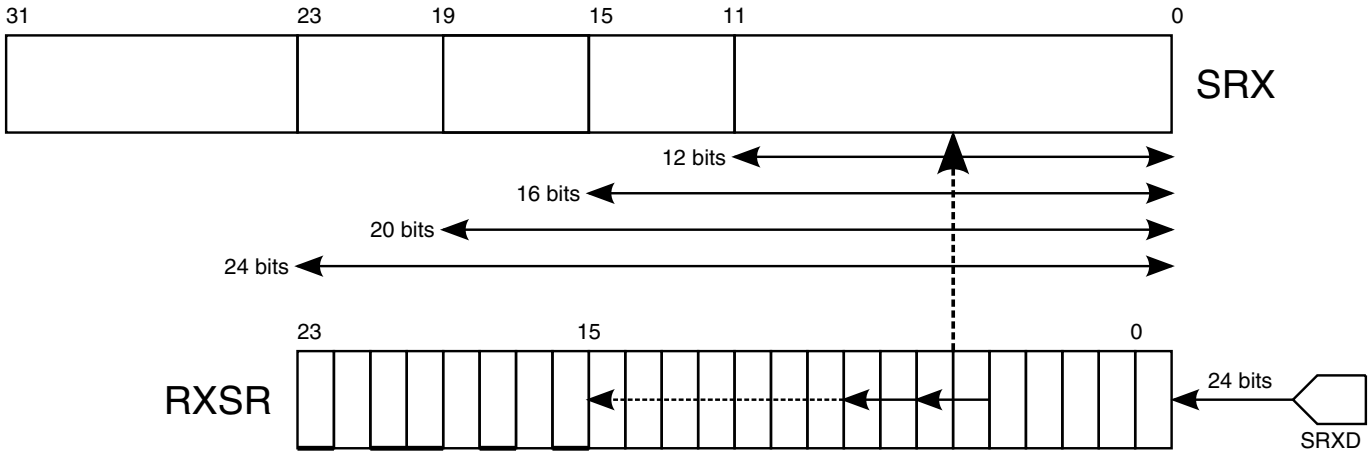


Figure 73-10. Receive Data Path (RXBIT0=1, RSHFD=0) (LSB Alignment)

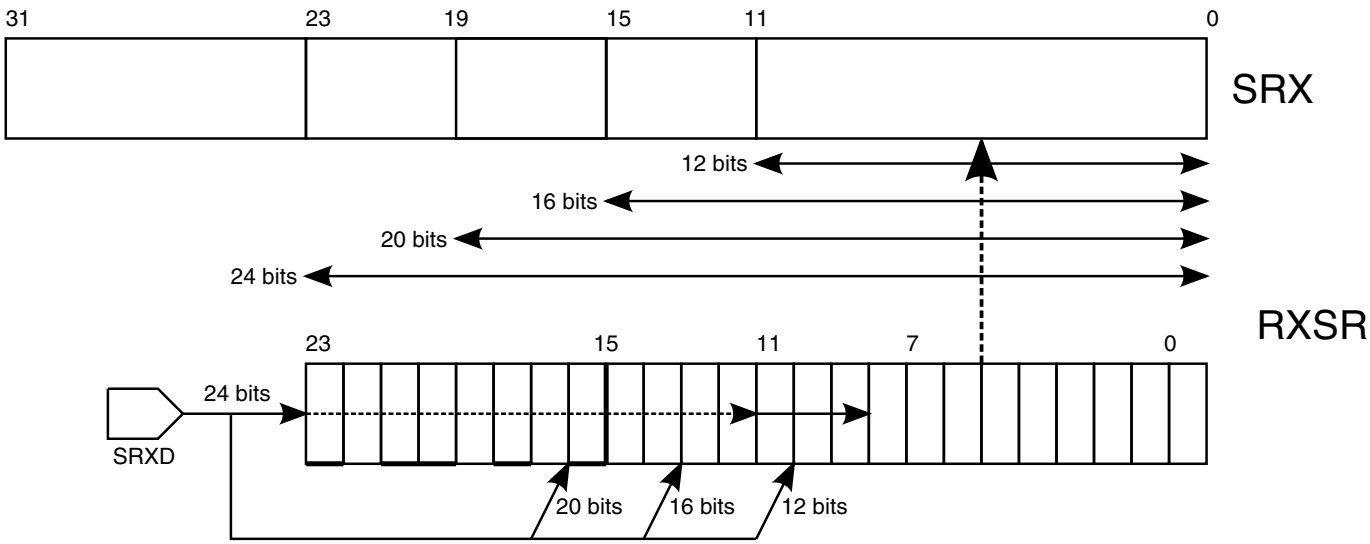


Figure 73-11. Receive Data Path (RXBIT0=1, RSHFD=1) (LSB Alignment)

73.7 Functional Description

73.7.1 Operating Modes

Different modes can be programmed by several bits in the SSI control registers.

See the table below for the list of SSI operating modes and some of the typical applications in which they can be used:

Table 73-3. SSI Operating Modes

TX, RX Sections	Serial Clock	Mode	Typical Application
Asynchronous	Continuous	Normal	Multiple synchronous CODECs
Asynchronous	Continuous	Network	TDM CODEC or DSP networks
Synchronous	Continuous	Normal	Multiple synchronous CODECs
Synchronous	Continuous	Network	TDM CODEC or DSP network
Synchronous	Gated	Normal	SPI-type devices; DSP to ARM platform

The transmit and receive sections of the SSI can be synchronous or asynchronous. In Synchronous mode, the transmitter and the receiver use a common clock and frame synchronization signal. Masking of slots for Transmit and Receive section can differ in synchronous mode. Also the shifting of data in independent and for receive section depends on RXBIT0 and RSHFD bits in SSI.SRCR register. In Asynchronous mode, the transmitter and receiver each has its own clock and frame synchronization signals.

Normal or Network mode can be selected. In Normal mode, the SSI functions with one data word of I/O per frame. In Network mode, any number from two to thirty-two data words of I/O per frame can be used. Network mode is typically used in star time division multiplex networks with other processors or CODECs, allowing interface to time division multiplexed networks without additional logic. Gated clock mode option can be selected in Normal synchronous mode only. During Gated mode the clock is not-continuous and runs only during data-transmission. These distinctions result in the basic operating modes that allow the SSI to communicate with a wide variety of devices.

The SSI supports both Normal and Network modes, and these can be selected independently of whether the transmitter and receiver are synchronous or asynchronous. Typically these protocols are used in a periodic manner, where data is transferred at regular intervals, such as at the sampling rate of an external CODEC. Both modes use the concept of a frame. The beginning of the frame is marked with a frame sync when programmed with continuous clock. The frame sync occurs at a periodic interval. The length of the frame is determined by the DC[4:0] bits in either the SSI.SRCCR or SSI.STCCR register, depending on whether data is being transmitted or received. The number of words transferred per frame depends on the mode of the SSI.

In Normal mode, one data word is transferred per frame. In Network mode, the frame is divided into anywhere between two and thirty-two time slots, where in each time slot one data word can optionally be transferred.

Apart from the above basic modes of operation, SSI supports the following modes which require some specific programming.

- I2S mode
- AC97 mode
  - AC97 Fixed mode
  - AC97 Variable mode

In (non-I2S) slave modes (external frame sync), the programmed word length setting of the SSI should be equal to the word length setting of the master. In I2S slave mode, the programmed word length setting of the SSI can be lesser than or equal to the word length setting of the I2S master (external CODEC).

In slave modes, the programmed frame length setting (DC bits) of the SSI can be lesser than or equal to the frame length setting of the master (external CODEC).

The following sections provide detailed descriptions of the above modes.

73.7.1.1 Normal Mode

Normal mode is the simplest mode of the SSI. It is used to transfer data in one time slot per frame. A time slot is a unit of data and the WL[3:0] bits define the number of bits in a time slot. In Continuous Clock mode, a frame sync occurs at the beginning of each frame.

The length of the frame is determined by the following factors:

- The period of the Serial Bit Clock (DIV2, PSR, PM[7:0] bits for internal clock or the frequency of the external clock on the STCK port)
- The number of bits per time slot (WL[3:0] bits)
- The number of time slots per frame (DC[4:0] bits)

If Normal mode is configured with more than one time slot per frame, data is transferred only in the first time slot. No data is transferred in subsequent time slots. In Normal mode, DC[4:0] values corresponding to more than a single time slot in a frame, only results in lengthening of the frame.

73.7.1.1.1 Normal Mode Transmit

The conditions for data transmission from the SSI in Normal mode are:

- SSI Enabled (SSIEN = 1)
- Write data to Transmit Data Register (STX)
- Transmitter Enabled (TE = 1)
- Frame sync active (for continuous clock case)
- Bit clock begins

When the above conditions occur in Normal mode, the next data word is transferred into the Transmit Shift Register (SSI.TXSR) from the Transmit Data Register 0 (SSI.STX0), or from the Transmit FIFO 0 Register, if transmit FIFO 0 is enabled. The new data word is transmitted on arrival of frame-sync preceded by clocks in continuous clock mode. In gated-external mode, data word in transmitted on arrival of frame-sync. In gated-internal mode, data word is transmitted whenever data is available in Tx-FIFO.

If Transmit FIFO 0 is not enabled and the transmit data register empty enable (TDE0_EN) and transmit interrupt enable (TIE) bits are set, transmit interrupt occurs when the word in SSI_STX0 is shifted to Transmit Shift (SSI.TXSR) register for shifting.

If Transmit FIFO 0 is enabled and the transmit fifo empty enable (TFE0_EN) and transmit interrupt enable (TIE) bits are set, transmit interrupt occurs when the number of empty slots in Transmit Fifo 0 exceed or are equal to the selected threshold value i.e. Transmit Fifo 0 Watermark (TFWM0) value. If transmit FIFO 0 is enabled and filled with data, 15 data words can be transferred before the core must write new data to the SSI.STX0 register.

The STXD port is disabled except during the data transmission period. For a continuous clock, the optional frame sync output and clock outputs are not disabled, even if both receiver and transmitter are disabled.

73.7.1.1.2 Normal Mode Receive

The conditions for data reception from the SSI are:

- SSI enabled (SSIEN = 1)
- Receiver enabled (RE = 1)
- Frame sync active (for continuous clock case)
- Bit clock begins

With the above conditions in Normal mode with a continuous clock, each time the frame sync signal is generated (or detected) a data word is clocked in. With the above conditions and a gated clock, each time the clock begins, a data word is clocked in.

If Receive FIFO 0 is not enabled and Receive Interrupt enable (RIE) and Received Data 0 Ready enable (RDR0_EN) bits are set, receive interrupt occurs when received data word is transferred from the Receive Shift Register (SSI.RXSR) to the Receive Data Register 0 (SSI.SRX0), thus setting the Receive Data Ready 0 (RDR0) flag.

If Receive FIFO 0 is enabled, and Receive Interrupt enable (RIE) and Received Fifo 0 full enable (RFF0_EN) bits are set, receive interrupt occurs when the received data word is transferred to the Receive FIFO 0 and Receive FIFO 0 reaches the selected threshold and results in Receive FIFO Full 0 (RFF0) flag to get set.

The core program has to read the data from the Receive Data Register 0 (SSI.SRX0) (in case Receive FIFO0 is disabled) before a new data word is transferred from the Receive Shift Register (SSI.RXSR), otherwise the Receive Overrun Error 0 (ROE0) bit is set. If receive FIFO 0 is enabled, the Receive Overrun Error 0 (ROE0) bit is set when the Receive FIFO 0 data level reaches the selected threshold and a new data word is ready to be transferred to the Receive FIFO 0.

See the following figure for an illustration of transmitter and receiver timing for an 8-bit word in the first time slot in Normal mode, continuous clock with a late word length frame sync. The Tx Data register is loaded with the data to be transmitted. On arrival of the clock, this data is transferred to the Transmit Shift Register, which gets transmitted on arrival of the frame-sync on the STXD output. Simultaneously, the Receive Shift Register shifts in the received data available on the SRXD input and, at the end of the time slot, this data is transferred to the Rx Data Register.

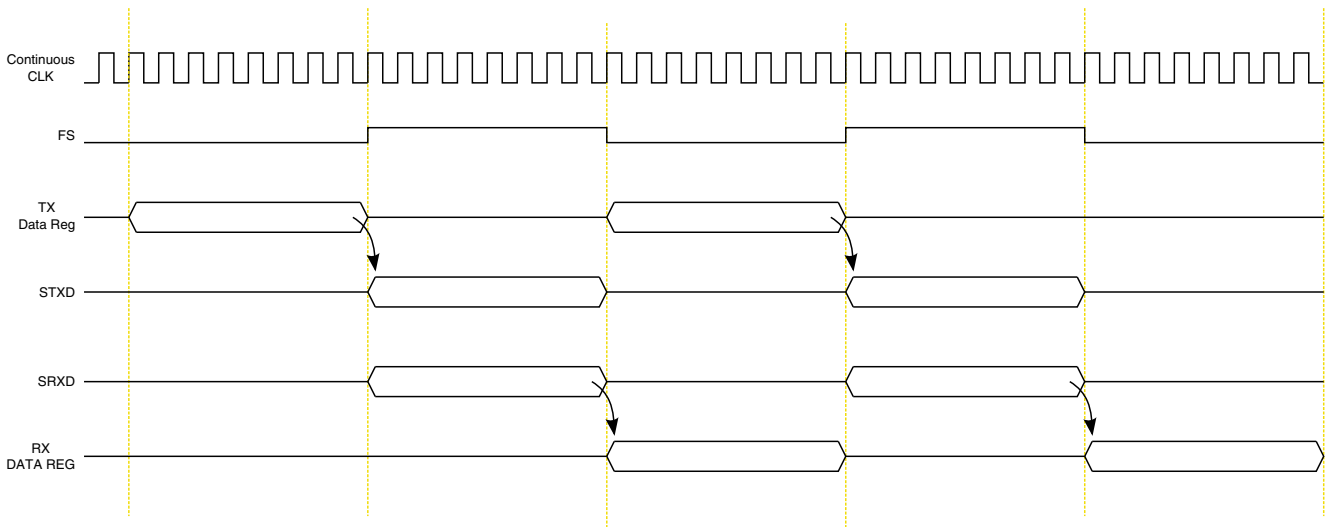


Figure 73-12. Normal Mode Timing - Continuous Clock

The following figure shows a similar case for internal (SSI generates clock) gated clock mode and Figure 73-14 shows a case for external (SSI receives clock) gated clock mode.

NOTE

A pull-down resistor is required in the gated clock case because the clock port is disabled between transmissions.

The Tx Data register is loaded with the data to be transmitted. On arrival of the clock, this data is transferred to the Transmit Shift Register, which gets transmitted on arrival of the frame-sync on the STXD output. Simultaneously, the Receive Shift Register shifts in the received data available on the SRXD input and, at the end of the time slot, this data is transferred to the Rx Data Register. In case of Internal Gated clock mode, the Tx Data line and clock output port are put in the high-impedance state at the end of transmission

of the last bit (at the completion of the complete clock cycle), whereas, in External Gated clock mode, the Tx Data line is tri-stated at the last inactive edge of the incoming bit clock (during the last bit in a data word).

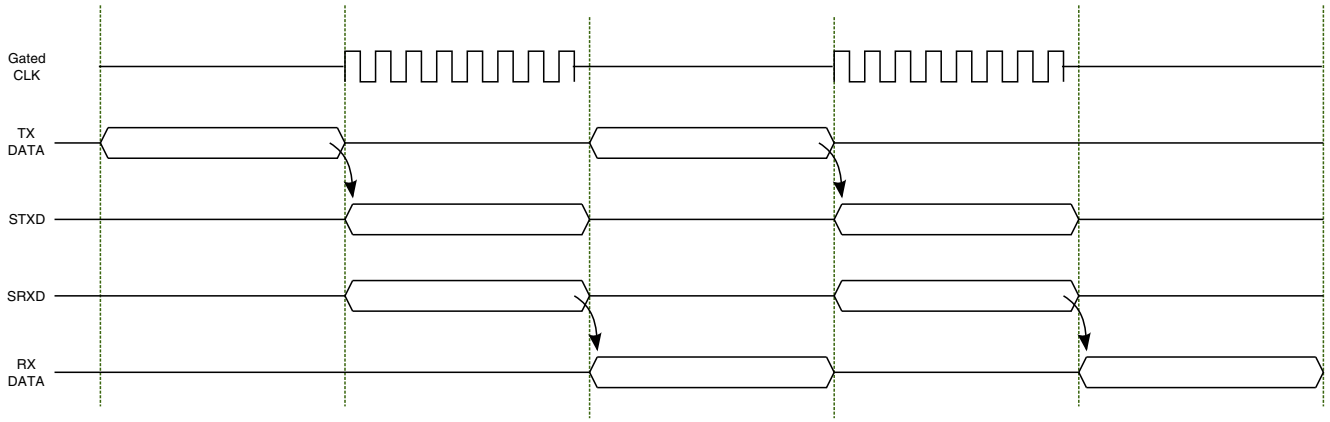


Figure 73-13. Normal Mode Timing - Internal Gated Clock

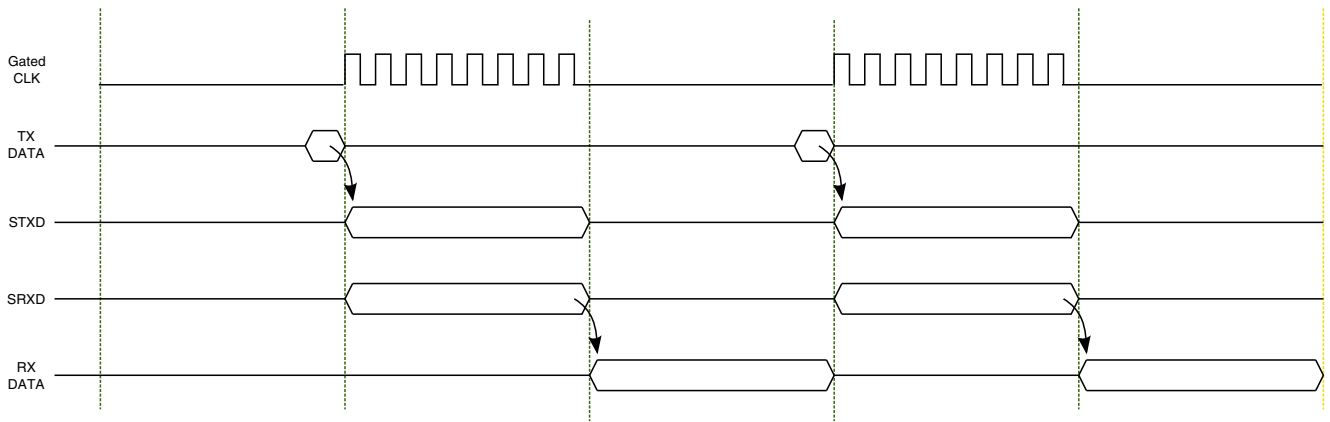


Figure 73-14. Normal Mode Timing - External Gated Clock

73.7.1.2 Network Mode

Network mode is used for creating a Time Division Multiplexed (TDM) network, such as a TDM CODEC network or a network of DSPs. In Continuous Clock mode, a frame sync occurs at the beginning of each frame.

In this mode, the frame is divided into more than one time slot. During each time slot, one data word can be transferred. Each time slot is then assigned to an appropriate CODEC or DSP on the network. The DSP can be a master device that controls its own private network, or a slave device that is connected to an existing TDM network and occupies a few time slots.

The frame sync signal indicates the beginning of a new data frame. Each data frame is divided into time slots and transmission and/or reception of one data word can occur in each time slot (rather than in just the frame sync time slot as in Normal mode). The frame rate dividers, controlled by the DC[4:0] bits, select two to thirty-two time slots per frame. The length of the frame is determined by the following factors:

- The period of the serial bit clock (PSR, PM[7:0] bits for internal clock, or the frequency of the external clock on the STCK port)
- The number of bits per sample (WL[3:0] bits)
- The number of time slots per frame (DC[4:0] bits)

In Network mode, data can be transmitted in any time slot. The distinction of the Network mode is that each time slot is identified with respect to the frame sync (data word time). This time slot identification allows the option of transmitting data during the time slot by writing to the STX registers or ignoring the time slot as determined by SSI.STMSK register bits. The receiver is treated in the same manner and received data is only transferred to the receive data register/fifo if the corresponding time slot is enabled (through SSI.SRMSK).

By utilizing the SSI.STMSK and SSI.SRMSK registers, software only has to service the SSI during valid time slots. This eliminates any overhead associated with unused time slots. Refer to SSI Transmit Time Slot Mask Register (SSI_SSI_STMSK) and SSI Receive Time Slot Mask Register (SSI_SSI_SRMSK) for more information on SSI.STMSK and SSI.SRMSK.

In the Two-Channel mode of operation, the second set of Transmit and Receive FIFOs and Data Registers are used to create two separate channels. These channels are completely independent, with a their own set of Core interrupts and DMA requests, which are identical to the ones available for the default channel. In this mode, data is transmitted/received in enabled time slots alternately from/to FIFO 0 and FIFO 1, starting from FIFO 0. The first data word is taken from FIFO 0 and transmitted in the first enabled time slot and subsequently, data is loaded from FIFO 1 and FIFO 0 alternately and transmitted. Similarly, the first received data is sent to FIFO 0 and subsequent data is sent to FIFO 1 and FIFO 0 alternately. Time slots can be selected through the Transmit and Receive Time Slot Mask registers (SSI.STMSK and SSI.SRMSK). For using this mode of operation, the TCH_EN bit (SCR[8]) needs to be set.

73.7.1.2.1 Network Mode Transmit

The transmit portion of SSI is enabled when the SSIEN and the TE bits in the SSI.SCR are both set. However, for continuous clock, when the TE bit is set, the transmitter is enabled only after detection of a new frame sync (transmission starts from the next frame boundary).

Normal start-up sequence for transmission is to perform the following:

1. Enable Network Mode.
2. Enable SSI
3. Write the data to be transmitted to the SSI.STX register. This clears the TDE flag
4. Set the TE bit to enable the transmitter on the next frame boundary (for continuous clock case).
5. Enable transmit interrupts.

(Alternatively, the programmer may decide not to transmit in a time slot by configuring the STMSK.) TDE flag is set as data is shifted from SSI.STX register to TXSR, but the STXD port remains disabled during the time slots. When the next frame sync is detected or generated (continuous clock), the data word in TXSR and is shifted out (transmitted). When the SSI.STX register is empty, the TDE bit is set, which causes a transmitter interrupt (in case the FIFO is disabled) to be sent if the TIE bit is set. Software can poll the TDE bit or use interrupts to reload the STX register with new data for the next time slot. Failing to reload the SSI.STX register before the TXSR is finished shifting (empty) causes a transmitter underrun and the TUE error bit is set. In case the FIFO is enabled, the TFE flag is set in accordance with the watermark setting and this flag causes the transmitter interrupt to occur.

The operation of clearing the TE bit disables the transmitter after completion of transmission of the current frame. Setting the TE bit enables transmission from the next frame. During that time the STXD port is disabled. The TE bit should be cleared after the TDE bit is set to ensure that all pending data is transmitted.

To summarize, the Network mode transmitter generates interrupts every enabled time slot (when FIFO is disabled) and requires the core program to respond to each enabled time slot. These responses from the core are one of the following:

- Write data in data register to enable transmission in the next time slot.
- Configure the time slot register to disable transmission in the next time slot (unless time slot is already masked by SSI.STMSK register bit).
- Do nothing-transmit underrun occurs at the beginning of the next time slot and the previous data is re-transmitted.

In the Two-Channel mode of operation, both the channels (Data Registers, FIFOs, Interrupts and DMA requests) operate in the same manner, as described above. The only difference in case of the second channel is that the Interrupts related to this channel are generated only in case this mode of operation is selected (TDE1 is low by default).

73.7.1.2.2 Network Mode Receive

The receiver portion of the SSI is enabled when both the SSIEN and the RE bits in the SSI.SCR are set. However, the receive enable only takes place during that time slot if RE is enabled before the second to last bit of the word. If the RE bit is cleared, the receiver is disabled at the end of the current frame.

SSI is capable of finding the start of the next frame automatically. When the word is completely received, it is transferred to the SRX register, which sets the RDR bit (Receive Data Ready). Setting the RDR bit causes a receive interrupt to occur if the receiver interrupt is enabled (the RIE bit is set) and (Receive data ready enable) RDR_EN bit is set. The second data word (second time slot in the frame), begins shifting in immediately after the transfer of the first data word to the SSI.SRX register. The core program has to read the data from the Receive Data Register (which clears RDR) before the second data word is completely received (ready to transfer to RX data register) or a receive overrun error occurs (the ROE bit is set).

An interrupt can occur after the reception of each enabled data word or the programmer can poll the RDR flag. The core program response can be one of the following:

- Read RX and use the data.
- Read RX and ignore the data.
- Do nothing-the receiver overrun exception occurs at the end of the current time slot.

NOTE

For a continuous clock, the optional frame sync output and clock output signals are not affected, even if the transmitter or receiver is disabled. TE and RE do not disable the bit clock or the frame sync generation. To disable the bit clock and the frame sync generation, the SSIEN bit in the SSI.SCR can be cleared or TFR_CLK_DIS/RFR_CLK_DIS bits can be set, or the port control logic external to the SSI (for example, in the IOMUXC) can be re configured.

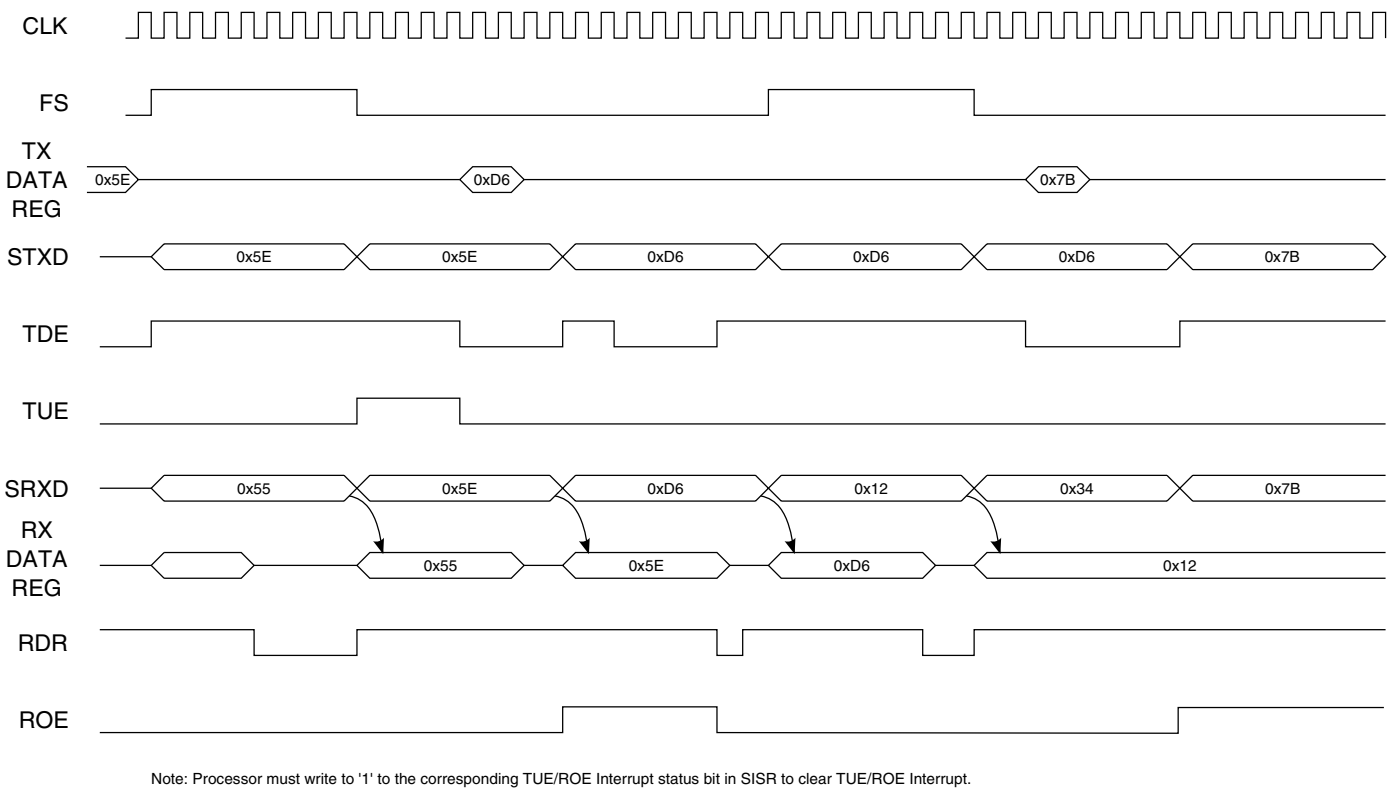
In the Two-Channel mode of operation, both the channels (Data Registers, FIFOs, Interrupts and DMA requests) operate in the same manner, as described above. The only difference in case of the second channel is that the Interrupts related to this channel are generated only in case this mode of operation is selected.

The transmitter and receiver timing for an 8-bit word with continuous clock, FIFO disabled, three words per frame sync in Network mode is shown in the figure below).

NOTE

The transmitter repeats the value 0x5E because of an underrun condition

For the receive section, data received on the SRXD pin gets transferred to the Rx Data register at the end of each time slot. If the FIFO is disabled, the RDR flag gets set and causes a receiver interrupt if RE, RIE and RDR_EN bits are set. If the FIFO is enabled, then the RFF flag is used for interrupt generation (this flag is set in accordance with the watermark settings). Here all time slots are enabled. The receive data ready flag is set after reception of the first data (0x55). Since the flag is not cleared (Rx Data Register is not read by core), the Receive Overrun Error (ROE) flag is set on reception of the next data (0x5E). ROE flag is cleared on writing '1' to the corresponding interrupt status bit in SSI Status Register.



Note: Processor must write to '1' to the corresponding TUE/ROE Interrupt status bit in SISR to clear TUE/ROE Interrupt.

Figure 73-15. Network Mode Timing - Continuous Clock

73.7.1.3 Gated Clock Mode

Gated Clock mode is often used to hook up to SPI-type interfaces on Micro controller Units (MCUs) or external peripheral chips. In Gated Clock mode, the presence of the clock indicates that valid data is on the STXD or SRXD ports. For this reason, no frame sync is needed in this mode.

Once transmission of data has completed, the clock is pulled to the inactive state. Gated clocks are allowed for both the transmit and receive sections with either internal or external clock in Normal mode. Gated clocks are not allowed in Network mode. See Table 73-2 ("Clock Pin Configurations") for SSI configuration for gated-mode operation.

The clock runs when the TE bit and/or the RE bit are appropriately enabled. For the case of internally generated clock, all internal bit clocks, word clocks, and frame clocks continue to operate. When a valid time slot occurs (such as the first time slot in Normal mode), the internal bit clock is enabled onto the appropriate clock port. This allows data to be transferred out in periodic intervals in Gated Clock mode. With an external clock, the SSI waits for a clock signal to be received. Once the clock begins, valid data is shifted in. Care should be taken to clear all DC bits (0x00000) when SSI is used in Gated mode. In gated mode of operation the TFS, RFS, TLS, RLS, TFRC and RFRC bits of SSI.AISR register are not generated.

For Gated clock operated in external clock mode, a proper clock signalling must be applied to the SSI STCK in order for it to function properly. When TSCKP is 0, CLK_IST value should be 1. When TSCKP is 1, CLK_IST value should be 0. If the SSI uses rising edge transition to clock data (TSCKP=0) and the falling edge transition to latch data (RSCKP=0), the clock must be in an active low state when idle. If the SSI uses falling edge transition to clock data (TSCKP=1) and the rising edge transition to latch data (RSCKP=1), the clock must be in a active high state when idle. The figures below illustrate the different edge clocking/latching.

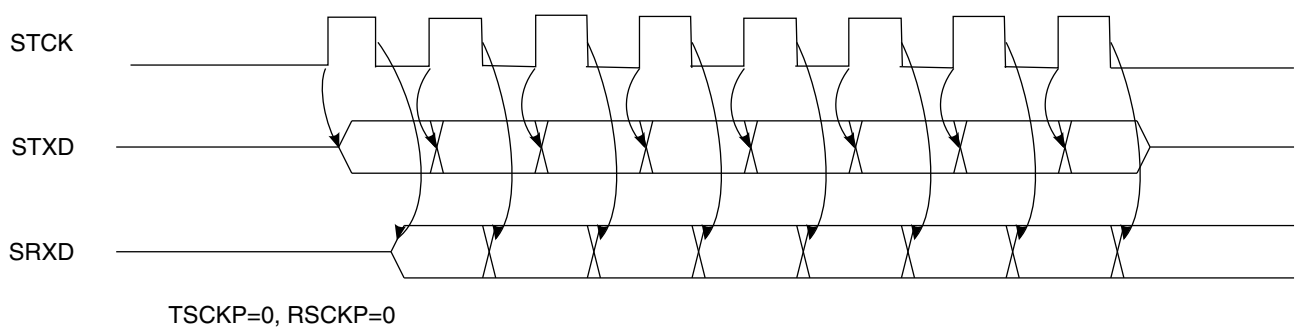


Figure 73-16. Internal Gated Mode Timing - Rising Edge Clocking / Falling Edge Latching

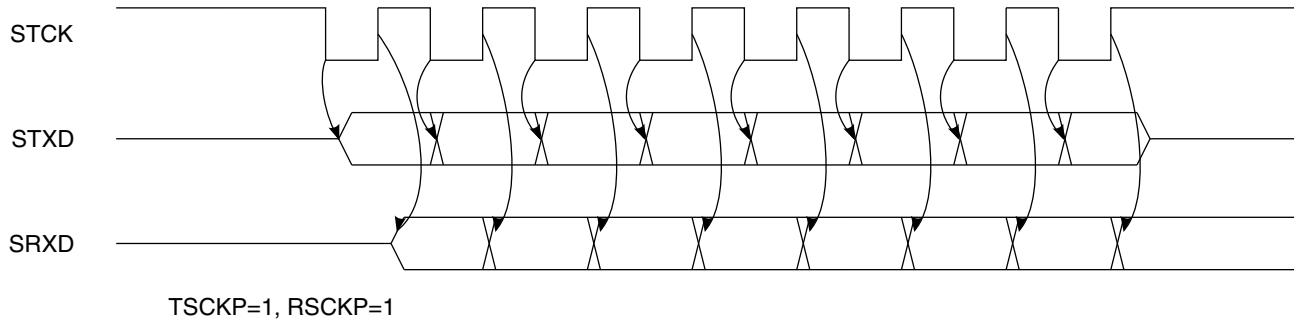


Figure 73-17. Internal Gated Mode Timing - Falling Edge Clocking / Rising Edge Latching

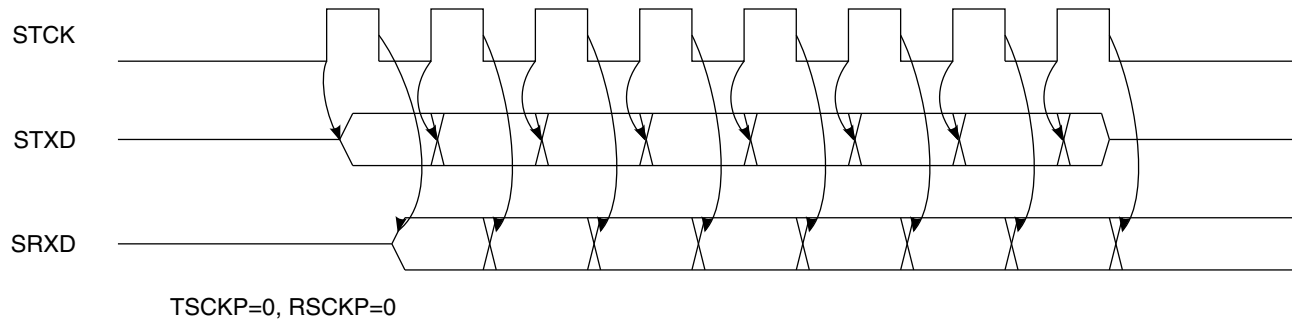


Figure 73-18. External Gated Mode Timing - Rising Edge Clocking / Falling Edge Latching

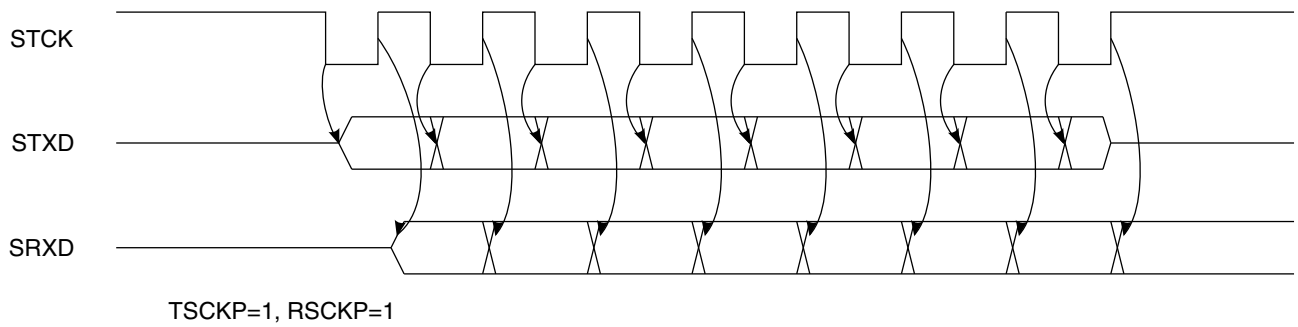


Figure 73-19. External Gated Mode Timing - Falling Edge clocking / Rising Edge Latching

The bit clock ports must be kept free of timing glitches. If a single glitch occurs, all ensuing transfers will be out of synchronization.

In case of External Gated Mode, even though the Tx Data line is put in the high-impedance state at the last non-active edge of the bit clock, the round trip delay should be sufficient to take care of hold time requirements at the external receiver.

73.7.1.4 I2S Mode

The SSI is compliant to the Inter-IC Sound (I2S) bus specification from Philips Semiconductors (February 1986, Revised June 5, 1996). For more information on I2S, refer to the latest version of NXP Semiconductor's I2S specification.

See the following figure for an illustration of the basic I2S protocol timing.

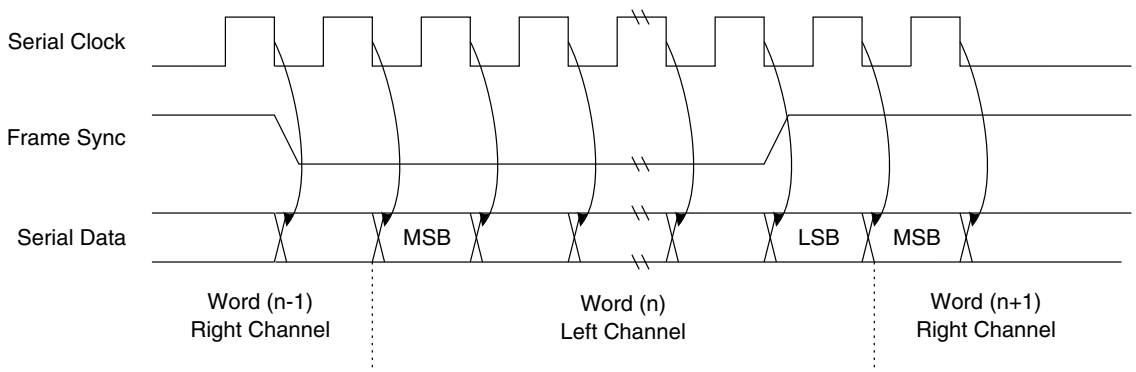


Figure 73-20. I2S Mode Timing - Serial Clock, Frame Sync and Serial Data

Select I2S mode using the options listed in the table below.

Table 73-4. I2S Mode Selection

I2S_MODE[1]	I2S_MODE[0]	Mode Type
0	0	Normal mode
0	1	I2S master mode
1	0	I2S slave mode
1	1	Normal mode

In normal mode operation, no register bits are forced to any particular state internally and the user can program the SSI to work in any operating condition.

When I2S modes are entered (I2S master (01) or I2S slave (10)), the following settings are recommended:

- Sync mode (SSI_SCR[4] =1)
- Tx shift direction: MSB transmitted first (SSI_STCR[4]=0)
- Rx shift direction: MSB received first (SSI_SRCR[4]=0)
- Tx data clocked at falling edge of the clock (SSI_STCR[3]=1)
- Rx data latched at rising edge of the clock (SSI_SRCR[3]=1)
- Tx frame sync active low (SSI_STCR[2]=1)
- Rx frame sync active low (SSI_SRCR[2]=1)

- Tx frame sync initiated one bit before data is transmitted (SSI_STCR[0]=1)
- Rx frame sync initiated one bit before data is received (SSI_SRCR[0]=1)
- TX Frame Rate should be 2 (SSI_STCCR[12:8] = 1)
- RX Frame Rate should be 2 (SSI_SRCCR[12:8] = 1)

In I2S master mode (SSI_SCR[6:5]=01), the following additional settings are recommended:

- TXDIR bit (SSI_STCR[5]) set to 1 to select internal generated bit clock
- TFDIR bit (SSI_STCR[6]) set to 1 to select internal generated frame sync

In I2S master mode(SSI.SCR[6:5]=01), the following settings are internally overridden by the hardware:

- Network mode is selected (SSI_SCR[3]=1)
- Tx frame sync length set to one-word-long-frame (SSI_STCR[1]=0)
- Rx frame sync length set to one-word-long-frame (SSI_SRCR[1]=0)
- Tx shifting w.r.t. bit 0 of TXSR (SSI.STCR[9]=1)
- Rx shifting w.r.t. bit 0 of RXSR (SSI.SRCR[9]=1)

The user needs to set the following control bits to configure the bit clock and frame sync:

- PM (SSI_STCCR[7:0])
- PSR (SSI_STCCR[17])
- DIV2(SSI_STCCR[18]
- WL (SSI_STCCR[16:13]
- DC (SSI_STCCR[12:8])

The word length is fixed to 32 in I2S Master mode and the WL bits determine the number of bits that will contain valid data (out of the 32 transmitted/received bits in each channel).

In I2S slave mode(SSI_SCR[6:5]=10), the following additional settings are recommended:

- TXDIR bit(SSI_STCR[5]) set to 0 to select external generated bit clock
- TFDIR bit(SSI_STCR[6]) set to 0 to select external generated frame sync

In I2S slave mode(SSI_SCR[6:5]=10), the following settings are internally overridden by the hardware:

- Normal mode is selected (SSI_SCR[3]=0)
- Tx frame sync length set to one-bit-long-frame (SSI_STCR[1]=1)
- Rx frame sync length set to one-bit-long-frame (SSI_SRCR[1]=1)
- Tx shifting w.r.t. bit 0 of TXSR (SSI_STCR[9]=1)
- Rx shifting w.r.t. bit 0 of RXSR (SSI_SRCR[9]=1)

The user needs to set the following control bits to configure the data transmission:

- WL (SSI_STCCR[16:13])
- DC (SSI_STCCR[12:8])

The word length is variable in I2S slave mode and the WL bits determine the number of bits that will contain valid data. The actual word length is determined by the external CODEC. The external I2S Master still sends frame sync according to the I2S protocol (early, word wide and active low), the SSI internally operates so that each frame sync transition is the start of a new frame (the WL bits determine the number of bits to be transmitted/received). After one data word has been transferred, the SSI waits for the next frame sync transition to start operation in the next time slot. Transmit (STMSK) and receive (SRMSK) mask bits should not be used in I2S Slave mode of operation. Masking is supported only for network mode of operation.

73.7.1.5 AC97 Mode

In AC97 mode of operation, the SSI transmits a 16-bit Tag Slot at the start of a frame and the rest of the slots (in that frame) are all 20-bits wide. The same sequence is followed while receiving data. Refer to the AC97 specification for details regarding transmit and receive sequences and data formats.

NOTE

The Audio Codec specification released in 1997 [AC '97] defines the Architecture and Digital Interface, specifically designed for implementing audio and modem I/O functionality in personal computers. Companion specifications include the Modem Codec [MC '97], and the combined Audio/Modem Codec standard [AMC '97]. The current version of AC '97 was produced in 2002. The AC-97 specification defines a recommended 48-pin QFP IC package.

Note that the SSI only has one RxDATA pin so the SSI can only support one codec. Secondary codecs are not supported.

When AC97 mode is enabled, the following settings are internally overridden by the hardware. The programmed register values are not changed by entering AC97 mode but they no longer apply to the block's operation. Writing to the programmed register fields will update their values; these updates can be seen by reading back the register fields. However, these settings will not take effect until AC97 mode is turned off.

The register bits within the bracket are the equivalent settings:

- Sync mode is entered (SSI.SCR[4] =1)

- Network mode is selected (SSI.SCR[3]=1)
- Tx shift direction is MSB transmitted first (SSI.STCR[4]=0)
- Rx shift direction is MSB received first (SSI.SRCR[4]=0)
- Tx data is clocked at rising edge of the clock (SSI.STCR[3]=0)
- Rx data is latched at falling edge of the clock (SSI.SRCR[3]=0)
- Tx frame sync is active high (SSI.STCR[2]=0)
- Rx frame sync is active high (SSI.SRCR[2]=0)
- Tx frame sync length is one-word-long-frame (SSI.STCR[1]=0)
- Rx frame sync length is one-word-long-frame (SSI.SRCR[1]=0)
- Tx frame sync initiated one bit before data is transmitted (SSI.STCR[0]=1)
- Rx frame sync initiated one bit before data is received (SSI.SRCR[0]=1)
- Tx shifting w.r.t. bit 0 of TXSR (SSI.STCR[9]=1)
- Rx shifting w.r.t. bit 0 of RXSR (SSI.SRCR[9]=1)
- Tx FIFO is enabled (SSI.STCR[7]=1)
- Rx FIFO is enabled (SSI.SRCR[7]=1)
- TFDIR bit (SSI.STCR[6]) is forced to 1 internally to select internal generated frame sync
- TXDIR bit (SSI.STCR[5]) is forced to 0 internally to select external generated bit clock

Any alteration of these bits individually will not affect the operational conditions of the SSI unless AC97 mode is deselected.

Hence, the only control bits needed to be set by the user to configure the data transmission/reception are the WL (SSI.STCCR[16:13]) and DC (SSI.STCCR[12:8]) bits. In AC97 mode, the WL bits can only legally take the values corresponding to 16-bit (truncated data) or 20-bit time slots. In case WL bits are set to select 16-bit time slots, the SSI pads the transmit data (four least significant bits) with zeros and while receiving, stores only the most significant 16 bits in the Rx FIFO.

Follow the sequence for programming the SSI to work in AC97 mode:

1. Program the WL bits to a value corresponding to either 16 or 20 bits. The WL bit setting is only for the data portion of the AC97 frame (Slots #3 through #12). The Tag slot (Slot #0) is always 16 bits wide and the Command Address and Command Data slots (Slots #1 and #2) are always 20 bits wide.
2. Select the number of time slots by programming the DC bits. For AC97 operation, DC bits should be set to a value of '0xC', resulting in 13 time slots per frame.
3. Write data to be transmitted, in Tx FIFO 0 (through Tx Data Register 0) and Tx FIFO 1 while using Two-Channel Mode (TCH_EN = 1).
4. Program the FV, TIF, RD, WR and FRDIV bits in SSI.SACNT register
5. Update the contents of SSI.SACADD, SSI.SACDAT and SSI.SATAG (for Fixed mode only) registers

6. Enable the AC97 mode of operation (AC97EN bit in SSI.SACNT register)

Once the SSI starts transmitting and receiving data (after being configured in AC97 mode), the programmer needs to service the interrupts, as and when they are raised (updates to command address/data or tag registers, reading of received data and writing more data for transmission). Further details regarding fixed and variable mode implementation are provided in the following sections.

While using AC97 in Two-Channel Mode (TCH_EN=1), it is recommended that the received tag is not stored in the Rx FIFO (TIF=0). In case the programmer needs to update the SSI.SATAG register and also issue a RD/WR command (in a single frame), it is recommended that the SSI.SATAG register be updated prior to issuing a RD/WR command.

73.7.1.5.1 AC97 Fixed Mode (SSI.SACNT[1]=0)

In fixed mode of operation, SSI transmits in accordance with the AC97 Frame Rate Divider bits (i.e. FRDIV in SACNT) which decides the number of frames for which the SSI should be idle, after operating for one frame.

In a valid frame, TAG Value (written by Core) will be transmitted in Slot #0, Command Address will be transmitted in Slot #1 in case of RD/WR Command, and Command Data will be transmitted in Slot #2 in case of a WR Command. The data from TX-FIFO is transmitted in Slot #3 - Slot #12 depending on the valid slots indicated by the TAG value.

While receiving, bit 15 of the TAG Value (Slot #0) is checked to see if the CODEC is ready. If this bit is set, the frame is received. The received TAG provides the information about Slots containing valid data. The the corresponding TAG bit is valid, the Command Address (Slot #1) and Command Data (Slot #2) values are stored in the corresponding registers. The received data (Slot #3 - Slot #12) is then stored in the Rx-FIFO (for valid slots).

73.7.1.5.2 AC97 Variable Mode (SSI.SACNT[1]=1)

In Variable Mode, the transmit slots which should contain data in the current frame are determined by SLOTREQ bits received in the previous frame. While receiving, if the CODEC is ready, the frame is received and the SLOTREQ bits (contained in Slot #1) are stored for scheduling transmission in the next frame.

The SSI.SACCST, SSI.SACCEN and SSI.SACCDIS registers helps in determining which transmit slots are active. This information is used to ensure that SSI does not transmit data for powered-down/inactive channels.

73.7.2 External Frame and Clock Operation

When applying external frame sync and clock signals to SSI, there should be at least 4-bit clock cycles between the enabling of the transmit or receive section and the rising edge of the corresponding frame sync signal. The transition of STFS or SRFS should be synchronized with the rising edge of external clock signal, STCK or SRCK.

73.7.2.1 Data Alignment Formats Supported

The SSI supports three data formats in order to provide flexibility with handling data. These formats dictate how data is written to (and read from) the data registers. Therefore, data can appear in different places in SSI.STX0/1 and SSI.SRX0/1 based on the data format and the number of bits per word. Independent data formats are supported for both the transmitter and receiver (that is, the transmitter and receiver can use different data formats).

The supported data formats are:

- MSB alignment
- LSB alignment
  - Zero-extended (receive data only)
  - Sign-extended (receive data only)

With MSB alignment, the most significant byte is bits 31 through 24 of the data register if the word length is larger than or equal to 16 bits. If the word length is less than 16 bits and MSB alignment is chosen, the most significant byte is bits 15 through 8. With LSB alignment, the least significant byte is bits 7 through 0. Data alignment is controlled by the TXBIT0 bit in the SSI.STCR and the RXBIT0 bit in the SSI.SRCR. See the table below for the bit assignment for all the data formats supported by the SSI.

Table 73-5. Data Alignment

Format	Bit Number																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
8-bit LSB Aligned																									7	6	5	4	3	2	1	0
8-bit MSB Aligned																	7	6	5	4	3	2	1	0								
10-bit LSB Aligned																							9	8	7	6	5	4	3	2	1	0
10-bit MSB Aligned																	9	8	7	6	5	4	3	2	1	0						
12-bit LSB Aligned																					11	10	9	8	7	6	5	4	3	2	1	0

Table continues on the next page...

Table 73-5. Data Alignment (continued)

12-bit MSB Aligned																	11	10	9	8	7	6	5	4	3	2	1	0				
16-bit LSB Aligned																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit MSB Aligned	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
18-bit LSB Aligned															17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
18-bit MSB Aligned	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
20-bit LSB Aligned													19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
20-bit MSB Aligned	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
22-bit LSB Aligned											21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
22-bit MSB Aligned	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
24-bit LSB Aligned									23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
24-bit MSB Aligned	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								

In addition, receive data can either be zero-extended or sign-extended if LSB alignment is selected. With zero-extension, all bits above the most significant bit are 0's. This format is useful when data is stored in a pure integer format. With sign-extension, all bits above the most significant bit are equal to the most significant bit. This format is useful when data is stored in a fixed-point integer format (which implies fractional values). Receive data extension is controlled by the RXEXT bit in the SSI.SRCR. Transmit data used with LSB alignment has no concept of sign/zero-extension. Unused bits above the most significant bit are simply ignored.

When configured in I2S or AC97 mode, the SSI forces the selection of LSB alignment. However, RXEXT still permits a choice between zero-extension and sign-extension.

Refer to SSI Transmit Configuration Register (SSI_SSI_STCR) and SSI Receive Configuration Register (SSI_SSI_SRCR) for more details on the relevant bits in the SSI.STCR and SSI.SRCR registers.

73.7.3 SSI Architecture

The Synchronous Serial Interface (SSI) is connected to chip pads through the Digital Audio Mux (AUDMUX) block or directly as well. The AUDMUX can be configured to connect the SSI to the chip pads in various ways.

Refer to Figure 73-1 for a block diagram of the SSI.

73.7.4 SSI Clocking

The SSI uses the following clocks:

- Bit clock - Used to serially clock the data bits in and out of the SSI port. This clock is either generated internally (from SSI's sys clock) or taken from external clock source (through the Tx/Rx clock ports).
- Word clock - Used to count the number of data bits per word (8, 10, 12, 16, 18, 20, 22 or 24 bits). This clock is generated internally from the bit clock.
- Frame clock (Frame Sync) - Used to count the number of words in a frame. This signal can be generated internally from the bit clock, or taken from external source (from the Tx/Rx frame sync ports).
- Network clock -In master mode, this is an integer multiple of frame clock. This is oversampling clock. It is used in cases when SSI has to provide the clock.

Care should be taken to ensure that the bit clock frequency (either internally generated by dividing the SSI's sys clock or sourced from external device through Tx/Rx clock ports) is never greater than 1/5 of the ipg_clk (from CCM) frequency.

In Normal mode (SCR[6:5]=00), the bit clock, used to serially clock the data, is visible on the Serial Transmit Clock (STCK) and Serial Receive Clock (SRCK) ports. The word clock is an internal clock used to determine when transmission of an 8, 10, 12, 16, 18, 20, 22 or 24 bit word has completed. The word clock in turn then clocks the frame clock, which counts the number of words in the frame. The frame clock can be viewed on the STFS and SRFS frame sync ports, because a frame sync is generated after the correct number of words in the frame have passed. In master and synchronous mode, the unused port SRCK is used as network clock (oversampling clock) enabled by the SCR register bit 15, SYS_CLK_EN. This network clock is an oversampling clock of the frame sync clock (STFS). In this mode, the word length (WL), Prescaler Range (PSR), Prescaler Modulus (PM) and Frame rate (DC) selects the ratio of network clock to sampling clock STFS. In case of I2S mode, the network clock (oversampling clock) can be made available on this port if the SYS_CLK_EN bit is set. The relationship between the clocks and the dividers is shown in the figure below ("SSI Clocking"). The bit clock can be received from an SSI clock port or can be generated from the network clock through a divider, as shown in Figure 73-22 ("SSI Transmit Clock Generator Block Diagram").

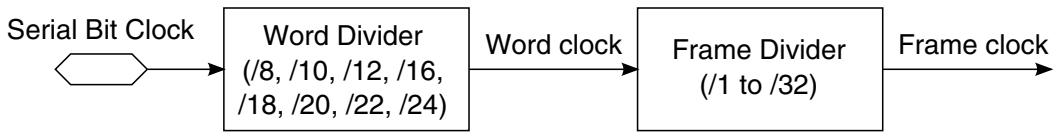


Figure 73-21. SSI Clocking

73.7.4.1 SSI Clock and Frame Sync Generation

Data clock and frame sync signals can be generated internally, or can be obtained from external sources. If internally generated, the SSI clock generator is used to derive bit clock and frame sync signals from the SSI's sys clock. The SSI clock generator consists of a selectable, fixed prescaler and a programmable prescaler for bit rate clock generation.

In Gated Clock mode, the data clock is valid only when data is being transmitted. Otherwise the clock port is pulled to the inactive state. A programmable frame rate divider and a word length divider are used for frame rate sync signal generation.

The following figure shows a block diagram of the clock generator for the transmit section. The serial bit clock can be internal or external, depending on the Transmit Direction (TXDIR) bit in the SSI Transmit Configuration Register (SSI.STCR). The receive section contains an equivalent clock generator circuit.

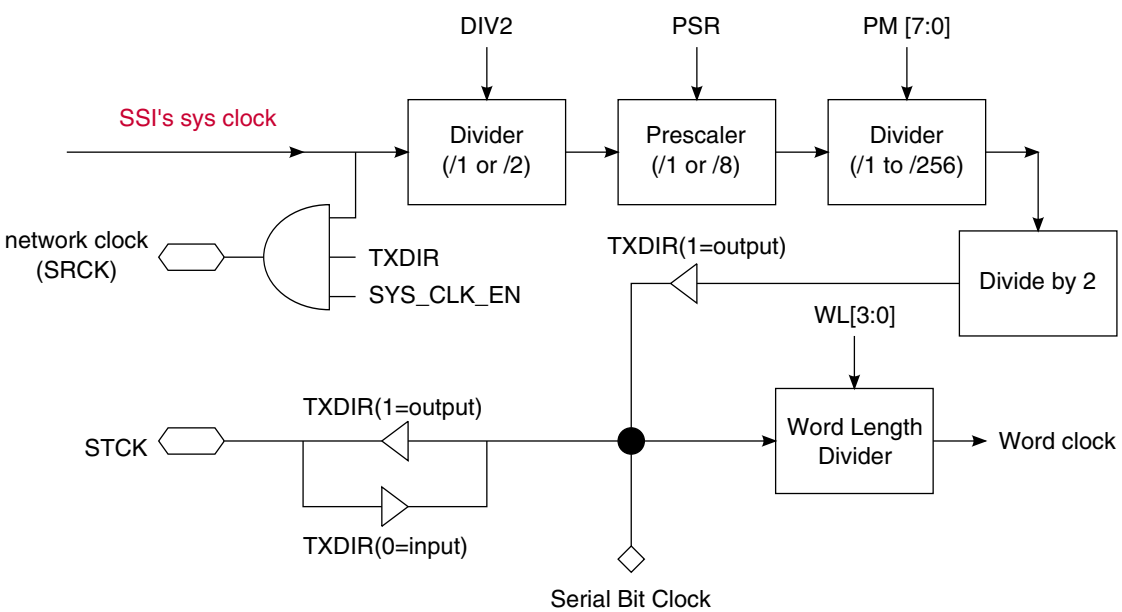


Figure 73-22. SSI Transmit Clock Generator Block Diagram

The figure below shows the Frame Sync Generator block for the transmit section. When internally generated, both receive and transmit frame sync are generated from the word clock and are defined by the Frame Rate Divider (DC[4:0]) bits and the Word Length (WL[3:0]) bits of the SSI Transmit Clock Control Register (SSI.STCCR). The receive section contains an equivalent circuit for the Frame Sync Generator.

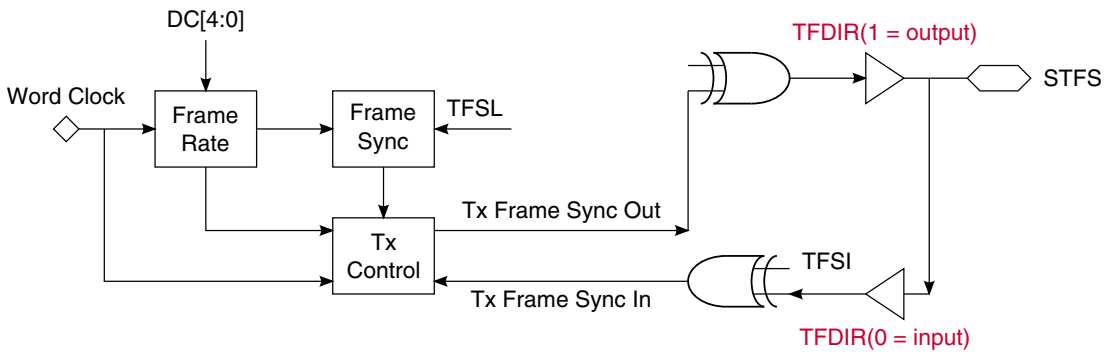


Figure 73-23. SSI Transmit Frame Sync Generator Block Diagram

73.7.4.2 DIV2, PSR and PM Bit Description

The bit clock frequency can be calculated from the SSI's sys clock using the equation in the following figure.

NOTE

You must ensure that the bit-clock frequency must be never greater than 1/5 of the peripheral clock frequency. The oversampling clock frequency can go up to peripheral clock frequency. Bits DIV2, PSR and PM should not be all set to zero at the same time.

$$f_{INT\_BIT\_CLK} = f_{SSI's\ sys\ clock} / [(DIV2 + 1) \times (7 \times PSR + 1) \times (PM + 1) \times 2]$$

where PM = PM[7:0]

$$f_{FRAME\_SYN\_CLK} = (F_{INT\_BIT\_CLK}) / [(DC + 1) \times WL]$$

where DC = DC[4:0] and WL = 8, 10, 12, 16, 18, 20, 22, 24

Figure 73-24. SSI Bit Clock Equation

For example, if the SSI's sys clock is 12.288, in 8-bit word Normal mode with DC[4:0] set to 0(00000), PM[7:0] set to 47 (0010 1111), the PSR bit cleared, DIV2 bit set to 1, a bit clock rate of 12.288 Mhz / [1 x 4 x 48] = 64 kHz is generated. Since the 8-bit word rate is equal to one (i.e. normal mode), the sampling rate (FS rate) would then be 64 kHz / [1 * 8] = 8 kHz.

In next example, SSI's sys clock is 11.2896 Mhz. A 16-bit word Network mode with DC[4:0] set to 1 (00001), PM[7:0] set to 3 (0000 0011), the PSR bit is set to 0, DIV2 bit set to 0, and a 11.2896 MHz oversampling clock, a bit clock rate of 11.2896 Mhz / [1 x 2 x 4] = 1.4112 MHz is generated. Since the 16-bit word rate is equal to two, the sampling rate (FS rate) would be 1.4112 MHz / [2 * 16] = 44.1 kHz.

The table below shows programming examples for the clock dividers in the CCM and the SSI to support various bit clock (STCK) frequencies.

Table 73-6. SSI Bit Clock and Frame Rate as a Function of PSR, PM, and DIV2

Bits/ Word	Words / Frame	Ideal Frame Rate (kHz)	PLL Freq (Mhz)	SSIDIV (in CCM)	SSI's sys clock Freq (Mhz)	DIV 2	PS R	P M	W L	D C	Actual Serial bit clock Freq (kHz) STCK	Target Serial bit clock Freq (kHz) STCK	Erro r (Hz)
16	1	8	294.912	48	12.288	0	0	47	7	0	128	128	0
16	2	8	294.912	48	12.288	0	0	23	7	1	256	256	0
16	4	8	294.912	48	12.288	0	0	11	7	3	512	512	0
16	1	12	294.912	48	12.288	0	0	31	7	0	192	192	0
16	2	12	294.912	48	12.288	0	0	15	7	1	384	384	0
16	4	12	294.912	48	12.288	0	0	7	7	3	768	768	0
16	1	16	294.912	48	12.288	0	0	23	7	0	256	256	0
16	2	16	294.912	48	12.288	0	0	11	7	1	512	512	0
16	4	16	294.912	48	12.288	0	0	5	7	3	1024	1024	0
16	1	24	294.912	48	12.288	0	0	15	7	0	384	384	0
16	2	24	294.912	48	12.288	0	0	7	7	1	768	768	0
16	4	24	294.912	48	12.288	0	0	3	7	3	1536	1536	0
16	1	32	294.912	48	12.288	0	0	11	7	0	512	512	0
16	2	32	294.912	48	12.288	0	0	5	7	1	1024	1024	0
16	4	32	294.912	48	12.288	0	0	2	7	3	2048	2048	0
16	1	48	294.912	48	12.288	0	0	15	7	0	768	768	0
16	2	48	294.912	48	12.288	0	0	3	7	1	1536	1536	0
16	4	48	294.912	48	12.288	0	0	1	7	3	3072	3072	0

Table continues on the next page...

Table 73-6. SSI Bit Clock and Frame Rate as a Function of PSR, PM, and DIV2 (continued)

Bits/ Word	Words / Frame	Ideal Frame Rate (kHz)	PLL Freq (Mhz)	SSIDIV (in CCM)	SSI's sys clock Freq (Mhz)	DIV 2	PS R	P M	W L	D C	Actual Serial bit clock Freq (kHz) STCK	Target Serial bit clock Freq (kHz) STCK	Erro r (Hz)
16	1	11.025	270.9504	48	11.2896	0	0	31	7	0	176.4	176.4	0
16	2	11.025	270.9504	48	11.2896	0	0	15	7	1	352.8	352.8	0
16	4	11.025	270.9504	48	11.2896	0	0	7	7	3	705.6	705.6	0
16	1	22.05	270.9504	48	11.2896	0	0	15	7	0	352.8	352.8	0
16	2	22.05	270.9504	48	11.2896	0	0	7	7	1	705.6	705.6	0
16	4	22.05	270.9504	48	11.2896	0	0	3	7	3	1411.2	1411.2	0
16	1	44.1	270.9504	48	11.2896	0	0	7	7	0	705.6	705.6	0
16	2	44.1	270.9504	48	11.2896	0	0	3	7	1	1411.2	1411.2	0
16	4	44.1	270.9504	48	11.2896	0	0	1	7	3	2822.4	2822.4	0

NOTE

The table above describes how various frame rates can be achieved with the PLLs supplying a frequency of 294.912 MHz and 270.9504 MHz (with WL and DC settings as shown). These clocks are recommended as convenient starting points but the system allows for other input clock frequencies as well.

Table 73-6 shows programming of the CCM and SSI dividers in order to generate the appropriate network clock and serial bit clock frequencies for various sampling rates. In these examples, the master mode is selected either by setting I2S master bit (SCR[6:5]=01) or individually programming the SSI in network, synchronous, transmit internal mode (the table specifically illustrates the I2S mode frequencies/sample rates). The network clock is oversampling clock.

Note that the I2S master mode requires that a word length of 32 bits be used (regardless of the actual data type). Consequently, the fixed I2S frame rate of 64 bits per frame (word length (WL) can be any value) and DC of 1 are assumed.

73.7.5 Receive Interrupt Enable Bit Description

When the RIE and RE bit are set, the processor is interrupted when either of the SSI Receive FIFO Full (RFF0/1) bits in SSI.SISR is set (if the corresponding Receive FIFO is enabled). If the Receive FIFO is not enabled, the interrupt is generated when the corresponding SSI Receive Data Ready (RDR0/1) bit in the SSI.SISR is set

When the receive FIFO is enabled, a maximum of 15 values are available to be read (15 values per channel in Two-Channel mode). If not enabled, then one value can be read from the SRX register (one each in case of Two-Channel mode). If the RIE bit is cleared, these interrupts are disabled. However, the RFF0/1 and RDR0/1 bits still indicate the receive data register full condition. Reading the SSI.SRX registers clears the RDR bits, thus clearing the pending interrupt. Two receive data interrupts (two per channel in case of Two-Channel mode) are available: receive data with exception status and receive data without exception. The tables below show the conditions under which these interrupts are generated.

Table 73-7. SSI Receive Data 1 Interrupts

Interrupt	RIE	ROE0	RFF0/RDR0
Receive Data 1(with Exception Status)	1	1	1
Receive Data 1(without exception)	1	0	1

Table 73-8. SSI Receive Data 0 Interrupts

Interrupt	RIE	ROE1	RFF1/RDR1
Receive Data 0 (with Exception Status)	1	1	1
Receive Data 0 (without exception)	1	0	1

73.7.6 Transmit Interrupt Enable Bit Description

The SSI Transmit Interrupt Enable (TIE) control bit determines whether the processor is interrupted when the SSI transmitter needs to be serviced.

When the TIE and TE bits are set, the program controller is interrupted when either of the SSI Transmit FIFO Empty (TFE0/1) flags in SISR are set (if corresponding Transmit FIFO is enabled). If the corresponding Transmit FIFO is not enabled, an interrupt is generated when the corresponding SSI Transmit Data Register Empty (TDE0/1) flag in the SISR is set and Transmit Enable (TE) bit is set.

When Transmit FIFO 0 is enabled, a maximum of 15 values can be written to the SSI (15 per channel in case of Two-Channel mode, using Tx FIFO 1). If not enabled, then one value can be written to the SSI.STX0 register (one per channel in case of Two-Channel mode using SSI.STX1). When the TIE bit is cleared, all transmit interrupts are disabled. However, the TDE0/1 bits always indicate the corresponding SSI.STX register empty condition, even when the transmitter is disabled by the Transmit Enable (TE) bit (in the SSI.SCR). Writing data to the STX clears the corresponding TDE bit, thus clearing the interrupt. Two transmit data interrupts are available (four in case of Two-

Channel mode, two per channel): transmit data with exception status and transmit data without exceptions. The tables below show the conditions under which these interrupts are generated.

Table 73-9. SSI Transmit Data 1 Interrupts

Interrupt	TIE	TUE1	TFE1/TDE1
Transmit Data 1 (with Exception Status)	1	1	1
Transmit Data 1 (without exception)	1	0	1

Table 73-10. SSI Transmit Data 0 Interrupts

Interrupt	TIE	TUE0	TFE0/TDE0
Transmit Data 0 (with Exception Status)	1	1	1
Transmit Data 0 (without exception)	1	0	1

73.7.7 Internal Frame and Clock Shutdown

During transmit/receive operation, disabling TE/RE will ensure that data transmission/ reception stops after current frame ends following which TFRC/RFRC Status bits will get set to indicate the Frame Completion State.

If TE is disabled 4 clock cycles before the next frame, extra frame generated are invalid frames. TFR_CLK_DIS/RFR_CLK_DIS bit is set in the current or any of the previous frames, SSI will stop driving the STFS/SRFS and STCK/SRCK signals after the current frame ends.

If TFR_CLK_DIS/RFR_CLK_DIS bit is not set, SSI will continue generating STFS/ SRFS and STCK/SRCK signals (in case direction is from SSI), which then can be disabled by writing '1' to TFR_CLK_DIS/RFR_CLK_DIS bit. SSI will then stop driving these signals after end of frame is reached following which TFRC/RFRC status bits will get set to indicate the Frame Completion State.

The following figure is an illustration of transmission case where TXDIR and TFDIR are both set to '1'. In this case TE is disabled with TFR_CLK_DIS bit set in current or any of the previous frames.

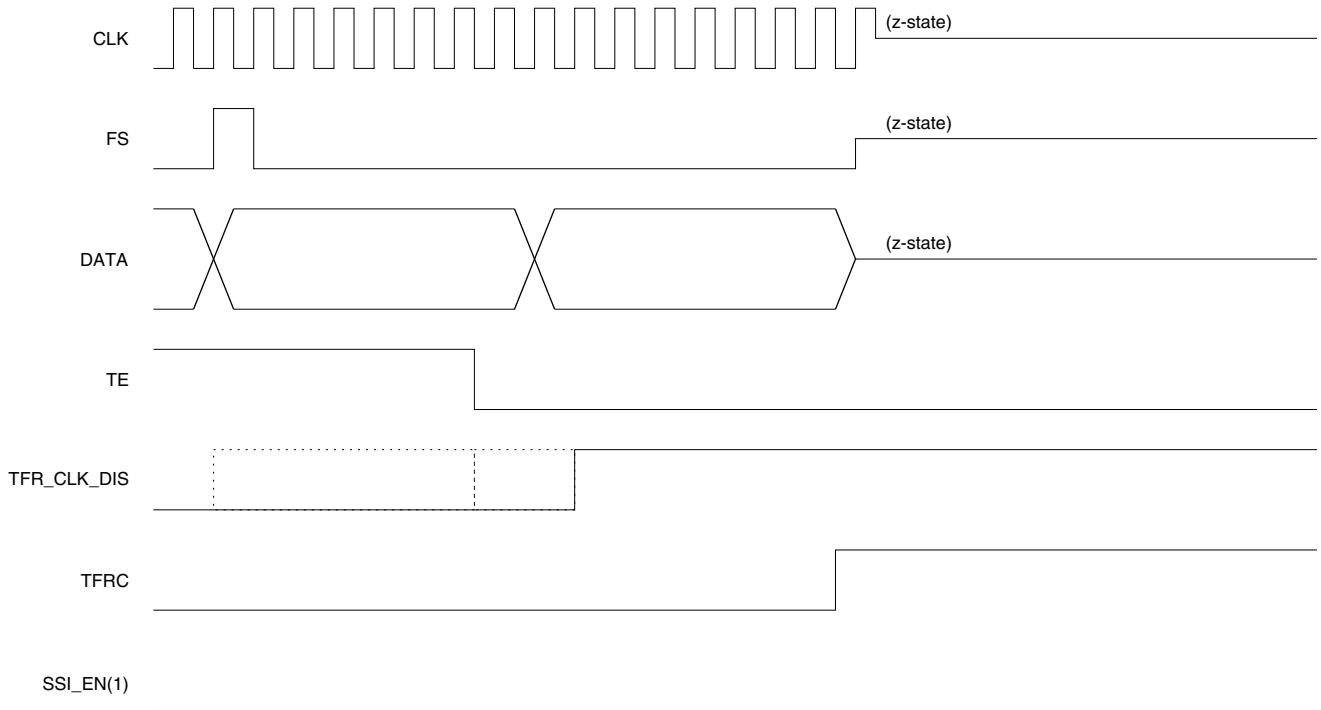


Figure 73-25. TFR_CLK_DIS assertion in current or previous frame as TE disable

The figure below is an illustration of transmission case where TXDIR and TFDIR are both set to '1'. In this case TFR_CLK_DIS bit is set after few frames of disabling TE. TFRC (Transmit Frame Complete) is set at frame boundary after TE is cleared. Once software services this interrupt and sets TFR_CLK_DIS bit later, TFRC bit is again set at next frame boundary.

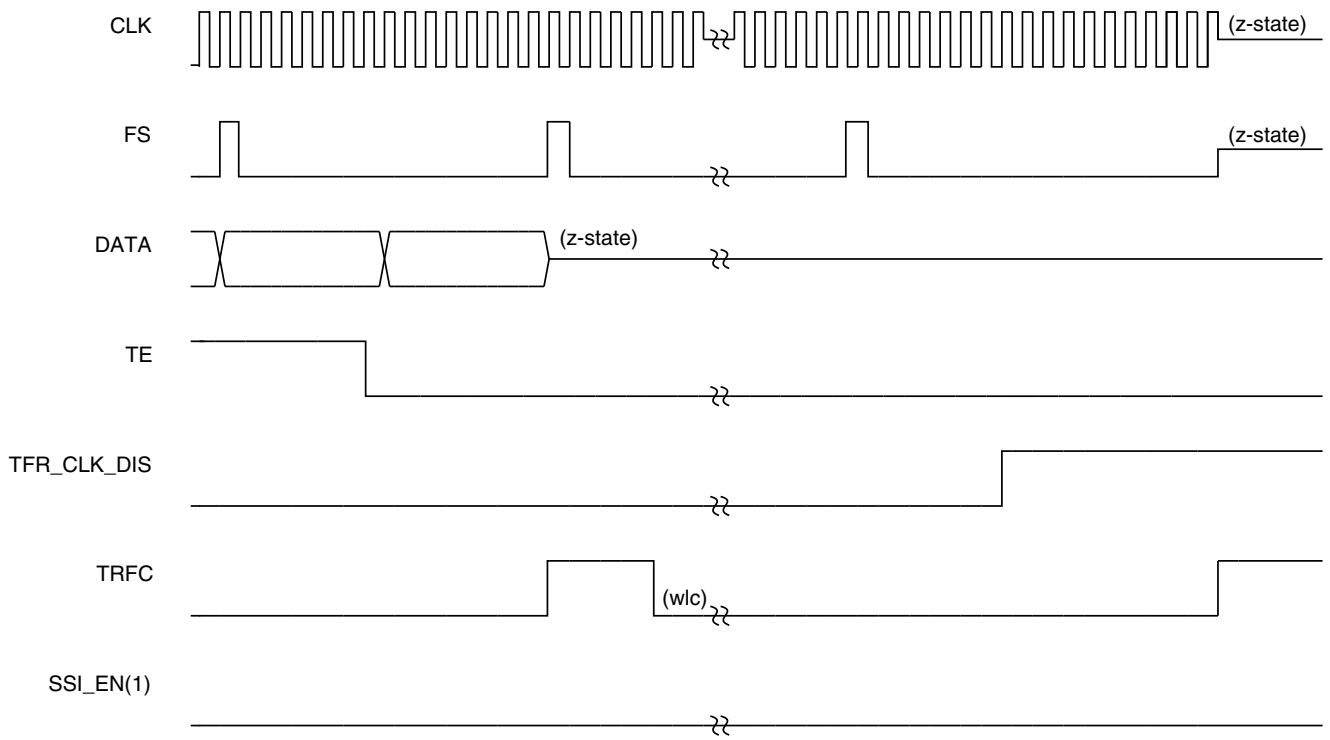


Figure 73-26. TFR_CLK_DIS assertion in subsequent frame after disabling TE

73.7.8 Peripheral Bus Interface

The SSI has a Peripheral Bus interface to provide a control and data interface. This interface is used by both the processor and DMA controller.

73.7.8.1 Transfer Lengths Supported

The Peripheral Bus interface of the SSI only supports 32-bit transfers with all SSI registers other than SSI.STX0, SSI.STX1, SSI.SRX0, and SSI.SRX1 (that is, the data registers).

With the exception of the data registers, using 8-bit and 16-bit transactions could result in undesired behavior but will not result in a transfer bus error. The data registers (SSI.STX0, SSI.STX1, SSI.SRX0, and SSI.SRX1) support 8-bit, 16-bit, and 32-bit transfer lengths without restrictions.

73.7.8.2 Transfer Bus Errors

Transfer bus errors are generated upon response to the following:

- Write transfer to a read-only register.
- Read or write access to a register space beyond the last populated register of the SSI in its memory map (up until the end of the allocated memory address range of the SSI).

73.7.8.3 Clock Rate

The Peripheral Bus clock frequency must be at least five times the serial bit clock frequency.

73.7.9 Reset

The SSI is affected by the following types of reset:

- Power-on Reset-The Power-on reset clears the SSIEN bit in SSI.SCR, which disables the SSI. All other status and control bits in the SSI are affected as described in SSI Programming Model in the "Memory Map and Register Definition section".
- SSI Reset-The SSI reset is generated when the SSIEN bit in the SSI.SCR is cleared. The SSI status bits are preset to the same state produced by the Power-on reset. The SSI control bits are unaffected. The control bits in the SSI.SCR are also unaffected. The SSI reset is useful for selective reset of the SSI without changing the present SSI control bits and without affecting the other peripherals.

73.8 Programmable Registers

SSI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5001_4000	SSI Transmit Data Register n (SSI-2_SSI_STX0)	32	R/W	0000_0000h	73.8.1/ 4486
5001_4004	SSI Transmit Data Register n (SSI-2_SSI_STX1)	32	R/W	0000_0000h	73.8.1/ 4486
5001_4008	SSI Receive Data Register n (SSI-2_SSI_SRX0)	32	R	0000_0000h	73.8.2/ 4486
5001_400C	SSI Receive Data Register n (SSI-2_SSI_SRX1)	32	R	0000_0000h	73.8.2/ 4486

Table continues on the next page...

SSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5001_4010	SSI Control Register (SSI-2_SSI_SCR)	32	R/W	0000_0000h	73.8.3/ 4487
5001_4014	SSI Interrupt Status Register (SSI-2_SSI_SISR)	32	w1c	0000_3003h	73.8.4/ 4489
5001_4018	SSI Interrupt Enable Register (SSI-2_SIER)	32	R/W	0000_3003h	73.8.5/ 4494
5001_401C	SSI Transmit Configuration Register (SSI-2_SSI_STCR)	32	R/W	0000_0200h	73.8.6/ 4496
5001_4020	SSI Receive Configuration Register (SSI-2_SSI_SRCR)	32	R/W	0000_0200h	73.8.7/ 4498
5001_4024	SSI Transmit Clock Control Register (SSI-2_SSI_STCCR)	32	R/W	0004_0000h	73.8.8/ 4500
5001_4028	SSI Receive Clock Control Register (SSI-2_SRCCR)	32	R/W	0004_0000h	73.8.9/ 4502
5001_402C	SSI FIFO Control/Status Register (SSI-2_SSI_SFCSR)	32	R/W	0081_0081h	73.8.10/ 4503
5001_4038	SSI AC97 Control Register (SSI-2_SSI_SACNT)	32	R/W	0000_0000h	73.8.11/ 4507
5001_403C	SSI AC97 Command Address Register (SSI-2_SSI_SACADD)	32	R/W	0000_0000h	73.8.12/ 4508
5001_4040	SSI AC97 Command Data Register (SSI-2_SSI_SACDAT)	32	R/W	0000_0000h	73.8.13/ 4509
5001_4044	SSI AC97 Tag Register (SSI-2_SATAG)	32	R/W	0000_0000h	73.8.14/ 4509
5001_4048	SSI Transmit Time Slot Mask Register (SSI-2_SSI_STMSK)	32	R/W	0000_0000h	73.8.15/ 4510
5001_404C	SSI Receive Time Slot Mask Register (SSI-2_SSI_SRMSK)	32	R/W	0000_0000h	73.8.16/ 4510
5001_4050	SSI AC97 Channel Status Register (SSI-2_SSI_SACCST)	32	R	0000_0000h	73.8.17/ 4511
5001_4054	SSI AC97 Channel Enable Register (SSI-2_SSI_SACCEN)	32	W	0000_0000h	73.8.18/ 4511
5001_4058	SSI AC97 Channel Disable Register (SSI-2_SSI_SACCDIS)	32	W	0000_0000h	73.8.19/ 4512
63FC_C000	SSI Transmit Data Register n (SSI-1_SSI_STX0)	32	R/W	0000_0000h	73.8.1/ 4486
63FC_C004	SSI Transmit Data Register n (SSI-1_SSI_STX1)	32	R/W	0000_0000h	73.8.1/ 4486
63FC_C008	SSI Receive Data Register n (SSI-1_SSI_SRX0)	32	R	0000_0000h	73.8.2/ 4486

Table continues on the next page...

SSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FC_C00C	SSI Receive Data Register n (SSI-1_SSI_SRX1)	32	R	0000_0000h	73.8.2/ 4486
63FC_C010	SSI Control Register (SSI-1_SSI_SCR)	32	R/W	0000_0000h	73.8.3/ 4487
63FC_C014	SSI Interrupt Status Register (SSI-1_SSI_SISR)	32	w1c	0000_3003h	73.8.4/ 4489
63FC_C018	SSI Interrupt Enable Register (SSI-1_SIER)	32	R/W	0000_3003h	73.8.5/ 4494
63FC_C01C	SSI Transmit Configuration Register (SSI-1_SSI_STCR)	32	R/W	0000_0200h	73.8.6/ 4496
63FC_C020	SSI Receive Configuration Register (SSI-1_SSI_SRCR)	32	R/W	0000_0200h	73.8.7/ 4498
63FC_C024	SSI Transmit Clock Control Register (SSI-1_SSI_STCCR)	32	R/W	0004_0000h	73.8.8/ 4500
63FC_C028	SSI Receive Clock Control Register (SSI-1_SRCCR)	32	R/W	0004_0000h	73.8.9/ 4502
63FC_C02C	SSI FIFO Control/Status Register (SSI-1_SSI_SFCSR)	32	R/W	0081_0081h	73.8.10/ 4503
63FC_C038	SSI AC97 Control Register (SSI-1_SSI_SACNT)	32	R/W	0000_0000h	73.8.11/ 4507
63FC_C03C	SSI AC97 Command Address Register (SSI-1_SSI_SACADD)	32	R/W	0000_0000h	73.8.12/ 4508
63FC_C040	SSI AC97 Command Data Register (SSI-1_SSI_SACDAT)	32	R/W	0000_0000h	73.8.13/ 4509
63FC_C044	SSI AC97 Tag Register (SSI-1_SATAG)	32	R/W	0000_0000h	73.8.14/ 4509
63FC_C048	SSI Transmit Time Slot Mask Register (SSI-1_SSI_STMSK)	32	R/W	0000_0000h	73.8.15/ 4510
63FC_C04C	SSI Receive Time Slot Mask Register (SSI-1_SSI_SRMSK)	32	R/W	0000_0000h	73.8.16/ 4510
63FC_C050	SSI AC97 Channel Status Register (SSI-1_SSI_SACCST)	32	R	0000_0000h	73.8.17/ 4511
63FC_C054	SSI AC97 Channel Enable Register (SSI-1_SSI_SACCEN)	32	W	0000_0000h	73.8.18/ 4511
63FC_C058	SSI AC97 Channel Disable Register (SSI-1_SSI_SACCDIS)	32	W	0000_0000h	73.8.19/ 4512
63FE_8000	SSI Transmit Data Register n (SSI-3_SSI_STX0)	32	R/W	0000_0000h	73.8.1/ 4486
63FE_8004	SSI Transmit Data Register n (SSI-3_SSI_STX1)	32	R/W	0000_0000h	73.8.1/ 4486

Table continues on the next page...

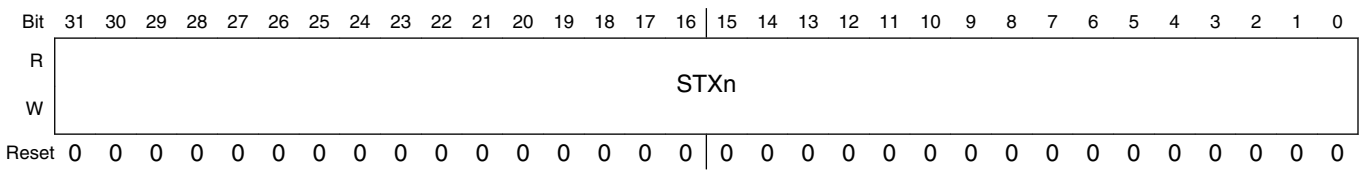
SSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FE_8008	SSI Receive Data Register n (SSI-3_SSI_SRX0)	32	R	0000_0000h	73.8.2/ 4486
63FE_800C	SSI Receive Data Register n (SSI-3_SSI_SRX1)	32	R	0000_0000h	73.8.2/ 4486
63FE_8010	SSI Control Register (SSI-3_SSI_SCR)	32	R/W	0000_0000h	73.8.3/ 4487
63FE_8014	SSI Interrupt Status Register (SSI-3_SSI_SISR)	32	w1c	0000_3003h	73.8.4/ 4489
63FE_8018	SSI Interrupt Enable Register (SSI-3_SIER)	32	R/W	0000_3003h	73.8.5/ 4494
63FE_801C	SSI Transmit Configuration Register (SSI-3_SSI_STCR)	32	R/W	0000_0200h	73.8.6/ 4496
63FE_8020	SSI Receive Configuration Register (SSI-3_SSI_SRCR)	32	R/W	0000_0200h	73.8.7/ 4498
63FE_8024	SSI Transmit Clock Control Register (SSI-3_SSI_STCCR)	32	R/W	0004_0000h	73.8.8/ 4500
63FE_8028	SSI Receive Clock Control Register (SSI-3_SRCCR)	32	R/W	0004_0000h	73.8.9/ 4502
63FE_802C	SSI FIFO Control/Status Register (SSI-3_SSI_SFCSR)	32	R/W	0081_0081h	73.8.10/ 4503
63FE_8038	SSI AC97 Control Register (SSI-3_SSI_SACNT)	32	R/W	0000_0000h	73.8.11/ 4507
63FE_803C	SSI AC97 Command Address Register (SSI-3_SSI_SACADD)	32	R/W	0000_0000h	73.8.12/ 4508
63FE_8040	SSI AC97 Command Data Register (SSI-3_SSI_SACDAT)	32	R/W	0000_0000h	73.8.13/ 4509
63FE_8044	SSI AC97 Tag Register (SSI-3_SATAG)	32	R/W	0000_0000h	73.8.14/ 4509
63FE_8048	SSI Transmit Time Slot Mask Register (SSI-3_SSI_STMSK)	32	R/W	0000_0000h	73.8.15/ 4510
63FE_804C	SSI Receive Time Slot Mask Register (SSI-3_SSI_SRMSK)	32	R/W	0000_0000h	73.8.16/ 4510
63FE_8050	SSI AC97 Channel Status Register (SSI-3_SSI_SACCST)	32	R	0000_0000h	73.8.17/ 4511
63FE_8054	SSI AC97 Channel Enable Register (SSI-3_SSI_SACCEN)	32	W	0000_0000h	73.8.18/ 4511
63FE_8058	SSI AC97 Channel Disable Register (SSI-3_SSI_SACCDIS)	32	W	0000_0000h	73.8.19/ 4512

73.8.1 SSI Transmit Data Register n (SSIx_SSI_STX*n*)

Enable SSI (SSIEN=1) before writing to SSI Transmit Data Registers.

Addresses: SSI-2_SSI_STX0 is 5001_4000h base + 0h offset = 5001_4000h
SSI-2_SSI_STX1 is 5001_4000h base + 4h offset = 5001_4004h
SSI-1_SSI_STX0 is 63FC_C000h base + 0h offset = 63FC_C000h
SSI-1_SSI_STX1 is 63FC_C000h base + 4h offset = 63FC_C004h
SSI-3_SSI_STX0 is 63FE_8000h base + 0h offset = 63FE_8000h
SSI-3_SSI_STX1 is 63FE_8000h base + 4h offset = 63FE_8004h

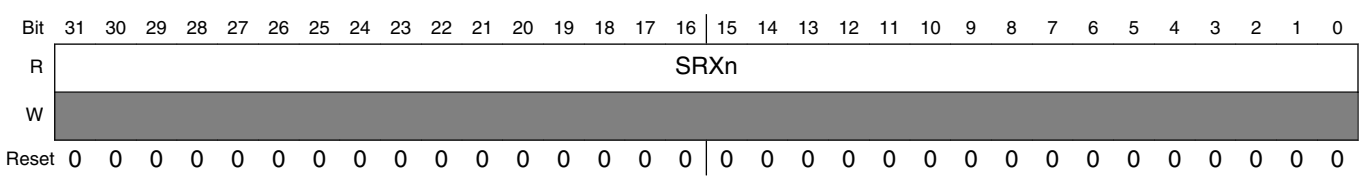


SSIx_SSI_STX*n* field descriptions

Field	Description
31–0 STXn	SSI Transmit Data. These bits store the data to be transmitted by the These are implemented as the first word of their respective Tx FIFOs. Data written to these registers is transferred to the Transmit Shift Register (TXSR), when shifting of the previous data is complete. If both FIFOs are in use, data is alternately transferred from STX0 and STX1, to TXSR. Multiple writes to the STX registers will not result in the previous data being over-written by the subsequent data. STX1 can only be used in Two-Channel mode of operation. Protection from over-writing is present irrespective of whether the transmitter is enabled or not. Example 1: If Tx FIFO0 is in use and user writes Data1...Data16 to STX0,Data16 will not over-write Data1. Data1...Data15 are stored in the FIFO whileData16 is discarded. Example 2: If Tx FIFO0 is not in use and user writes Data1, Data2 to STX0, then Data2 will not over-write Data1 and will be discarded.

73.8.2 SSI Receive Data Register n (SSIx_SSI_SRX*n*)

Addresses: SSI-2_SSI_SRX0 is 5001_4000h base + 8h offset = 5001_4008h
SSI-2_SSI_SRX1 is 5001_4000h base + Ch offset = 5001_400Ch
SSI-1_SSI_SRX0 is 63FC_C000h base + 8h offset = 63FC_C008h
SSI-1_SSI_SRX1 is 63FC_C000h base + Ch offset = 63FC_C00Ch
SSI-3_SSI_SRX0 is 63FE_8000h base + 8h offset = 63FE_8008h
SSI-3_SSI_SRX1 is 63FE_8000h base + Ch offset = 63FE_800Ch



SSIx_SSI_SRX*n* field descriptions

Field	Description
31–0 SRXn	SSI Receive Data. These bits store the data received by the These are implemented as the first word of their respective Rx FIFOs. These bits receive data from the RXSR depending on the mode of operation. In case both FIFOs are in use, data is transferred to each data register alternately. SRX1 can only be used in Two-Channel mode of operation.

73.8.3 SSI Control Register (SSIx_SSI_SCR)

The SSI Control Register (SSI_SCR) sets up the SSI reset is controlled by bit 0 in the SSI_SCR. SSI operating modes are also selected in this register (except AC97 mode which is selected in the SSI_SACNT register).

Addresses: SSI-2_SSI_SCR is 5001_4000h base + 10h offset = 5001_4010h
SSI-1_SSI_SCR is 63FC_C000h base + 10h offset = 63FC_C010h
SSI-3_SSI_SCR is 63FE_8000h base + 10h offset = 63FE_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			SYNC_TX_FS	RFR_CLK_DIS	TFR_CLK_DIS	CLK_IST	TCH_EN	SYS_CLK_EN	I2S MODE[1:0]		SYN	NET	RE	TE	SSIEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SSIx_SSI_SCR field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value zero. Reserved
12 SYNC_TX_FS	SYNC_FS_TX bit provides a safe window for TE to be visible to the internal circuit which is just after FS occurrence. When SYNC_TX_FS is set, TE(SCR[1]) gets latched on FS occurrence & latched TE is used to enable/disable SSI transmitter. TE needs setup of 2 bit-clock cycles before occurrence of FS. If TE is changed within 2 bit-clock cycles of FS occurrence, there is high probability that TE will be latched on next FS. Note: With TFR_CLK_DIS feature on, TE is used directly to enable transmitter in following cases (i) Sync mode & Rx disabled (ii) Async Mode. Latched-TE is used to disable the transmitter. This bit has no relevance in gated mode and AC97 mode. 0 - TE not latched with FS occurrence & used directly for transmitter enable/disable. 1 - TE latched with FS occurrence & latched-TE used for transmitter enable/disable.

Table continues on the next page...

SSIx_SSI_SCR field descriptions (continued)

Field	Description
11 RFR_CLK_DIS	Receive Frame Clock Disable. This bit provides the option to keep the Frame-sync and Clock enabled or to disable them after the receive frame in which the receiver is disabled. Writing to this bit has effect only when RE is disabled.The receiver is disabled by clearing the RE bit. 0 Continue Frame-sync/Clock generation after current frame during which RE is cleared. This may be required when Frame-sync and Clocks are required from SSI, even when no data is to be received. 1 Stop Frame-sync/Clock generation at next frame boundary. This will be effective also in case where receiver is already disabled in current or previous frames.
10 TFR_CLK_DIS	Transmit Frame Clock Disable. This bit provide option to keep the Frame-sync and Clock enabled or disabled after current transmit frame, in which transmitter is disabled by clearing TE bit. Writing to this bit has effect only when SSI is enabled TE is disabled. 0 Continue Frame-sync/Clock generation after current frame during which TE is cleared. This may be required when Frame-sync and Clocks are required from SSI, even when no data is to be received. 1 Stop Frame-sync/Clock generation at next frame boundary. This will be effective also in case where transmitter is already disabled in current or previous frames.
9 CLK_IST	Clock Idle State. This bit controls the idle state of the transmit clock port during SSI internal gated mode. Note: When Clock idle state is '1' the clock polarity should always be negedge triggered and when Clock idle = '0' the clock polarity should always be positive edge triggered. 0 Clock idle state is '0'. 1 Clock idle state is '1'.
8 TCH_EN	Two-Channel Operation Enable. This bit allows SSI to operate in the two-channel mode.In this mode while receiving, the RXSR transfers data to SRX0 and SRX1 alternately and while transmitting, data is alternately transferred from STX0 and STX1 to TXSR. For an even number of slots, Two-Channel Operation can be enabled to optimize usage of both FIFOs or disabled as in the case of odd number of active slots. This feature is especially useful in I2S mode, where data for Left Speaker can be placed in Tx-FIFO0 and for Right speaker in Tx-FIFO1. 0 Two-channel mode disabled. 1 Two-channel mode enabled.
7 SYS_CLK_EN	Network Clock (Oversampling Clock) Enable. When set, this bit allows the SSI to output the network clock at the SRCK port, provided that synchronous mode, and transmit internal clock mode are set. The relationship between bit clock and network clock is determined by DIV2, PSR, and PM bits. This feature is especially useful in I2S Master mode to output network clock (oversampling clock) on SRCK port. 0 network clock not output on SRCK port. 1 network clock output on SRCK port.
6–5 I2S MODE[1:0]	I2S Mode Select. These bits allow the SSI to operate in Normal, I2S Master or I2S Slave mode. Refer to I2S Mode for a detailed description of I2S Mode of operation. Refer to Table 73-4 for details regarding settings.
4 SYN	Synchronous Mode. This bit controls whether SSI is in synchronous mode or not. In synchronous mode, the transmit and receive sections of SSI share a common clock port (STCK) and frame sync port (STFS). 0 Asynchronous mode selected. 1 Synchronous mode selected.

Table continues on the next page...

SSIx_SSI_SCR field descriptions (continued)

Field	Description
3 NET	Network Mode. This bit controls whether SSI is in network mode or not. 0 Network mode not selected. 1 Network mode selected.
2 RE	Receive Enable. This control bit enables the receive section of the When this bit is enabled, data reception starts with the arrival of the next frame sync. If data is being received when this bit is cleared, data reception continues until the end of the current frame and then stops. If this bit is set again before the second to last bit of the last time slot in the current frame, then reception continues without interruption. RE should not be toggled in the same frame. 0 Receive section disabled. 1 Receive section enabled.
1 TE	Transmit Enable. This control bit enables the transmit section of the It enables the transfer of the contents of the STX registers to the TXSR and also enables the internal transmit clock. The transmit section is enabled when this bit is set and a frame boundary is detected. When this bit is cleared, the transmitter continues to send data until the end of the current frame and then stops. Data can be written to the STX registers with the TE bit cleared (the corresponding TDE bit will be cleared). If the TE bit is cleared and then set again before the second to last bit of the last time slot in the current frame, data transmission continues without interruption. The normal transmit enable sequence is to write data to the STX register(s) and then set the TE bit. The normal disable sequence is to clear the TE and TIE bits after the TDE bit is set. In gated clock mode, clearing the TE bit results in the clock stopping after the data currently in TXSR has shifted out. When the TE bit is set, the clock starts immediately (for internal gated clock mode). TE should not be toggled in the same frame. After enabling/disabling transmission, SSI expects 4 setup clock cycles before arrival of frame-sync for frame-sync to be accepted/rejected by In case of fewer clock cycles, there is high probability of the frame-sync to get missed. Note: If continuos clock is not provided, SSI expects 6 clock cycles before arrival of frame-sync for frame-sync to be accepted by 0 Transmit section disabled. 1 Transmit section enabled.
0 SSIEN	SSIEN - SSI Enable This bit is used to enable/disable the When disabled, all SSI status bits are preset to the same state produced by the power-on reset, all control bits are unaffected, the contents of Tx and Rx FIFOs are cleared. When SSI is disabled, all internal clocks are disabled (except register access clock). 0 SSI is disabled. 1 SSI is enabled.

73.8.4 SSI Interrupt Status Register (SSIx_SSI_SISR)

The SSI Interrupt Status Register (SSI_SISR) is used to monitor the This register is used by the core to interrogate the status of the In gated mode of operation the TFS, RFS, TLS, RLS, TFRC and RFRC bits of AISR register are not generated. The status bits are described in the following table.

- SSI Status flags are valid when SSI is enabled.
- See Receive Interrupt Enable Bit Description and Transmit Interrupt Enable Bit Description for interrupt source mapping.
- All the flags in the SSI_SISR are updated after the first bit of the next SSI word has completed transmission or reception. Certain status bits (ROE0/1 and TUE0/1) are cleared by writing 1 to the corresponding interrupt status bit in SSI_SISR.

Addresses: SSI-2_SSI_SISR is 5001_4000h base + 14h offset = 5001_4014h
SSI-1_SSI_SISR is 63FC_C000h base + 14h offset = 63FC_C014h
SSI-3_SSI_SISR is 63FE_8000h base + 14h offset = 63FE_8014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							RFRC	TFRC	0				CMDAU	CMDDU	RXT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDR1	RDR0	TDE1	TDE0	ROE1	ROE0	TUE1	TUE0	TFS	RFS	TLS	RLS	RFF1	RFF0	TFE1	TFE0
W					w1c	w1c	w1c	w1c								
Reset	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1

SSIx_SSI_SISR field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved
24 RFRC	Receive Frame Complete. This flag is set at the end of the frame during which Receiver is disabled. If Receive Frame & Clock are not disabled in the same frame, this flag is also set at the end of the frame in which Receive Frame & Clock are disabled. See description of RFR_CLK_DIS (SSI Control Register (SSI_SSI_SCR)) bit for more details on how to disable Receiver Frame & Clock or keep them enabled after receiver is disabled. 0 End of Frame not reached 1 End of frame reached after disabling RE or disabling RFR_CLK_DIS, when receiver is already disabled.
23 TFRC	Transmit Frame Complete. This flag is set at the end of the frame during which Transmitter is disabled. If Transmit Frame & Clock are not disabled in the same frame, this flag is also set at the end of the frame in which Transmit Frame & Clock are disabled. See description of TFR_CLK_DIS (SSI Control Register (SSI_SSI_SCR)) bit for more details on how to disable Transmit Frame & Clock or keep them enabled after transmitter is disabled.

Table continues on the next page...

SSIx_SSI_SISR field descriptions (continued)

Field	Description
	0 End of Frame not reached 1 End of frame reached after disabling TE or disabling TFR_CLK_DIS, when transmitter is already disabled.
22–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18 CMDAU	Command Address Register Updated. This bit causes the Command Address Updated interrupt (when CMDAU_EN bit is set). This status bit is set each time there is a difference in the previous and current value of the received Command Address. This bit is cleared on reading the SACADD register. 0 No change in SACADD register. 1 SACADD register updated with different value.
17 CMDDU	Command Data Register Updated. This bit causes the Command Data Updated interrupt (when CMDDU_EN bit is set). This status bit is set each time there is a difference in the previous and current value of the received Command Data. This bit is cleared on reading the SACDAT register. 0 No change in SACDAT register. 1 SACDAT register updated with different value.
16 RXT	Receive Tag Updated. This status bit is set each time there is a difference in the previous and current value of the received tag. It causes the Receive Tag Interrupt (if RXT_EN bit is set). This bit is cleared on reading the SATAG register. 0 No change in SATAG register. 1 SATAG register updated with different value.
15 RDR1	Receive Data Ready 1. This flag bit is set when SRX1 or Rx FIFO 1 is loaded with a new value and Two-Channel mode is selected. RDR1 is cleared when the Core reads the SRX1 register. If Rx FIFO 1 is enabled, RDR1 is cleared when the FIFO is empty. If RIE and RDR1_EN are set, a Receive Data 1 interrupt request is issued on setting of RDR1 bit in case Rx FIFO1 is disabled, if the FIFO is enabled, the interrupt is issued on RFF1 assertion. The RDR1 bit is cleared by POR and SSI reset. 0 No new data for Core to read. 1 New data for Core to read.
14 RDR0	Receive Data Ready 0. This flag bit is set when SRX0 or Rx FIFO 0 is loaded with a new value. RDR0 is cleared when the Core reads the SRX0 register. If Rx FIFO 0 is enabled, RDR0 is cleared when the FIFO is empty. If RIE and RDR0_EN are set, a Receive Data 0 interrupt request is issued on setting of RDR0 bit in case Rx FIFO0 is disabled, if the FIFO is enabled, the interrupt is issued on RFF0 assertion. The RDR0 bit is cleared by POR and SSI reset. 0 No new data for Core to read. 1 New data for Core to read.
13 TDE1	Transmit Data Register Empty 1. This flag is set whenever data is transferred to TXSR from STX1 register and Two-Channel mode is selected. If Tx FIFO1 is enabled, this occurs when there is at least one empty slot in STX1 or Tx FIFO1. If Tx FIFO1 is not enabled, this occurs when the contents of STX1 are transferred to TXSR. The TDE1 bit is cleared when the Core writes to STX1. If TIE and TDE1_EN are set, an SSI Transmit Data 1 interrupt request is issued on setting of TDE1 bit. The TDE1 bit is cleared by POR and SSI reset.

Table continues on the next page...

SSIx_SSI_SISR field descriptions (continued)

Field	Description
	0 Data available for transmission. 1 Data needs to be written by the Core for transmission.
12 TDE0	Transmit Data Register Empty 0. This flag is set whenever data is transferred to TXSR from STX0 register. If Tx FIFO 0 is enabled, this occurs when there is at least one empty slot in STX0 or Tx FIFO 0. If Tx FIFO 0 is not enabled, this occurs when the contents of STX0 are transferred to TXSR. The TDE0 bit is cleared when the Core writes to STX0. If TIE and TDE0_EN are set, an SSI Transmit Data 0 interrupt request is issued on setting of TDE0 bit. The TDE0 bit is cleared by POR and SSI reset. 0 Data available for transmission. 1 Data needs to be written by the Core for transmission.
11 ROE1	Receiver Overrun Error 1. This flag is set when the RXSR is filled and ready to transfer to SRX1 register or to Rx FIFO 1 (when enabled) and these are already full and Two-Channel mode is selected. If Rx FIFO 1 is enabled, this is indicated by RFF1 flag, else this is indicated by the RDR1 flag. The RXSR is not transferred in this case. The ROE1 flag causes an interrupt if RIE and ROE1_EN are set. The ROE1 bit is cleared by POR and SSI reset. It is also cleared by writing '1' to this bit. Clearing the RE bit does not affect the ROE1 bit. 0 Default interrupt issued to the Core. 1 Exception interrupt issued to the Core.
10 ROE0	Receiver Overrun Error 0. This flag is set when the RXSR is filled and ready to transfer to SRX0 register or to Rx FIFO 0 (when enabled) and these are already full. If Rx FIFO 0 is enabled, this is indicated by RFF0 flag, else this is indicated by the RDR0 flag. The RXSR is not transferred in this case. The ROE0 flag causes an interrupt if RIE and ROE0_EN are set. The ROE0 bit is cleared by POR and SSI reset. It is also cleared by writing '1' to this bit. Clearing the RE bit does not affect the ROE0 bit. 0 Default interrupt issued to the Core. 1 Exception interrupt issued to the Core.
9 TUE1	Transmitter Underrun Error 1. This flag is set when the TXSR is empty (no data to be transmitted), the TDE1 flag is set, a transmit time slot occurs and the SSI is in Two-Channel mode. When a transmit underrun error occurs, the previous data is retransmitted. In Network mode, each time slot requires data transmission (unless masked through STMSK register), when the transmitter is enabled (TE is set). The TUE1 flag causes an interrupt if TIE and TUE1_EN are set. The TUE1 bit is cleared by POR and SSI reset. It is also cleared by writing '1' to this bit. 0 Default interrupt issued to the Core. 1 Exception interrupt issued to the Core.
8 TUE0	Transmitter Underrun Error 0. This flag is set when the TXSR is empty (no data to be transmitted), the TDE0 flag is set and a transmit time slot occurs. When a transmit underrun error occurs, the previous data is retransmitted. In Network mode, each time slot requires data transmission (unless masked through STMSK register), when the transmitter is enabled (TE is set). The TUE0 flag causes an interrupt if TIE and TUE0_EN are set. The TUE0 bit is cleared by POR and SSI reset. It is also cleared by writing '1' to this bit.

Table continues on the next page...

SSIx_SSI_SISR field descriptions (continued)

Field	Description
	0 Default interrupt issued to the Core. 1 Exception interrupt issued to the Core.
7 TFS	Transmit Frame Sync. This flag indicates the occurrence of transmit frame sync. Data written to the STX registers during the time slot when the TFS flag is set, is sent during the second time slot (in Network mode) or in the next first time slot (in Normal mode). In Network mode, the TFS bit is set during transmission of the first time slot of the frame and is then cleared when starting transmission of the next time slot. In Normal mode, this bit is high for the first time slot. This flag causes an interrupt if TIE and TFS_EN are set. The TFS bit is cleared by POR and SSI reset. 0 No Occurrence of Transmit frame sync. 1 Transmit frame sync occurred during transmission of last word written to STX registers.
6 RFS	Receive Frame Sync. This flag indicates the occurrence of receive frame sync. In Network mode, the RFS bit is set when the first slot of the frame is being received. It is cleared when the next slot begins to be received. In Normal mode, this bit is always high (When DC = 0). This flag causes an interrupt if RIE and RFS_EN are set. The RFS bit is cleared by POR and SSI reset. 0 No Occurrence of Receive frame sync. 1 Receive frame sync occurred during reception of next word in SRX registers.
5 TLS	Transmit Last Time Slot. This flag indicates the last time slot in a frame. When set, it indicates that the current time slot is the last time slot of the frame. TLS is set at the start of the last transmit time slot and causes the SSI to issue an interrupt (if TIE and TLS_EN are set). TLS is not generated when frame rate is 1 in normal mode of operation. TLS is cleared when the SISR is read with this bit set. The TLS bit is cleared by POR and SSI reset. 0 Current time slot is not last time slot of frame. 1 Current time slot is the last transmit time slot of frame.
4 RLS	Receive Last Time Slot. This flag indicates the last time slot in a frame. When set, it indicates that the current time slot is the last receive time slot of the frame. RLS is set at the end of the last time slot and causes the SSI to issue an interrupt (if RIE and RLS_EN are set). RLS is cleared when the SISR is read with this bit set. The RLS bit is cleared by POR and SSI reset. 0 Current time slot is not last time slot of frame. 1 Current time slot is the last receive time slot of frame.
3 RFF1	Receive FIFO Full 1. This flag is set when Rx FIFO1 is enabled, the data level in Rx FIFO1 reaches the selected Rx FIFO WaterMark 1 (RFWM1) threshold and the SSI is in Two-Channel mode. The setting of RFF1 only causes an interrupt when RIE and RFF1_EN are set, Rx FIFO1 is enabled and the Two-Channel mode is selected. RFF1 is automatically cleared when the amount of data in Rx FIFO1 falls below the threshold. The RFF1 bit is cleared by POR and SSI reset. When Rx FIFO1 contains15 words, the maximum it can hold, all further data received (for storage in this FIFO) is ignored until the FIFO contents are read. 0 Space available in Receive FIFO1. 1 Receive FIFO1 is full.
2 RFF0	Receive FIFO Full 0. This flag is set when Rx FIFO0 is enabled and the data level in Rx FIFO0 reaches the selected Rx FIFO WaterMark 0 (RFWM0) threshold. The setting of RFF0 only causes an interrupt when RIE and RFF0_EN are set and Rx FIFO0 is enabled. RFF0 is automatically cleared when the amount of data in Rx FIFO0 falls below the threshold. The RFF0 bit is cleared by POR and SSI reset. When Rx FIFO0 contains15 words, the maximum it can hold, all further data received (for storage in this FIFO) is ignored until the FIFO contents are read.

Table continues on the next page...

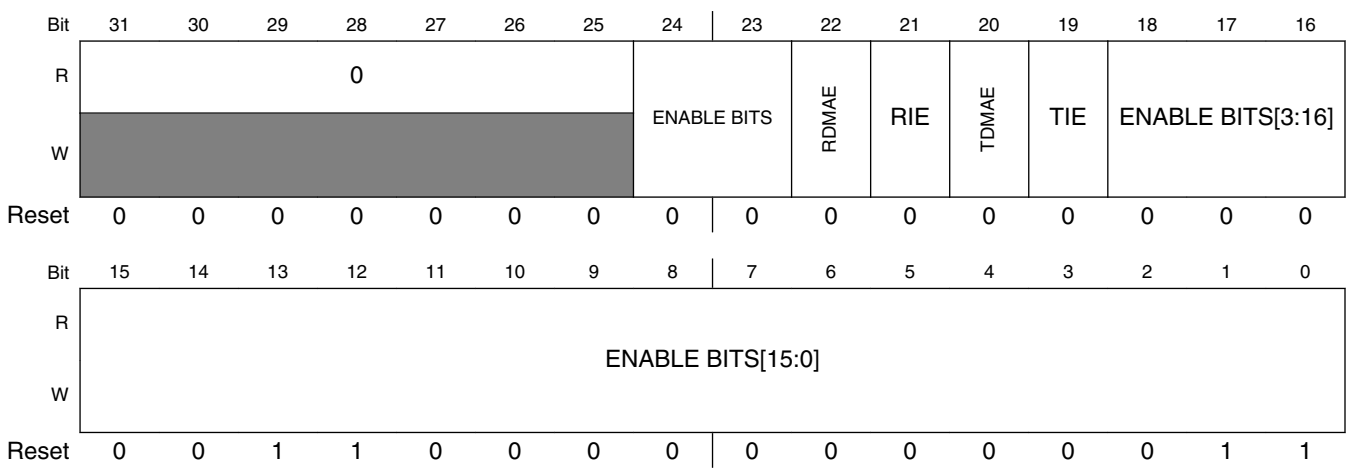
SSIx_SSI_SISR field descriptions (continued)

Field	Description
	0 Space available in Receive FIFO0. 1 Receive FIFO0 is full.
1 TFE1	Transmit FIFO Empty 1. This flag is set when the empty slots in Tx FIFO exceed or are equal to the selected Tx FIFO WaterMark 1 (TFWM1) threshold and the Two-Channel mode is selected. The setting of TFE1 only causes an interrupt when TIE and TFE1_EN are set, Tx FIFO1 is enabled and Two-Channel mode is selected. The TFE1 bit is automatically cleared when the data level in Tx FIFO1 becomes more than the amount specified by the watermark bits. The TFE1 bit is set by POR and SSI reset. 0 Transmit FIFO1 has data for transmission. 1 Transmit FIFO1 is empty.
0 TFE0	Transmit FIFO Empty 0. This flag is set when the empty slots in Tx FIFO exceed or are equal to the selected Tx FIFO WaterMark 0 (TFWM0) threshold. The setting of TFE0 only causes an interrupt when TIE and TFE0_EN are set and Tx FIFO0 is enabled. The TFE0 bit is automatically cleared when the data level in Tx FIFO0 becomes more than the amount specified by the watermark bits. The TFE0 bit is set by POR and SSI reset. 0 Transmit FIFO0 has data for transmission. 1 Transmit FIFO0 is empty.

73.8.5 SSI Interrupt Enable Register (SSIx_SIER)

The SSI Interrupt Enable Register (SIER) is a 25-bit register used to set up the SSI interrupts and DMA requests.

Addresses: SSI-2_SIER is 5001_4000h base + 18h offset = 5001_4018h
SSI-1_SIER is 63FC_C000h base + 18h offset = 63FC_C018h
SSI-3_SIER is 63FE_8000h base + 18h offset = 63FE_8018h



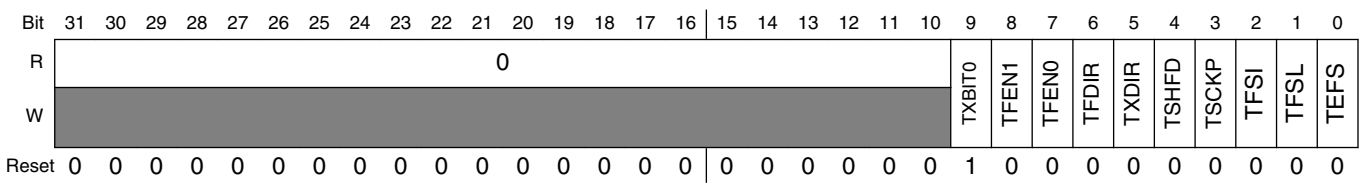
SSIx_SIER field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved
24–23 ENABLE BITS	Enable Bit. Each bit controls whether the corresponding status bit in SISR can issue an interrupt to the Core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.
22 RDMAE	Receive DMA Enable. This bit allows SSI to request for DMA transfers. When enabled, DMA requests are generated when any of the RFF0/1 bits in the SISR are set and if the corresponding RFEN bit is also set. If the corresponding FIFO is disabled, a DMA request is generated when the corresponding RDR bit is set. 0 SSI Receiver DMA requests disabled. 1 SSI Receiver DMA requests enabled.
21 RIE	Receive Interrupt Enable. This control bit allows the SSI to issue receiver related interrupts to the Core. Refer to Receive Interrupt Enable Bit Description for a detailed description of this bit. 0 SSI Receiver Interrupt requests disabled. 1 SSI Receiver Interrupt requests enabled.
20 TDMAE	Transmit DMA Enable. This bit allows SSI to request for DMA transfers. When enabled, DMA requests are generated when any of the TFE0/1 bits in the SISR are set and if the corresponding TFEN bit is also set. If the corresponding FIFO is disabled, a DMA request is generated when the corresponding TDE bit is set. 0 SSI Transmitter DMA requests disabled. 1 SSI Transmitter DMA requests enabled.
19 TIE	Transmit Interrupt Enable. This control bit allows the SSI to issue transmitter data related interrupts to the Core. Refer to Transmit Interrupt Enable Bit Description for a detailed description of this bit. 0 SSI Transmitter Interrupt requests disabled. 1 SSI Transmitter Interrupt requests enabled.
18–0 ENABLE BITS	Enable Bit. Each bit controls whether the corresponding status bit in SISR can issue an interrupt to the Core or not. 0 Corresponding status bit cannot issue interrupt. 1 Corresponding status bit can issue interrupt.

73.8.6 SSI Transmit Configuration Register (SSIx_SSI_STCR)

The SSI Transmit Configuration Register (SSI_STCR) is a read/write control registers used to direct the transmit operation of the STCR controls the direction of the bit clock and frame sync ports, STCK and STFS. Interrupt enable bit for the transmit sections is provided in this control register. The Power-on reset clears all SSI_STCR bits. However, SSI reset does not affect the SSI_STCR bits. The SSI_STCR bits are described in the following paragraphs. See the Programmable Registers section for the programming model of the SSI. The SSI Control Register (SSI_SCR) must first be set to enable interrupts. Next, the SSI interrupt bit in the Interrupt Enable Register (SSI_SIER) must be set to enable the interrupt. Finally, the interrupt can be enabled from within the

Addresses: SSI-2_SSI_STCR is 5001_4000h base + 1Ch offset = 5001_401Ch
SSI-1_SSI_STCR is 63FC_C000h base + 1Ch offset = 63FC_C01Ch
SSI-3_SSI_STCR is 63FE_8000h base + 1Ch offset = 63FE_801Ch



SSIx_SSI_STCR field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9 TXBIT0	Transmit Bit 0. This control bit allows SSI to transmit the data word from bit position 0 or 15/31 in the transmit shift register. The shifting data direction can be MSB or LSB first, controlled by the TSHFD bit. 0 Shifting with respect to bit 31 (if word length = 16, 18, 20, 22 or 24) or bit 15 (if word length = 8, 10 or 12) of transmit shift register (MSB aligned). 1 Shifting with respect to bit 0 of transmit shift register (LSB aligned).
8 TFEN1	Transmit FIFO Enable 1. This bit enables transmit FIFO 1. When enabled, the FIFO allows15 samples to be transmitted by the SSI (per channel) (a 9th sample can be shifting out) before TDE1 bit is set. When the FIFO is disabled, an interrupt is generated when a single sample is transferred to the transmit shift register (provided the interrupt is enabled). 0 Transmit FIFO 1 disabled. 1 Transmit FIFO 1 enabled.
7 TFEN0	Transmit FIFO Enable 0. This bit enables transmit FIFO 0. When enabled, the FIFO allows15 samples to be transmitted by the SSI per channel (a 9th sample can be shifting out) before TDE0 bit is set. When the FIFO is disabled, an interrupt is generated when a single sample is transferred to the transmit shift register (provided the interrupt is enabled). 0 Transmit FIFO 0 disabled. 1 Transmit FIFO 0 enabled.

Table continues on the next page...

SSIx_SSI_STCR field descriptions (continued)

Field	Description
6 TFDIR	Transmit Frame Direction. This bit controls the direction and source of the transmit frame sync signal. Internally generated frame sync signal is sent out through the STFS port and external frame sync is taken from the same port. 0 Frame Sync is external. 1 Frame Sync generated internally.
5 TXDIR	Transmit Clock Direction. This bit controls the direction and source of the clock signal used to clock the TXSR. Internally generated clock is output through the STCK port. External clock is taken from this port. Refer to Table 73-2 for details of clock pin configurations. 0 Transmit Clock is external. 1 Transmit Clock generated internally.
4 TSHFD	Transmit Shift Direction. This bit controls whether the MSB or LSB will be transmitted first in a sample. **NOTE:** The CODEC device labels the MSB as bit 0, whereas the Core labels the LSB as bit 0. Therefore, when using a standard CODEC, Core MSB (CODEC LSB) is shifted in first (TSHFD cleared). 0 Data transmitted MSB first. 1 Data transmitted LSB first.
3 TSCKP	Transmit Clock Polarity. This bit controls which bit clock edge is used to clock out data for the transmit section. Note: TSCKP is 0 CLK_IST = 0; TSCKP is 1CLK_IST = 1 0 Data clocked out on rising edge of bit clock. 1 Data clocked out on falling edge of bit clock.
2 TFSI	Transmit Frame Sync Invert. This bit controls the active state of the frame sync I/O signal for the transmit section of SSI. 0 Transmit frame sync is active high. 1 Transmit frame sync is active low.
1 TFSL	Transmit Frame Sync Length. This bit controls the length of the frame sync signal to be generated or recognized for the transmit section. The length of a word-long frame sync is same as the length of the data word selected by WL[3:0]. 0 Transmit frame sync is one-word long. 1 Transmit frame sync is one-clock-bit long.
0 TEFS	Transmit Early Frame Sync. This bit controls when the frame sync is initiated for the transmit section. The frame sync signal is deasserted after one bit-for-bit length frame sync and after one word-for-word length frame sync. In case of synchronous operation, the frame sync can also be initiated on receiving the first bit of data. 0 Transmit frame sync initiated as the first bit of data is transmitted. 1 Transmit frame sync is initiated one bit before the data is transmitted.

73.8.7 SSI Receive Configuration Register (SSIx_SSI_SRCR)

The SSI Receive Configuration Register (SSI_SRCR) is a read/write control registers used to direct the receive operation of the SSI. SSI.SRCR controls the direction of the bit clock and frame sync ports, SRCK and SRFS. Interrupt enable bit for the transmit sections is provided in this control register. The Power-on reset clears all SSI_SRCR bits. However, SSI reset does not affect the SSI_SRCR bits.

Addresses: SSI-2_SSI_SRCR is 5001_4000h base + 20h offset = 5001_4020h
SSI-1_SSI_SRCR is 63FC_C000h base + 20h offset = 63FC_C020h
SSI-3_SSI_SRCR is 63FE_8000h base + 20h offset = 63FE_8020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																					RXEXT	RXBIT0	RFEN1	RFEN0	RFDIR	RXDIR	RSHFD	RSCKP	RFSI	RFSL	REFS
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

SSIx_SSI_SRCR field descriptions

Field	Description
31–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10 RXEXT	Receive Data Extension. This control bit allows SSI to store the received data word in sign extended form. This bit affects data storage only in case received data is LSB aligned (SRCR[9]=1) 0 Sign extension turned off. 1 Sign extension turned on.
9 RXBIT0	Receive Bit 0. This control bit allows SSI to receive the data word at bit position 0 or 15/31 in the receive shift register. The shifting data direction can be MSB or LSB first, controlled by the RSHFD bit. 0 Shifting with respect to bit 31 (if word length = 16, 18, 20, 22 or 24) or bit 15 (if word length = 8, 10 or 12) of receive shift register (MSB aligned). 1 Shifting with respect to bit 0 of receive shift register (LSB aligned).
8 RFEN1	Receive FIFO Enable 1. This bit enables receive FIFO 1. When enabled, the FIFO allows15 samples to be received by the SSI per channel (a 16th sample can be shifting in) before RDR1 bit is set. When the FIFO is disabled, an interrupt is generated when a single sample is received by the SSI (provided the interrupt is enabled). 0 Receive FIFO 1 disabled. 1 Receive FIFO 1 enabled.
7 RFEN0	Receive FIFO Enable 0. This bit enables receive FIFO 0. When enabled, the FIFO allows15 samples to be received by the SSI (per channel) (a 16th sample can be shifting in) before RDR0 bit is set. When the FIFO is disabled, an interrupt is generated when a single sample is received by the SSI (provided the interrupt is enabled). 0 Receive FIFO 0 disabled. 1 Receive FIFO 0 enabled.

Table continues on the next page...

SSIx_SSI_SRCR field descriptions (continued)

Field	Description
6 RFDIR	Receive Frame Direction. This bit controls the direction and source of the receive frame sync signal. Internally generated frame sync signal is sent out through the SRFS port and external frame sync is taken from the same port. 0 Frame Sync is external. 1 Frame Sync generated internally.
5 RXDIR	Receive Clock Direction. This bit controls the direction and source of the clock signal used to clock the RXSR. Internally generated clock is output through the SRCK port. External clock is taken from this port. Refer to Table 73-2 for details on clock pin configurations. 0 Receive Clock is external. 1 Receive Clock generated internally.
4 RSHFD	Receive Shift Direction. This bit controls whether the MSB or LSB will be received first in a sample. **NOTE:** The CODEC device labels the MSB as bit 0, whereas the Core labels the LSB as bit 0. Therefore, when using a standard CODEC, Core MSB (CODEC LSB) is shifted in first (RSHFD cleared). 0 Data received MSB first. 1 Data received LSB first.
3 RSCKP	Receive Clock Polarity. This bit controls which bit clock edge is used to latch in data for the receive section. 0 Data latched on falling edge of bit clock. 1 Data latched on rising edge of bit clock.
2 RFSI	Receive Frame Sync Invert. This bit controls the active state of the frame sync I/O signal for the receive section of SSI. 0 Receive frame sync is active high. 1 Receive frame sync is active low.
1 RFSL	Receive Frame Sync Length. This bit controls the length of the frame sync signal to be generated or recognized for the receive section. The length of a word-long frame sync is same as the length of the data word selected by WL[3:0]. 0 Receive frame sync is one-word long. 1 Receive frame sync is one-clock-bit long.
0 REFS	Receive Early Frame Sync. This bit controls when the frame sync is initiated for the receive section. The frame sync is disabled after one bit-for-bit length frame sync and after one word-for-word length frame sync. 0 Receive frame sync initiated as the first bit of data is received. 1 Receive frame sync is initiated one bit before the data is received.

73.8.8 SSI Transmit Clock Control Register (SSIx_SSI_STCCR)

The SSI Transmit and Receive Control (SSI_STCCR and SSI_SRCCR) registers are 19-bit, read/write control registers used to direct the operation of the SSI. The Clock Controller Module (CCM) can source the SSI clock (SSI's sys clock-from CCM's ssi_clk_root) from multiple sources and perform fractional division to support commonly used audio bit rates. The CCM can maintain the SSI's sys clock frequency at a constant rate even in cases where the ipg_clk (from CCM) frequency changes. These registers control the SSI clock generator, bit and frame sync rates, word length, and number of words per frame for the serial data. The SSI.STCCR register is dedicated to the transmit section, and the SSI_SRCCR register is dedicated to the receive section except in Synchronous mode, in which the SSI_STCCR register controls both the receive and transmit sections. Power-on reset clears all SSI_STCCR and SSI_SRCCR bits. SSI reset does not affect the SSI_STCCR and SSI_SRCCR bits. The control bits are described in the following paragraphs. Although the bit patterns of the SSI_STCCR and SSI_SRCCR registers are the same, the contents of these two registers can be programmed differently.

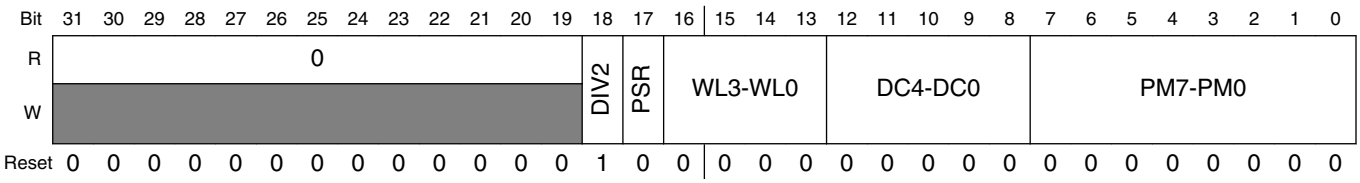
Table 73-51. SSI Data Length

WL3	WL2	WL1	WL0	Number of Bits/Word	Supported in Implementation
0	0	0	0	2	No
0	0	0	1	4	No
0	0	1	0	6	No
0	0	1	1	8	Yes
0	1	0	0	10	Yes
0	1	0	1	12	Yes
0	1	1	0	14	No
0	1	1	1	16	Yes
1	0	0	0	18	Yes
1	0	0	1	20	Yes
1	0	1	0	22	Yes
1	0	1	1	24	Yes
1	1	0	0	26	No
1	1	0	1	28	No
1	1	1	0	30	No
1	1	1	1	32	No

Addresses: SSI-2_SSI_STCCR is 5001_4000h base + 24h offset = 5001_4024h

SSI-1_SSI_STCCR is 63FC_C000h base + 24h offset = 63FC_C024h

SSI-3_SSI_STCCR is 63FE_8000h base + 24h offset = 63FE_8024h



SSIx_SSI_STCCR field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18 DIV2	Divide By 2. This bit controls a divide-by-two divider in series with the rest of the prescalers. 0 Divider bypassed. 1 Divider used to divide clock by 2.
17 PSR	Prescaler Range. This bit controls a fixed divide-by-eight prescaler in series with the variable prescaler. It extends the range of the prescaler for those cases where a slower bit clock is required. 0 Prescaler bypassed. 1 Prescaler used to divide clock by 8.
16–13 WL3-WL0	Word Length Control. These bits are used to control the length of the data words being transferred by the SSI. These bits control the Word Length Divider in the Clock Generator. They also control the frame sync pulse length when the FSL bit is cleared. In I2S Master mode, the SSI works with a fixed word length of 32, and the WL bits are used to control the amount of valid data in those 32 bits. In AC97 Mode of operation, if word length is set to any value other than 16 bits, it will result in a word length of 20 bits.
12–8 DC4-DC0	Frame Rate Divider Control. These bits are used to control the divide ratio for the programmable frame rate dividers. The divide ratio works on the word clock. In Normal mode, this ratio determines the word transfer rate. In Network mode, this ratio sets the number of words per frame. The divide ratio ranges from 1 to 32 in Normal mode and from 2 to 32 in Network mode. In Normal mode, a divide ratio of 1 (DC=00000) provides continuous periodic data word transfer. A bit-length frame sync must be used in this case. These bits can be programmed with values ranging from "00000" to "11111" to control the number of words in a frame.
7–0 PM7-PM0	Prescaler Modulus Select. These bits control the prescale divider in the clock generator. This prescaler is used only in Internal Clock mode to divide the internal clock. The bit clock output is available at the clock port. A divide ratio from 1 to 256 (PM[7:0] = 0x00 to 0xFF) can be selected. Refer to DIV2, PSR and PM Bit Description for details regarding settings.

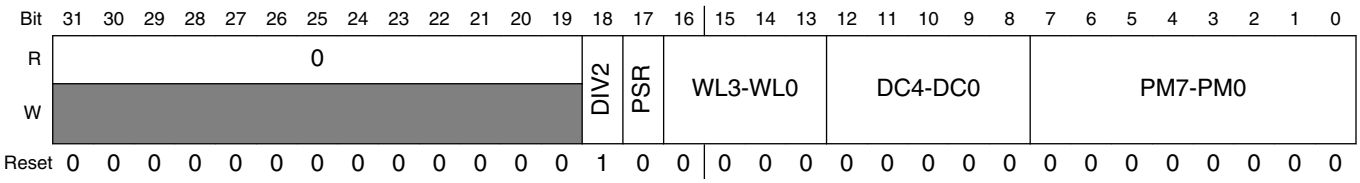
73.8.9 SSI Receive Clock Control Register (SSIx_SRCCR)

The SSI Transmit and Receive Control (SSI_STCCR and SSI_SRCCR) registers are 19-bit, read/write control registers used to direct the operation of the SSI. The Clock Controller Module (CCM) can source the SSI clock (SSI's sys clock-from CCM's ssi_clk_root) from multiple sources and perform fractional division to support commonly used audio bit rates. The CCM can maintain the SSI's sys clock frequency at a constant rate even in cases where the ipg_clk from CCM frequency changes. These registers control the SSI clock generator, bit and frame sync rates, word length, and number of words per frame for the serial data. The SSI_STCCR register is dedicated to the transmit section, and the SSI_SRCCR register is dedicated to the receive section except in Synchronous mode, in which the SSI_STCCR register controls both the receive and transmit sections. Power-on reset clears all SSI_STCCR and SSI_SRCCR bits. SSI reset does not affect the SSI_STCCR and SSI_SRCCR bits. The control bits are described in the following paragraphs. Although the bit patterns of the SSI_STCCR and SSI_SRCCR registers are the same, the contents of these two registers can be programmed differently.

Table 73-53. SSI Data Length

WL3	WL2	WL1	WL0	Number of Bits/Word	Supported in Implementation
0	0	0	0	2	No
0	0	0	1	4	No
0	0	1	0	6	No
0	0	1	1	8	Yes
0	1	0	0	10	Yes
0	1	0	1	12	Yes
0	1	1	0	14	No
0	1	1	1	16	Yes
1	0	0	0	18	Yes
1	0	0	1	20	Yes
1	0	1	0	22	Yes
1	0	1	1	24	Yes
1	1	0	0	26	No
1	1	0	1	28	No
1	1	1	0	30	No
1	1	1	1	32	No

Addresses: SSI-2_SRCCR is 5001_4000h base + 28h offset = 5001_4028h
SSI-1_SRCCR is 63FC_C000h base + 28h offset = 63FC_C028h
SSI-3_SRCCR is 63FE_8000h base + 28h offset = 63FE_8028h



SSIx_SRCCR field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18 DIV2	Divide By 2. This bit controls a divide-by-two divider in series with the rest of the prescalers. 0 Divider bypassed. 1 Divider used to divide clock by 2.
17 PSR	Prescaler Range. This bit controls a fixed divide-by-eight prescaler in series with the variable prescaler. It extends the range of the prescaler for those cases where a slower bit clock is required. 0 Prescaler bypassed. 1 Prescaler used to divide clock by 8.
16–13 WL3-WL0	Word Length Control. These bits are used to control the length of the data words being transferred by the SSI. These bits control the Word Length Divider in the Clock Generator. They also control the frame sync pulse length when the FSL bit is cleared. In I2S Master mode, the SSI works with a fixed word length of 32, and the WL bits are used to control the amount of valid data in those 32 bits. In AC97 Mode of operation, if word length is set to any value other than 16 bits, it will result in a word length of 20 bits.
12–8 DC4-DC0	Frame Rate Divider Control. These bits are used to control the divide ratio for the programmable frame rate dividers. The divide ratio works on the word clock. In Normal mode, this ratio determines the word transfer rate. In Network mode, this ratio sets the number of words per frame. The divide ratio ranges from 1 to 32 in Normal mode and from 2 to 32 in Network mode. In Normal mode, a divide ratio of 1 (DC=00000) provides continuous periodic data word transfer. A bit-length frame sync must be used in this case. These bits can be programmed with values ranging from "00000" to "11111" to control the number of words in a frame.
7–0 PM7-PM0	Prescaler Modulus Select. These bits control the prescale divider in the clock generator. This prescaler is used only in Internal Clock mode to divide the internal clock. The bit clock output is available at the clock port. A divide ratio from 1 to 256 (PM[7:0] = 0x00 to 0xFF) can be selected. Refer to DIV2, PSR and PM Bit Description for details regarding settings.

73.8.10 SSI FIFO Control/Status Register (SSIx_SSI_SFCSR)

The SSI FIFO Control / Status Register indicates the status of the Transmit FIFO Empty flag, with different settings of the Transmit FIFO WaterMark bits and varying amounts of data in the Tx FIFO

.

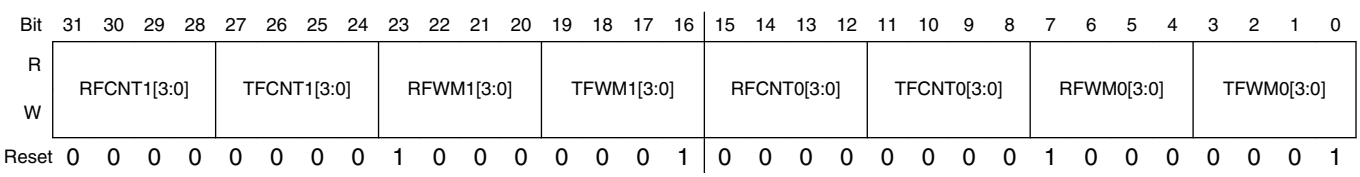
Table 73-55. Status of Transmit FIFO Empty Flag

Transmit FIFO Watermark (TFWM)	Number of data in Tx-Fifo														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
2	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
3	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
4	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
5	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
6	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
7	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
8	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
9	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
10	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
11	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
12	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
13	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Addresses: SSI-2_SSI_SFCSR is 5001_4000h base + 2Ch offset = 5001_402Ch

SSI-1_SSI_SFCSR is 63FC_C000h base + 2Ch offset = 63FC_C02Ch

SSI-3_SSI_SFCSR is 63FE_8000h base + 2Ch offset = 63FE_802Ch



SSIx_SSI_SFCSR field descriptions

Field	Description
31–28 RFCNT1[3:0]	Receive FIFO Counter1. These bits indicate the number of data words in Receive FIFO 1. 0000 0 data word in receive FIFO 0001 1 data word in receive FIFO 0010 2 data word in receive FIFO 0011 3 data word in receive FIFO 0100 4 data word in receive FIFO 0101 5 data word in receive FIFO 0110 6 data word in receive FIFO 0111 7 data word in receive FIFO

Table continues on the next page...

SSIx_SSI_SFCSR field descriptions (continued)

Field	Description
	1000 8 data word in receive FIFO 1001 9 data word in receive FIFO 1010 10 data word in receive FIFO 1011 11 data word in receive FIFO 1100 12 data word in receive FIFO 1101 13 data word in receive FIFO 1110 14 data word in receive FIFO 1111 15 data word in receive FIFO
27–24 TFCNT1[3:0]	Transmit FIFO Counter1. These bits indicate the number of data words in Transmit FIFO. 0000 0 data word in transmit FIFO 0001 1 data word in transmit FIFO 0010 2 data word in transmit FIFO 0011 3 data word in transmit FIFO 0100 4 data word in transmit FIFO 0101 5 data word in transmit FIFO 0110 6 data word in transmit FIFO 0111 7 data word in transmit FIFO 1000 8 data word in transmit FIFO 1001 9 data word in transmit FIFO 1010 10 data word in transmit FIFO 1011 11 data word in transmit FIFO 1100 12 data word in transmit FIFO 1101 13 data word in transmit FIFO 1110 14 data word in transmit FIFO 1111 15 data word in transmit FIFO
23–20 RFWM1[3:0]	Receive FIFO Full WaterMark 1. These bits control the threshold at which the RFF1 flag will be set. The RFF1 flag is set whenever the data level in Rx FIFO 1 reaches the selected threshold. 0000 Reserved 0001 RFF set when at least one data word has been written to the Receive FIFO. Set when RxFIFO = 1,2.....15 data words 0010 RFF set when 2 or more data words have been written to the Receive FIFO. Set when RxFIFO = 2,3.....15 data words 0011 RFF set when 3 or more data words have been written to the Receive FIFO. Set when RxFIFO = 3,4.....15 data words 0100 RFF set when 4 or more data words have been written to the Receive FIFO. Set when RxFIFO = 4,5.....15 data words 0101 RFF set when 5 or more data words have been written to the Receive FIFO. Set when RxFIFO = 5,6.....15 data words 0110 RFF set when 6 or more data words have been written to the Receive.. Set when RxFIFO = 6,7.....15 data words 0111 RFF set when 7 or more data words have been written to the Receive FIFO. Set when RxFIFO = 7,8.....15 data words 1000 RFF set when 8 or more data words have been written to the Receive FIFO. Set when RxFIFO =8,9.....15 data words 1001 RFF set when 9 or more data words have been written to the Receive FIFO. Set when RxFIFO = 9,10.....15 data words

Table continues on the next page...

SSIx_SSI_SFCSR field descriptions (continued)

Field	Description
	1010 RFF set when 10 or more data words have been written to the Receive FIFO. Set when RxFIFO = 10,11.....15 data words 1011 RFF set when 11 or more data words have been written to the Receive FIFO. Set when RxFIFO = 11,12.....15 data words 1100 RFF set when 12 or more data words have been written to the Receive FIFO. Set when RxFIFO = 12,13.....15 data words 1101 RFF set when 13 or more data words have been written to the Receive FIFO. Set when RxFIFO = 13,14,15data words 1110 RFF set when 14 or more data words have been written to the Receive FIFO. Set when RxFIFO = 14,15 data words 1111 RFF set when 15 data words have been written to the Receive FIFO (default). Set when RxFIFO = 15 data words
19–16 TFWM1[3:0]	Transmit FIFO Empty WaterMark 1. These bits control the threshold at which the TFE1 flag will be set. The TFE1 flag is set whenever the empty slots in Tx FIFO exceed or are equal to the selected threshold. 0000 Reserved 0001 TFE set when there are more than or equal to 1 empty slots in Transmit FIFO (default). Transmit FIFO empty is set when TxFIFO <= 14 data. 0010 TFE set when there are more than or equal to 2 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <=13 data. 0011 TFE set when there are more than or equal to 3 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <=12 data. 0100 TFE set when there are more than or equal to 4 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <=11 data. 0101 TFE set when there are more than or equal to 5 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <=10 data. 0110 TFE set when there are more than or equal to 6 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <=9 data. 0111 TFE set when there are more than or equal to 7 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <=8 data. 1000 TFE set when there are more than or equal to 8 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <=7 data. 1001 TFE set when there are more than or equal to 9 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <= 6 data. 1010 TFE set when there are more than or equal to 10 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <= 5 data. 1011 TFE set when there are more than or equal to 11 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <= 4 data. 1100 TFE set when there are more than or equal to 12 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <= 3 data. 1101 TFE set when there are more than or equal to 13 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <= 2 data. 1110 TFE set when there are more than or equal to 14 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO <= 1 data. 1111 TFE set when there are 15 empty slots in Transmit FIFO. Transmit FIFO empty is set when TxFIFO = 0 data.
15–12 RFCNT0[3:0]	Receive FIFO Counter 0. These bits indicate the number of data words in Receive FIFO 0. Refer to RFCNT1[3:0] for details regarding settings for receive FIFO counter bits.
11–8 TFCNT0[3:0]	Transmit FIFO Counter 0. These bits indicate the number of data words in Transmit FIFO 0. Refer to TFCNT1[3:0] for details regarding settings for transmit FIFO counter bits.

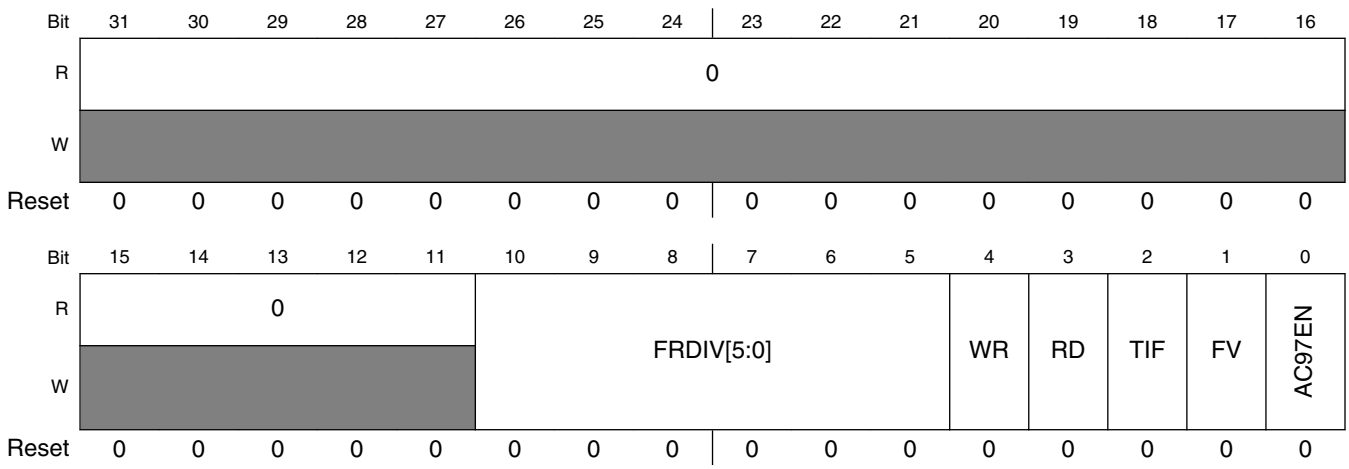
Table continues on the next page...

SSIx_SSI_SFCSR field descriptions (continued)

Field	Description
7–4 RFWM0[3:0]	Receive FIFO Full WaterMark 0. These bits control the threshold at which the RFF0 flag will be set. The RFF0 flag is set whenever the data level in Rx FIFO 0 reaches the selected threshold. Refer to RFWM1[3:0] for details regarding settings for receive FIFO watermark bits.
3–0 TFWM0[3:0]	Transmit FIFO Empty WaterMark 0. These bits control the threshold at which the TFE0 flag will be set. The TFE0 flag is set whenever the empty slots in Tx FIFO exceed or are equal to the selected threshold. Refer to TFWM1[3:0] for details regarding settings for transmit FIFO watermark bits.

73.8.11 SSI AC97 Control Register (SSIx_SSI_SACNT)

Addresses: SSI-2_SSI_SACNT is 5001_4000h base + 38h offset = 5001_4038h
SSI-1_SSI_SACNT is 63FC_C000h base + 38h offset = 63FC_C038h
SSI-3_SSI_SACNT is 63FE_8000h base + 38h offset = 63FE_8038h



SSIx_SSI_SACNT field descriptions

Field	Description
31–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10–5 FRDIV[5:0]	Frame Rate Divider. These bits control the frequency of AC97 data transmission/reception. They are programmed with the number of frames for which the SSI should be idle, after operating in one frame. Through these bits, AC97 frequency of operation, from 48 KHz (000000) to 1 KHz (101111) can be achieved. Sample Value: 001010 (10 Decimal) = SSI will operate once every 11 frames.
4 WR	Write Command. This bit specifies whether the next frame will carry an AC97 Write Command or not. The programmer should take care that only one of the bits (WR or RD) is set at a time. When this bit is set, the corresponding tag bits (corresponding to Command Address and Command Data slots of the next Tx frame) are automatically set. This bit is automatically cleared by the SSI after completing transmission of a frame. 0 Next frame will not have a Write Command. 1 Next frame will have a Write Command.

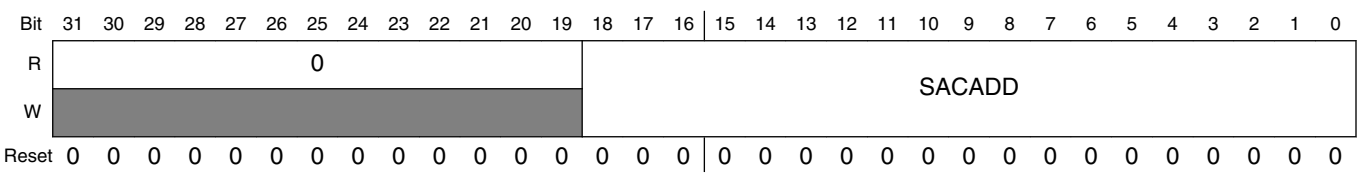
Table continues on the next page...

SSIx_SSI_SACNT field descriptions (continued)

Field	Description
3 RD	Read Command. This bit specifies whether the next frame will carry an AC97 Read Command or not. The programmer should take care that only one of the bits (WR or RD) is set at a time. When this bit is set, the corresponding tag bit (corresponding to Command Address slot of the next Tx frame) is automatically set. This bit is automatically cleared by the SSI after completing transmission of a frame. 0 Next frame will not have a Read Command. 1 Next frame will have a Read Command.
2 TIF	Tag in FIFO. This bit controls the destination of the information received in AC97 tag slot (Slot #0). 0 Tag info stored in SATAG register. 1 Tag info stored in Rx FIFO 0.
1 FV	Fixed/Variable Operation. This bit selects whether the SSI is in AC97 Fixed mode or AC97 Variable mode. 0 AC97 Fixed Mode. 1 AC97 Variable Mode.
0 AC97EN	AC97 Mode Enable. This bit is used to enable SSI AC97 operation. Refer to AC97 Mode for details of AC97 operation. 0 AC97 mode disabled. 1 SSI in AC97 mode.

73.8.12 SSI AC97 Command Address Register (SSIx_SSI_SACADD)

Addresses: SSI-2_SSI_SACADD is 5001_4000h base + 3Ch offset = 5001_403Ch
SSI-1_SSI_SACADD is 63FC_C000h base + 3Ch offset = 63FC_C03Ch
SSI-3_SSI_SACADD is 63FE_8000h base + 3Ch offset = 63FE_803Ch

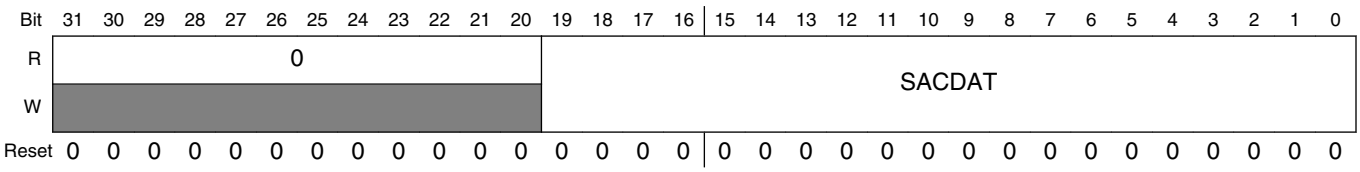


SSIx_SSI_SACADD field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero. Reserved
18–0 SACADD	AC97 Command Address. These bits store the Command Address Slot information (bit 19 of the slot is sent in accordance with the Read and Write Command bits in SSI_SACNT register). These bits can be updated by a direct write from the Core. They are also updated with the information received in the incoming Command Address Slot. If the contents of these bits change due to an update, the CMDAU bit in SISR is set.

73.8.13 SSI AC97 Command Data Register (SSIx_SSI_SACDAT)

Addresses: SSI-2_SSI_SACDAT is 5001_4000h base + 40h offset = 5001_4040h
SSI-1_SSI_SACDAT is 63FC_C000h base + 40h offset = 63FC_C040h
SSI-3_SSI_SACDAT is 63FE_8000h base + 40h offset = 63FE_8040h

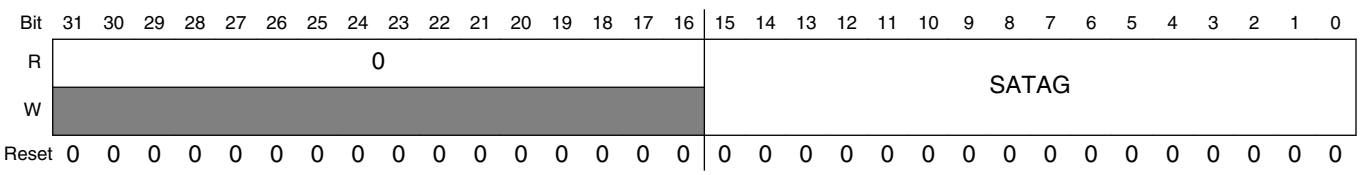


SSIx_SSI_SACDAT field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value zero. Reserved
19–0 SACDAT	AC97 Command Data. The outgoing Command Data Slot carries the information contained in these bits. These bits can be updated by a direct write from the Core. They are also updated with the information received in the incoming Command Data Slot. If the contents of these bits change due to an update, the CMDDU bit in SISR is set. These bits are transmitted only during AC97 Write Command. During AC97 Read Command, 0x00000 is transmitted in time slot #2.

73.8.14 SSI AC97 Tag Register (SSIx_SATAG)

Addresses: SSI-2_SATAG is 5001_4000h base + 44h offset = 5001_4044h
SSI-1_SATAG is 63FC_C000h base + 44h offset = 63FC_C044h
SSI-3_SATAG is 63FE_8000h base + 44h offset = 63FE_8044h



SSIx_SATAG field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–0 SATAG	AC97 Tag Value. Writing to this register (by the Core) sets the value of the Tx-Tag in AC97 fixed mode of operation. On a read, the Core gets the Rx-Tag Value received (in the last frame) from the Codec. If TIF bit in SSI_SACNT register is set, the TAG value is also stored in Rx-FIFO in addition to SATAG register. When the received Tag value changes, the RXT bit in SISR register is set. Bits SATAG[1:0] convey the Codec -ID. In current implementation only Primary Codecs are supported. Thus writing value 2'b00 to this field is mandatory.

73.8.15 SSI Transmit Time Slot Mask Register (SSIx_SSI_STMSK)

Addresses: SSI-2_SSI_STMSK is 5001_4000h base + 48h offset = 5001_4048h
SSI-1_SSI_STMSK is 63FC_C000h base + 48h offset = 63FC_C048h
SSI-3_SSI_STMSK is 63FE_8000h base + 48h offset = 63FE_8048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	STMSK																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SSIx_SSI_STMSK field descriptions

Field	Description
31–0 STMSK	Transmit Mask. These bits indicate which slot has been masked in the current frame. The Core can write to this register to control the time slots in which the SSI transmits data. Each bit has info corresponding to the respective time slot in the frame. Transmit mask bits should not be used in I2S Slave mode of operation. SSI_STMSK register value must be set before enabling Transmission. 0 Valid Time Slot. 1 Time Slot masked (no data transmitted in this time slot).

73.8.16 SSI Receive Time Slot Mask Register (SSIx_SSI_SRMSK)

Addresses: SSI-2_SSI_SRMSK is 5001_4000h base + 4Ch offset = 5001_404Ch
SSI-1_SSI_SRMSK is 63FC_C000h base + 4Ch offset = 63FC_C04Ch
SSI-3_SSI_SRMSK is 63FE_8000h base + 4Ch offset = 63FE_804Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SRMSK																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SSIx_SSI_SRMSK field descriptions

Field	Description
31–0 SRMSK	Receive Mask. These bits indicate which slot has been masked in the current frame. The Core can write to this register to control the time slots in which the SSI receives data. Each bit has info corresponding to the respective time slot in the frame. SSI_SRMSK register value must be set before enabling Receiver. Receive mask bits should not be used in I2S Slave mode of operation. 0 Valid Time Slot. 1 Time Slot masked (no data received in this time slot).

73.8.17 SSI AC97 Channel Status Register (SSIx_SSI_SACCST)

Addresses: SSI-2_SSI_SACCST is 5001_4000h base + 50h offset = 5001_4050h
SSI-1_SSI_SACCST is 63FC_C000h base + 50h offset = 63FC_C050h
SSI-3_SSI_SACCST is 63FE_8000h base + 50h offset = 63FE_8050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																						SACCST									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SSIx_SSI_SACCST field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–0 SACCST	AC97 Channel Status. These bits indicate which data slot has been enabled in AC97 variable mode operation. This register is updated in case the core enables/disables a channel through a write to SSI_SACCEN/SSI_SACCDIS register or the external codec enables a channel by sending a '1' in the corresponding SLOTREQ bit. Bit [0] corresponds to the first data slot in an AC97 frame (Slot #3) and Bit [9] corresponds to the tenth data slot (slot #12). The contents of this register only have relevance while the SSI is operating in AC97 variable mode. Writes to this register result in an error response on the block interface. 0 Data channel disabled. 1 Data channel enabled.

73.8.18 SSI AC97 Channel Enable Register (SSIx_SSI_SACCEN)

Addresses: SSI-2_SSI_SACCEN is 5001_4000h base + 54h offset = 5001_4054h
SSI-1_SSI_SACCEN is 63FC_C000h base + 54h offset = 63FC_C054h
SSI-3_SSI_SACCEN is 63FE_8000h base + 54h offset = 63FE_8054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W																							SACCEN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SSIx_SSI_SACCEN field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–0 SACCEN	AC97 Channel Enable. The Core writes a '1' to these bits to enable an AC97 data channel. Writing a '0' has no effect. Bit [0] corresponds to the first data slot in an AC97 frame (Slot #3) and Bit [9] corresponds

Table continues on the next page...

SSIx_SSI_SACCEN field descriptions (continued)

Field	Description
	to the tenth data slot (slot #12). Writes to these bits only have effect in the AC97 Variable mode of operation. These bits are always read as '0' by the Core. 0 Write Has no effect. 1 Write Enables the corresponding data channel.

73.8.19 SSI AC97 Channel Disable Register (SSIx_SSI_SACCDIS)

Addresses: SSI-2_SSI_SACCDIS is 5001_4000h base + 58h offset = 5001_4058h
SSI-1_SSI_SACCDIS is 63FC_C000h base + 58h offset = 63FC_C058h
SSI-3_SSI_SACCDIS is 63FE_8000h base + 58h offset = 63FE_8058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W																							SACCDIS								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SSIx_SSI_SACCDIS field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–0 SACCDIS	AC97 Channel Disable. The Core writes a '1' to these bits to disable an AC97 data channel. Writing a '0' has no effect. Bit [0] corresponds to the first data slot in an AC97 frame (Slot #3) and Bit [9] corresponds to the tenth data slot (slot #12). Writes to these bits only have effect in the AC97 Variable mode of operation. These bits are always read as '0' by the Core. 0 Write Has no effect. 1 Write Disables the corresponding data channel.

Chapter 74
Television Encoder (TVEv2)

74.1 Introduction

The Television Encoder (TVE) is designed to provide direct connection between an Application Processor, or ARM platform, and a TV set via analog interfaces. The ARM platform may be both a standalone product and a part of a Wireless Baseband Platform (WBP). Integration of the TVE into the ARM platform and the WBP allows minimizing system complexity and cost. The TV Encoder supports different Standard Definition (SD) and High Definition (HD) television standards.

The block is based on mixed (digital and analog) signal processing. It includes three main components:

- TV Signal Processor (TVSP) is a digital block which is responsible for modulation, filtering, generation synchronization signals, upsampling, cable detection control, copy protection and other processing tasks.
- Triple Video Digital-to-Analog Converter (TVDAC) is an analog block which is designed to convert up to three digital video signals to a current.
- Cable Detection Circuit (CDC) is an analog block which provides monitoring of TVDAC output signals for Cable Detection (CD).

The TVE can operate in two modes:

1. Encoding the video data according to the selected TV standard
2. Generating RGB analog signals according to the VGA specification.

The main features of the TVE in TV encoding mode are:

1. SD mode features:
    - Supported TV standards:
        - NTSC
        - PAL B,D,G,H, I/M/N
        - 480i
        - 576i

- Supported output formats:
  - Composite video (CVBS) - up to two simultaneous identical outputs are supported
  - S-video (Y/C), composite video and S-video signals may be output simultaneously
  - YPrPb component
  - RGB component
- Programmable notch/pre-comb filter for CVBS
- Switchable pedestal
- Copy Generation Management System (CGMS) support according to
  - EIA-608b
  - IEC 61880-1
  - EIA-J CPR-1204-1
- Wide-Screen Signaling (WSS) support according to ITU-R BT.1119, ETSI EN 300 294
- Closed Caption insertion capability according to EIA-608b
- Macrovision™ 7.1 copy protection
- Output oversampling up to x16 for elimination of external analog filters

2. HD mode features:
   - Supported TV standards:
     - 720p 60Hz
     - 720p 50Hz
     - 720p 30Hz
     - 720p 25Hz
     - 720p 24Hz
     - 1080i 60Hz
     - 1080i 50Hz
     - 1035i 60Hz (1920x1035)
     - 1080p 30Hz
     - 1080p 30Hz25Hz
     - 1080p 25Hz24Hz
     - 1080p 50Hz
     - 1080p 60Hz
   - Supported output formats:
     - YPrPb component
     - RGB component
   - CGMS support according to
     - EIAJ CPR-1204-2 (CGMS)
     - CEA-805-A (CGMS type B)
   - Output oversampling up to x4 for elimination of external analog filters
3. Common SD/HD mode features

- Flexible timing and gain control mechanism allowing non-standard parameters ("user mode")
- Programmable Chroma digital filters
- Programmable adaptive Luma digital filters for
  - horizontal and vertical sharpening
  - horizontal and vertical noise reduction
  - de-flickering in interlaced modes
- Programmable YCrCb to RGB color matrix
- Output resolution - 10 bits (single-ended analog output loaded by a double terminated 75-Ohm cable)
- Cable Detection functionality including
  - Separate cable connection monitoring separately for each TVDAC channel both during normal operation and in standby mode
  - Automatic TVE power on when a TV cable has been connected
  - Detection of connection type (composite, S-video, component)
  - Detection of cable short to the ground

In VGA mode, the TVE provides transfer of RGB video data from the ARM platform to a monitor. The Interface supports RGBHV video data format with three analog color signals (RGB) and separate horizontal (HSYNC) and vertical (VSYNC) synchronization signals. The standard VGA DE-15 connector includes signal lines listed in Table 74-1.

Table 74-1. VGA DE-15 Connector Signals

##	Name	Description
Pin 1	**RED**	**Red video**
Pin 2	GREEN	Green video
Pin 3	BLUE	Blue video
Pin 4	N/C	Not connected
Pin 5	GND	Ground (HSync)
Pin 6	RED_RTN	Red return
Pin 7	GREEN_RTN	Green return
Pin 8	BLUE_RTN	Blue return
Pin 9	+5 V	+5 V DC
Pin 10	GND	Ground (VSync, DDC)
Pin 11	N/C	Not connected
Pin 12	SDA	I²C data
Pin 13	HSync	Horizontal sync
Pin 14	VSync	Vertical sync
Pin 15	SCL	I²C clock

Figure 74-1 depicts the TVE integration scheme into the ARM platfor

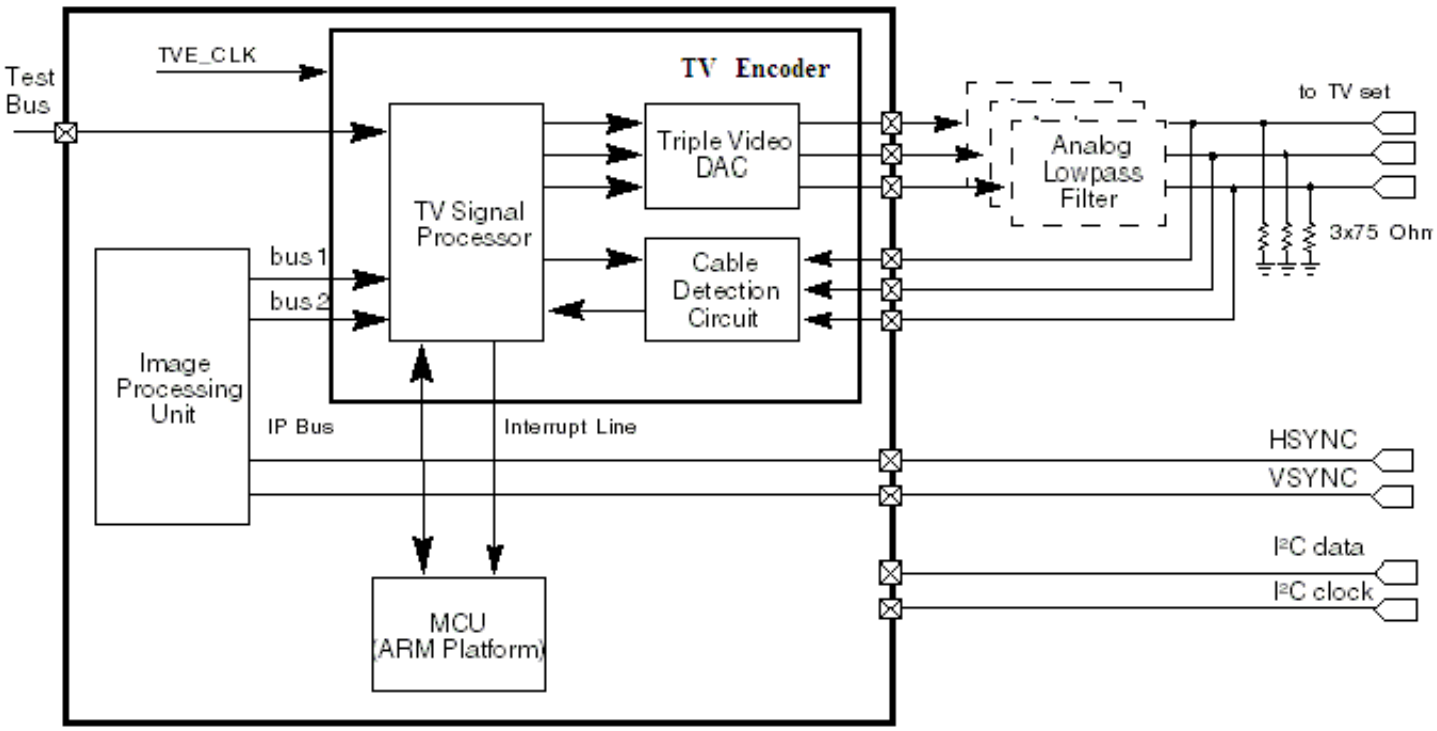


Figure 74-1. TVE Integration in an Application Processor

The TVE receives a 30-bit video data stream either via the VIDEO_DATA_IN_1 bus or via the VIDEO_DATA_IN_2 bus from the Image Processing Unit (IPU). The VIDEO_DATA_IN_1 bus is used for data transfer when the TVE operates in TV encoding mode or in test mode with the IPU driving the test data. The VIDEO_DATA_IN_2 bus is used in VGA mode.

In VGA mode, the IPU transfers data in RGB format (8 bits/component). The 30-bit VIDEO_DATA_IN_2 bus is connected to the 24-bit IPU data output bus as shown in Table 74-2.

Table 74-2. Connection between the IPU Data Output Bus and the VIDEO_DATA_IN_2 Bus

VIDEO_DATA_IN_2 bus pins	Connection	Description
0	ground	
8:1	IPU_DATA_OUT[7:0]	G component
9	ground	
10	ground	
18:11	IPU_DATA_OUT[15:8]	R component
19	ground	
20	ground	

Table continues on the next page...

Table 74-2. Connection between the IPU Data Output Bus and the VIDEO_DATA_IN_2 Bus (continued)

VIDEO_DATA_IN_2 bus pins	Connection	Description
28:21	IPU_DATA_OUT[23:16]	B component
29	ground	

The IPU provides also the sampling clocks VIDEO_DATA_CLK_1 and VIDEO_DATA_CLK_2 for each of the data buses. The clock frequency depends on the mode.

In SD mode, the TVE output is always in the interlaced format. For some TVE configuration, the TVE performs inter-field vertical filtering for de-flickering, fine sharpening and high-frequency noise reduction. In this case, the input data arrives in the progressive format at the doubled frequency Fsx2 = 27 MHz where Fs = 13.5 MHz is the pixel frequency for the interlaced format. In HD mode the input data rate is always Fs = 74.25 MHz or Fs = 74.25/1.001 = 74.1758 MHz. The Y component is sampled at the VIDEO_DATA_CLK_1(2) frequency. For the YCbCr 4:2:2 input format, the Cb/Cr components are sampled at a half of the VIDEO_DATA_CLK_1(2) frequency. For the YCbCr 4:4:4 input format, the Cb/Cr components are sampled at the VIDEO_DATA_CLK_1(2) frequency.

The VIDEO_DATA_CLK_2 clock frequency depends on current resolution of the VGA output. For more details see Operation in VGA Mode-Introduction.

The data stream via the VIDEO_DATA_IN_1 is also accompanied by two separated signals for horizontal synchronization (HSYNC) and vertical synchronization (VSYNC) and by data validness indicator VIDEO_DATA_EN. The existing HSYNC_2, VSYNC_2 and VIDEO_DATA_EN_2 pins are unused for VGA mode.

The second alternative for the TVE input source is the test bus driven by an external device through the chip pins. The test bus can be used for both in functional mode testing or in TVDAC testing. In both cases, the test bus replaces the IPU output.

The TVE generates a TV signal in one of the formats specified above in TV encoding mode. The signal is oversampled before digital-to-analog conversion. The oversampling factor is programmable and can vary from 2x to 16x in SD mode and from x2 to x4 in HD mode. In primary operation mode with maximal output oversampling, the TVDAC sampling frequency is 216 MHz for SD and 297 MHz or 297/1.001 MHz for HD. For SD, there are also 2 additional modes when the TVDAC sampling frequency is 108 or 54 MHz. For HD, there is one additional mode with the TVDAC sampling frequency of 148.5 MHz or 148.5/1.001 MHz. These modes should be used as a reserved option in the case if the performance of the used TVDAC degrades at high sampling frequency.

In VGA mode, the TVE can provide oversampling by 2 or 4. If the pixel rate is less than 75 Mpix/sec, x4 oversampling should be used. For faster pixel rates, x2 oversampling should be used.

The TVDAC is able to drive directly a 75-Ohm double terminated cable (the equivalent load is 37.5 Ohm). External Analog Lowpass Filters are optional. They are recommended in VGA mode with pixel rates faster than 75Mpix/sec when no oversampling can be done. The cutoff frequency of the filters should be 70 MHz.

The TVE is sampled by the TVE_CLK clock. The clock frequency is equal to the TVDAC sampling rate. All required internal clocks are produced by division of the TVE_CLK inside the TVE.

An output voltage of each TVE channel is monitored by Cable Detection (CD) system. The CD is able to distinguish between the following cases:

- Normal load impedance which is equal to 37.5 Ohm when the TVE drives directly a 75-Ohm double terminated cable
- Double load impedance (75 Ohm) when the cable is disconnected
- Zero load impedance when the output is shorted to the ground

The CD monitors each of the output channels separately. The monitoring result is sent to the ARM platform which decides on the TV output mode corresponding to the connected cable type.

The TVE is controlled by the ARM platform via the IP Bus interface. The TVE generates interrupts for the ARM platform via a singe interrupt line corresponding to the Indigo IP Interface.

74.1.1 Overview

This paragraph describes TVE architecture and functionality with focus on TV encoding mode. For more details on operation in VGA mode, see Operation in VGA Mode-Introduction.

The TVE block diagram is shown in Figure 74-2. The TVE consists of three sub-blocks: the TV Signal Processor (TVSP), the Triple Video DAC (TVDAC) and the Cable Detection Circuit (CDC).

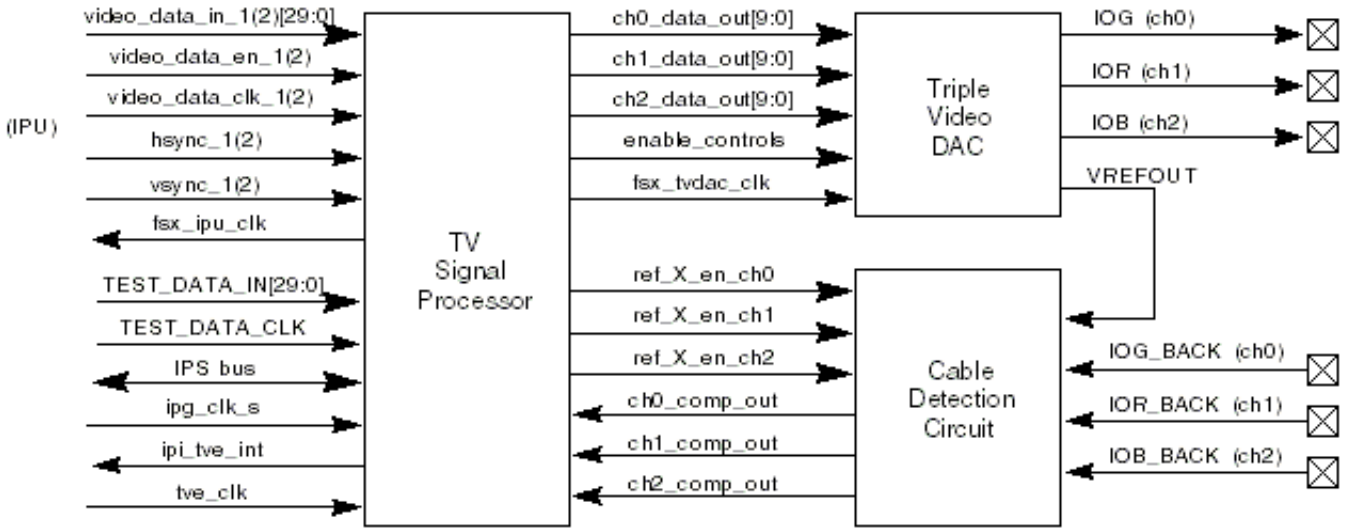


Figure 74-2. TVE Block Diagram

The TVDAC converts an input 10-bit code in the unsigned format to the current. The DAC is able to drive directly a double terminated 75-Ohm cable. The full scale output current is controlled by an external resistor to provide the required full-scale voltage of the video signal on the output load. The nominal sampling rate of the TVDAC is 216 MHz in SD mode and 297 MHz or 297/1.001 MHz in HD mode.

The CDC is an analog block which includes a reference voltage divider and three comparators (one for each TVDAC output). The reference voltage for the CDC is taken from the TVDAC VREFOUT pin. The CDC compares the TVDAC output voltage with the different reference levels derived from the VREFOUT voltage. The result is sent to the TVSP which decides whether a cable is connected.

The TVSP is a digital part of the TVE. The TVSP block diagram is shown in Figure 74-3.

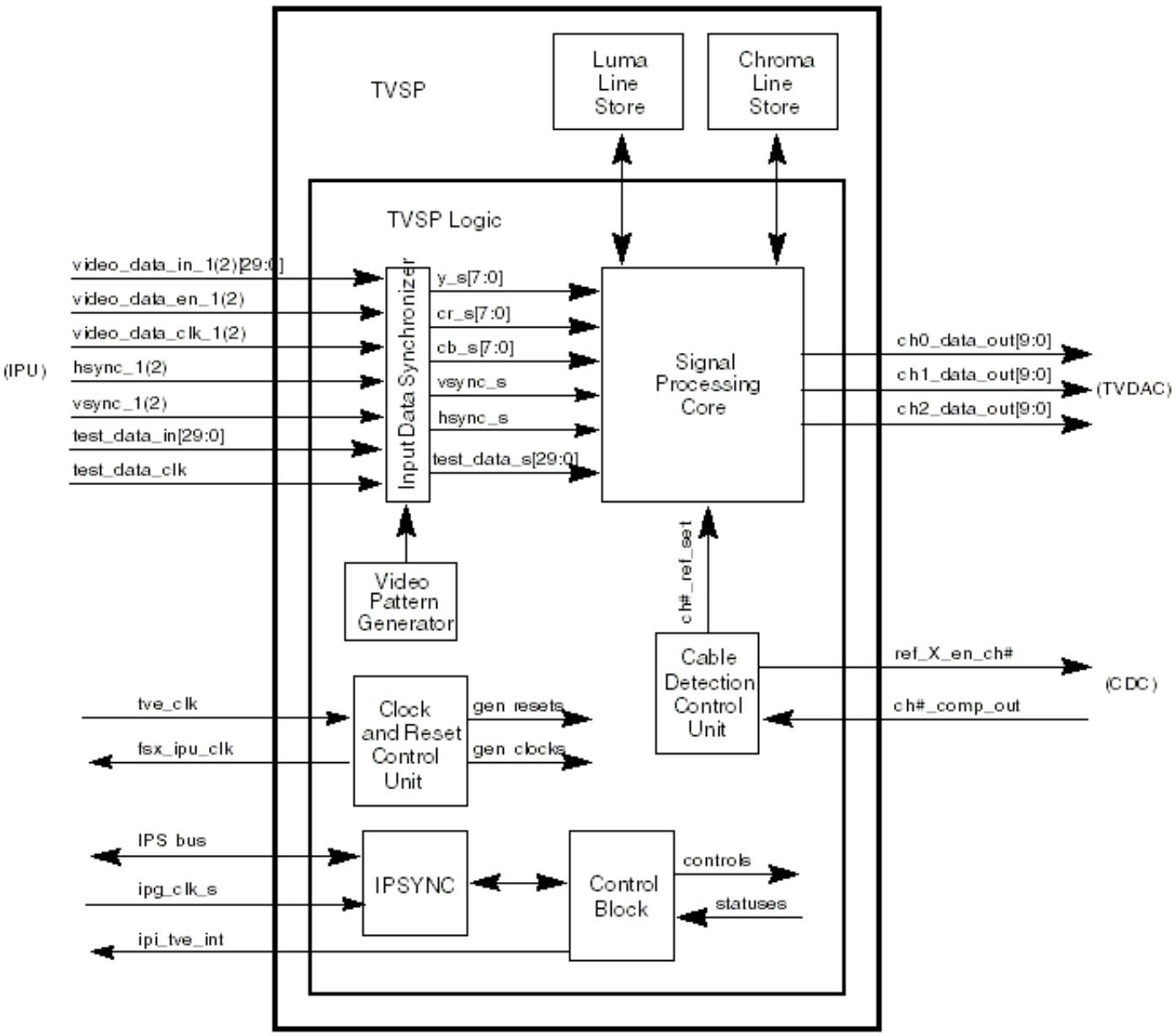


Figure 74-3. TVSP Block Diagram

The TVSP includes two parts: the memory-less TVSP Logic and two memories - the Luma Line Store and the Chroma Line Store.

The central sub-block of the TVSP is based on the Signal Processing Core (SPC) which is responsible for the following TV encoding tasks:

- Synchronization signals generation according to the used TV standard
- Digital non-adaptive and adaptive filtering of video signals
- Sampling rate conversion (upsampling)

- Color conversion YCrCb to RGB and color adjustment
- Color modulation in composite and S-video output modes required by the supported SD standards
- VBI data insertion
- Macrovision^{TM} signals generation for copy protection in SD mode

The SPC receives the input data from the Input Data Synchronizer (IDS) in the YCrCb 4:2:2 or the YCrCb 4:4:4 formats. The IDS is designed to synchronize the IPU output data to the TVE clock.

According to the IPU type used, two synchronization modes are possible:

1. If the IPU supports output data synchronization to an external clock, the TVE should provide the FSx_IPU_CLK clock (27 MHz in SD mode and 74.25 MHz or 74.25/1.001 MHz in HD mode) to the IPU. The IPU uses this clock to generate the output data stream. The IDS is transparent in this case. In functional mode, the sampling rate is 13.5 MHz in SD mode and 74.25 MHz or 74.25/1.001 MHz in HD mode (for the YCrCb 4:2:2 format, the Cb and Cr samples are repeated twice). In TVDAC test mode, the data sampling rate is correspondingly 27 MHz and 74.25 MHz or 74.25/1.001 MHz.
2. If the IPU does not support output data synchronization to an external clock, it should provide the sampling rates mentioned above (for example, to generate the display clock using fractional frequency division). The frequencies of the IPU output and TVE input clocks must be equal, but a phase jitter between two clocks is allowed. The maximal value of this jitter should be less than one period of one pixel clock. Also initial phase difference between two clocks may be unspecified. The IDS is designed to eliminate phase uncertainty and phase jitter between two clocks.

The IDS receives a 30-bit input data from the IPU. In functional mode, the least significant byte of the data bus include the Y component, next two bytes are the Cr and Cb components correspondingly. Six most significant bits are unused in functional mode. In TVDAC test mode 1, each 10 bits are fed to the corresponding TVDAC channel. In TVDAC test mode 2, only 10 least significant bits are fed to all TVDAC channels and 20 most significant bits are unused.

In addition to the data, the IDS resamples the vertical (VSYNC_1(2)) and horizontal (HSYNC_1(2)) synchronization signals from the IPU. It also gates off the invalid IPU output data according to the VIDEO_DATA_EN_1(2) data enable signal from the IPU.

The SPC has three output channels. Each channel is represented by 10 bits. The ouputs can be in composite (CVBS), S-video, YCrCb and RGB component formats according to the TV_OUT_MODE control parameter (see Table 74-3).

Table 74-3. TVE Output Data Formats

Output channel	Output data format (TV_OUT_MODE)							
	Composite (CVBS)				S-video (Y/C)	CVBS and Y/C	YCrCb component	RGB component
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Channel #0	0	CVBS	0	CVBS	Y	Y	Y	G
Channel #1	0	0	0	0	C	C	Cr	R
Channel #2	0	0	CVBS	CVBS	0	CVBS	Cb	B

The Cable Detection Control Unit (CDCU) is a digital part of the CD system. It sends periodical requests to the CDC to monitor TVDAC output voltage. The CDC returns the results of load resistance measurement and ground short detection for each of monitored channels. For TVE channels those are inactive for the used TV_OUT_MODE setting (zeros in Table 74-3), the CDCU forces the output current to the value required for the measurement.

The Clock and Reset Control Unit (CRCU) produces internal generated clocks by division of the TVE_CLK clock. The CRCU asserts also reset signals to the sub-blocks which are unused in the current configuration and enable signals for the TVDAC channels.

The Control Module (CM) is responsible for IP Bus interface and interrupt generation support. It contains a set of control and status registers. Because the CM is sampled by the 27-MHz clock in SD mode and by the 74.25-MHz or 74.25/1.001-MHz clock in HD mode, the IP Synchronizer (IPSYNC) is used to synchronize the frequency of the IP Bus interface (typically, 66 MHz) to the CM clock frequency.

The SPC block diagram is depicted in Figure 74-4.

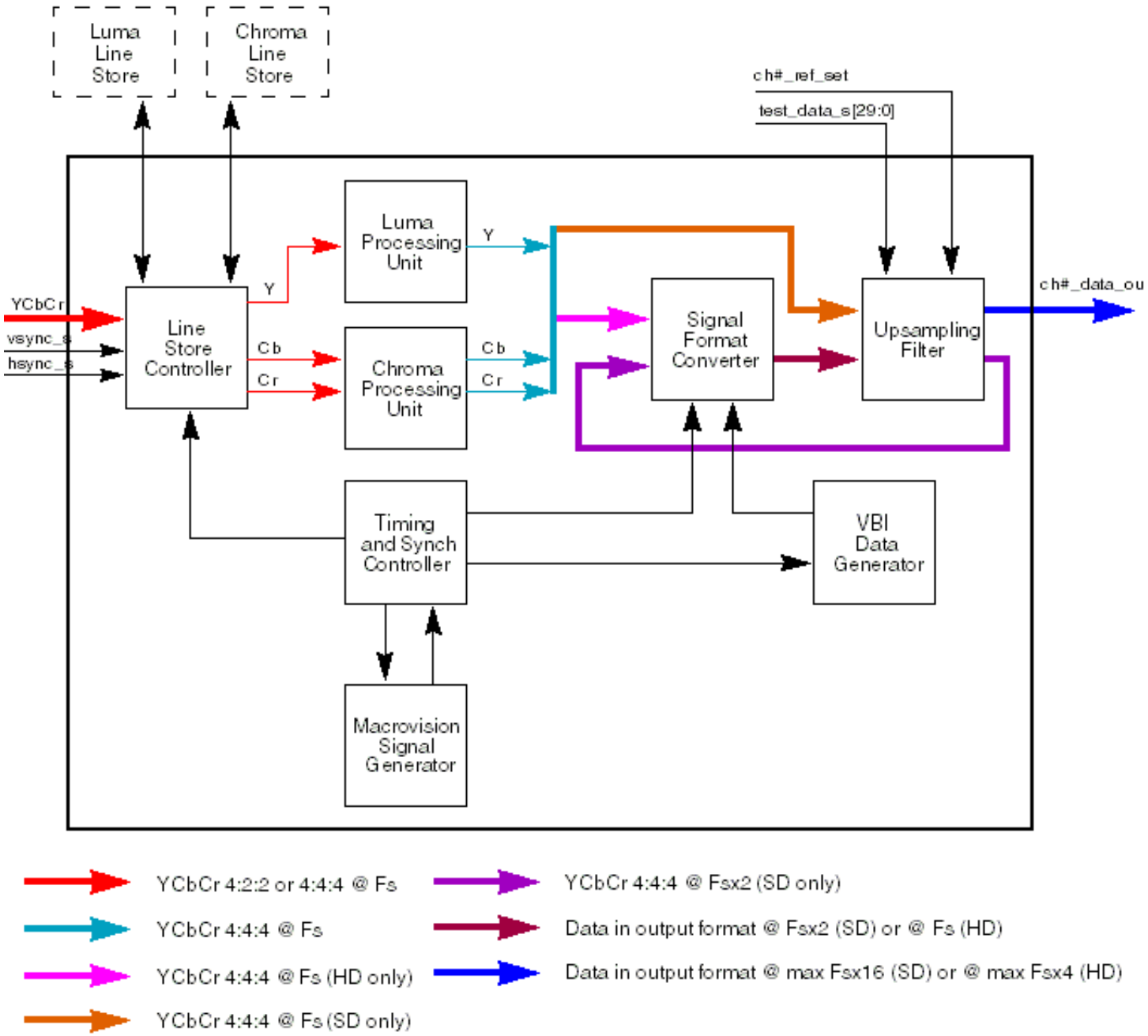


Figure 74-4. SPC Block Diagram

The SPC includes the Line Store Controller (LSC), The Luma Processing Unit (LPU), the Chroma Processing Unit, the Signal Format Converter (SFC), the Upsampling Filter (UF), The Timing and Synchronization Controller (TSC), the Macrovision Signal Generator and the VBI Data Generator (VDG).

The LSC receives the data in the YCbCr 4:0:0 or 4:4:4 format from the IDS. The data is written to the LLS (Y) and the CLS (Cb/Cr) line-by-line. The LLS size is 5 lines of 1920 Y samples. The CLS size is 4 lines of 1920 Cb/Cr pairs. The LSC reads 5 sample

columns of the Y component from the LLS and sends it to the LPU. In parallel, the LSC reads 4-sample columns of the Cb/Cr component from the CLS and sends it to the chroma processing unit.The columns are read sequentially so for generation of one output line, the whole line of pixel columns is read from the line stores.

The LPU contains a set of filters including:

- 5-tap vertical adaptive filter designed for
  - vertical coarse and fine sharpening
  - vertical mid-frequency and high-frequency noise reduction
  - de-flickering (required only for interlaced TV formats)
- 5-tap horizontal adaptive filter designed for
  - horizontal coarse and fine sharpening
  - horizontal mid-frequency and high-frequency noise reduction
  - horizontal de-ringing
- Notch filter for better Luma/Chroma separation in SD composite mode
- Lowpass filter for reducing out-of-band noise (it may enabled instead of the notch filter because both filters are implemented by the same hardware).

The adaptive filters boost (for sharpening) or suppress (for noise reduction, de-ringing and de-flickering) mid and high frequency components in the Luma spectrum. The filter operation (boost or suppression) depends on the Luma gradient in the corresponding spatial direction. The filters behavior is fully programmable. If the filter for vertical fine sharpening /vertical high-frequency noise reduction/de-flickering is enabled, the input data from the IPU should arrive in the progressive format.

The chroma processing unit performs the following operation on the Chroma samples from the CLS:

- Upsampling from Fs/2 to Fs (if the input data format is YCbCr 4:2:2)
- Lowpass filtering for reducing color noise and improving Luma/Chroma separation in SD composite mode
- Comb filtering for improving Luma/Chroma separation in SD composite mode.

The chroma processing unit produces the filtered Cb and Cr components sampled at Fs.

The following processing path of the LPU and chroma processing unit output samples depends on the mode. In SD mode, the samples are fed to the UF. The UF performs upsampling the data to Fsx2 = 27 MHz. The upsampled data enters the SFC. In HD mode, the LPU and chroma processing unit outputs sampled at Fs = 74.25 or 74.25/1.001 MHz are sent directly to the SFC.

The SFC is responsible for the following functions:

- Conversion the YCbCr data to the RGB format for RGB component output

- Subcarrier generation and Chroma modulation according to NTSC and PAL standards in SD composite and S-video modes
- Adjustment of brightness, saturation and hue (in user mode)
- Signal level normalization to the desired values
- Combining the synchronization signals and the VBI data with the video signal
- Composing three output streams according to Table 74-3.

Signal formats and waveforms on three SFC outputs correspond to the final TV signals. The UF provides upsampling of these SFC outputs. The UF consists of three identical channels. Each channel includes the following elements:

- The Gain and Offset Correction (GOC) block is designed to compensate the TVDAC gain and offset deviation from the nominal values. The gain and the offset are adjusted in the range +/-25% and +/-12.5% of the full scale separately for each channel. The TVDAC_#_GAIN and TVDAC_#_OFFSET parameters are responsible for gain and offset control. The GOC includes an input mux used for selection between the functional mode data, the test data and the CD reference levels.
- The Drop Correction Filter (DCF) provides compensation of the gain drop introduced by the TVDAC at high frequencies. The drop compensation factor is programmable via the TVDAC_#_DROP_COMP parameter separately for each channel.
- The Halfband Upsampling Filter 1 (HUF1) performs upsampling from Fs to Fsx2 (27 MHz in SD mode and 148.5 MHz or 148.5/1.001 MHz in HD mode). In SD mode, the HUF1 input is muxed to the LPU and chroma processing unit outputs and the HUF1 output is muxed to the SFC input, so the first upsampling is done before processing in the SFC.
- The Halfband Upsampling Filter 2 (HUF2) performs upsampling from Fsx2 to Fsx4 (54 MHz in SD mode and 297 MHz or 297/1.001 MHz in HD mode). The HUF2 may be bypassed in HD mode if the output sampling frequency is set to Fsx2 via the TVDAC_SAMP_RATE control parameter.
- The 3rd-order Interpolation Comb Filter (ICF) performs upsampling from Fsx4 to Fsx8 or Fsx16 in SD mode. The upsampling factor is controlled by the TVDAC_SAMP_RATE parameter. The ICF may be bypassed also in SD mode if the TVDAC_SAMP_RATE sets the output sampling frequency to 54 MHz.

In test mode, a test data is fed directly to the TVDAC (without upsampling) or through the UF.

The TSC is responsible for the following functions:

- Maintenance vertical and horizontal timing specified by the TV standards
- Generation of the synchronization signals, the color burst envelope, the active video data enable signal, the pedestal
- Control of the blanking level for each of three output channels

- Insertion of the Macrovision signal distortions
- Edge shaping for all generated signals
- Delays between the synchronization signals

The TSC is triggered by the VSYNC and HSYNC signals from the IPU. The TSC provides flexible control on horizontal timings in user mode than allows non-standard timing modes in addition to the standard configurations (including square pixel operation in SD mode). The MSG controls insertion of the Macrovision signal distortions in the TSC. The MSG gets all necessary triggers from the TSC and returns Macrovision controls to the TSC.

The VDG generates the following VBI control sequences:

- Closed Caption
- CGMS
- WSS

The VDG can be configured only to one of them at the moment because a common hardware is used for generation of all these sequences.

74.1.2 Operation in VGA Mode-Introduction

In VGA mode, the most of the processing is bypassed. The data is in the RGB888 format. It is synchronized by the IDS to the internal pixel clock. The IDS output is fed directly to the UF. To provide such data flow, the TVE should be configured to test mode with upsampling and the selected TV standard should be one of HD standards. The UF provides gain correction and upsampling by 2 or 4 depending on required display resolution. The gain should be 548/511=1.072 to provide 0.7V peak-to-peak amplitude on the TVDAC output. Required clock frequencies for some VGA modes are shown in Table 74-4.

Table 74-4. Some VGA Modes and Clock Frequencies

Display resolution	Frame rate, Hz	Pixel clock frequency, MHz	TVE_CLK frequency, MHz	Upsampling factor	TVDAC sampling frequency, MHz
VGA (640x480)	60	25.175	100.7	4	100.7
VGA (640x480)	75	31.5	126	4	126
SVGA (800x600)	60	40	160	4	160
SVGA (800x600)	75	49.5	198	4	198
XGA (1024x768)	60	65	260	4	260
XGA (1024x768)	75	78.8	157.6	2	157.6
WXGA (1368x768)	60	85.86	171.72	2	171.72

Table continues on the next page...

Table 74-4. Some VGA Modes and Clock Frequencies (continued)

Display resolution	Frame rate, Hz	Pixel clock frequency, MHz	TVE_CLK frequency, MHz	Upsampling factor	TVDAC sampling frequency, MHz
SXGA (1280x1024)	60	108	216	2	216
SXGA (1280x1024)	75	135	270	2	270
WSXGA (1440x900)	60	106.47	212.94	2	212.94
WSXGA+ (1680x1050)	60	147.14	294.28	2	294.28

74.1.3 Features

Common TVE features in both SD and HD modes are shown in Table 74-5.

Table 74-5. Common TVE Features in SD and HD Modes

Parameter	Value	Comments	TVEv1
Input data format	YCrCb 4:2:2 YCrCb 4:4:4	The IPU should support 24-bit/pixel parallel data output format	YCrCb 4:2:2
Input interface format and synchronization	Parallel interface with separate HSYNC and VSYNC	Two inputs from the IPU are available to allow transferring video data from one of two IPU display interfaces	Single input from the IPU
Input data resolution	8 bits/component		Same
Output data resolution	10 bits	-	Same
Output interface format	Parallel	-	Same
Non-standard horizontal timings support	Yes	Supported in user mode	No
Programmable adaptive Luma digital filter	Yes. The 5x5 adaptive filter is used for • horizontal and vertical sharpening • horizontal and vertical noise reduction • de-flickering in interlaced modes	The filter is fully programmable. The filter algorithm can be altered in user mode.	No
Programmable YCrCb to RGB color matrix	Yes	Five matrix coefficients are programmable. The can be also used for brightness, saturation and hue adjustment in user mode.	Fixed color matrix

Table continues on the next page...

Table 74-5. Common TVE Features in SD and HD Modes (continued)

Parameter	Value	Comments	TVEv1
Cable Detection functionality	Supported. It includes • separate cable connection monitoring separately for each TVDAC channel both during normal operation and in standby mode • automatic TVE power on when a TV cable has been connected • detection of connection type (composite, S-video, component) • detection of cable short to the ground	-	Same

Main parameters of the TVE in SD mode are given in Table 74-6.

Table 74-6. TVE Features in SD Mode

Parameter	Value	Comments	TVEv1
Supported color standards	PAL (B,D,G,H, I/M/N), NTSC	-	Same plus SECAM
Supported TV standards	480i (29.97Hz), 576i (25Hz)	-	Same
Input frame format	Interlaced if no inter-field filtering is performed. Progressive otherwise.	Inter-field filtering is used for vertical fine sharpening, vertical high-frequency noise reduction and de-flickering	Interlaced only
Input sampling frequency	If no inter-field filtering is performed: Y - 13.5 MHz Cb/Cr - 6.75 MHz for YCbCr 4:2:2 format, Cb/Cr - 13.5 MHz for YCbCr 4:4:4 format. If inter-field filtering is performed: Y - 27 MHz Cb/Cr - 13.5 MHz for YCbCr 4:2:2 format, Cb/Cr - 27 MHz for YCbCr 4:4:4 format.	-	Y - 13.5 MHz Cb/Cr - 6.75 MHz
Square pixel operation	Yes	Also other non-standard timings can be supported in user mode	No
Output data format	Composite (CVBS) or S-video (Y/C) or component (YPrPb or RGB)	Simultaneous output of two identical CVBS as well as CVBS and Y/C is possible	Same

Table continues on the next page...

Table 74-6. TVE Features in SD Mode (continued)

Parameter	Value	Comments	TVEv1
Output sampling frequency	216/108/54 MHz	Lower sampling frequencies should be used if the TVDAC performance degrades at high sampling frequencies	Same plus 27 MHz
Macrovision support	Yes. According to Macrovision 7.1	-	Same
Programmable chroma bandwidth	Yes. The possible bandwidth values are: • for YCbCr 4:2:2 input - 0.6, 1.3, 2 and 3 MHz • for YCbCr 4:4:4 input - 1.3, 2.7, 4 and 5.4 MHz	-	For composite and S-video output formats - 0.7, 1.0, 1.7 MHz For component output format - 1.8 MHz
Switchable pedestal	Yes	-	Same
Copy generation management system (CGMS) support	Yes. According to EIA-608b, IEC 61880-1, EIA-J CPR-1204-1	-	Same
Closed Caption support	Yes. According to EIA-608b	-	Same
Wide Screen Signaling (WSS) support	Yes. According to ITU-R BT.1119, ETSI EN 300 294.	-	Same
Luma notch filter	Yes	For better Luma/Chroma separation in composite mode	No
Chroma comb filter	Yes	For better Luma/Chroma separation in composite mode	No

Main parameters of the TVE in HD mode are given in Table 74-7.

Table 74-7. TVE Features in HD Mode

Parameter	Value	Comments
Supported HD TV standards	720p 60Hz 720p 50Hz 720p 30Hz 720p 25Hz 720p 24Hz 1080i 60Hz 1080i 50Hz 1035i 60Hz (1920x1035) 1080p 30Hz 1080p 25Hz 1080p 24Hz 1080p 50Hz 1080p60Hz	-
Input frame format	Interlaced for 1080i. Progressive otherwise.	No inter-field filtering is available for 1080i
Input sampling frequency	74.25 MHz -for 25/30/60Hz modes 74.176 MHz - for 29.97/59.94Hz modes	-
Output data format	Component (YCrCb or RGB)	-
Output sampling frequency	297 MHz -for 25/30/60Hz modes 296.704 MHz - for 29.97/59.94Hz modes	-
Programmable chroma bandwidth	Yes. The possible bandwidth values are: • for YCbCr 4:2:2 input - 3.3, 7.1, 11 and 16.5 MHz • for YCbCr 4:4:4 input - 6.7, 14.3, 22 and 29.7 MHz	-
Copy generation management system (CGMS) support	Yes. According to EIAJ CPR-1204-2 (CGMS) and CEA-805-A (CGMS type B)	-

Main parameters of the VGA mode are given in Table 74-8.

Table 74-8. TVE Features in VGA Mode

Parameter	Value	Comments
Input data format	RGB	The IPU should support 24-bit/ pixel parallel data output format
Input interface format and synchronization	Parallel interface	The TVE accepts video data only. The HSYNC and VSYNC are fed to the display directly from the IPU
Input color component resolution	8 bits/component	
Maximal screen resolution	Up to WSXGA+ (1680x1050 pixels)	
Output video signal format	RGB analog signals with separate HSYNC and VSYNC	

Table continues on the next page...

Table 74-8. TVE Features in VGA Mode (continued)

Parameter	Value	Comments
Output color component resolution	10 bits	
Output analog levels	0V - min., 0.7V max	
Maximal output signal bandwidth	Up to 150 MHz	
Maximal output sampling rate	Up to 300 MHz	
Output oversampling	2 or 4	

Main parameters of the TVDAC are given in Table 74-9.

Table 74-9. TVDAC Features

Parameter	Value	Comments
Video signal formats	Composite (CVBS) or S-Video (Y/C) or component (YPrPb or RGB)	-
Resolution	10 bits	-
Signal bandwidth	From 0 to 30 MHz	-
Sampling rate	From 54 to 300 MHz	-
Digital input	Parallel 10x3 bits in unsigned format	-
Output type	Single-end current output	-
Output drive characteristics	Full drive - double-terminated 75 Ohm load (equivalent to 37.5 Ohm) Low drive - 100 Ohm load	Drive with standard output video levels
Output Drive Range	Controllable full-scale current	Controlled by an external resistor
Channel enable control	Separate for each channel and common control enabling the voltage reference	-
Voltage reference	Build-in bandgap reference	-
Maximal gain mismatch	2%	Between any DAC
Maximal Integral Nonlinearity (INL)	+/-2 LSB	-
Maximal Differential Nonlinearity (DNL)	+/-1 LSB	-
Spurious-free Dynamic Range	> 50 dBc @ F=15 MHz, Fs=300 MHz	-
Power Supplies	Analog - 2.5-2.75 V, Digital - 1.2 V	-

74.1.4 Modes of Operation

74.1.4.1 Operation in TV Mode

In Normal Operation Mode, the TVE converts the input YCrCb 4:2:2 or 4:4:4 video stream to an analog TV signal. The TVE is in Normal Operation Mode, if the TV_OUT_MODE control parameter is in the range from 1 to 7 (see Table 74-3). This parameter also specifies the output TV signal type. During Normal Operation Mode, inactive TVDAC and UF channels, which outputs should be 0 according to Table 74-3, are disabled to save power.

The TV standard supported is specified by the TV_STAND control parameter as described in Table 74-10.

Table 74-10. TV_STAND Parameter

TV_STAND	Standard
0000	SD NTSC
0001	SD PALM
0010	SD Combination PALN
0011	SD PAL (B,D,G,H,I)
0100	HD 720p60
0101	HD 720p50
0110	HD 720p30
0111	HD 720p25
1000	HD 720p24
1001	HD 1080i60
1010	HD 1080i50
1011	HD 1035i60 (1920x1035)
1100	HD 1080p30
1101	HD 1080p25
1110	HD 1080p24
1111	Reserved

For the TV_STAND values corresponding to HD modes, only the YPbPr and RGB output formats are valid. For SD modes, the input data from the IPU can be in interlaced or progressive formats.

To provide HD 1080p50 and HD 1080p60 modes the following programming should be used:

- The TVDAC_SAMP_RATE should be set to 1 (the oversampling factor is x2).

- The TVE_CLK frequency should be 297 MHz or 297/0.001 MHz and the VIDEO_DATA_CLK frequency should be 148.5 MHz or 148.5/0.001 MHz.
- The rest of control registers should be configured as for HD 1080p25 and HD 1080p30 modes.

74.1.4.2 Operation in VGA Mode

In VGA mode, the most of the processing is bypassed. The data is in the RGB888 format. It is synchronized by the IDS to the internal pixel clock. The IDS output is fed directly to the UF. To provide such data flow, the TVE should be configured to test mode with upsampling and the selected TV standard should be one of HD standards. The UF provides gain correction and upsampling by 2 or 4 depending on required display resolution. The gain should be 548/511=1.072 to provide 0.7V peak-to-peak amplitude on the TVDAC output. Required clock frequencies for some VGA modes are shown in Table 74-4.

Table 74-11. Some VGA Modes and Clock Frequencies

Display resolution	Frame rate, Hz	Pixel clock frequency, MHz	TVE_CLK frequency, MHz	Upsampling factor	TVDAC sampling frequency, MHz
VGA (640x480)	60	25.175	100.7	4	100.7
VGA (640x480)	75	31.5	126	4	126
SVGA (800x600)	60	40	160	4	160
SVGA (800x600)	75	49.5	198	4	198
XGA (1024x768)	60	65	260	4	260
XGA (1024x768)	75	78.8	157.6	2	157.6
WXGA (1368x768)	60	85.86	171.72	2	171.72
SXGA (1280x1024)	60	108	216	2	216
SXGA (1280x1024)	75	135	270	2	270
WSXGA (1440x900)	60	106.47	212.94	2	212.94
WSXGA+ (1680x1050)	60	147.14	294.28	2	294.28

74.1.4.3 Standby Mode

The TVE is in Standby Mode, if the TV_OUT_MODE control parameter is 0. In this case, the most of internally generated clocks are gated off and internal resets are generated for the TVE sub-blocks. The TVDAC is disabled. If the CD is enabled, the UF, the CDC and the TVDAC are periodically turned on for a short time to monitor cable connection.

74.1.4.4 Cable Detection Modes

The CD operation is supported with both hardware and software involvement. Different types of the TV connection are distinguished by a number of cable lines connected to the TV output. If one or several cable lines are connected, the equivalent load impedance for these outputs is close to 37.5 Ohm and the voltage on them matches the nominal video levels. If no cable is connected, the load impedance is elevated to 75 Ohm. In this case, the output voltage is higher than the nominal video levels.

An additional feature of the CD system is detection of output line short to the ground. The short may appear if a headphone jack is plugged into the A/V connector and shorts the CVBS output to the ground.

Some natural limitations exist for CD operation. It is impossible to differentiate the mixed CVBS/S-video, the component YPrPb and the component RGB connections because in all these three cases a three-wire cable is used. So for these cases the software should set a default option or allow user choosing the interface type.

The CD is supported in both full-drive output mode (when the TV cable is connected directly to the TVDAC outputs) and low-drive output mode (when the TVDAC drives a high-resistance load and a buffer/filter is located between the TVDAC output and the TV connector). In both cases, three monitoring input pins should be connected on a board directly to the cable plug.

There are three CD modes:

1. Monitoring the current cable connection when the TVE is in Normal Operation Mode. The mode is enabled if TV_OUT_MODE is not 000 and the CD_TRIG_MODE is 0.
2. Periodical automatic pending the cable connection when the TVE is in Standby Mode. The mode is enabled if TV_OUT_MODE is 000 and the CD_TRIG_MODE is 0.
3. Checking the cable connection 'on demand' (on the trigger control generated by software). The mode is enabled if the CD_TRIG_MODE is 1.

In modes 1 and 2, the CDCU periodically initiates monitoring the output voltage. In mode 3, the ARM platform should set the CD_MAN_TRIG bit.

The CDC measures the voltage in two steps. At the first step, cable connection is checked. The voltage comparison results obtained at the first step are compared with the current TV output mode specified by the TV_OUT_MODE parameter. If the results do

not match the current output mode, an interrupt to the ARM platform is generated. At the second step, output short to ground is checked. If short has been detected, an interrupt to the ARM platform is generated.

The software is able to control for which output channels cable connection and/or output short should be checked. The CD_CH_#_LM_EN and CD_CH_#_SM_EN control bits are used for this. If an output load for one of the channels for which the CD_CH_#_LM_EN=1 is different from the value defined the TV_OUT_MODE, the load monitoring interrupt status bit (CD_LM_INT) is set and the corresponding interrupt is generated. If the output of one of the channels for which the CD_CH_#_SM_EN=1 is shorted to the ground, the short monitoring interrupt status bit (CD_SM_INT) is set and the corresponding interrupt is generated.

In the first CD mode, the monitoring period is a multiple of a TV field duration (for SD mode) or a TV frame duration (for HD mode). It is specified by the CD_MON_PER control parameter. The measurement is performed during the first half-line of the pre-equalization pulse in the vertical blanking period.

In the second CD mode, the monitoring period is a multiple of 0.31 sec. It is also specified by the CD_MON_PER control parameter. For output monitoring, the CDCU enables the monitored channels for a measurement time and forces its input to the desired value.

The reference voltage level used for comparison in the CDC can be equal to the Luma blanking level (about 320 mV) or to the Chroma blanking level (about 670 mV). The specific values of the reference levels depend on the CD_REF_MODE and the CD_CH_2_REF_LVL and the current TV output mode specified by the TV_OUT_MODE parameter. Table 74-12 shows the CD reference levels depending on these control parameters.

Table 74-12. CD Reference Level

Output channel	CD_REF_MODE	CD Reference Level[1] (TV_OUT_MODE)							
		Composite (CVBS)				S-video (Y/C)	CVBS and Y/C	YCrCb component	RGB component
		(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Channel #0	0	CD_CH_0_REF_LVL	0	CD_CH_0_REF_LVL	0	0	0	0	0
	1	CD_CH_0_REF_LVL	CD_CH_0_REF_LVL	CD_CH_0_REF_LVL	CD_CH_0_REF_LVL	CD_CH_0_REF_LVL	CD_CH_0_REF_LVL	CD_CH_0_REF_LVL	CD_CH_0_REF_LVL

Table continues on the next page...

Table 74-12. CD Reference Level (continued)

Output channel	CD_REF_MODE	CD Reference Level[1] (TV_OUT_MODE)							
		Composite (CVBS)				S-video (Y/C)	CVBS and Y/C	YCrCb component	RGB component
		(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Channel #1	0	CD_CH_1_REF_LVL	CD_CH_1_REF_LVL	CD_CH_1_REF_LVL	CD_CH_1_REF_LVL	1	1	CD_CH_1_REF_LVL	0
	1	CD_CH_1_REF_LVL	CD_CH_1_REF_LVL	CD_CH_1_REF_LVL	CD_CH_1_REF_LVL	CD_CH_1_REF_LVL	CD_CH_1_REF_LVL	CD_CH_1_REF_LVL	CD_CH_1_REF_LVL
Channel #2	0	CD_CH_2_REF_LVL	CD_CH_2_REF_LVL	0	0	CD_CH_2_REF_LVL	0	CD_CH_2_REF_LVL	0
	1	CD_CH_2_REF_LVL	CD_CH_2_REF_LVL	CD_CH_2_REF_LVL	CD_CH_2_REF_LVL	CD_CH_2_REF_LVL	CD_CH_2_REF_LVL	CD_CH_2_REF_LVL	CD_CH_2_REF_LVL

1. Luma Reference Level is 0, Chroma Reference Level is 1

74.2 External Signal Description

74.2.1 Detailed Signal Descriptions

The MMT pins are described in Table 74-13.

Table 74-13. Detailed Signal Descriptions

Signal	I/O	Description	
Clocks and resets			
tve_clk	I	High speed clock input	
		State Meaning	Asserted/Negated-Samples TVE flip-flops on positive edge
		Timing	Assertion/Negation - According to the required operation frequency. The allowed frequencies for SD are 216, 108, 54 MHz, for HD - 297 (297/1.001), 148.5 (148.5/1.001) MHz
ipg_clk_s	I	IPI SOC Sideband Bus gated clock for IP bus	
		State Meaning	According to the IP Bus protocol
		Timing	According to the IP Bus protocol. May be asynchronous with the tve_clk clock

Table continues on the next page...

Table 74-13. Detailed Signal Descriptions (continued)

Signal	I/O	Description	
fsx_ipu_clk	O	Internally generated clock fed to the IPU	
		State Meaning	Used for sampling of the output data interface in the IPU
		Timing	Derived by division of the TVE_CLK clock. Edge-synchronized with TVE_CLK
ipg_hard_async_reset_b	I	System HW reset	
		State Meaning	According to the IP Bus protocol
		Timing	According to the IP Bus protocol
ipg_soft_reset_b	I	System SW reset	
		State Meaning	According to the IP Bus protocol
		Timing	According to the IP Bus protocol
Input Data Interface			
video_data_clk_1	I	Clocking signal for input video data bus 1 from the IPU	
		State Meaning	Negative edge - Indicates that the IPU has updated the video_data_in_1 bus Positive edge - Indicates that the TVE can sample the video_data_in_1 bus
		Timing	The clock frequency is 13.5 MHz or 27 MHz in functional SD mode, 74.25 (74.25/1.001) MHz in functional HD mode. The clock frequency is 27 MHz (SD) and 74.25 (74.25/1.001) MHz (HD) in TVDAC test modes. The clock may be asynchronous with the TVE_CLK but the phase jitter between this clock and the clock generated in the TVE must be less than a half of video_data_clk cycle.
video_data_in_1[29:0]	I	Video data input bus 1 from the IPU	
		State Meaning	In functional mode - video_data_in_1[7:0]: Y video_data_in_1[15:8]: Cr video_data_in_1[23:16]: Cb video_data_in_1[29:24]: unused In TVDAC test modes 1 and 3 - video_data_in_1[9:0]: test data for TVDAC channel #0 video_data_in_1[19:10]: test data for TVDAC channel #1 video_data_in_1[29:20]: test data for TVDAC channel #2 In TVDAC test modes 2 and 4 - video_data_in_1[9:0]: test data for TVDAC channels #0 - #2 video_data_in_1[29:10]: unused
		Timing	The bus is sampled by negative edges of the video_data_clk_1 clock
video_data_en_1	I	Enable signal for input video data bus 1 from the IPU. In TVDAC test modes, the signal is unused.	
		State Meaning	Asserted-Indicates valid input video data on the bus 1 Negated- Indicates no valid input video data on the bus 1
		Timing	Assertion/Negation - Synchronized to negative edges of the video_data_clk_1 clock.

Table continues on the next page...

Table 74-13. Detailed Signal Descriptions (continued)

Signal	I/O	Description	
vsync_1	I	Vertical synchronization signal for input video data bus 1 from the IPU. In TVDAC test modes, the signal is unused.	
		State Meaning	Asserted-Indicates video frame start Negated- Indicates no video frame start
		Timing	Assertion/Negation- Synchronized to negative edges of the video_data_clk_1 clock. Remains asserted for one clock cycle
hsync_1	I	Horizontal synchronization signal for input video data bus 1 from the IPU. In TVDAC test modes, the signal is unused.	
		State Meaning	Asserted-Indicates video line start Negated- Indicates no video line start
		Timing	Assertion/Negation - Synchronized to negative edges of the video_data_clk_1 clock. Remains asserted for one clock cycle
video_data_clk_2	I	Clocking signal for input video data bus 2 from the IPU	
		State Meaning	Negative edge - Indicates that the IPU has updated the video_data_in_2 bus Positive edge - Indicates that the TVE can sample the video_data_in_2 bus
		Timing	The clock frequency is 13.5 MHz or 27 MHz in functional SD mode, 74.25 (74.25/1.001) MHz in functional HD mode. The clock frequency is 27 MHz (SD) and 74.25 (74.25/1.001) MHz (HD) in TVDAC test modes. The clock may be asynchronous with the TVE_CLK but the phase jitter between this clock and the clock generated in the TVE must be less than a half of video_data_clk cycle.
video_data_in_2[29:0]	I	Video data input bus 2 from the IPU	
		State Meaning	In functional TV encoding mode - video_data_in_2[7:0]: Y video_data_in_2[15:8]: Cr video_data_in_2[23:16]: Cb video_data_in_2[29:24]: unused In VGA mode - video_data_in_2[0]: ground video_data_in_2[8:1]: G video_data_in_2[10:9]: ground video_data_in_2[18:11]: R video_data_in_2[20:19]: ground video_data_in_2[28:21]: B video_data_in_2[29]: ground In TVDAC test modes 1 and 3 - video_data_in_2[9:0]: test data for TVDAC channel #0 video_data_in_2[19:10]: test data for TVDAC channel #1 video_data_in_2[29:20]: test data for TVDAC channel #2 In TVDAC test modes 2 and 4 - video_data_in_2[9:0]: test data for TVDAC channels #0 - #2 video_data_in_2[29:10]: unused
		Timing	The bus is sampled by negative edges of the video_data_clk_2 clock

Table continues on the next page...

Table 74-13. Detailed Signal Descriptions (continued)

Signal	I/O	Description	
video_data_en_2	I	Enable signal for input video data bus 2 from the IPU. In VGA mode and TVDAC test modes, the signal is unused.	
		State Meaning	Asserted-Indicates valid input video data on the bus 2 Negated- Indicates no valid input video data on the bus 2
		Timing	Assertion/Negation - Synchronized to negative edges of the video_data_clk_2 clock.
vsync_2	I	Vertical synchronization signal for input video data bus 2 from the IPU. In VGA mode and TVDAC test modes, the signal is unused.	
		State Meaning	Asserted-Indicates video frame start Negated- Indicates no video frame start
		Timing	Assertion/Negation- Synchronized to negative edges of the video_data_clk_2 clock. Remains asserted for one clock cycle
hsync_2	I	Horizontal synchronization signal for input video data bus 2 from the IPU. In VGA mode and TVDAC test modes, the signal is unused.	
		State Meaning	Asserted-Indicates video line start Negated- Indicates no video line start
		Timing	Assertion/Negation - Synchronized to negative edges of the video_data_clk_2 clock. Remains asserted for one clock cycle
External Test Data Interface			
IPP_IND_TEST_DATA_CLK	I	Clocking signal for input data from the external test bus	
		State Meaning	Negative edge - Indicates that an external source has updated the IPP_IND_TEST_DATA_IN bus Positive edge - Indicates that the TVE can sample the IPP_IND_TEST_DATA_IN bus
		Timing	The clock frequency is 13.5 or 27 MHz (SD) and 74.25 (74.25/1.001) MHz (HD) in TVE test mode. The clock frequency is 27-MHz (SD) and 74.25 (74.25/1.001) MHz (HD) in TVDAC test modes. The clock may be asynchronous with the TVE_CLK but the phase jitter between this clock and the clock generated in the TVE must be less than a half the clock cycle.

Table continues on the next page...

Table 74-13. Detailed Signal Descriptions (continued)

Signal	I/O	Description	
IPP_IND_TEST_DATA_IN[29:0]	I	Test data input from an external source	
		State Meaning	In TVE test mode - IPP_IND_TEST_DATA_IN[7:0]: Y IPP_IND_TEST_DATA_IN[15:8]: Cr IPP_IND_TEST_DATA_IN[23:16]: Cb IPP_IND_TEST_DATA_IN[24]: data enable IPP_IND_TEST_DATA_IN[25]: hsync IPP_IND_TEST_DATA_IN[26]: vsync IPP_IND_TEST_DATA_IN[29:27]: unused In TVDAC test modes 1 and 3 - IPP_IND_TEST_DATA_IN[9:0]: test data for TVDAC channel #0 IPP_IND_TEST_DATA_IN[19:10]: test data for TVDAC channel #1 IPP_IND_TEST_DATA_IN[29:20]: test data for TVDAC channel #2 In TVDAC test modes 2 and 4 - IPP_IND_TEST_DATA_IN[9:0]: test data for TVDAC channels #0 - #2 IPP_IND_TEST_DATA_IN[29:10]: unused
		Timing	The bus is sampled by negative edges of the IPP_IND_TEST_DATA_CLK clock
IPP_IBE_TEST_DATA_IN	O	Test data bus and clock pins direction control	
		State Meaning	Asserted - Indicates that the test data bus and clock pins direction should be set for test data input from an external source (the external test data is used) Negated - Indicates that the test data bus and clock pins direction should be set according to alternate functions of these pins (the external test data is unused)
		Timing	The signal is sampled by positive edges of the TVE clock
Output Analog Interface and Other Analog Pins			
IPP_ANALOGIO_TVCDC_IOG_BACK	I	Cable detection channel #0 (green) input	
		State Meaning	Voltage range is according to TVDAC specifications
		Timing	na
IPP_ANALOGIO_TVCDC_IOR_BACK	I	Cable detection channel #1 (red) input	
		State Meaning	Voltage range is according to TVDAC specifications
		Timing	na
IPP_ANALOGIO_TVCDC_IOB_BACK	I	Cable detection channel #2 (blue) input	
		State Meaning	Voltage range is according to TVDAC specifications
		Timing	na
IPP_ANALOGIO_TVDAC_VREFIN	I	Reference voltage input	
		State Meaning	Voltage range is according to TVDAC specifications
		Timing	na

Table continues on the next page...

Table 74-13. Detailed Signal Descriptions (continued)

Signal	I/O	Description	
IPP_ANALOGIO_TVDAC_IOG	O	TVDAC channel #0 (green) output	
		State Meaning	Voltage range is according to TVDAC specifications
		Timing	na
IPP_ANALOGIO_TVDAC_IOR	O	TVDAC channel #1 (red) output	
		State Meaning	Voltage range is according to TVDAC specifications
		Timing	na
IPP_ANALOGIO_TVDAC_IOB	O	TVDAC channel #2 (blue) output	
		State Meaning	Voltage range is according to TVDAC specifications
		Timing	na
IPP_ANALOGIO_TVDAC_VREFOUT	O	Reference voltage output	
		State Meaning	Voltage range is according to TVDAC specifications
		Timing	na
IPP_ANALOGIO_TVDAC_COMP	na	Reference voltage compensation	
		State Meaning	Voltage range is according to TVDAC specifications
		Timing	na
IPP_POWER_TVDAC_DHVDD	na	TVDAC 2.75V digital power supply	
		State Meaning	na
		Timing	na
IPP_POWER_TVDAC_DHVSS	na	TVDAC 2.75V digital ground	
		State Meaning	na
		Timing	na
IPP_POWER_TVDAC_AHVDD	na	TVDAC 2.75V analog power supply	
		State Meaning	na
		Timing	na
IPP_POWER_TVDAC_AHVSS	na	TVDAC analog ground	
		State Meaning	na
		Timing	na

Table continues on the next page...

Table 74-13. Detailed Signal Descriptions (continued)

Signal	I/O	Description	
IPP_POWER_TVDAC_AVDDG	na	TVDAC 2.75V analog power supply for output channel #0 (green)	
		State Meaning	na
		Timing	na
IPP_POWER_TVDAC_AVDDR	na	TVDAC 2.75V analog power supply for output channel #1 (red)	
		State Meaning	na
		Timing	na
IPP_POWER_TVDAC_AVDDB	na	TVDAC 2.75V analog power supply for output channel #2 (blue)	
		State Meaning	na
		Timing	na
IPP_POWER_TVDAC_AVSSG	na	TVDAC analog ground for output channel #0 (green)	
		State Meaning	na
		Timing	na
IPP_POWER_TVDAC_AVSSR	na	TVDAC analog ground for output channel #1 (red)	
		State Meaning	na
		Timing	na
IPP_POWER_TVDAC_AVSSB	na	TVDAC analog ground for output channel #2 (blue)	
		State Meaning	na
		Timing	na
IPP_POWER_TVCDC_HVDD	na	Cable detection circuit 2.75V analog power supply	
		State Meaning	na
		Timing	na
IPP_POWER_TVCDC_HVSS	na	Cable detection circuit analog ground	
		State Meaning	na
		Timing	na
IP Interrupt Bus Interface			

Table continues on the next page...

Table 74-13. Detailed Signal Descriptions (continued)

Signal	I/O	Description	
ipi_tve_int	O	Interrupt request	
		State Meaning	Asserted-Indicates TVE interrupt request to the ARM platform. Negated-Indicates no TVE interrupt request to the ARM platform.
		Timing	Assertion-Synchronized to S27_CLK. Remains asserted until the ARM platform will clear the corresponding status bit (according to the IP Bus protocol protocol) Negation- Synchronized to S27_CLK.
Miscellaneous Signals			
via_tvdac_clk_pol	I	TVDAC clock polarity	
		State Meaning	Asserted- Indicates that the TVDAC clock polarity is opposite to the tve_clk polarity. Negated- Indicates that the TVDAC clock polarity is equal to the tve_clk polarity.
		Timing	Assertion/Negation- Static signal assigned by TVE integration into SoC.
via_tvdac_en_pol	I	TVDAC global and channels enable signals polarity	
		State Meaning	Asserted- Indicates that the TVDAC enabling signals are active high. Negated- Indicates that the TVDAC enabling signals are active low.
		Timing	Assertion/Negation- Static signal assigned by TVE integration into SoC.

74.2.2 Interface between the IPU and the TVE

The timing diagrams of the interface signals between the IPU and the TVE are shown in Figure 74-5 and Figure 74-6.

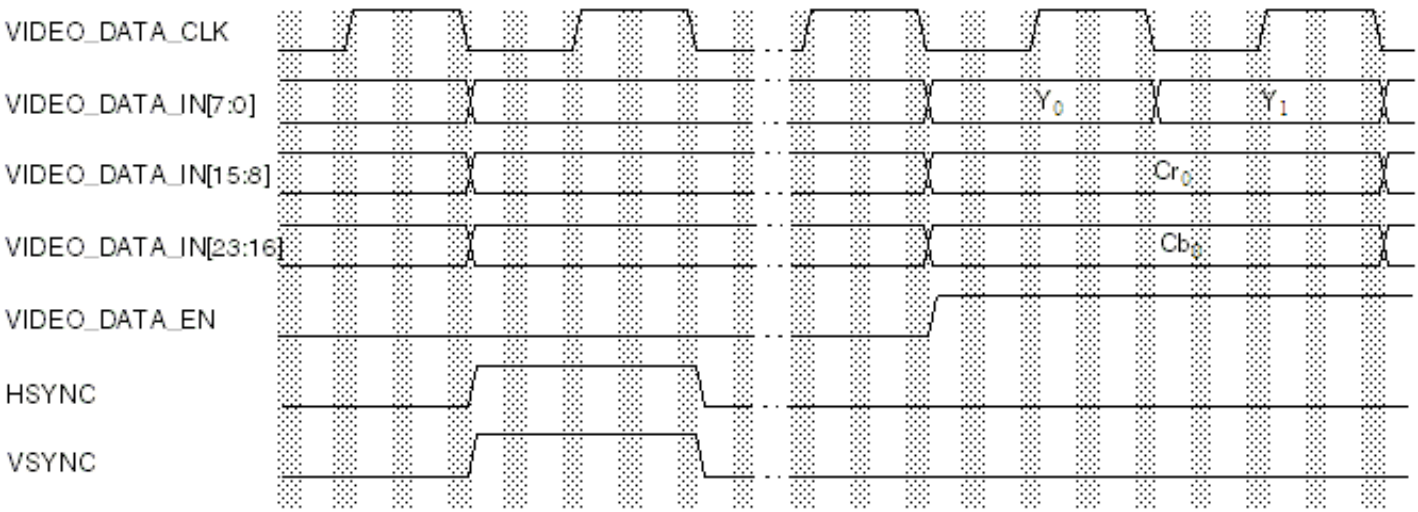


Figure 74-5. Interface between the IPU and the TVE for YCbCr 4:2:2 Input Format

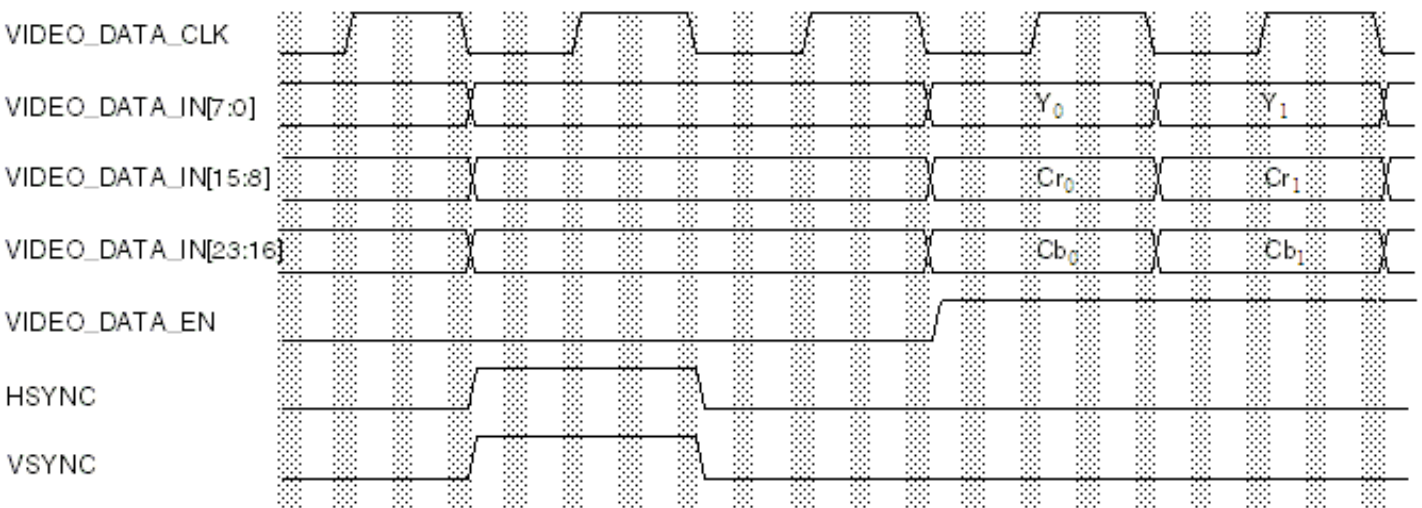


Figure 74-6. Interface between the IPU and the TVE for YCbCr 4:4:4 Input Format

74.2.3 Software Recommendations

74.2.3.1 PLL configuration

In order to obtain the best TV signal quality, the reference clock phase jitter should be minimized. For this purpose the PLL producing the TVE_CLK clock (the DPLLC4 in iMX53) should be configured with pre-divider factor of 1.

74.2.3.2 HSYNC and VSYNC in VGA Mode

GPIO pads driving the HSYNC and VSYNC signals in VGA mode should be configured for maximal driving strength.

74.3 Programmable Registers

TVE memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FF_0000	Common Configuration Register (TVE_COM_CONF_REG)	32	R/W	8004_2000h	74.3.1/ 4548
63FF_0004	Luma Filter Control Register 0 (TVE_LUMA_FILT_CONT_REG_0)	32	R/W	0000_0000h	74.3.2/ 4550
63FF_0008	Luma Filter Control Register 1 (TVE_LUMA_FILT_CONT_REG_1)	32	R/W	0000_0000h	74.3.3/ 4552
63FF_000C	Luma Filter Control Register 2 (TVE_LUMA_FILT_CONT_REG_2)	32	R/W	0000_0000h	74.3.4/ 4553
63FF_0010	Luma Filter Control Register 3 (TVE_LUMA_FILT_CONT_REG_3)	32	R/W	0000_0000h	74.3.5/ 4554
63FF_0014	Luma Statistic Analysis Control Register 0 (TVE_LUMA_SA_CONT_REG_0)	32	R/W	0000_0000h	74.3.6/ 4556
63FF_0018	Luma Statistic Analysis Control Register 1 (TVE_LUMA_SA_CONT_REG_1)	32	R/W	0000_0000h	74.3.7/ 4557
63FF_001C	Luma Statistic Analysis Status Register 0 (TVE_LUMA_SA_STAT_REG_0)	32	R	0000_0000h	74.3.8/ 4557
63FF_0020	Luma Statistic Analysis Status Register 1 (TVE_LUMA_SA_STAT_REG_1)	32	R/W	0000_0000h	74.3.9/ 4558
63FF_0024	Chroma Control Register (TVE_CHROMA_CONT_REG)	32	R/W	0000_0000h	74.3.10/ 4559
63FF_0028	TVDAC 0 Control Register (TVE_TVDAC_0_CONT_REG)	32	R/W	0000_0000h	74.3.11/ 4560
63FF_002C	TVDAC 1 Control Register (TVE_TVDAC_1_CONT_REG)	32	R/W	0000_0000h	74.3.12/ 4561
63FF_0030	TVDAC 2 Control Register (TVE_TVDAC_2_CONT_REG)	32	R/W	0000_0000h	74.3.13/ 4562
63FF_0034	Cable Detection Control Register (TVE_CD_CONT_REG)	32	R/W	0000_0000h	74.3.14/ 4563

Table continues on the next page...

TVE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FF_0038	VBI Data Control Register (TVE_VBI_DATA_CONT_REG)	32	R/W	0000_0000h	74.3.15/ 4565
63FF_003C	VBI Data Register 0 (TVE_VBI_DATA_REG_0)	32	R/W	0000_0000h	74.3.16/ 4567
63FF_0040	VBI Data Register 1 (TVE_VBI_DATA_REG_1)	32	R/W	0000_0000h	74.3.17/ 4567
63FF_0044	VBI Data Register 2 (TVE_VBI_DATA_REG_2)	32	R/W	0000_0000h	74.3.18/ 4568
63FF_0048	VBI Data Register 3 (TVE_VBI_DATA_REG_3)	32	R/W	0000_0000h	74.3.19/ 4568
63FF_004C	VBI Data Register 4 (TVE_VBI_DATA_REG_4)	32	R/W	0000_0000h	74.3.20/ 4568
63FF_0050	VBI Data Register 5 (TVE_VBI_DATA_REG_5)	32	R/W	0000_0000h	74.3.21/ 4569
63FF_0054	VBI Data Register 6 (TVE_VBI_DATA_REG_6)	32	R/W	0000_0000h	74.3.22/ 4569
63FF_0058	VBI Data Register 7 (TVE_VBI_DATA_REG_7)	32	R/W	0000_0000h	74.3.23/ 4570
63FF_005C	VBI Data Register 8 (TVE_VBI_DATA_REG_8)	32	R/W	0000_0000h	74.3.24/ 4570
63FF_0060	VBI Data Register 9 (TVE_VBI_DATA_REG_9)	32	R/W	0000_0000h	74.3.25/ 4570
63FF_0064	Interrupt Control Register (TVE_INT_CONT_REG)	32	R/W	0000_0000h	74.3.26/ 4571
63FF_0068	Status Register (TVE_STAT_REG)	32	R/W	0000_0000h	74.3.27/ 4573
63FF_006C	Test Mode Register (TVE_TST_MODE_REG)	32	R/W	0000_0000h	74.3.28/ 4576
63FF_0070	User Mode Control Register (TVE_USER_MODE_CONT_REG)	32	R/W	0000_0000h	74.3.29/ 4578
63FF_0074	SD Timing User Control Register 0 (TVE_SD_TIMING_USR_CONT_REG_0)	32	R/W	0000_0000h	74.3.30/ 4579
63FF_0078	SD Timing User Control Register 1 (TVE_SD_TIMING_USR_CONT_REG_1)	32	R/W	0000_0000h	74.3.31/ 4579
63FF_007C	SD Timing User Control Register 2 (TVE_SD_TIMING_USR_CONT_REG_2)	32	R/W	0000_0000h	74.3.32/ 4580
63FF_0080	HD Timing User Control Register 0 (TVE_HD_TIMING_USR_CONT_REG_0)	32	R/W	0000_0000h	74.3.33/ 4581
63FF_0084	HD Timing User Control Register 1 (TVE_HD_TIMING_USR_CONT_REG_1)	32	R/W	0000_0000h	74.3.34/ 4582

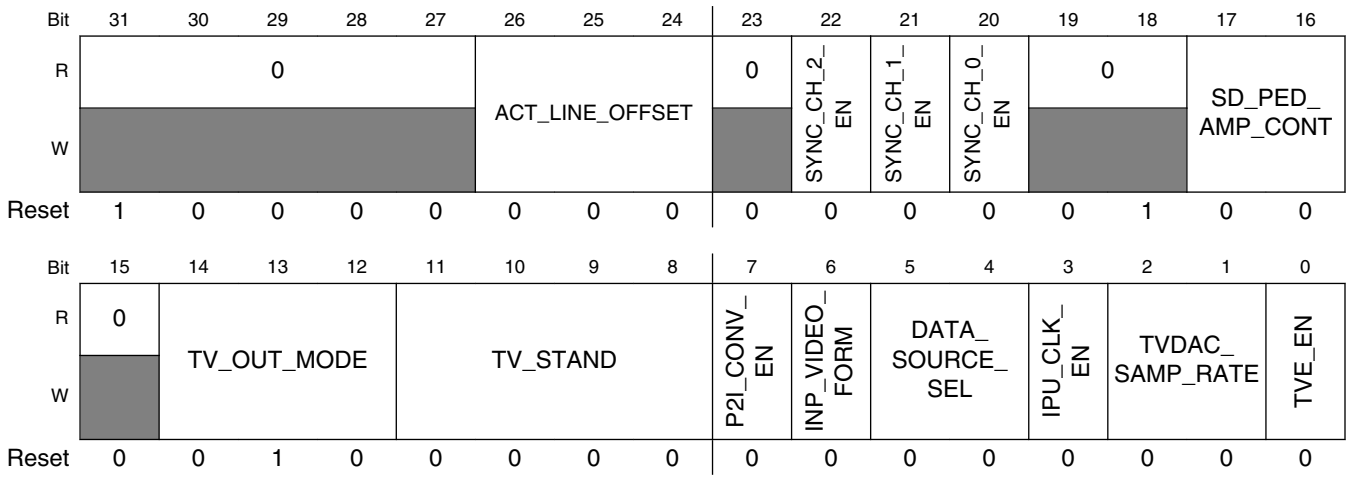
Table continues on the next page...

TVE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FF_0088	HD Timing User Control Register 2 (TVE_HD_TIMING_USR_CONT_REG_2)	32	R/W	0000_0000h	74.3.35/ 4583
63FF_008C	Luma User Control Register 0 (TVE_LUMA_USR_CONT_REG_0)	32	R/W	0000_0000h	74.3.36/ 4583
63FF_0090	Luma User Control Register 1 (TVE_LUMA_USR_CONT_REG_1)	32	R/W	0000_0000h	74.3.37/ 4584
63FF_0094	Luma User Control Register 2 (TVE_LUMA_USR_CONT_REG_2)	32	R/W	0000_0000h	74.3.38/ 4584
63FF_0098	Luma User Control Register 3 (TVE_LUMA_USR_CONT_REG_3)	32	R/W	0000_0000h	74.3.39/ 4585
63FF_009C	Color Space Conversion User Control Register 0 (TVE_CSC_USR_CONT_REG_0)	32	R/W	0000_0000h	74.3.40/ 4585
63FF_00A0	Color Space Conversion User Control Register 1 (TVE_CSC_USR_CONT_REG_1)	32	R/W	0000_0000h	74.3.41/ 4586
63FF_00A4	Color Space Conversion User Control Register 2 (TVE_CSC_USR_CONT_REG_2)	32	R/W	0000_0000h	74.3.42/ 4587
63FF_00A8	Blanking Level User Control Register (TVE_BLANK_USR_CONT_REG)	32	R/W	0000_0000h	74.3.43/ 4588
63FF_00AC	SD Modulation User Control Register (TVE_SD_MOD_USR_CONT_REG)	32	R/W	0000_0000h	74.3.44/ 4588
63FF_00B0	VBI Data User Control Register 0 (TVE_VBI_DATA_USR_CONT_REG_0)	32	R/W	0000_0000h	74.3.45/ 4589
63FF_00B4	VBI Data User Control Register 1 (TVE_VBI_DATA_USR_CONT_REG_1)	32	R/W	0000_0000h	74.3.46/ 4590
63FF_00B8	VBI Data User Control Register 2 (TVE_VBI_DATA_USR_CONT_REG_2)	32	R/W	0000_0000h	74.3.47/ 4590
63FF_00BC	VBI Data User Control Register 3 (TVE_VBI_DATA_USR_CONT_REG_3)	32	R/W	0000_0000h	74.3.48/ 4591
63FF_00C0	VBI Data User Control Register 4 (TVE_VBI_DATA_USR_CONT_REG_4)	32	R/W	0000_0000h	74.3.49/ 4592
63FF_00C4	Drop Compensation User Control Register (TVE_DROP_COMP_USR_CONT_REG)	32	R/W	0000_0000h	74.3.50/ 4592

74.3.1 Common Configuration Register (TVE_COM_CONF_REG)

Address: TVE_COM_CONF_REG is 63FF_0000h base + 0h offset = 63FF_0000h



TVE_COM_CONF_REG field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26–24 ACT_LINE_OFFSET	Offset of the first active video data line relative to a standard value. Defined in lines. 000 No offset 001 1-line offset 010 2-line offset 011 3-line offset 100 3-line offset 101 4-line offset 110 5-line offset 111 7-line offset
23 Reserved	This read-only field is reserved and always has the value zero. Reserved.
22 SYNC_CH_2_EN	Enable sync on output channel 2. 0 Disable 1 Enable
21 SYNC_CH_1_EN	Enable sync on output channel 1. 0 Disable 1 Enable
20 SYNC_CH_0_EN	Enable sync on output channel 0. 0 Disable 1 Enable

Table continues on the next page...

TVE_COM_CONF_REG field descriptions (continued)

Field	Description
19–18 Reserved	This read-only field is reserved and always has the value zero. Reserved.
17–16 SD_PED_AMP_CONT	Control pedestal, Y/R/G/B black-to-white and synch output amplitude. 00 714mV (black-to-white)/286mV (synch), no pedestal 01 714mV (black-to-white)/286mV (synch), with pedestal 10 700mV (black-to-white)/300mV (synch), no pedestal 11 Reserved
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14–12 TV_OUT_MODE	Select TV output mode. For HD TV standards, only YPbPr and RGB operation modes are valid. 000 Disable output 001 CVBS on channel #0 010 CVBS on channel #2 011 CVBS on both channels #0 and #2 100 S-video on channels #0 and #1 101 S-video on channels #0 and #1 and CVBS on channel #2 110 YPbPr component 111 RGB component
11–8 TV_STAND	Select TV standard. 0000 SD NTSC 0001 SD PALM 0010 SD Combination PALN 0011 SD PAL (B,D,G,H,I) 0100 HD 720p60 0101 HD 720p50 0110 HD 720p30 0111 HD 720p25 1000 HD 720p24 1001 HD 1080i60 1010 HD 1080i50 1011 HD 1035i60 (1920x1035) 1100 HD 1080p30 1101 HD 1080p25 1110 HD 1080p24 1111 Reserved
7 P2I_CONV_EN	Enable progressive to interlaced conversion. Valid only in for interlaced output mode. Must be 0 for progressive output mode. 0 Disable 1 Enable
6 INP_VIDEO_FORM	Select input video format. 0 YCbCr 4:2:2 1 YCbCr 4:4:4

Table continues on the next page...

TVE_COM_CONF_REG field descriptions (continued)

Field	Description
5–4 DATA_ SOURCE_SEL	Select data source. 00 Video data bus 1 from the IPU 01 Video data bus 2 from the IPU 10 External test data bus 11 Internal Color Bar Generator
3 IPU_CLK_EN	Enable clock output for the IPU. 0 Disable 1 Enable
2–1 TVDAC_SAMP_ RATE	Select TVDAC sampling rate. Must match the TVE_CLK clock frequency. 00 216 MHz (SD mode) or 297 MHz (HD mode) 01 108 MHz (SD mode) or 148.5 MHz (HD mode) 10 54 MHz (SD mode) or reserved (HD mode) 11 Reserved
0 TVE_EN	Enable the When low, resets all internal registers excluding the IPS registers and gates off internal clocks. 0 Disable 1 Enable

74.3.2 Luma Filter Control Register 0 (TVE_LUMA_FILT_CONT_REG_0)

Address: TVE_LUMA_FILT_CONT_REG_0 is 63FF_0000h base + 4h offset = 63FF_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DEFLICK_HIGH_THRESH								DEFLICK_MID_THRESH							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEFLICK_LOW_THRESH								0	DEFLICK_COEF			0		DEFLICK_MEAS_WIN	DEFLICK_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_LUMA_FILT_CONT_REG_0 field descriptions

Field	Description
31–24 DEFLICK_ HIGH_THRESH	Vertical deflickering/ fine sharpening/high-frequency noise reduction high threshold. 00000000 0

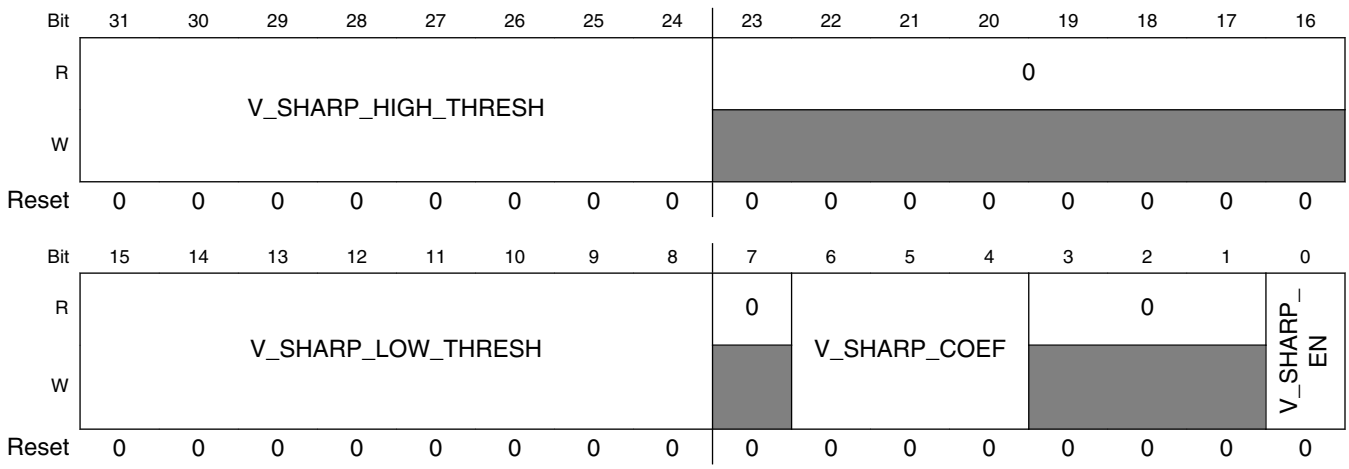
Table continues on the next page...

TVE_LUMA_FILT_CONT_REG_0 field descriptions (continued)

Field	Description
	00000001 1 11111111 255
23–16 DEFLICK_MID_ THRESH	Vertical deflickering/ fine sharpening/high-frequency noise reduction mid threshold. 00000000 0 00000001 1 11111111 255
15–8 DEFLICK_LOW_ THRESH	Vertical deflickering/ fine sharpening/high-frequency noise reduction low threshold. 00000000 0 00000001 1 11111111 255
7 Reserved	This read-only field is reserved and always has the value zero. Reserved.
6–4 DEFLICK_COEF	Vertical deflickering/ fine sharpening/high-frequency noise reduction coefficient. 000 0.125 001 0.25 010 0.375 011 0.5 100 0.625 101 0.75 110 0.875 111 1.0
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved.
1 DEFLICK_ MEAS_WIN	Select vertical deflickering/ fine sharpening/high-frequency noise reduction measurement window. 0 1-pixel window 1 3-pixels window
0 DEFLICK_EN	Enable vertical deflickering/ fine sharpening/high-frequency noise reduction filter. 0 Disable 1 Enable

74.3.3 Luma Filter Control Register 1 (TVE_LUMA_FILT_CONT_REG_1)

Address: TVE_LUMA_FILT_CONT_REG_1 is 63FF_0000h base + 8h offset = 63FF_0008h



TVE_LUMA_FILT_CONT_REG_1 field descriptions

Field	Description
31–24 V_SHARP_ HIGH_THRESH	Vertical coarse sharpening/ mid-frequency noise reduction high threshold. 00000000 0 00000001 1 11111111 255
23–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 V_SHARP_ LOW_THRESH	Vertical coarse sharpening/ mid-frequency noise reduction low threshold. 00000000 0 00000001 1 11111111 255
7 Reserved	This read-only field is reserved and always has the value zero. Reserved.
6–4 V_SHARP_ COEF	Vertical coarse sharpening/ mid-frequency noise reduction coefficient. 000 0.125 001 0.25 010 0.375 011 0.5 100 0.625 101 0.75 110 0.875 111 1.0
3–1 Reserved	This read-only field is reserved and always has the value zero. Reserved.

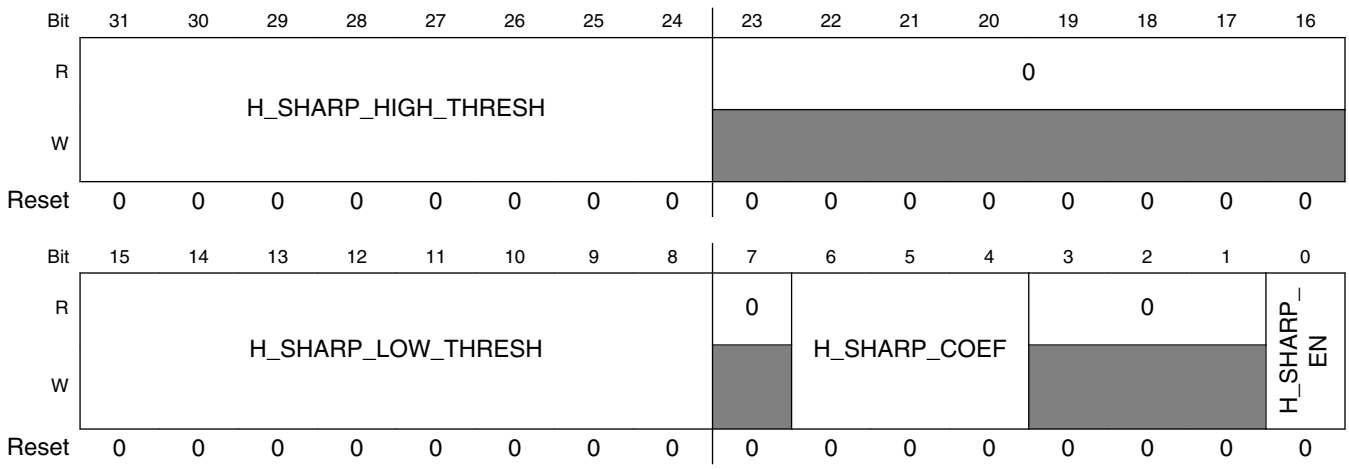
Table continues on the next page...

TVE_LUMA_FILT_CONT_REG_1 field descriptions (continued)

Field	Description
0 V_SHARP_EN	Enable vertical coarse sharpening/ mid-frequency noise reduction filter. 0 Disable 1 Enable

74.3.4 Luma Filter Control Register 2 (TVE_LUMA_FILT_CONT_REG_2)

Address: TVE_LUMA_FILT_CONT_REG_2 is 63FF_0000h base + Ch offset = 63FF_000Ch



TVE_LUMA_FILT_CONT_REG_2 field descriptions

Field	Description
31–24 H_SHARP_ HIGH_THRESH	Horizontal coarse sharpening/ mid-frequency noise reduction high threshold. 00000000 0 00000001 1 11111111 255
23–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 H_SHARP_ LOW_THRESH	Horizontal coarse sharpening/ mid-frequency noise reduction low threshold. 00000000 0 00000001 1 11111111 255
7 Reserved	This read-only field is reserved and always has the value zero. Reserved.
6–4 H_SHARP_ COEF	Horizontal coarse sharpening/ mid-frequency noise reduction coefficient. 000 0.125 001 0.25

Table continues on the next page...

TVE_LUMA_FILT_CONT_REG_2 field descriptions (continued)

Field	Description
	010 0.375 011 0.5 100 0.625 101 0.75 110 0.875 111 1.0
3–1 Reserved	This read-only field is reserved and always has the value zero. Reserved.
0 H_SHARP_EN	Enable horizontal coarse sharpening/ mid-frequency noise reduction filter. 0 Disable 1 Enable

74.3.5 Luma Filter Control Register 3 (TVE_LUMA_FILT_CONT_REG_3)

Address: TVE_LUMA_FILT_CONT_REG_3 is 63FF_0000h base + 10h offset = 63FF_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DERING_HIGH_THRESH								DERING_MID_THRESH							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DERING_LOW_THRESH								0	DERING_COEF			0	SUPP_ FILTER_ TYPE		DERING_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_LUMA_FILT_CONT_REG_3 field descriptions

Field	Description
31–24 DERING_HIGH_ THRESH	Horizontal deringing/ fine sharpening/high-frequency noise reduction high threshold. 00000000 0 00000001 1 11111111 255
23–16 DERING_MID_ THRESH	Horizontal deringing/ fine sharpening/high-frequency noise reduction mid threshold. 00000000 0

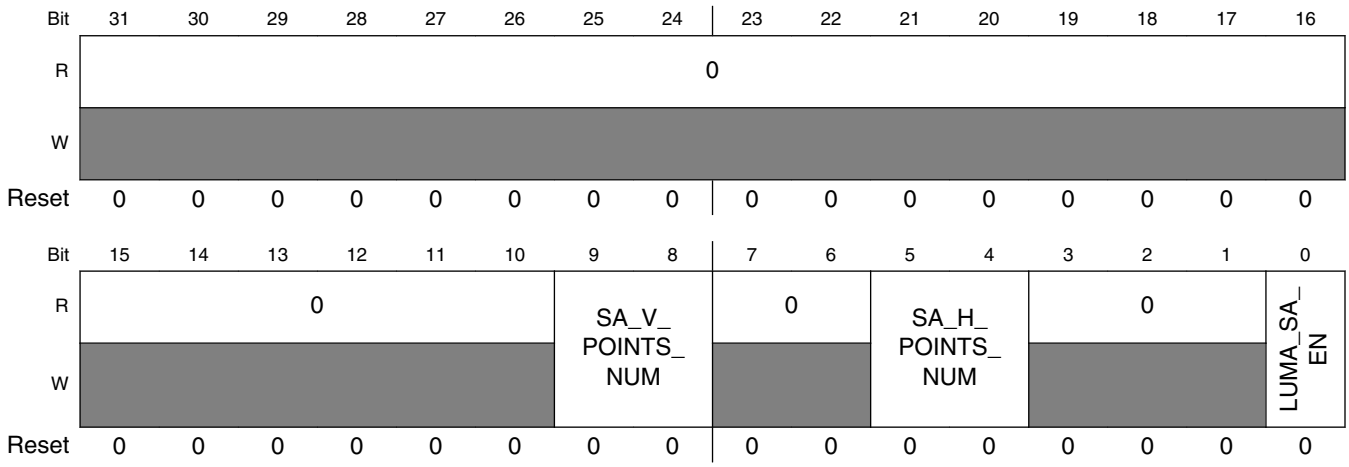
Table continues on the next page...

TVE_LUMA_FILT_CONT_REG_3 field descriptions (continued)

Field	Description
	00000001 1 11111111 255
15–8 DERING_LOW_THRESH	Horizontal deringing/ fine sharpening/high-frequency noise reduction low threshold. 00000000 0 00000001 1 11111111 255
7 Reserved	This read-only field is reserved and always has the value zero. Reserved.
6–4 DERING_COEF	Horizontal deringing/ fine sharpening/high-frequency noise reduction coefficient. 000 0.125 001 0.25 010 0.375 011 0.5 100 0.625 101 0.75 110 0.875 111 1.0
3 Reserved	This read-only field is reserved and always has the value zero. Reserved.
2–1 SUPP_FILTER_TYPE	Select supplement filter type. 00 No supplement filter 01 Lowpass filter 10 PAL notch filter (4.43 MHz) 11 NTSC notch filter (3.58 MHz)
0 DERING_EN	Enable horizontal deringing/ fine sharpening/high-frequency noise reduction filter. 0 Disable 1 Enable

74.3.6 Luma Statistic Analysis Control Register 0 (TVE_LUMA_SA_CONT_REG_0)

Address: TVE_LUMA_SA_CONT_REG_0 is 63FF_0000h base + 14h offset = 63FF_0014h



TVE_LUMA_SA_CONT_REG_0 field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value zero. Reserved.
9–8 SA_V_POINTS_NUM	Select number of vertical points (lines) used for statistic analysis. 00 256 points 01 128 points 10 64 points 11 32 points
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved.
5–4 SA_H_POINTS_NUM	Select number of horizontal points (pixels in a line) used for statistic analysis. 00 256 points 01 128 points 10 64 points 11 32 points
3–1 Reserved	This read-only field is reserved and always has the value zero. Reserved.
0 LUMA_SA_EN	Enable luma statistic analysis. 0 Disable 1 Enable

74.3.7 Luma Statistic Analysis Control Register 1 (TVE_LUMA_SA_CONT_REG_1)

Address: TVE_LUMA_SA_CONT_REG_1 is 63FF_0000h base + 18h offset = 63FF_0018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SA_WIN_V_OFFSET								SA_WIN_H_OFFSET								SA_WIN_HEIGHT								SA_WIN_WIDTH							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_LUMA_SA_CONT_REG_1 field descriptions

Field	Description
31–24 SA_WIN_V_OFFSET	Statistic analysis window vertical offset relative to start of an input active video data frame/field. A sum of this parameter and SA_WIN_HEIGHT should not exceed the height of the input video data frame/field. 00000000 0 lines 00000001 8 lines 11111111 2040 lines
23–16 SA_WIN_H_OFFSET	Statistic analysis window horizontal offset relative to start of an input active video data line. A sum of this parameter and SA_WIN_WIDTH should not exceed the length of the input video data line. 00000000 0 pixels 00000001 8 pixels 11111111 2040 pixels
15–8 SA_WIN_HEIGHT	Statistic analysis window height. 00000000 0 lines 00000001 8 lines 11111111 2040 lines
7–0 SA_WIN_WIDTH	Statistic analysis window width. 00000000 0 pixels 00000001 8 pixels 11111111 2040 pixels

74.3.8 Luma Statistic Analysis Status Register 0 (TVE_LUMA_SA_STAT_REG_0)

Address: TVE_LUMA_SA_STAT_REG_0 is 63FF_0000h base + 1Ch offset = 63FF_001Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DERING_MEAS_MEAN								H_SHARP_MEAS_MEAN								V_SHARP_MEAS_MEAN								DEFLICK_MEAS_MEAN							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_LUMA_SA_STAT_REG_0 field descriptions

Field	Description
31–24 DERING_ MEAS_MEAN	Mean absolute value of horizontal deringing/ fine sharpening/high-frequency noise reduction measurement filter output. 00000000 0 00000001 1 11111111 255
23–16 H_SHARP_ MEAS_MEAN	Mean absolute value of horizontal coarse sharpening/ mid-frequency noise reduction measurement filter output. 00000000 0 00000001 1 11111111 255
15–8 V_SHARP_ MEAS_MEAN	Mean absolute value of vertical coarse sharpening/ mid-frequency noise reduction measurement filter output. 00000000 0 00000001 1 11111111 255
7–0 DEFLICK_ MEAS_MEAN	Mean absolute value of vertical deflickering/ fine sharpening/high-frequency noise reduction measurement filter output. 00000000 0 00000001 1 11111111 255

74.3.9 Luma Statistic Analysis Status Register 1 (TVE_LUMA_SA_STAT_REG_1)

Address: TVE_LUMA_SA_STAT_REG_1 is 63FF_0000h base + 20h offset = 63FF_0020h

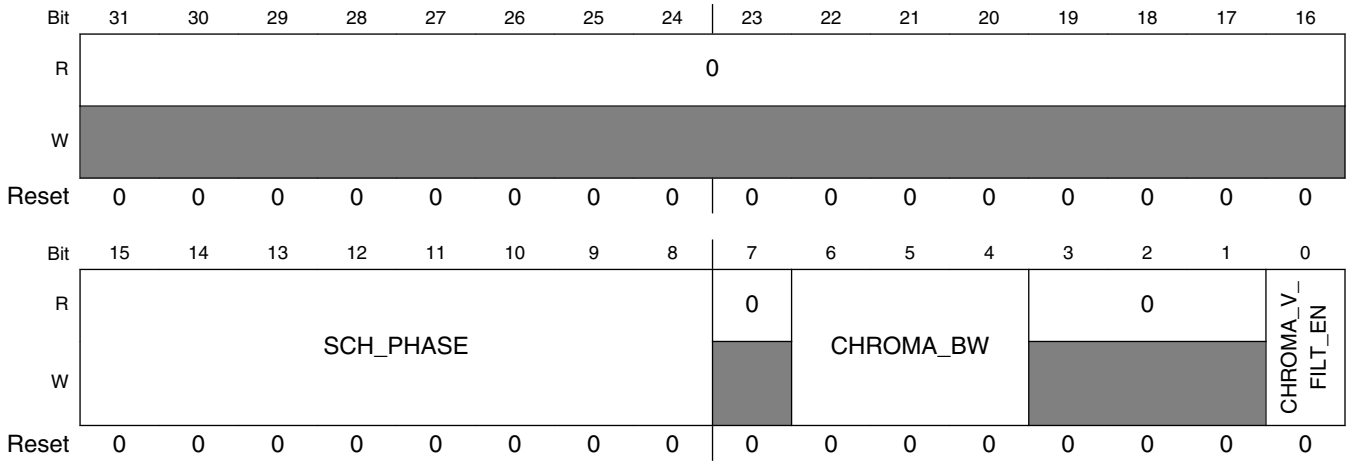
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								LUMA_MEAN							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_LUMA_SA_STAT_REG_1 field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved.
7–0 LUMA_MEAN	Mean value of Luma. 00000000 0 00000001 1 11111111 255

74.3.10 Chroma Control Register (TVE_CHROMA_CONT_REG)

Address: TVE_CHROMA_CONT_REG is 63FF_0000h base + 24h offset = 63FF_0024h



TVE_CHROMA_CONT_REG field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 SCH_PHASE	Subcarrier initial phase offset (SD only). The SCH phase is loaded into the 8 MSBs of the phase accumulator and the 22 LSBs are cleared upon reset and every 4 fields for 525-line standards and every 8 fields for 625-line standards. 00000000 0 degrees 00000001 360/256 degrees 11111111 255*360/256 degrees
7 Reserved	This read-only field is reserved and always has the value zero. Reserved.
6–4 CHROMA_BW	Select Chroma filter bandwidth. 000 0.6 MHz for SD mode and YCbCr 4:2:2 input format 1.2 MHz for SD mode and YCbCr 4:4:4 input format 3.3 MHz for HD mode and YCbCr 4:2:2 input format 6.6 MHz for HD mode and YCbCr 4:4:4 input format 001 1.0 MHz for SD mode and YCbCr 4:2:2 input format 2.0 MHz for SD mode and YCbCr 4:4:4 input format 5.5 MHz for HD mode and YCbCr 4:2:2 input format 11.0 MHz for HD mode and YCbCr 4:4:4 input format 010 1.3 MHz for SD mode and YCbCr 4:2:2 input format 3.0 MHz for SD mode and YCbCr 4:4:4 input format 7.1 MHz for HD mode and YCbCr 4:2:2 input format 16.5 MHz for HD mode and YCbCr 4:4:4 input format 011 2.0 MHz for SD mode and YCbCr 4:2:2 input format 4.0 MHz for SD mode and YCbCr 4:4:4 input format 11.0 MHz for HD mode and YCbCr 4:2:2 input format 22.0 MHz for HD mode and YCbCr 4:4:4 input format

Table continues on the next page...

TVE_CHROMA_CONT_REG field descriptions (continued)

Field	Description
	100 3.0 MHz for SD mode and YCbCr 4:2:2 input format 6.4 MHz for SD mode and YCbCr 4:4:4 input format 16.5 MHz for HD mode and YCbCr 4:2:2 input format 35.2 MHz for HD mode and YCbCr 4:4:4 input format 101 Reserved 110 Reserved 111 Reserved
3–1 Reserved	This read-only field is reserved and always has the value zero. Reserved.
0 CHROMA_V_FILT_EN	Enable vertical chroma filter. Must be 0 for HD mode. 0 Disable 1 Enable

74.3.11 TVDAC 0 Control Register (TVE_TVDAC_0_CONT_REG)

Address: TVE_TVDAC_0_CONT_REG is 63FF_0000h base + 28h offset = 63FF_0028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								BG_RDY_TIME								TVDAC_0_OFFSET								0		TVDAC_0_GAIN					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_TVDAC_0_CONT_REG field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved.
23–16 BG_RDY_TIME	TVDAC bandgap reference ready time. Specified with resolution of 1 msec. This parameter characterizes the TVDAC ready time after enabling the TVDAC. 00000000 1 msec 00000001 2 msec 11111110 255 msec 11111111 0 msec
15–8 TVDAC_0_OFFSET	Offset value in the two's complement format for the TVDAC channel #0. Added to the signal after the gain is corrected according to TVDAC_0_GAIN. 10000000 128 10000001 127 11111111 1 00000000 0 00000001 1 01111111 127

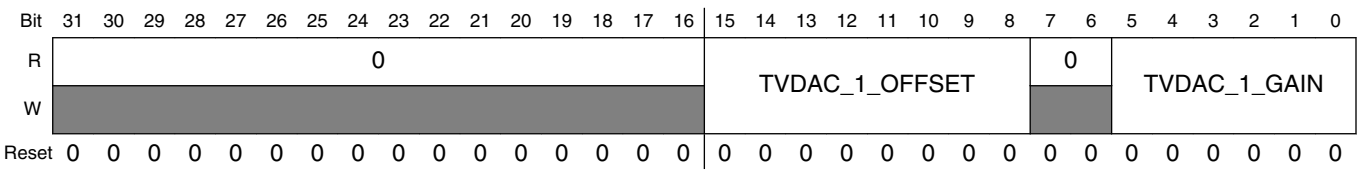
Table continues on the next page...

TVE_TVDAC_0_CONT_REG field descriptions (continued)

Field	Description
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved.
5–0 TVDAC_0_GAIN	Select gain value for the TVDAC channel #0. 100000 1-32/128 100001 1-31/128 111111 1-1/128 000000 1 000001 1+1/128 011111 1+31/128

74.3.12 TVDAC 1 Control Register (TVE_TVDAC_1_CONT_REG)

Address: TVE_TVDAC_1_CONT_REG is 63FF_0000h base + 2Ch offset = 63FF_002Ch

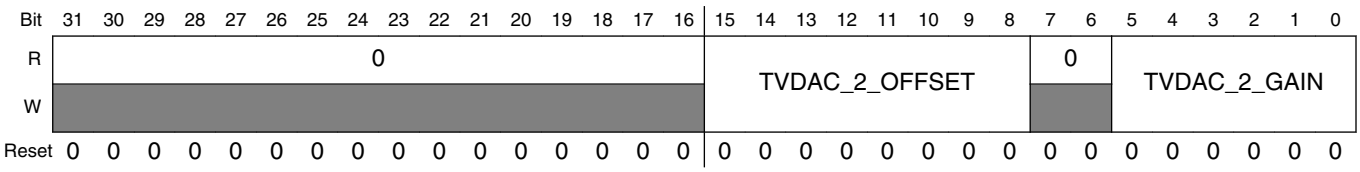


TVE_TVDAC_1_CONT_REG field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 TVDAC_1_ OFFSET	Offset value in the two's complement format for the TVDAC channel #1. Added to the signal after the gain is corrected according to TVDAC_0_GAIN. 10000000 128 10000001 -127 11111111 1 00000000 0 00000001 1 01111111 127
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved.
5–0 TVDAC_1_GAIN	Select gain value for the TVDAC channel #1. 100000 1-32/128 100001 1-31/128 111111 1-1/128 000000 1 000001 1+1/128 011111 1+31/128

74.3.13 TVDAC 2 Control Register (TVE_TVDAC_2_CONT_REG)

Address: TVE_TVDAC_2_CONT_REG is 63FF_0000h base + 30h offset = 63FF_0030h



TVE_TVDAC_2_CONT_REG field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved.
15–8 TVDAC_2_ OFFSET	Offset value in the two's complement format for the TVDAC channel #2. Added to the signal after the gain is corrected according to TVDAC_0_GAIN. 10000000 128 10000001 127 11111111 -1 00000000 0 00000001 1 01111111 127
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved.
5–0 TVDAC_2_GAIN	Select gain value for the TVDAC channel #2. 100000 1-32/128 100001 1-31/128 111111 1-1/128 000000 1 000001 1+1/128 011111 1+31/128

74.3.14 Cable Detection Control Register (TVE_CD_CONT_REG)

Address: TVE_CD_CONT_REG is 63FF_0000h base + 34h offset = 63FF_0034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0									CD_CH_2_SM_EN	CD_CH_1_SM_EN	CD_CH_2_SM_EN	0	CD_CH_2_LM_EN	CD_CH_1_LM_EN	CD_CH_2_LM_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				CD_REF_MODE	CD_CH_2_REF_LVL	CD_CH_1_REF_LVL	CD_CH_0_REF_LVL	CD_MON_PER				0		CD_TRIG_MODE	CD_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_CD_CONT_REG field descriptions

Field	Description
31–23 Reserved	This read-only field is reserved and always has the value zero. Reserved.
22 CD_CH_2_SM_EN	TVDAC channel #2 short monitoring enable. 0 Disable 1 Enable
21 CD_CH_1_SM_EN	TVDAC channel #1 short monitoring enable. 0 Disable 1 Enable
20 CD_CH_2_SM_EN	TVDAC channel #0 short monitoring enable. 0 Disable 1 Enable
19 Reserved	This read-only field is reserved and always has the value zero. Reserved.
18 CD_CH_2_LM_EN	TVDAC channel #2 load impedance monitoring enable. 0 Disable 1 Enable
17 CD_CH_1_LM_EN	TVDAC channel #1 load impedance monitoring enable. 0 Disable 1 Enable
16 CD_CH_2_LM_EN	TVDAC channel #0 load impedance monitoring enable.

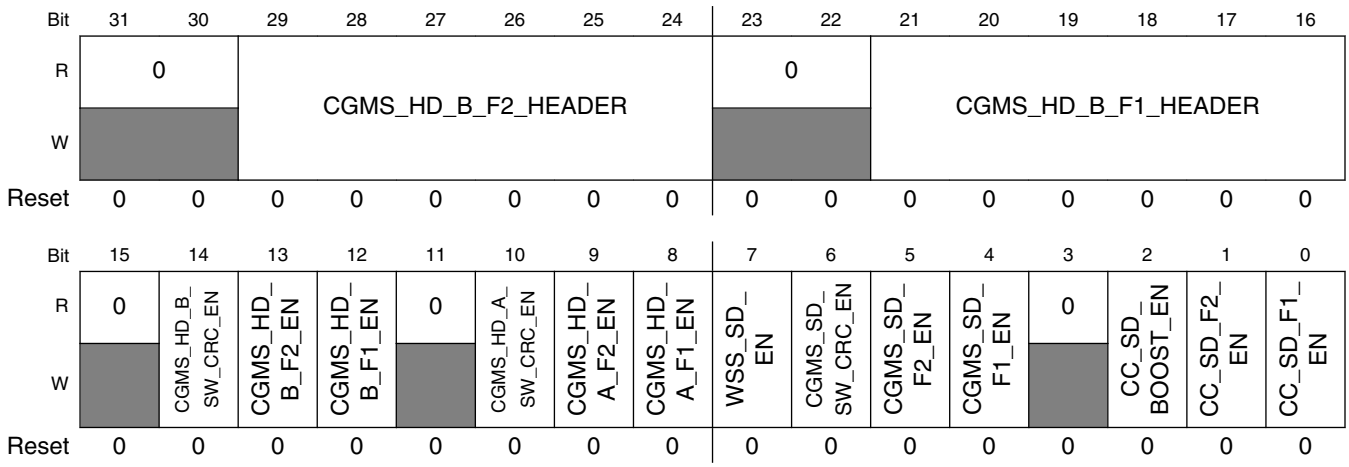
Table continues on the next page...

TVE_CD_CONT_REG field descriptions (continued)

Field	Description
	0 Disable 1 Enable
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved.
11 CD_REF_MODE	CD reference mode. Specifies how the reference voltage level for CD comparators is set for active TVDAC channels. The bit is ignored for inactive channels or in standby mode (manual mode assumed). The bit is also ignored for channels# 1 and #2 in YPrPb mode (TV_OUT_MODE=110). 0 Automatic mode when the reference voltage is set according to the channel signal type defined by the TV_OUT_MODE 1 Manual mode when the reference voltage is set according to the CD_CH_#_REF_LVL
10 CD_CH_2_REF_LVL	CD reference level for TVDAC channel #2. Specified the expected level of the channel output voltage when the output is loaded by nominal 37.5-Ohm impedance. This bit is ignored if the channel is active and CD_REF_MODE is 0 excluding YPrPb mode (TV_OUT_MODE=110). 0 The reference level corresponds to the Luma blanking level 1 The reference level corresponds to the Chroma blanking level
9 CD_CH_1_REF_LVL	CD reference level for TVDAC channel #1. Specified the expected level of the channel output voltage when the output is loaded by nominal 37.5-Ohm impedance. This bit is ignored if the channel is active and CD_REF_MODE is 0 excluding YPrPb mode (TV_OUT_MODE=110). 0 The reference level corresponds to the Luma blanking level 1 The reference level corresponds to the Chroma blanking level
8 CD_CH_0_REF_LVL	CD reference level for TVDAC channel #0. Specified the expected level of the channel output voltage when the output is loaded by nominal 37.5-Ohm impedance. This bit is ignored if the channel is active and CD_REF_MODE is 0. 0 The reference level corresponds to the Luma blanking level 1 The reference level corresponds to the Chroma blanking level
7–4 CD_MON_PER	Cable detection monitoring period. If TV_OUT_MODE is not 000 (during normal operation), specified in TV fields for SD or in TV frames for HD. If TV_OUT_MODE is 000 (standby mode), specified in 0.31-sec units. 0000 1 field (SD) or 1 frame (HD) for normal operation or 0.31 sec for standby mode 0001 2 fields (SD) or 2 frames (HD) for normal operation or 0.62 sec for standby mode 1111 16 fields (SD) or 16 frames (HD) for normal operation or 4.96 sec for standby mode
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved.
1 CD_TRIG_MODE	CD trigger mode. 0 Automatic (periodical) trigger 1 Manual (single shot) trigger
0 CD_EN	Enable cable detection. When low, resets the CD Control Unit and the CD Circuit. 0 Disable 1 Enable

74.3.15 VBI Data Control Register (TVE_VBI_DATA_CONT_REG)

Address: TVE_VBI_DATA_CONT_REG is 63FF_0000h base + 38h offset = 63FF_0038h



TVE_VBI_DATA_CONT_REG field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved.
29–24 CGMS_HD_B_F2_HEADER	HD CGMS Type B header for field 2.
23–22 Reserved	This read-only field is reserved and always has the value zero. Reserved.
21–16 CGMS_HD_B_F1_HEADER	HD CGMS Type B header for field 1.
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14 CGMS_HD_B_SW_CRC_EN	Enable HD CGMS Type B CRC value calculated by software. Otherwise the CRC value is calculated by hardware. 0 Disable 1 Enable
13 CGMS_HD_B_F2_EN	Enable HD CGMS Type B data insertion for field 2. 0 Disable 1 Enable
12 CGMS_HD_B_F1_EN	Enable HD CGMS Type B data insertion for field 1. 0 Disable 1 Enable
11 Reserved	This read-only field is reserved and always has the value zero. Reserved.

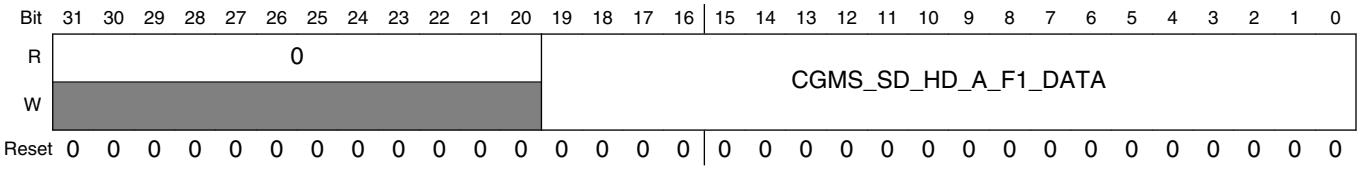
Table continues on the next page...

TVE_VBI_DATA_CONT_REG field descriptions (continued)

Field	Description
10 CGMS_HD_A_SW_CRC_EN	Enable HD CGMS Type A CRC value calculated by software. Otherwise the CRC value is calculated by hardware. 0 Disable 1 Enable
9 CGMS_HD_A_F2_EN	Enable HD CGMS Type A data insertion for field 2. 0 Disable 1 Enable
8 CGMS_HD_A_F1_EN	Enable HD CGMS Type A data insertion for field 1. 0 Disable 1 Enable
7 WSS_SD_EN	Enable SD WSS data insertion. 0 Disable 1 Enable
6 CGMS_SD_SW_CRC_EN	Enable SD CGMS CRC value calculated by software. Otherwise the CRC value is calculated by hardware. 0 Disable 1 Enable
5 CGMS_SD_F2_EN	Enable SD CGMS data insertion for field 2. 0 Disable 1 Enable
4 CGMS_SD_F1_EN	Enable SD CGMS data insertion for field 1. 0 Disable 1 Enable
3 Reserved	This read-only field is reserved and always has the value zero. Reserved.
2 CC_SD_BOOST_EN	Enable SD closed caption level boost by a factor of 1.7. May be used in RGB TV output mode. 0 Disable 1 Enable
1 CC_SD_F2_EN	Enable SD closed caption data insertion for field 2. 0 Disable 1 Enable
0 CC_SD_F1_EN	Enable SD closed caption data insertion for field 1. 0 Disable 1 Enable

74.3.16 VBI Data Register 0 (TVE_VBI_DATA_REG_0)

Address: TVE_VBI_DATA_REG_0 is 63FF_0000h base + 3Ch offset = 63FF_003Ch

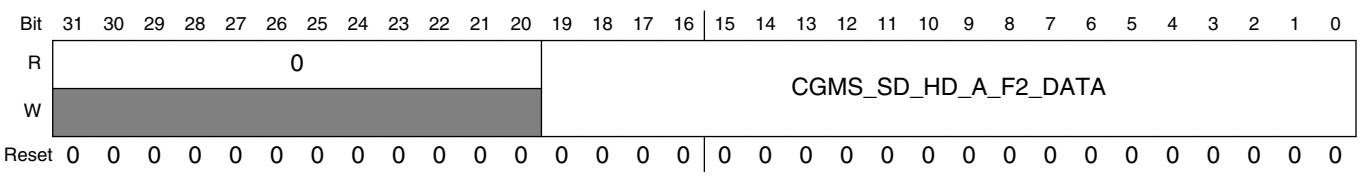


TVE_VBI_DATA_REG_0 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–0 CGMS_SD_HD_A_F1_DATA	CGMS data (in SD mode) or CGMS Type A data (in HD mode) for field 1.

74.3.17 VBI Data Register 1 (TVE_VBI_DATA_REG_1)

Address: TVE_VBI_DATA_REG_1 is 63FF_0000h base + 40h offset = 63FF_0040h



TVE_VBI_DATA_REG_1 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value zero. Reserved.
19–0 CGMS_SD_HD_A_F2_DATA	CGMS data (in SD mode) or CGMS Type A data (in HD mode) for field 2.

74.3.18 VBI Data Register 2 (TVE_VBI_DATA_REG_2)

Address: TVE_VBI_DATA_REG_2 is 63FF_0000h base + 44h offset = 63FF_0044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CC_SD_CGMS_HD_B_F1_DATA_0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_VBI_DATA_REG_2 field descriptions

Field	Description
31–0 CC_SD_CGMS_HD_B_F1_DATA_0	Closed Caption data (in SD mode) or CGMS Type B word 0 data (in HD mode) for field 1. In SD mode only bits CC_SD_CGMS_HD_B_F1_DATA_0[14:8] and CC_SD_CGMS_HD_B_F1_DATA_0[6:0] can be used respectively as MSB and LSB parts of the Closed Caption data.

74.3.19 VBI Data Register 3 (TVE_VBI_DATA_REG_3)

Address: TVE_VBI_DATA_REG_3 is 63FF_0000h base + 48h offset = 63FF_0048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WSS_SD_CGMS_HD_B_F1_DATA_1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_VBI_DATA_REG_3 field descriptions

Field	Description
31–0 WSS_SD_CGMS_HD_B_F1_DATA_1	Wide Screen Signaling data (in SD mode) or CGMS Type B word 1 data (in HD mode) for field 1. In SD mode only bits WSS_SD_CGMS_HD_B_F1_DATA_1[19:0] are used as the WSS data.

74.3.20 VBI Data Register 4 (TVE_VBI_DATA_REG_4)

Address: TVE_VBI_DATA_REG_4 is 63FF_0000h base + 4Ch offset = 63FF_004Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CGMS_HD_B_F1_DATA_2																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_VBI_DATA_REG_4 field descriptions

Field	Description
31–0 CGMS_HD_B_ F1_DATA_2	CGMS Type B word 2 data (in HD mode) for field 1.

74.3.21 VBI Data Register 5 (TVE_VBI_DATA_REG_5)

Address: TVE_VBI_DATA_REG_5 is 63FF_0000h base + 50h offset = 63FF_0050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CGMS_HD_B_F1_DATA_3																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_VBI_DATA_REG_5 field descriptions

Field	Description
31–0 CGMS_HD_B_ F1_DATA_3	CGMS Type B word 3 data (in HD mode) for field 1.

74.3.22 VBI Data Register 6 (TVE_VBI_DATA_REG_6)

Address: TVE_VBI_DATA_REG_6 is 63FF_0000h base + 54h offset = 63FF_0054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CC_SD_CGMS_HD_B_F2_DATA_0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_VBI_DATA_REG_6 field descriptions

Field	Description
31–0 CC_SD_CGMS_ HD_B_F2_ DATA_0	Closed Caption data (in SD mode) or CGMS Type B word 0 data (in HD mode) for field 2. In SD mode only bits CC_SD_CGMS_HD_B_F2_DATA_0[14:8] and CC_SD_CGMS_HD_B_F2_DATA_0[6:0] can be used respectively as MSB and LSB parts of the Closed Caption data.

74.3.23 VBI Data Register 7 (TVE_VBI_DATA_REG_7)

Address: TVE_VBI_DATA_REG_7 is 63FF_0000h base + 58h offset = 63FF_0058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CGMS_HD_B_F2_DATA_1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_VBI_DATA_REG_7 field descriptions

Field	Description
31–0 CGMS_HD_B_F2_DATA_1	CGMS Type B word 1 data (in HD mode) for field 2.

74.3.24 VBI Data Register 8 (TVE_VBI_DATA_REG_8)

Address: TVE_VBI_DATA_REG_8 is 63FF_0000h base + 5Ch offset = 63FF_005Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CGMS_HD_B_F2_DATA_2																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_VBI_DATA_REG_8 field descriptions

Field	Description
31–0 CGMS_HD_B_F2_DATA_2	CGMS Type B word 2 data (in HD mode) for field 2.

74.3.25 VBI Data Register 9 (TVE_VBI_DATA_REG_9)

Address: TVE_VBI_DATA_REG_9 is 63FF_0000h base + 60h offset = 63FF_0060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CGMS_HD_B_F2_DATA_3																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_VBI_DATA_REG_9 field descriptions

Field	Description
31–0 CGMS_HD_B_F2_DATA_3	CGMS Type B word 3 data (in HD mode) for field 2.

74.3.26 Interrupt Control Register (TVE_INT_CONT_REG)

Address: TVE_INT_CONT_REG is 63FF_0000h base + 64h offset = 63FF_0064h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SA_MEAS_END_IEN	TVE_FRAME_END_IEN	TVE_FIELD_END_IEN	CGMS_HD_B_F2_DONE_IEN	CGMS_HD_B_F1_DONE_IEN	CGMS_HD_A_F2_DONE_IEN	CGMS_HD_A_F1_DONE_IEN	WSS_SD_DONE_IEN	CGMS_SD_F2_DONE_IEN	CGMS_SD_F1_DONE_IEN	CC_SD_F2_DONE_IEN	CC_SD_F1_DONE_IEN	CD_MON_END_IEN	CD_SM_IEN	CD_LM_IEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_INT_CONT_REG field descriptions

Field	Description
31–15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14 SA_MEAS_END_IEN	Enable Luma statistic measurement end interrupt. 0 Disable 1 Enable
13 TVE_FRAME_END_IEN	Enable end-of-field interrupt. 0 Disable 1 Enable
12 TVE_FIELD_END_IEN	Enable end-of-frame interrupt. 0 Disable 1 Enable
11 CGMS_HD_B_F2_DONE_IEN	Enable HD CGMS Type B done interrupt for field 2. 0 Disable 1 Enable

Table continues on the next page...

TVE_INT_CONT_REG field descriptions (continued)

Field	Description
10 CGMS_HD_B_ F1_DONE_IEN	Enable HD CGMS Type B done interrupt for field 1. 0 Disable 1 Enable
9 CGMS_HD_A_ F2_DONE_IEN	Enable HD CGMS Type A done interrupt for field 2. 0 Disable 1 Enable
8 CGMS_HD_A_ F1_DONE_IEN	Enable HD CGMS Type A done interrupt for field 1. 0 Disable 1 Enable
7 WSS_SD_ DONE_IEN	Enable SD WSS done interrupt. 0 Disable 1 Enable
6 CGMS_SD_F2_ DONE_IEN	Enable SD CGMS done interrupt for field 2. 0 Disable 1 Enable
5 CGMS_SD_F1_ DONE_IEN	Enable SD CGMS done interrupt for field 1. 0 Disable 1 Enable
4 CC_SD_F2_ DONE_IEN	Enable SD closed caption done interrupt for field 2. 0 Disable 1 Enable
3 CC_SD_F1_ DONE_IEN	Enable SD closed caption done interrupt for field 1. 0 Disable 1 Enable
2 CD_MON_END_ IEN	Enable CD end-of-monitoring interrupt. 0 Disable 1 Enable
1 CD_SM_IEN	Enable CD short monitoring interrupt. 0 Disable 1 Enable
0 CD_LM_IEN	Enable CD load resistance monitoring interrupt. 0 Disable 1 Enable

74.3.27 Status Register (TVE_STAT_REG)

Address: TVE_STAT_REG is 63FF_0000h base + 68h offset = 63FF_0068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						BG_READY	CD_MAN_TRIG	0	CD_CH_2_SM_ST	CD_CH_1_SM_ST	CD_CH_0_SM_ST	0	CD_CH_2_LM_ST	CD_CH_1_LM_ST	CD_CH_0_LM_ST
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SA_MEAS_END_INT	TVE_FRAME_END_INT	TVE_FIELD_END_INT	CGMS_HD_B_F2_DONE_IEN	CGMS_HD_B_F1_DONE_IEN	CGMS_HD_A_F2_DONE_IEN	CGMS_HD_A_F1_DONE_IEN	WSS_SD_DONE_IEN	CGMS_SD_F2_DONE_IEN	CGMS_SD_F1_DONE_IEN	CC_SD_F2_DONE_IEN	CC_SD_F1_DONE_IEN	CD_MON_END_INT	CD_SM_INT	CD_LM_INT
W		w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_STAT_REG field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value zero. Reserved.
25 BG_READY	Status bit of bandgap reference of the TVDAC. The ready time is set by the BG_RDY_TIME parameter. 0 Bandgap reference is not ready. 1 Bandgap reference is ready
24 CD_MAN_TRIG	Control/status bit for manual (single shot) trigger of CD monitors. Set by the ARM platform, cleared by the TVE. Has no effect if CD_TRIG_MODE is 0. This bit can be set only if CD_EN=1. 0 No CD trigger 1 Trigger CD
23 Reserved	This read-only field is reserved and always has the value zero. Reserved.
22 CD_CH_2_SM_ST	Status bit of TVDAC channel #2 short monitoring by CD. Valid only if CD_CH_2_SM_EN is set. 0 Short detected 1 No short detected
21 CD_CH_1_SM_ST	Status bit of TVDAC channel #1 short monitoring by CD. Valid only if CD_CH_1_SM_EN is set.

Table continues on the next page...

TVE_STAT_REG field descriptions (continued)

Field	Description
	0 Short detected 1 No short detected
20 CD_CH_0_SM_ST	Status bit of TVDAC channel #0 short monitoring by CD. Valid only if CD_CH_0_SM_EN is set. 0 Short detected 1 No short detected
19 Reserved	This read-only field is reserved and always has the value zero. Reserved.
18 CD_CH_2_LM_ST	Status bit of TVDAC channel #2 load impedance monitoring by CD. Valid only if CD_CH_2_LM_EN is set. 0 Cable detected 1 No cable detected
17 CD_CH_1_LM_ST	Status bit of TVDAC channel #1 load impedance monitoring by CD. Valid only if CD_CH_1_LM_EN is set. 0 Cable detected 1 No cable detected
16 CD_CH_0_LM_ST	Status bit of TVDAC channel #0 load impedance monitoring by CD. Valid only if CD_CH_0_LM_EN is set. 0 Cable detected 1 No cable detected
15 Reserved	This read-only field is reserved and always has the value zero. Reserved.
14 SA_MEAS_END_INT	Status bit of the Luma statistic measurement end interrupt. 0 No statistic measurement end event has occurred 1 Statistic measurement end event has occurred
13 TVE_FRAME_END_INT	Status bit of the end-of-frame interrupt. 0 No end-of-frame event has occurred 1 End-of-frame event has occurred
12 TVE_FIELD_END_INT	Status bit of the end-of-field interrupt. 0 No end-of-field event has occurred 1 End-of-field event has occurred
11 CGMS_HD_B_F2_DONE_IEN	Status bit of the HD CGMS Type B done interrupt for field 2. 0 No HD CGMS Type B done 1 HD CGMS Type B done
10 CGMS_HD_B_F1_DONE_IEN	Status bit of the HD CGMS Type B done interrupt for field 1. 0 No HD CGMS Type B done 1 HD CGMS Type B done
9 CGMS_HD_A_F2_DONE_IEN	Status bit of the HD CGMS Type A done interrupt for field 2. 0 No HD CGMS Type B done 1 HD CGMS Type B done

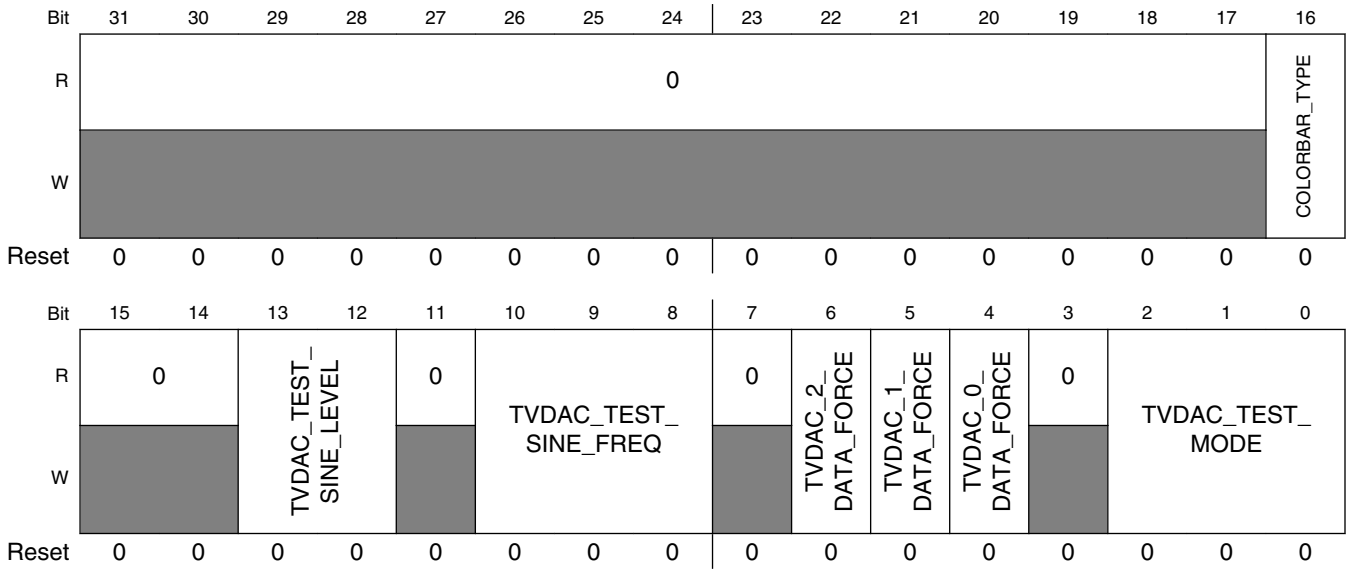
Table continues on the next page...

TVE_STAT_REG field descriptions (continued)

Field	Description
8 CGMS_HD_A_F1_DONE_IEN	Status bit of the HD CGMS Type A done interrupt for field 1. 0 No HD CGMS Type B done 1 HD CGMS Type B done
7 WSS_SD_DONE_IEN	Status bit of the SD WSS done interrupt. 0 No SD WSS done 1 SD WSS done
6 CGMS_SD_F2_DONE_IEN	Status bit of the SD CGMS done interrupt for field 2. 0 No SD CGMS done 1 SD CGMS done
5 CGMS_SD_F1_DONE_IEN	Status bit of the SD CGMS done interrupt for field 1. 0 No SD CGMS done 1 SD CGMS done
4 CC_SD_F2_DONE_IEN	Status bit of the SD Closed Caption done interrupt for field 2. 0 No SD Closed Caption done 1 SD Closed Caption done
3 CC_SD_F1_DONE_IEN	Status bit of the SD Closed Caption done interrupt for field 1. 0 No SD Closed Caption done 1 SD Closed Caption done
2 CD_MON_END_INT	Status bit of the end-of-monitoring interrupt by CD. 0 No end-of-monitoring event has occurred 1 End-of-monitoring event has occurred
1 CD_SM_INT	Status bit of the short monitoring interrupt by CD. 0 No short detected 1 Short detected
0 CD_LM_INT	Status bit of the load impedance monitoring interrupt by CD. 0 No mismatch between current output mode and real output load configuration 1 Current output mode does not match real output load configuration

74.3.28 Test Mode Register (TVE_TST_MODE_REG)

Address: TVE_TST_MODE_REG is 63FF_0000h base + 6Ch offset = 63FF_006Ch



TVE_TST_MODE_REG field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value zero. Reserved.
16 COLORBAR_TYPE	Select the type of internally generated color bar. 0 100% 1 75%
15–14 Reserved	This read-only field is reserved and always has the value zero. Reserved.
13–12 TVDAC_TEST_SINE_LEVEL	Select level of internally generated sine wave pattern. 00 Sine wave swing is 100% of TVDAC full scale (from 1 to 1023) 01 Sine wave swing is 75% of TVDAC full scale 10 Sine wave swing is 50% of TVDAC full scale 11 Sine wave swing is 25% of TVDAC full scale
11 Reserved	This read-only field is reserved and always has the value zero. Reserved.
10–8 TVDAC_TEST_SINE_FREQ	Select frequency of internally generated sine wave pattern. Defined ratio (division factor) between the TVDAC sampling frequency and the test sine wave frequency. 000 8 001 16 010 32 011 64 100 128

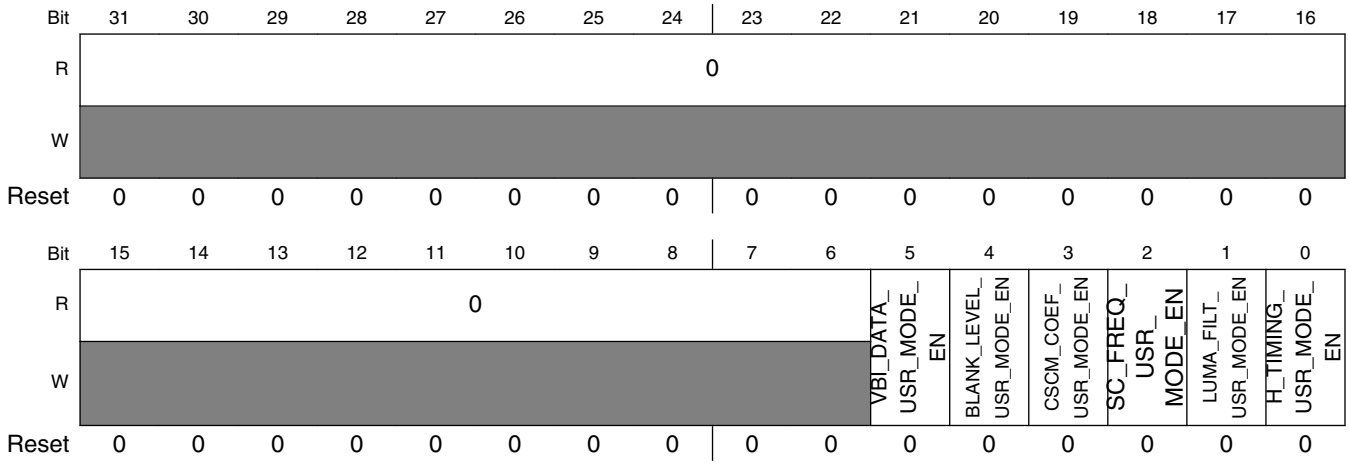
Table continues on the next page...

TVE_TST_MODE_REG field descriptions (continued)

Field	Description
	101 256 110 Reserved 111 Reserved
7 Reserved	This read-only field is reserved and always has the value zero. Reserved.
6 TVDAC_2_ DATA_FORCE	Enable forcing input of TVDAC channel #2 by value defined in BLANKING_CH_2_USR. 0 Disable 1 Enable
5 TVDAC_1_ DATA_FORCE	Enable forcing input of TVDAC channel #1 by value defined in BLANKING_CH_1_USR. 0 Disable 1 Enable
4 TVDAC_0_ DATA_FORCE	Enable forcing input of TVDAC channel #0 by value defined in BLANKING_CH_0_USR. 0 Disable 1 Enable
3 Reserved	This read-only field is reserved and always has the value zero. Reserved.
2–0 TVDAC_TEST_ MODE	Select TVDAC test mode. 000 Disable test mode (functional mode) 001 TVDAC test mode 1 - different data on TVDAC channels and UF is in data path 010 TVDAC test mode 2 - equal data on TVDAC channels and UF is in data path 011 TVDAC test mode 3 - different data on TVDAC channels and UF is bypassed 100 TVDAC test mode 4 - equal data on TVDAC channels and UF is bypassed 101 TVDAC test mode 5 - sine wave data from the internal generator directly to all TVDAC channels 110 Reserved 111 Reserved

74.3.29 User Mode Control Register (TVE_USER_MODE_CONT_REG)

Address: TVE_USER_MODE_CONT_REG is 63FF_0000h base + 70h offset = 63FF_0070h

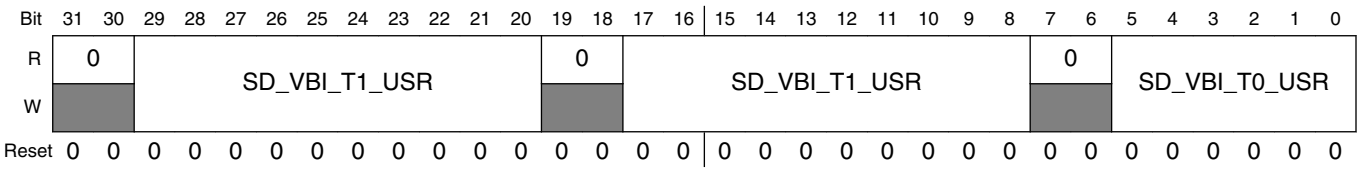


TVE_USER_MODE_CONT_REG field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero. Reserved.
5 VBI_DATA_ USR_MODE_EN	Enable user mode for VBI data. 0 Disable 1 Enable
4 BLANK_LEVEL_ USR_MODE_EN	Enable user mode for blank level. 0 Disable 1 Enable
3 CSCM_COEF_ USR_MODE_EN	Enable user mode for color space conversion matrix. 0 Disable 1 Enable
2 SC_FREQ_ USR_MODE_EN	Enable user mode for subcarrier frequency. 0 Disable 1 Enable
1 LUMA_FILT_ USR_MODE_EN	Enable user mode for Luma filtering. 0 Disable 1 Enable
0 H_TIMING_ USR_MODE_EN	Enable user mode for horizontal timing. 0 Disable 1 Enable

74.3.30 SD Timing User Control Register 0 (TVE_SD_TIMING_USR_CONT_REG_0)

Address: TVE_SD_TIMING_USR_CONT_REG_0 is 63FF_0000h base + 74h offset = 63FF_0074h

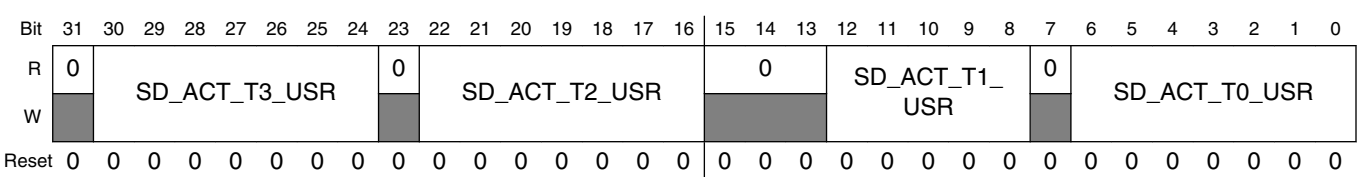


TVE_SD_TIMING_USR_CONT_REG_0 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved.
29–20 SD_VBI_T1_ USR	User defined horizontal timing interval t2 for VBI lines in SD mode. 0000000000 1 pixels 0000000001 2 pixels 1111111111 1024 pixels
19–18 Reserved	This read-only field is reserved and always has the value zero. Reserved.
17–8 SD_VBI_T1_ USR	User defined horizontal timing interval t1 for VBI lines in SD mode. 0000000000 1 pixels 0000000001 2 pixels 1111111111 1024 pixels
7–6 Reserved	This read-only field is reserved and always has the value zero. Reserved.
5–0 SD_VBI_T0_ USR	User defined horizontal timing interval t0 for VBI lines in SD mode. 000000 1 pixels 000001 2 pixels 111111 64 pixels

74.3.31 SD Timing User Control Register 1 (TVE_SD_TIMING_USR_CONT_REG_1)

Address: TVE_SD_TIMING_USR_CONT_REG_1 is 63FF_0000h base + 78h offset = 63FF_0078h

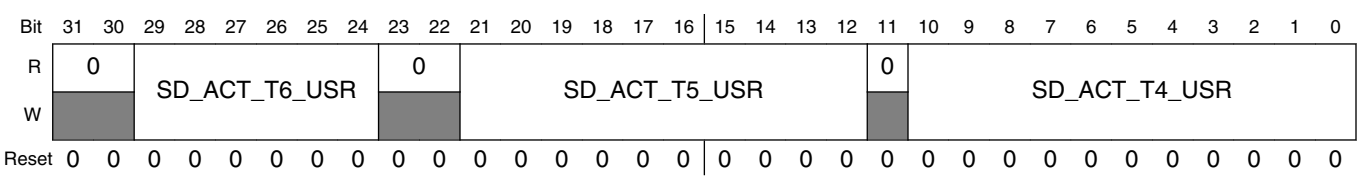


TVE_SD_TIMING_USR_CONT_REG_1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–24 SD_ACT_T3_USR	User defined horizontal timing interval t3 for active data lines in SD mode. 0000000 1 pixels 0000001 2 pixels 1111111 128 pixels
23 Reserved	This read-only field is reserved and always has the value zero. Reserved.
22–16 SD_ACT_T2_USR	User defined horizontal timing interval t2 for active data lines in SD mode. 0000000 1 pixels 0000001 2 pixels 1111111 128 pixels
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved.
12–8 SD_ACT_T1_USR	User defined horizontal timing interval t1 for active data lines in SD mode. 00000 1 pixels 00001 2 pixels 11111 32 pixels
7 Reserved	This read-only field is reserved and always has the value zero. Reserved.
6–0 SD_ACT_T0_USR	User defined horizontal timing interval t0 for active data lines in SD mode. 0000000 1 pixels 0000001 2 pixels 1111111 128 pixels

74.3.32 SD Timing User Control Register 2 (TVE_SD_TIMING_USR_CONT_REG_2)

Address: TVE_SD_TIMING_USR_CONT_REG_2 is 63FF_0000h base + 7Ch offset = 63FF_007Ch



TVE_SD_TIMING_USR_CONT_REG_2 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved.

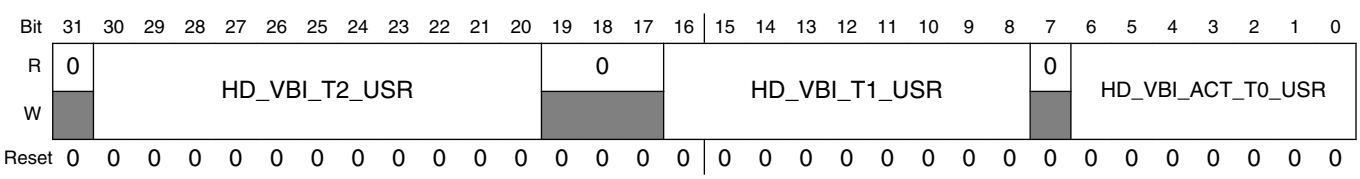
Table continues on the next page...

TVE_SD_TIMING_USR_CONT_REG_2 field descriptions (continued)

Field	Description
29–24 SD_ACT_T6_ USR	User defined horizontal timing interval t6 for active data lines in SD mode. 000000 1 pixels 000001 2 pixels 111111 64 pixels
23–22 Reserved	This read-only field is reserved and always has the value zero. Reserved.
21–12 SD_ACT_T5_ USR	User defined horizontal timing interval t5 for active data lines in SD mode. 0000000000 1 pixels 0000000001 2 pixels 1111111111 1024 pixels
11 Reserved	This read-only field is reserved and always has the value zero. Reserved.
10–0 SD_ACT_T4_ USR	User defined horizontal timing interval t4 for active data lines in SD mode. 00000000000 1 pixels 00000000001 2 pixels 11111111111 2048 pixels

74.3.33 HD Timing User Control Register 0 (TVE_HD_TIMING_USR_CONT_REG_0)

Address: TVE_HD_TIMING_USR_CONT_REG_0 is 63FF_0000h base + 80h offset = 63FF_0080h



TVE_HD_TIMING_USR_CONT_REG_0 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero. Reserved.
30–20 HD_VBI_T2_ USR	User defined horizontal timing interval t2 for VBI lines in HD mode. 00000000000 1 pixels 00000000001 2 pixels 11111111111 2048 pixels
19–17 Reserved	This read-only field is reserved and always has the value zero. Reserved.

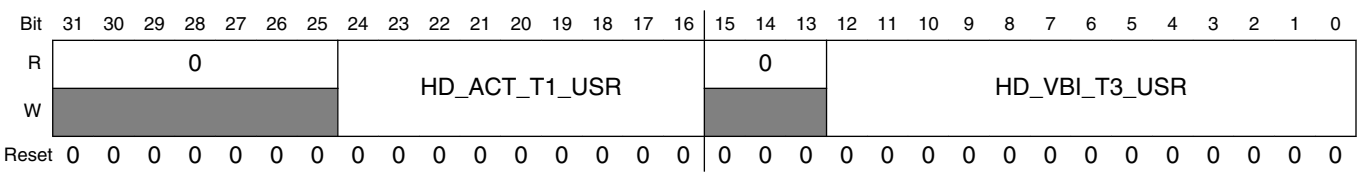
Table continues on the next page...

TVE_HD_TIMING_USR_CONT_REG_0 field descriptions (continued)

Field	Description
16–8 HD_VBI_T1_ USR	User defined horizontal timing interval t1 for VBI lines in HD mode. 000000000 1 pixels 000000001 2 pixels 111111111 512 pixels
7 Reserved	This read-only field is reserved and always has the value zero. Reserved.
6–0 HD_VBI_ACT_ T0_USR	User defined horizontal timing interval t0 for both VBI and active data lines in HD mode. 0000000 1 pixels 0000001 2 pixels 1111111 128 pixels

74.3.34 HD Timing User Control Register 1 (TVE_HD_TIMING_USR_CONT_REG_1)

Address: TVE_HD_TIMING_USR_CONT_REG_1 is 63FF_0000h base + 84h offset = 63FF_0084h

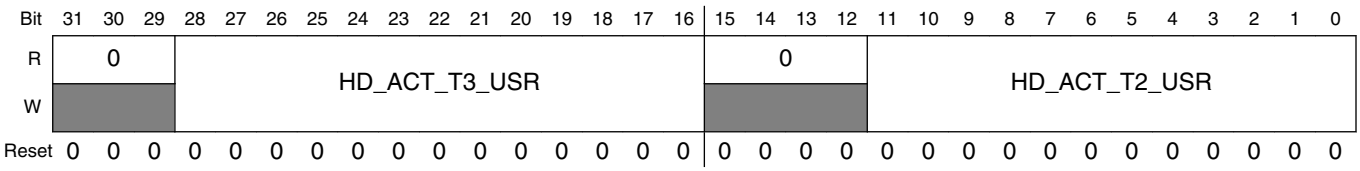


TVE_HD_TIMING_USR_CONT_REG_1 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero. Reserved.
24–16 HD_ACT_T1_ USR	User defined horizontal timing interval t1 for active data lines in HD mode. 000000000 1 pixels 000000001 2 pixels 111111111 512 pixels
15–13 Reserved	This read-only field is reserved and always has the value zero. Reserved.
12–0 HD_VBI_T3_ USR	User defined horizontal timing interval t3 for VBI lines in HD mode. 000000000000 1 pixels 000000000001 2 pixels 111111111111 4096 pixels

74.3.35 HD Timing User Control Register 2 (TVE_HD_TIMING_USR_CONT_REG_2)

Address: TVE_HD_TIMING_USR_CONT_REG_2 is 63FF_0000h base + 88h offset = 63FF_0088h

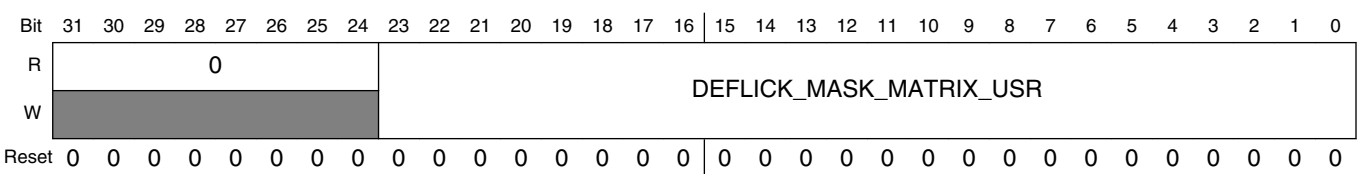


TVE_HD_TIMING_USR_CONT_REG_2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–16 HD_ACT_T3_USR	User defined horizontal timing interval t1 for active data lines in HD mode. 0000000000000 1 pixels 0000000000001 2 pixels 1111111111111 8192 pixels
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved.
11–0 HD_ACT_T2_USR	User defined horizontal timing interval t3 for active data lines in HD mode. 000000000000 1 pixels 000000000001 2 pixels 111111111111 4096 pixels

74.3.36 Luma User Control Register 0 (TVE_LUMA_USR_CONT_REG_0)

Address: TVE_LUMA_USR_CONT_REG_0 is 63FF_0000h base + 8Ch offset = 63FF_008Ch



TVE_LUMA_USR_CONT_REG_0 field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

TVE_LUMA_USR_CONT_REG_0 field descriptions (continued)

Field	Description
23–0 DEFLICK_ MASK_ MATRIX_USR	User defined decision matrix for vertical deflickering/ fine sharpening/high-frequency noise reduction luma filter.

74.3.37 Luma User Control Register 1 (TVE_LUMA_USR_CONT_REG_1)

Address: TVE_LUMA_USR_CONT_REG_1 is 63FF_0000h base + 90h offset = 63FF_0090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								V_SHARP_MASK_MATRIX_USR																									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_LUMA_USR_CONT_REG_1 field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved.
23–0 V_SHARP_ MASK_ MATRIX_USR	User defined decision matrix for vertical coarse sharpening/mid-frequency noise reduction luma filter.

74.3.38 Luma User Control Register 2 (TVE_LUMA_USR_CONT_REG_2)

Address: TVE_LUMA_USR_CONT_REG_2 is 63FF_0000h base + 94h offset = 63FF_0094h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								H_SHARP_MASK_MATRIX_USR																									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_LUMA_USR_CONT_REG_2 field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved.

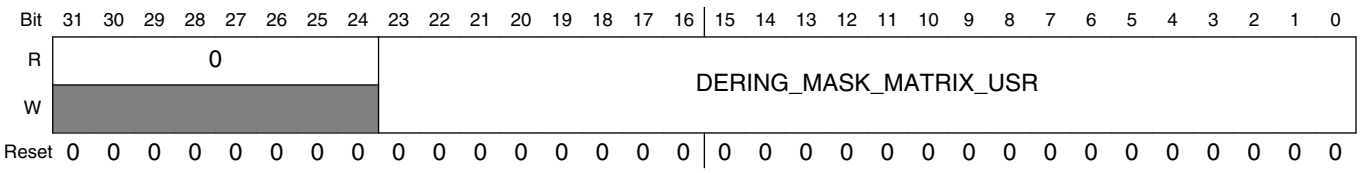
Table continues on the next page...

TVE_LUMA_USR_CONT_REG_2 field descriptions (continued)

Field	Description
23–0 H_SHARP_ MASK_ MATRIX_USR	User defined decision matrix for horizontal coarse sharpening/mid-frequency noise reduction luma filter.

74.3.39 Luma User Control Register 3 (TVE_LUMA_USR_CONT_REG_3)

Address: TVE_LUMA_USR_CONT_REG_3 is 63FF_0000h base + 98h offset = 63FF_0098h

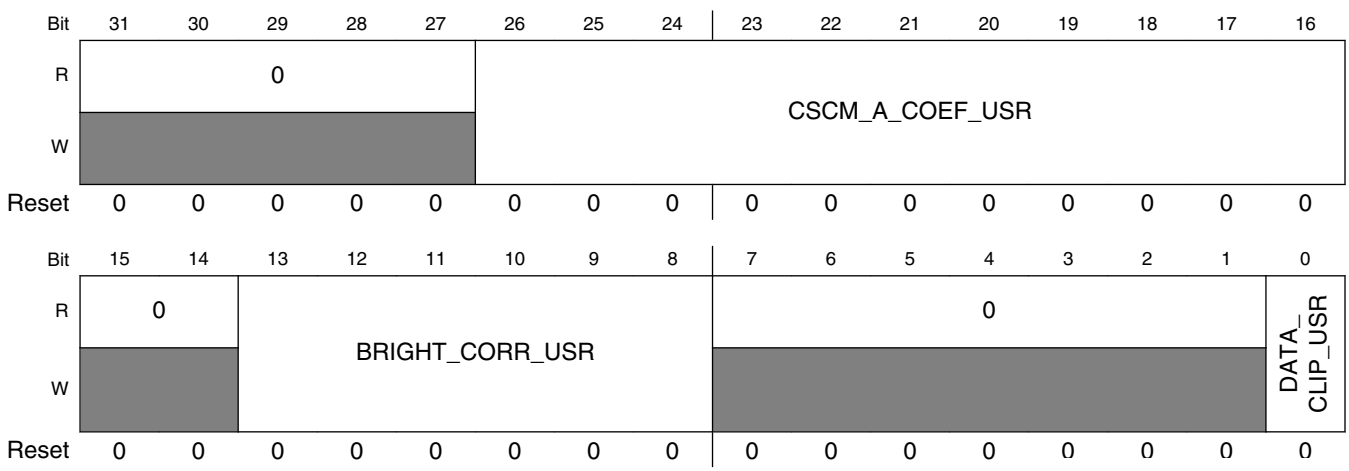


TVE_LUMA_USR_CONT_REG_3 field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved.
23–0 DERING_ MASK_ MATRIX_USR	User defined decision matrix for horizontal deringing/ fine sharpening/high-frequency noise reduction luma filter.

74.3.40 Color Space Conversion User Control Register 0 (TVE_CSC_USR_CONT_REG_0)

Address: TVE_CSC_USR_CONT_REG_0 is 63FF_0000h base + 9Ch offset = 63FF_009Ch



TVE_CSC_USR_CONT_REG_0 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26–16 CSCM_A_ COEF_USR	User defined CSC coefficient A. Represented in unsigned format with 10 fractional bits and 1 integer bit. 00000000000 0 00000000001 1/1024 11111111111 1+1023/1024
15–14 Reserved	This read-only field is reserved and always has the value zero. Reserved.
13–8 BRIGHT_ CORR_USR	User defined brightness correction offset in two's complement format. 100000 32 100001 31 000000 0 011111 31
7–1 Reserved	This read-only field is reserved and always has the value zero. Reserved.
0 DATA_CLIP_ USR	User defined lower clipping level for CSC output data. Relevant only for Luma, composite or RGB ouputs. 0 128 1 64

74.3.41 Color Space Conversion User Control Register 1 (TVE_CSC_USR_CONT_REG_1)

Address: TVE_CSC_USR_CONT_REG_1 is 63FF_0000h base + A0h offset = 63FF_00A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					CSCM_C_COEF_USR											0				CSCM_B_COEF_USR											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_CSC_USR_CONT_REG_1 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26–16 CSCM_C_ COEF_USR	User defined CSC coefficient C. Represented in signed two's complement format with 10 fractional bits and 1 sign bit. 10000000000 1 10000000001 1+1/1024 11111111111 1/1024 00000000000 0

Table continues on the next page...

TVE_CSC_USR_CONT_REG_1 field descriptions (continued)

Field	Description
	00000000001 1/1024 01111111111 1023/1024
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved.
11–0 CSCM_B_ COEF_USR	User defined CSC coefficient B. Represented in signed two's complement format with 10 fractional bits, 1 integer bit and 1 sign bit. 100000000000 2 100000000001 2+1/1024 111111111111 1/1024 000000000000 0 000000000001 1/1024 011111111111 1+1023/1024

74.3.42 Color Space Conversion User Control Register 2 (TVE_CSC_USR_CONT_REG_2)

Address: TVE_CSC_USR_CONT_REG_2 is 63FF_0000h base + A4h offset = 63FF_00A4h

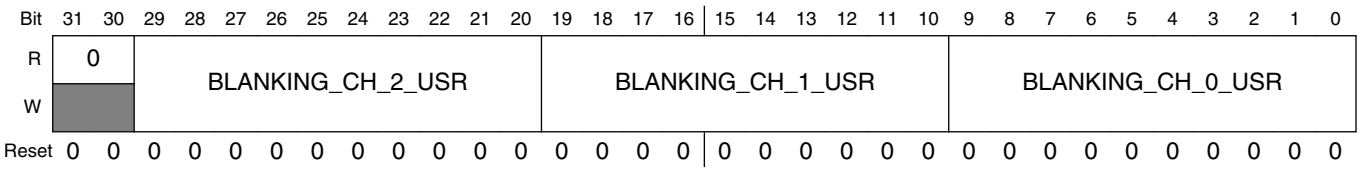
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			CSCM_E_COEF_USR													0				CSCM_D_COEF_USR											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_CSC_USR_CONT_REG_2 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–16 CSCM_E_ COEF_USR	User defined CSC coefficient E. Represented in unsigned format with 10 fractional bits and 3 integer bits. 0000000000000 0 0000000000001 1/1024 1111111111111 7+1023/1024
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved.
11–0 CSCM_D_ COEF_USR	User defined CSC coefficient D. Represented in unsigned format with 10 fractional bits and 2 integer bits. 000000000000 0 000000000001 1/1024 111111111111 3+1023/1024

74.3.43 Blanking Level User Control Register (TVE_BLANK_USR_CONT_REG)

Address: TVE_BLANK_USR_CONT_REG is 63FF_0000h base + A8h offset = 63FF_00A8h

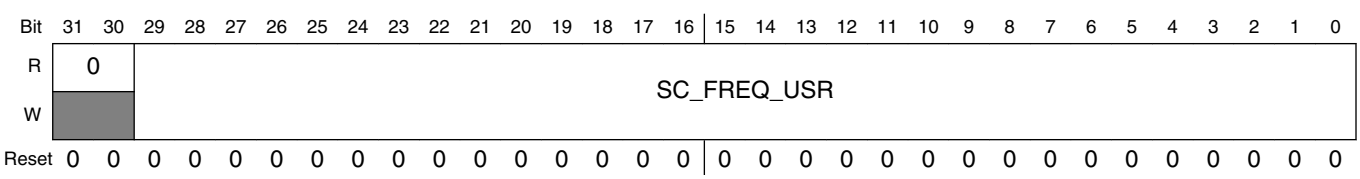


TVE_BLANK_USR_CONT_REG field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved.
29–20 BLANKING_CH_2_USR	User defined blanking level for channel #2. 0000000000 0 0000000001 1 1111111111 1023
19–10 BLANKING_CH_1_USR	User defined blanking level for channel #1. 0000000000 0 0000000001 1 1111111111 1023
9–0 BLANKING_CH_0_USR	User defined blanking level for channel #0. 0000000000 0 0000000001 1 1111111111 1023

74.3.44 SD Modulation User Control Register (TVE_SD_MOD_USR_CONT_REG)

Address: TVE_SD_MOD_USR_CONT_REG is 63FF_0000h base + ACh offset = 63FF_00ACh



TVE_SD_MOD_USR_CONT_REG field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero. Reserved.
29–0 SC_FREQ_USR	User defined subcarrier frequency factor. The real frequency is SC_FREQ_USR*27/2^{30} MHz.

74.3.45 VBI Data User Control Register 0 (TVE_VBI_DATA_USR_CONT_REG_0)

Address: TVE_VBI_DATA_USR_CONT_REG_0 is 63FF_0000h base + B0h offset = 63FF_00B0h

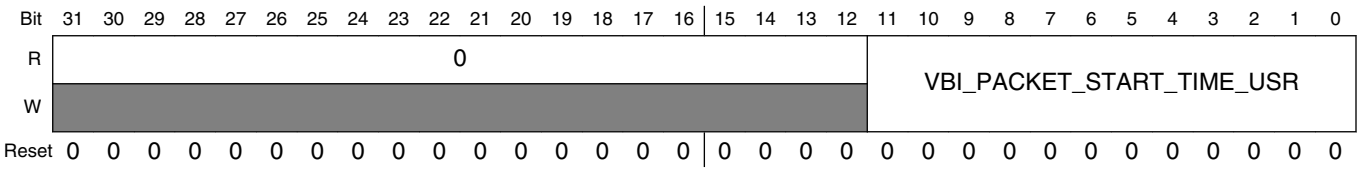
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				VBI_DATA_STOP_TIME_USR												0				VBI_DATA_START_TIME_USR											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_VBI_DATA_USR_CONT_REG_0 field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero. Reserved.
27–16 VBI_DATA_ STOP_TIME_ USR	User defined stop time for switching from VBI data mode to video mode. Defined relative to line start point. 000000000000 1 pixels 000000000001 2 pixels 111111111111 4096 pixels
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved.
11–0 VBI_DATA_ START_TIME_ USR	User defined start time for switching from video mode to VBI data mode. Defined relative to line start point. 000000000000 1 pixels 000000000001 2 pixels 111111111111 4096 pixels

74.3.46 VBI Data User Control Register 1 (TVE_VBI_DATA_USR_CONT_REG_1)

Address: TVE_VBI_DATA_USR_CONT_REG_1 is 63FF_0000h base + B4h offset = 63FF_00B4h

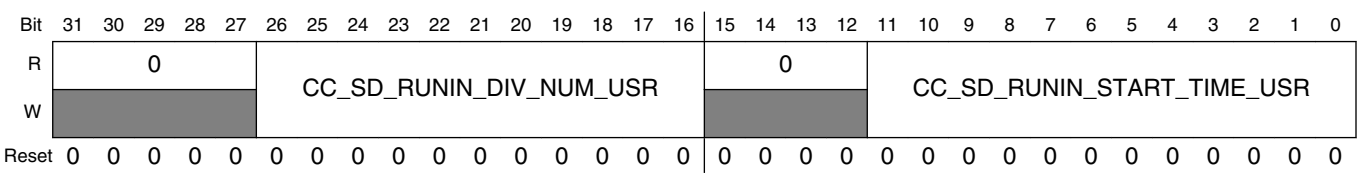


TVE_VBI_DATA_USR_CONT_REG_1 field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero. Reserved.
11–0 VBI_PACKET_START_TIME_USR	User defined VBI data packet start time. Defined relative to line start point. 000000000000 1 pixels 000000000001 2 pixels 111111111111 4096 pixels

74.3.47 VBI Data User Control Register 2 (TVE_VBI_DATA_USR_CONT_REG_2)

Address: TVE_VBI_DATA_USR_CONT_REG_2 is 63FF_0000h base + B8h offset = 63FF_00B8h



TVE_VBI_DATA_USR_CONT_REG_2 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero. Reserved.
26–16 CC_SD_RUNIN_DIV_NUM_USR	User defined numerator of the run-in frequency division factor for Closed Caption in SD mode. 00000000000 0 00000000001 1 11111111111 2047
15–12 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

TVE_VBI_DATA_USR_CONT_REG_2 field descriptions (continued)

Field	Description
11–0 CC_SD_RUNIN_START_TIME_USR	User defined run-in sequence start time for Closed Caption in SD mode. Defined relative to line start point. 000000000000 1 pixels 000000000001 2 pixels 111111111111 4096 pixels

74.3.48 VBI Data User Control Register 3 (TVE_VBI_DATA_USR_CONT_REG_3)

Address: TVE_VBI_DATA_USR_CONT_REG_3 is 63FF_0000h base + BCh offset = 63FF_00BCh

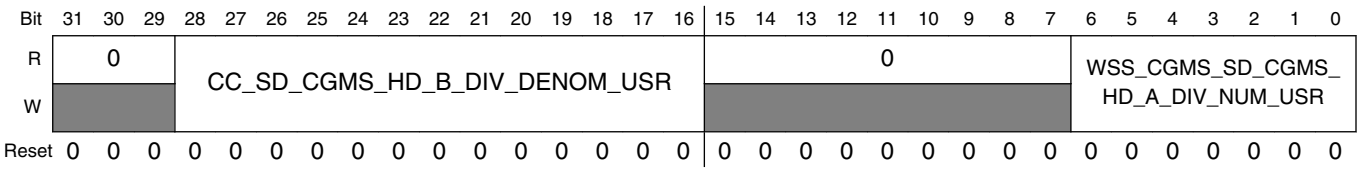
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			CC_SD_CGMS_HD_B_DIV_DENOM_USR													0									CC_SD_CGMS_HD_B_DIV_NUM_USR							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TVE_VBI_DATA_USR_CONT_REG_3 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–16 CC_SD_CGMS_HD_B_DIV_DENOM_USR	User defined denominator of the data rate division factor for Closed Caption in SD mode and CGMS Type B in HD mode. 0000000000000 0 0000000000001 1 1111111111111 8191
15–7 Reserved	This read-only field is reserved and always has the value zero. Reserved.
6–0 CC_SD_CGMS_HD_B_DIV_NUM_USR	User defined numerator of the data rate division factor for Closed Caption in SD mode and CGMS Type B in HD mode. 0000000 0 0000001 1 1111111 127

74.3.49 VBI Data User Control Register 4 (TVE_VBI_DATA_USR_CONT_REG_4)

Address: TVE_VBI_DATA_USR_CONT_REG_4 is 63FF_0000h base + C0h offset = 63FF_00C0h

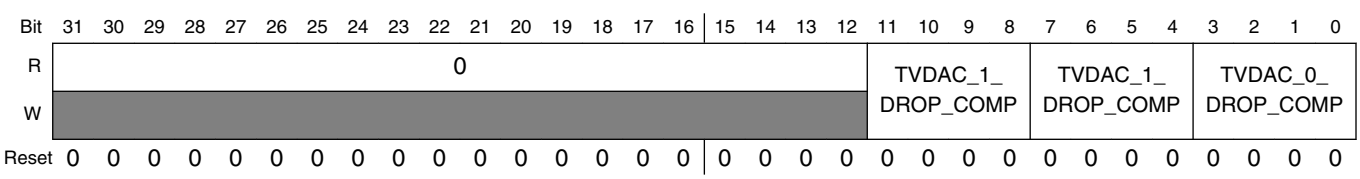


TVE_VBI_DATA_USR_CONT_REG_4 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero. Reserved.
28–16 CC_SD_CGMS_HD_B_DIV_DENOM_USR	User defined denominator of the data rate division factor for WSS/CGMS in SD mode and CGMS Type A in HD mode. 0000000000000 0 0000000000001 1 1111111111111 8191
15–7 Reserved	This read-only field is reserved and always has the value zero. Reserved.
6–0 WSS_CGMS_SD_CGMS_HD_A_DIV_NUM_USR	User defined numerator of the data rate division factor for WSS/CGMS in SD mode and CGMS Type A in HD mode. 0000000 0 0000001 1 1111111 127

74.3.50 Drop Compensation User Control Register (TVE_DROP_COMP_USR_CONT_REG)

Address: TVE_DROP_COMP_USR_CONT_REG is 63FF_0000h base + C4h offset = 63FF_00C4h



TVE_DROP_COMP_USR_CONT_REG field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero. Reserved.

Table continues on the next page...

TVE_DROP_COMP_USR_CONT_REG field descriptions (continued)

Field	Description
11–8 TVDAC_1_ DROP_COMP	User defined frequency response drop compensation coefficient K_{dc} for the TVDAC channel #2. 0000 0 0001 1/256 1111 15/256
7–4 TVDAC_1_ DROP_COMP	User defined frequency response drop compensation coefficient K_{dc} for the TVDAC channel #1. 0000 0 0001 1/256 1111 15/256
3–0 TVDAC_0_ DROP_COMP	User defined frequency response drop compensation coefficient K_{dc} for the TVDAC channel #0. 0000 0 0001 1/256 1111 15/256

Chapter 75
TrustZone Aware Interrupt Controller (TZIC)

75.1 Overview

The interrupt controller is a TrustZone enabled interrupt controller with an AXI interface. It allows complete and independent control over every interrupt connected to the controller, including enable, security, and priority functionality. It provides secure and non-secure transaction access to those interrupts, restricting non-secure read and write transactions to only those interrupts configured as non-secure and allowing secure transactions read and write capability to all interrupts regardless of security configuration. It supports priority masking, overall enable/disable of secure or nonsecure interrupts, and access to raw interrupt state and pending transactions.

NOTE

Chip-specific system interrupts are defined and described in the Interrupts and SDMA Events chapter.

NOTE

On i.MX53 board, power fail indication can be connected to the interrupt controller via GPIO_16 (ALT2) PAD. This can generate an interrupt (IRQ #7, as described in Table 3-1) if properly configured (priority, masking, etc.).

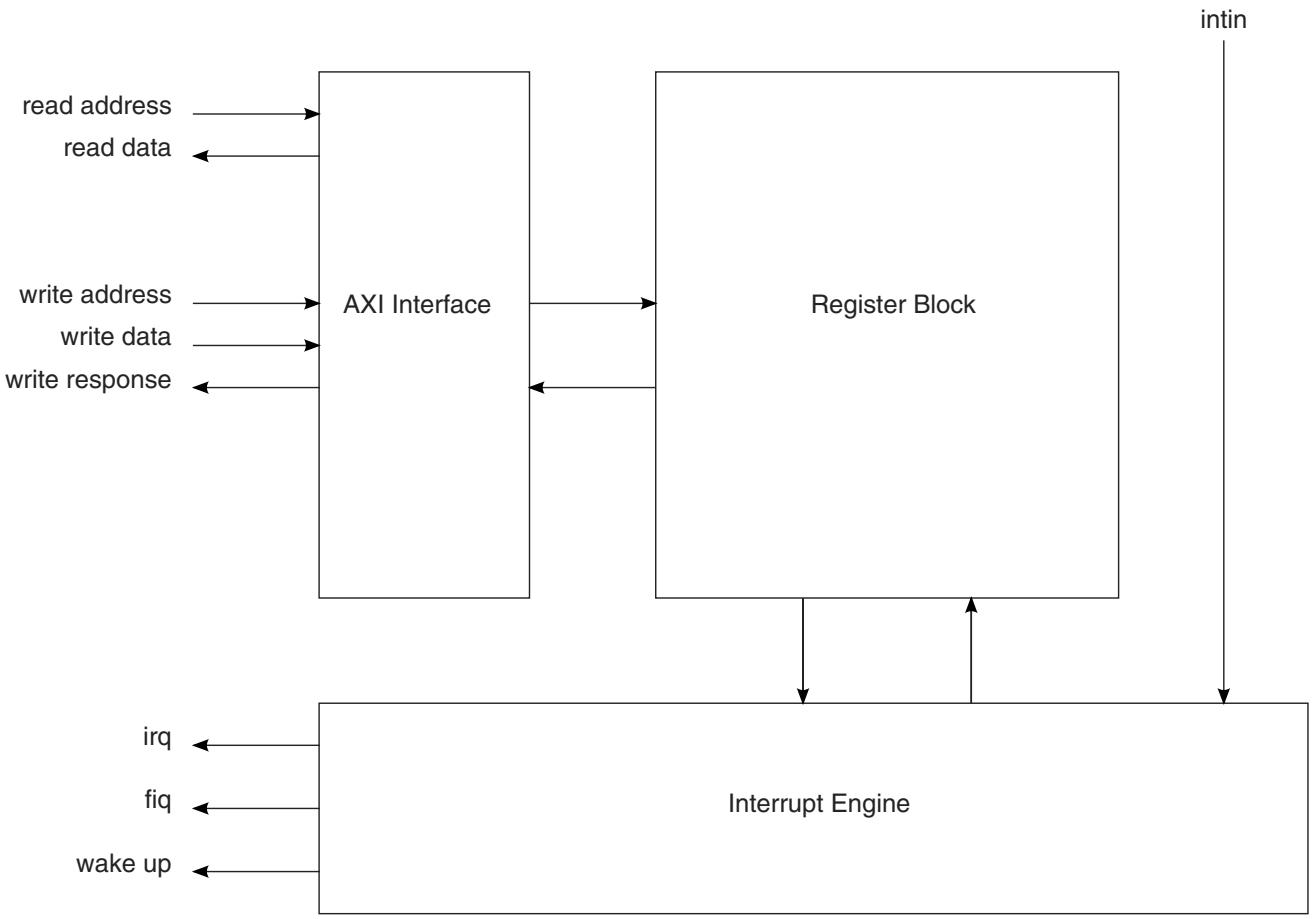


Figure 75-1. Block Diagram

75.2 Features

The following features are included in the interrupt controller

- AXI interface
- Support for up to 128 interrupts
- Secure configuration available on all interrupts
- Software triggered interrupts supported
- High priority pending registers for autovectored interrupt service routines
- Set/Clear registers for interrupt enable, force, and secure registers
- 16/32 programmable priority levels (nonsecure/secure)
- Wakeup configuration available on all interrupts

75.3 External Signal Description

The interrupt controller has no external signals.

75.4 Functional Description

The TZIC is an interrupt controllersupporting trustzone . This implementation allows up to 128 interrupts. It supports autovectoring solutions and provides a high priority pending register for software to directly access the interrupts currently pending at the highest active priority level (based on security configuration).

The interrupt controller is made up of three primary components:

1. The AXI Interface
2. The Register Block
3. The Interrupt Engine

75.4.1 Security Configurability

The TZIC is designed to handle two levels of secure interrupts using trustzone functionality included in the AXI specifications. It is also possible to use it in a single security level, non-trustzone aware system. At integration the TZIC can be connected such that all transactions are treated as secure, providing the ARM platform with full access to all registers. The TZIC_INTSEC registers reset to all interrupts being secure, which translates to all interrupts being available to the system without needing to be configured, therefore the non-secure software running as secure does not need to be aware of the security registers for configuration.

The TZIC_INTSEC registers determine which interrupts are accessible by non-secure transactions. Interrupts configured as secure are not accessible or configurable by non-secure transactions. Any information associated with an interrupt configured as secure is not accessible by non-secure transactions. Read transactions will return reset values for any bits tied to a secure interrupt, writes to these bits will be ignored. This includes all registers. The TZIC_INTSEC registers are not visible to non-secure transactions.

75.4.1.1 TrustZone and Interrupt Priority

The priority scheme implemented in the interrupt controller allows for flexible prioritization of secure and non-secure interrupts, and ease of use with legacy software.

Each interrupt can be assigned a priority value up to 8 bits (depending on the PRIORITY_MASK parameter). The highest priority value is 0, the next is 1, and so on. For designs where not all 8 bits are used, the lower bits are masked off, such that an implementation with only 5 bits would use values 0 (8'h00), 8 (8'h08), 16 (8'h10), 24 (8'h18), and so on. The number of bits available can be discovered by software through writing 8'hFF to a priority register and reading back the resulting value actually stored in the register.

Secure and non-secure transactions have different views of the priority registers. A secure transaction can see all implemented bits and has access to the full range of priority values. A non-secure transaction is limited to the lower half of the possible priorities. This is accomplished through shifting the values written and read during a non-secure transaction.

A non-secure write to a priority register will automatically be shifted one bit to the right, with the MSB assigned 1'b1. This shift is reversed when read, with the priority value shifted one bit to the left and the LSB assigned 1'b0. This is shown in the following equations:

Non-Secure write to priority register: priority_reg[7:0] = {1'b1, wdata[7:1]}

Non-Secure read from priority register: rdata[7:0] = {priority_reg[6:0], 1'b0}

This guarantees the highest priority a non-secure transaction can write is 128 (8'h80), which is the highest priority of the lower half of possible priorities. It also shows the non-secure software will not be aware of this limitation, because reading back a priority register with a value of 128 will result in a read data value of 0.

This scheme will allow for non-secure interrupts to function in their own priority scheme without knowledge of secure interrupt priorities while allowing secure interrupts to guarantee priority over non-secure interrupts. If all secure interrupts are assigned priorities in the upper half of the possible priorities, secure interrupts will always be of higher priority than non-secure interrupts.

Note that this priority value shifting is determined by the security of the transaction, not the security of the interrupt whose priority is being configured. A non-secure transaction will only have access to non-secure interrupt priority values, but a secure transaction will have access to secure and non-secure priority values, and will be accessing the actual value, not the shifted value. This means it is possible for a secure transaction to configure a non-secure interrupt priority to be in the upper half of the priority levels, or to configure a secure interrupt priority to be below a non-secure interrupt. In the first case, the non-

secure interrupt priority will remain set until written again by another secure transaction or a non-secure transaction (which would automatically limit the priority level to 128 or lower). A non-secure read will shift the value as specified above but would not alter the value. This is not recommended because it could impact the non-secure software in an unpredictable manner. In the second case, it may be desirable to configure a secure interrupt to be of lower priority than some non-secure interrupts, although such behavior should be used with care.

75.4.2 AXI Interface

The AXI interface allows access to all functionality of the interrupt controller. It is fully compatible with the AMBA 3 AXI specification from ARM.

Read transactions are generally returned with one initial wait state and two clock cycles per read data. All registers can be read with 32-bit transactions, the priority registers allow 8-bit transactions. 16-bit transactions are not supported and will result in an error. Exclusive reads are not supported. The interrupt controller will accept one read address at a time. If a second read transaction is valid before completion of the previous read, it will be held off until the interrupt controller is finished returning data from the first read.

Write transactions are generally accepted with one initial wait state, the one clock cycle per write data. Registers are word writeable only with the exception of the priority and priority mask registers which support byte writes. Exclusive writes are not supported. The interrupt controller will accept one write address at a time. If a second write transaction is valid before completion of the previous write, it will be held off until the interrupt controller is finished returning the write response from the first write. The interrupt controller will also hold off the write data bus until it receives a write address.

75.4.3 Interrupt Engine

The interrupt engine performs all processing on the raw interrupt information. It receives configuration information from the Register Block, interrupts from the system, and based on that information determines when to assert interrupts to the core.

An interrupt is asserted to the core (IRQ or FIQ) when all the following criteria are met:

- The EN bit in the TZIC_INTCTRL register is enabled, secure for FIQ or non-secure for normal IRQ.
- An interrupt is asserted, either on an interrupt input (intin) or by writing the TZIC_SRCSET or TZIC_SWINT registers.

- The interrupt is configured appropriately for the desired interrupt in the TZIC_INTSEC register, set as secure for FIQ or non-secure for IRQ.
- The asserted interrupt is enabled in the TZIC_ENSET/TZIC_ENCLEAR register.
- The priority of the asserted interrupt is greater than the TZIC_PRIOMASK register value. Note: this requires the actual priority value to be greater than the TZIC_PRIOMASK register value, after taking into account shifting for non-secure transactions.

This interrupt controller can handle up to 128 interrupts (intin). Each interrupt has an associated interrupt ID number 0 - 127.

The interrupt synchronizers are part of the interrupt engine. A two-flop synchronizer is implemented for each interrupt input. Clock gating is available for the synchronizers and is configured through the Synchronizer Control (TZIC_SYNCCTRL) Register. For more information see Synchronizer Control (TZIC_SYNCCTRL).

The interrupt engine handles the wakeup functionality of the interrupt controller, processing the dsm_int_holdoff signal input from the ARM platform and asserting tzic_wakeup_request to the clock controller module as necessary.

tzic_wakeup_request is an asynchronous, combinational output determined by the asynchronous interrupt sources (intin) and the TZIC_WAKEUP registers. It asserts when any interrupt is asserted on the intin input while configured for wakeup. It negates when there are no more interrupts asserted that are configured for wakeup. A software interrupt will not assert this output, because a software interrupt would require a clock to be triggered and the tzic_wakeup_request is intended for enabling clocks and/or power during low power states.

dsm_int_holdoff causes the interrupt synchronizers to stop updating. This is for shutting down clocks in the system. While preparing for low power modes, the ARM platform may be in a position where it expects no more interrupts to be occurring, but has not yet executed a standby-wait-for-interrupt. The ARM platform must write to the dsm_int_holdoff bit, then read it back to verify that it was successfully set. If not set, an interrupt occurred during the process before the synchronizers could be stopped and the ARM platform must process the interrupt and write to the dsm_int_holdoff bit again.

NOTE

Above internal decription is for information only, because the mechanisms are automated through the clock controller module when going into a low power mode.

75.4.4 Auto-Vectored Interrupt Handling

The TZIC is intended to be used with auto-vectored interrupt service routines. Once interrupts have been enabled using the TZIC_ENSET/TZIC_ENCLEAR registers and the TZIC_INTCTRL register, the interrupt engine is live and will signal an interrupt to the ARM platform when an enabled interrupt is asserted. Once the ARM platform interrupt is triggered, a read to the TZIC_PND or TZIC_HIPND registers will determine which interrupts are currently pending.

The TZIC_PND registers will contain information on every interrupt currently pending, whether asserted by the interrupt input or forced by software. This includes interrupts of priority lower than the TZIC_PRIOMASK value. The TZIC_PRIOMASK register limits the assertion of the IRQ/FIQ signals based on priority, it does not affect which interrupts are visible in the pending registers.

The TZIC_HIPND registers will contain information on which interrupts are currently pending at the highest pending priority, allowing autovectoring software to choose between the highest priority interrupts without having to process such information on its own. As with the TZIC_PNDPND register, the TZI.HIPND register will return pending values even if the highest pending priority is of equal or lower priority than the TZIC_PRIOMASK value, although in such a case the interrupt signals will not be asserted.

After choosing a pending interrupt to service, writing the priority of the interrupt being serviced to the TZIC_PRIOMASK register will allow for interrupt pre-emption by a higher priority interrupt being asserted later.

75.4.5 Reset

Coming out of reset, all interrupts are configured as secure, disabled, and set to priority level 0. Also, the TZIC_INTCTRL register is set to disable all secure and non-secure interrupts.

75.5 Initialization Information

Initialization procedures will vary based on whether the interrupt controller is set up in a one or two security domain configuration. This description will handle the case where the interrupt controller is set up to take advantage of both secure and non-secure interrupts, the single security level case is a subset of this scenario.

Note: When writing to any registers affecting the interrupt outputs to the ARM platform it is recommended the ARM platform interrupt inputs be masked to prevent interrupts asserting and negating due to register writes.

When the interrupt controller comes out of reset, all interrupts are disabled and set as secure. It is advised that the TZIC_INTCTRL register enables are not activated until all other configuration options have been set up. To set up the interrupt controller for autovectoring functionality:

1. Secure software sets TZIC_INTSEC registers appropriately - This configures all interrupts as either secure or non-secure, allowing the non-secure software to configure the non-secure interrupts.Until these registers are configured, non-secure software will not be able to configure any registers.

   At this point secure and non-secure software are free to configure their associated interrupts with minimal conflict. Non-secure software cannot affect any register bits associated with secure interrupts. It is the responsibility of the secure software not to overwrite any non-secure interrupt configuration bits.

2. Configure TZIC_ENSET/TZIC_ENCLEAR - Secure and non-secure software must enable all interrupts. At this point interrupt functionality has yet to be enabled in the TZIC_INTCTRL register so this will not cause interrupts to occur during initialization.
3. Configure TZIC_PRIORITY - Secure and non-secure software must set TZIC_PRIORITY registers to appropriate levels, otherwise all interrupts are set to the lowest priority level as determined by the PRIORITY_MASK parameter.
4. Set TZIC_PRIOMASK - The TZIC_PRIOMASK register must be set to allow interrupts to occur, out of reset it is set to the lowest priority level which masks all priority levels.
5. Set TZIC_INTCTRL - Secure and non-secure software must enable interrupt functionality in the TZIC_INTCTRL register. This will enable the interrupt controller to send interrupts to the ARM platform.

At this point the interrupt controller will generate interrupt signals to the ARM platform. The ARM platform itself must enable interrupts as well. In this configuration the interrupt controller is enabled for autovectoring software to access the TZIC_PND or TZIC_HIPND registers and determine which interrupt to service.

75.6 Programmable Registers

This is the memory map for all possible registers in the interrupt controller. The presence of some registers will be dependent upon the implementation configuration. Address space not specified is reserved.

This section contains the detailed register descriptions for the interrupt controler registers.

The read and write behavior of many of the interrupt controller regsiters is impacted by the security configuration of each individual interrupt, as determined by the INTSEC registers. In all cases, non-secure transactions will only read back information about interrupts configured as non-secure and writes will have no effect on secure interrupts. Secure transactions will have full read and write access to all interrupt bits.

Write bursts are taken with one clock per write data after an initial one clock delay. Read bursts are returned with two clocks per read data.

The specific behavior for each register set is described in the field descriptions. In some cases where the register described spans multiple 32-bit addressible locations, the variable X is used to denote which particular register corresponds to which interrupts. For instance, a register with 1 bit associated with each interrupt will have one 32-bit register for each set of 32 interrupts. The registers can be numbered such that REGISTER0 corresponds to interrupts 0-31, REGISTER1 corresponds to interrupts 32-63, and so on. This relationship can be expressed using the following equation:

REGISTER(X)[Y] corresponds to interrupt (32 x (X))+Y

Such that if X is 2 and Y is 17:

REGISTER(2)[17] -> (32 x (2))+17 = 81

The interrupt would be number 81.

This terminology is used to express the location and relationship for several registers in this section.

All registers are accessible with 32-bit accesses. The PRIOMASK register and PRIORITY registers support 8-bit transactions, all others are not supported and will yield unpredictable results. 16-bit transactions are not supported to any registers and will result in a slave error. Any access larger than 32-bit will also result in an error.

All registers are bufferable, non-cacheable, data, supervisor only. Transactions marked as cacheable, allocate, user, or instruction will result in error responses.

TZIC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0FFF_C000	Control Register (TZIC_INTCTRL)	32	R/W	0000_0000h	75.6.1/ 4607
0FFF_C004	Interrupt Controller Type Register (TZIC_INTTYPE)	32	R	0000_0403h	75.6.2/ 4609
0FFF_C00C	Priority Mask Register (TZIC_PRIOMASK)	32	R/W	0000_0000h	75.6.3/ 4610
0FFF_C010	Synchronizer Control (TZIC_SYNCCTRL)	32	R/W	0000_0000h	75.6.4/ 4611
0FFF_C014	DSM Interrupt Holdoff (TZIC_DSMINT)	32	R/W	0000_0000h	75.6.5/ 4612
0FFF_C080	Interrupt Security (FIQ) Register: Irq 0 to 31 (TZIC_INTSEC0)	32	R/W	0000_0000h	75.6.6/ 4613
0FFF_C084	Interrupt Security (FIQ) Register: Irq 0 to 31 (TZIC_INTSEC1)	32	R/W	0000_0000h	75.6.6/ 4613
0FFF_C088	Interrupt Security (FIQ) Register: Irq 0 to 31 (TZIC_INTSEC2)	32	R/W	0000_0000h	75.6.6/ 4613
0FFF_C08C	Interrupt Security (FIQ) Register: Irq 0 to 31 (TZIC_INTSEC3)	32	R/W	0000_0000h	75.6.6/ 4613
0FFF_C100	Enable Set Register: Irq 0 to 31 (TZIC_ENSET0)	32	R/W	0000_0000h	75.6.7/ 4614
0FFF_C104	Enable Set Register: Irq 0 to 31 (TZIC_ENSET1)	32	R/W	0000_0000h	75.6.7/ 4614
0FFF_C108	Enable Set Register: Irq 0 to 31 (TZIC_ENSET2)	32	R/W	0000_0000h	75.6.7/ 4614
0FFF_C10C	Enable Set Register: Irq 0 to 31 (TZIC_ENSET3)	32	R/W	0000_0000h	75.6.7/ 4614
0FFF_C180	Enable Clear Register: Irq 0 to 31 (TZIC_ENCLEAR0)	32	R/W	0000_0000h	75.6.8/ 4615
0FFF_C184	Enable Clear Register: Irq 0 to 31 (TZIC_ENCLEAR1)	32	R/W	0000_0000h	75.6.8/ 4615
0FFF_C188	Enable Clear Register: Irq 0 to 31 (TZIC_ENCLEAR2)	32	R/W	0000_0000h	75.6.8/ 4615
0FFF_C18C	Enable Clear Register: Irq 0 to 31 (TZIC_ENCLEAR3)	32	R/W	0000_0000h	75.6.8/ 4615
0FFF_C200	Source Set Register: Irq 0 to 31 (TZIC_SRCSET0)	32	R/W	0000_0000h	75.6.9/ 4616
0FFF_C204	Source Set Register: Irq 0 to 31 (TZIC_SRCSET1)	32	R/W	0000_0000h	75.6.9/ 4616
0FFF_C208	Source Set Register: Irq 0 to 31 (TZIC_SRCSET2)	32	R/W	0000_0000h	75.6.9/ 4616

Table continues on the next page...

TZIC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0FFF_C20C	Source Set Register: Irq 0 to 31 (TZIC_SRCSET3)	32	R/W	0000_0000h	75.6.9/ 4616
0FFF_C280	Source Clear Register: Irq 0 to 31 (TZIC_SRCCLEAR0)	32	R/W	0000_0000h	75.6.10/ 4617
0FFF_C284	Source Clear Register: Irq 0 to 31 (TZIC_SRCCLEAR1)	32	R/W	0000_0000h	75.6.10/ 4617
0FFF_C288	Source Clear Register: Irq 0 to 31 (TZIC_SRCCLEAR2)	32	R/W	0000_0000h	75.6.10/ 4617
0FFF_C28C	Source Clear Register: Irq 0 to 31 (TZIC_SRCCLEAR3)	32	R/W	0000_0000h	75.6.10/ 4617
0FFF_C400	Priority Register: Irq 0 to 3 (TZIC_PRIORITY0)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C404	Priority Register: Irq 0 to 3 (TZIC_PRIORITY1)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C408	Priority Register: Irq 0 to 3 (TZIC_PRIORITY2)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C40C	Priority Register: Irq 0 to 3 (TZIC_PRIORITY3)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C410	Priority Register: Irq 0 to 3 (TZIC_PRIORITY4)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C414	Priority Register: Irq 0 to 3 (TZIC_PRIORITY5)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C418	Priority Register: Irq 0 to 3 (TZIC_PRIORITY6)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C41C	Priority Register: Irq 0 to 3 (TZIC_PRIORITY7)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C420	Priority Register: Irq 0 to 3 (TZIC_PRIORITY8)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C424	Priority Register: Irq 0 to 3 (TZIC_PRIORITY9)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C428	Priority Register: Irq 0 to 3 (TZIC_PRIORITY10)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C42C	Priority Register: Irq 0 to 3 (TZIC_PRIORITY11)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C430	Priority Register: Irq 0 to 3 (TZIC_PRIORITY12)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C434	Priority Register: Irq 0 to 3 (TZIC_PRIORITY13)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C438	Priority Register: Irq 0 to 3 (TZIC_PRIORITY14)	32	R/W	0000_0000h	75.6.11/ 4618

Table continues on the next page...

TZIC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0FFF_C43C	Priority Register: Irq 0 to 3 (TZIC_PRIORITY15)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C440	Priority Register: Irq 0 to 3 (TZIC_PRIORITY16)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C444	Priority Register: Irq 0 to 3 (TZIC_PRIORITY17)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C448	Priority Register: Irq 0 to 3 (TZIC_PRIORITY18)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C44C	Priority Register: Irq 0 to 3 (TZIC_PRIORITY19)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C450	Priority Register: Irq 0 to 3 (TZIC_PRIORITY20)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C454	Priority Register: Irq 0 to 3 (TZIC_PRIORITY21)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C458	Priority Register: Irq 0 to 3 (TZIC_PRIORITY22)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C45C	Priority Register: Irq 0 to 3 (TZIC_PRIORITY23)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C460	Priority Register: Irq 0 to 3 (TZIC_PRIORITY24)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C464	Priority Register: Irq 0 to 3 (TZIC_PRIORITY25)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C468	Priority Register: Irq 0 to 3 (TZIC_PRIORITY26)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C46C	Priority Register: Irq 0 to 3 (TZIC_PRIORITY27)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C470	Priority Register: Irq 0 to 3 (TZIC_PRIORITY28)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C474	Priority Register: Irq 0 to 3 (TZIC_PRIORITY29)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C478	Priority Register: Irq 0 to 3 (TZIC_PRIORITY30)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_C47C	Priority Register: Irq 0 to 3 (TZIC_PRIORITY31)	32	R/W	0000_0000h	75.6.11/ 4618
0FFF_CD00	Pending Register: Irq 0 to 31 (TZIC_PND0)	32	R	0000_0000h	75.6.12/ 4620
0FFF_CD04	Pending Register: Irq 0 to 31 (TZIC_PND1)	32	R	0000_0000h	75.6.12/ 4620
0FFF_CD08	Pending Register: Irq 0 to 31 (TZIC_PND2)	32	R	0000_0000h	75.6.12/ 4620

Table continues on the next page...

TZIC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0FFF_CD0C	Pending Register: Irq 0 to 31 (TZIC_PND3)	32	R	0000_0000h	75.6.12/ 4620
0FFF_CD80	High Priority Pending Register: Irq 0 to 31 (TZIC_HIPND0)	32	R	0000_0000h	75.6.13/ 4621
0FFF_CD84	High Priority Pending Register: Irq 0 to 31 (TZIC_HIPND1)	32	R	0000_0000h	75.6.13/ 4621
0FFF_CD88	High Priority Pending Register: Irq 0 to 31 (TZIC_HIPND2)	32	R	0000_0000h	75.6.13/ 4621
0FFF_CD8C	High Priority Pending Register: Irq 0 to 31 (TZIC_HIPND3)	32	R	0000_0000h	75.6.13/ 4621
0FFF_CE00	Wakeup Config Register: Irq 0 to 31 (TZIC_WAKEUP0)	32	R	0000_0000h	75.6.14/ 4622
0FFF_CE04	Wakeup Config Register: Irq 0 to 31 (TZIC_WAKEUP1)	32	R	0000_0000h	75.6.14/ 4622
0FFF_CE08	Wakeup Config Register: Irq 0 to 31 (TZIC_WAKEUP2)	32	R	0000_0000h	75.6.14/ 4622
0FFF_CE0C	Wakeup Config Register: Irq 0 to 31 (TZIC_WAKEUP3)	32	R	0000_0000h	75.6.14/ 4622
0FFF_CF00	Software Interrupt Trigger Register (TZIC_SWINT)	32	W	0000_0000h	75.6.15/ 4623

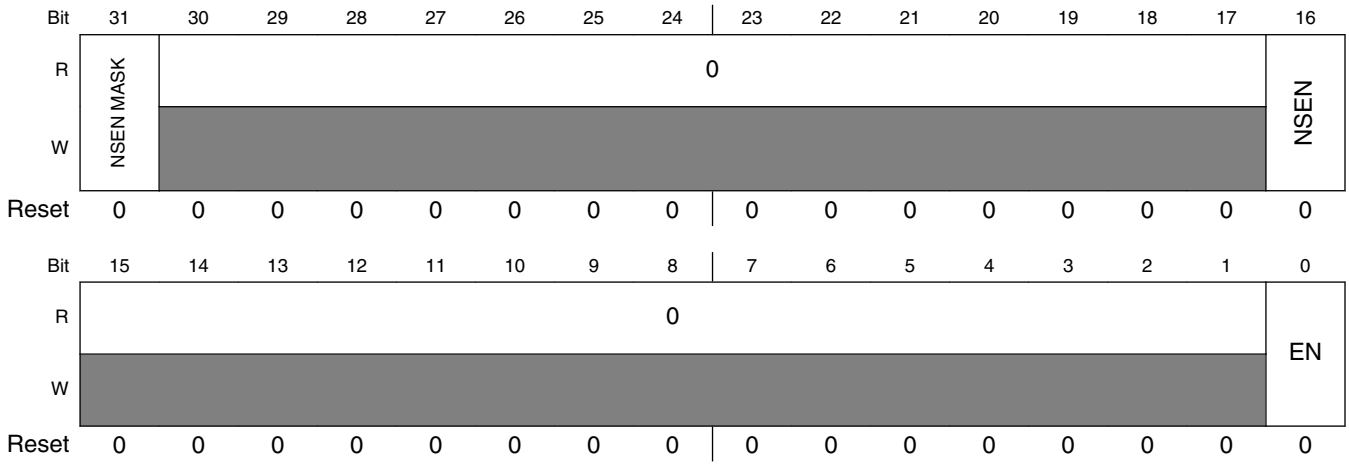
75.6.1 Control Register (TZIC_INTCTRL)

The interrupt control register (INTCTRL) can enable or disable all normal or secure interrupts to the ARM platform, depending on the state of the write transaction used. A secure read or write will access the enable bit for secure interrupts, a non-secure read or write will access the enable bit for non-secure interrupts. This only affects the outputs to the core, it has no impact on which interrupts are pending in the PND and HIPND registers.

A secure read can access the non-secure enable through bit 16, which will only be written if bit 31 is written at the same time. This allows secure software to ignore the non-secure enable bit and update the secure enable bit without a read-modify-write sequence.

This register can only be accessed by a 32-bit supervisor transaction.

Address: TZIC_INTCTRL is FFFC000h base + 0h offset = 0FFF_C000h



TZIC_INTCTRL field descriptions

Field	Description
31 NSEN MASK	Non-Secure Enable Mask - This bit indicates the secure write transaction is updating the NSEN bit. When written as 1'b0 the NSEN bit will not be updated. When written as 1'b1 the NSEN bit will be updated. This allows the secure software to ignore the NSEN bit and update only the Secure EN bit without requiring a read/modify/write to determine the current state of the NSEN. Note: Available for secure transactions only. If written with AWPROT[1] == 1'b0: If written with AWPROT[0] == 1'b1: no effect 0 NSEN bit unaffected 1 NSEN bit updated
30–17 Reserved	This read-only field is reserved and always has the value zero. Reserved
16 NSEN	Non-Secure Enable. This bit is available in secure mode only to enable/disable nonsecure interrupts. This allows for using both security levels in a single security level system where only secure transactions are available. A secure write to this bit will only take effect if the NSEN MASK bit is also set. Note: Available for secure transactions only. If written or read with AxPROT[1] == 1'b1: On reads: 1'b0 On writes: no effect If written or read with AxPROT[1] == 1'b0: 0 Non-Secure interrupts disabled 1 Non-Secure interrupts enabled
15–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 EN	Interrupt Enable. This bit, when set, enables normal or secure interrupts, depending on the secure state of the transaction. If written or read with AxPROT[1] == 1'b0: and If written or read with AxPROT[1] == 1'b1: 0 Secure Interrupts disabled

Table continues on the next page...

TZIC_INTCTRL field descriptions (continued)

Field	Description
	1 Secure Interrupts enabled 0 Non-Secure interrupts disabled 1 Non-Secure interrupts enabled

75.6.2 Interrupt Controller Type Register (TZIC_INTTYPE)

The interrupt controller type register (INTTYPE) contains information about the type of interrupt controller that has been implemented. It can be used to determine the number of interrupts, ARM cores, and if security is enabled for a given instantiation.

The current version only supports configurations with one ARM core and two security domain(s).

This register can only be accessed by a 32-bit supervisor read transaction.

Address: TZIC_INTTYPE is FFFC000h base + 4h offset = 0FFF_C004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					DOM	0		CPUS			ITLINES				
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1

TZIC_INTTYPE field descriptions

Field	Description
31–11 Reserved	This read-only field is reserved and always has the value zero. Reserved
10 DOM	Security Domains. This bit indicates the number of domains supported by the current configuration. 0 Supports 1 Domain 1 Supports 2 Domains: Secure and Non-Secure
9–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–5 CPUS	ARM core Count. This field indicates the number of ARM cores supported by the current configuration. Currently only 1 ARM core is supported, this field is specified for future compatability. 000 Supports 1 ARM core 001 Supports 2 ARM cores

Table continues on the next page...

TZIC_INTTYPE field descriptions (continued)

Field	Description
	010 Supports 3 ARM cores 111 Supports 8 ARM cores
4–0 ITLINES	Interrupt Lines. This field indicates the total number of interrupt lines supported by the current configuration. 00000 up to 32 interrupt lines supported 00001 up to 64 interrupt lines supported 00010 up to 96 interrupt lines supported 11110 up to 992 interrupt lines supported 11111 up to 1020 interrupt lines supported

75.6.3 Priority Mask Register (TZIC_PRIOMASK)

The Priority Mask Register (PRIOMASK) is a single register used to mask both secure and non-secure interrupts based on their priority configuration. The ARM platform will only receive an interrupt if there is a pending interrupt with a priority higher than the value in this register. If an interrupt source asserts and its priority is equal to or lower than the priority mask value, it will not cause an interrupt to the ARM platform.

The lowest priority the priority mask value can be set to is the lowest priority level in the controller. Therefore any interrupt set to the lowest allowed priority will never cause an interrupt since it is not possible for its priority to be greater than the mask value.

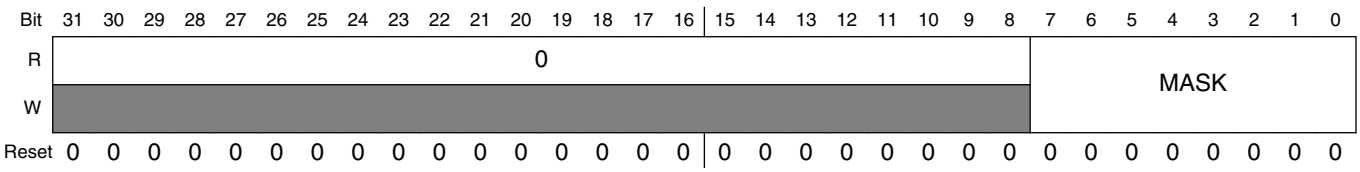
This register is affected by an integration parameter which varies the available resolution of priority values. The number of bits available can vary from four to eight, with the lowest resolution only implementing bits [7:4], and the highest resolution implementing bits [7:0]. Which bits are implemented can be determined by writing 8'hFF to bits [7:0] and reading back the result. The bits set high will indicate which register bits are implemented in this register as well as the PRIORITY registers.

Secure and non-secure transactions to this register will have different behaviors. A secure write will result in the register receiving the write data bits as written. A non-secure write will results in wdata bits [7:1] being shifted to the right one bit and bit [7] being set to 1'b1. A secure read will return the actual value stored in the register while a non-secure read will return the register value shifted to the left one bit and bit [0] being filled with a 1'b0. This is subject to the actual register bits implemented according to the integration parameter.

For more information regarding priorities, see TrustZone and Interrupt Priority.

This register can only be accessed by 8-bit or 32-bit supervisor transactions.

Address: TZIC_PRIOMASK is FFFC000h base + Ch offset = 0FFF_C00Ch



TZIC_PRIOMASK field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–0 MASK	Priority Mask - This field stores the priority mask value. Interrupts with a priority level set equal to or below this value are masked. For READS: 00000000 If ARPROT[1] == 1'b0 (secure), then priority mask is set to 0. If ARPROT[1] == 1'b1 (nonsecure), then priority mask is set to 128. 00000001 If ARPROT[1] == 1'b0 (secure), then the priority mask is set to 1. If ARPROT[1] == 1'b1 (nonsecure), then priority mask is set to 128. 11111111 The priority mask is set to 255.

75.6.4 Synchronizer Control (TZIC_SYNCCTRL)

The Synchronizer Control Register (SYNCCTRL) is used to control latency and power usage in the interrupt synchronizers. There are three configurations for synchronization:

- Low Latency-The synchronizer is clocked continuously and interrupts are synchronized as they arrive.
- Low Power-The synchronizer is not clocked until a difference is detected between the inputs and outputs of the synchronizer. When a difference is detected, the detection result is synchronized and used as a clock enable for the synchronizer, which introduces additional latency. Total latency is four clocks, two for the enable and two for the interrupt.

This register can only be accessed by 32-bit secure supervisor transactions.

Address: TZIC_SYNCCTRL is FFFC000h base + 10h offset = 0FFF_C010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														SYNCMODE	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TZIC_SYNCCTRL field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero. Reserved
1–0 SYNCMODE	Synchronizer Mode. 00 Low Latency 01 Low Power 10 Reserved 11 Reserved

75.6.5 DSM Interrupt Holdoff (TZIC_DSMINT)

The DSM Interrupt Holdoff Register (DSMINT) is used to stop interrupts to the ARM platform from occuring until a wakeup signal occurs. This is used for entering Deep Sleep Mode. When written, the DSM bit will cause the interrupt synchronizer to stop updating. If an interrupt occurs while this register is being written the bit will not be set and the ARM platform will have to try to write again. If no interrupt to the ARM platform occurs while the register is being written the write will succeed and the interrupt synchronizers will not update until the DSM bit clears. The DSM bit is cleared by either a write to the DSMINT register or assertion of the dsm_wakeup_request output of the interrupt controller. When a wakeup request is asserted, the interrupt controller will remove the DSM interrupt holdoff bit and interrupts will resume.

This register can only be accessed by 32-bit secure supervisor transactions.

Address: TZIC_DSMINT is FFFC000h base + 14h offset = 0FFF_C014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															DSM
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TZIC_DSMINT field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 DSM	DSM Interrupt Holdoff 0 Interrupt synchronizer updates 1 Interrupt synchronizer does not update

75.6.6 Interrupt Security (FIQ) Register: Irq 0 to 31 (TZIC_INTSEC*n*)

The Interrupt Security Registers (INTSEC) are used to set interrupts as secure or non-secure. Each bit in the register corresponds to an interrupt source available in the system. For each interrupt, if its secure bit is set to 1'b0, the interrupt is secure and will cause a secure interrupt to the ARM platform. When set as secure, a particular interrupt will also be invisible to non-secure accesses, its configuration hidden during non-secure reads. If its bit is set to 1'b1, the interrupt is non-secure and will cause a non-secure interrupt to the ARM platform.

The values read in the INTSEC register represent the interrupts currently set as non-secure. Out of reset, all interrupts are configured as secure, the intention being that when the interrupt controller is used in a system that does not use TrustZone, the secure bit of ARPROT/AWPROT will be tied to secure and the interrupt controller will act as always in secure mode.

These registers can only be accessed by 32-bit secure supervisor transactions.

Addresses: TZIC_INTSEC0 is FFFC000h base + 80h offset = 0FFF_C080h
TZIC_INTSEC1 is FFFC000h base + 84h offset = 0FFF_C084h
TZIC_INTSEC2 is FFFC000h base + 88h offset = 0FFF_C088h
TZIC_INTSEC3 is FFFC000h base + 8Ch offset = 0FFF_C08Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SECURE																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TZIC_INTSEC*n* field descriptions

Field	Description
31–0 SECURE	Interrupt Secure Status. When read, this register returns the status bit values of the associated interrupts. When written, this register sets the configuration for the associated interrupts. The corresponding interrupts for an INTSEC(X) are determined through the following formula: SECURE(X)[31] -> (32*X)+31

TZIC_INTSEC*n* field descriptions (continued)

Field	Description
	SECURE(X)[0] -> (32*X)+0 0 Interrupt is secure 1 Interrupt is non-secure

75.6.7 Enable Set Register: Irq 0 to 31 (TZIC_ENSET*n*)

The Enable Set Registers (ENSET) are used to enable interrupt requests to the core. Each bit in the register corresponds to an interrupt source available in the system. For each interrupt, when read, if its enable bit is set to 1'b1, the interrupt is enabled and will be transmitted to the ARM platform. If its bit is set to 1'b0, any activity on that interrupt will have no effect.

Writing a bit to 1'b1 in this register will enable the interrupt associated with that bit. Writing a bit to 1'b0 will have no effect. To clear an enable bit, the Enable Clear Registers must be used.

The values read in the ENSET register represent the currently enabled interrupts.

These registers can only be accessed by 32-bit supervisor transactions.

Addresses: TZIC_ENSET0 is FFFC000h base + 100h offset = 0FFF_C100h
TZIC_ENSET1 is FFFC000h base + 104h offset = 0FFF_C104h
TZIC_ENSET2 is FFFC000h base + 108h offset = 0FFF_C108h
TZIC_ENSET3 is FFFC000h base + 10Ch offset = 0FFF_C10Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INTENSET																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TZIC_ENSET*n* field descriptions

Field	Description
31–0 INTENSET	Interrupt Enable Set. When written, this register sets the enable bits of the associated interrupts. When read, this register returns the enable bit values of the associated interrupts. The corresponding interrupts for register ENSETX are determined through the following formula: INTENSET[31] -> (32*X)+31 INTENSET[0] -> (32*X)+0 If ARPROT[1] == 1'b0 (secure) 0 Interrupt is disabled 1 Interrupt is enabled If ARPROT[1] == 1'b1 (nonsecure)

TZIC_ENSET*n* field descriptions (continued)

Field	Description
	0 Interrupt is disabled or configured as secure 1 Interrupt is enabled If AWPROT[1] == 1'b0 (secure) 0 no effect 1 Interrupt enable bit set If AWPROT[1] == 1'b1 (nonsecure) 0 no effect 1 Interrupt enable bit set if bit is configured as non-secure

75.6.8 Enable Clear Register: Irq 0 to 31 (TZIC_ENCLEAR*n*)

The Enable Clear Registers (ENCLEAR) are used to disable interrupt requests to the core. Each bit in the register corresponds to an interrupt source available in the system. For each interrupt, when read, if its enable bit is set to 1'b1, the interrupt is enabled and will be transmitted to the ARM platform. If its bit is set to 1'b0, any activity on that interrupt will have no effect.

Writing a bit to 1'b1 in this register will disable the interrupt associated with that bit. Writing a bit to 1'b0 will have no effect. To set an enable bit, the Enable Set Registers must be used.

The values read in the ENCLEAR register represent the currently enabled interrupts.

These registers can only be accessed by 32-bit supervisor accesses.

Addresses: TZIC_ENCLEAR0 is FFFC000h base + 180h offset = 0FFF_C180h
TZIC_ENCLEAR1 is FFFC000h base + 184h offset = 0FFF_C184h
TZIC_ENCLEAR2 is FFFC000h base + 188h offset = 0FFF_C188h
TZIC_ENCLEAR3 is FFFC000h base + 18Ch offset = 0FFF_C18Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INTENCLEAR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TZIC_ENCLEAR*n* field descriptions

Field	Description
31–0 INTENCLEAR	Interrupt Enable Cleared. When written, this register clears the enable bits of the associated interrupts. When read, this register returns the enable bit values of the associated interrupts. The corresponding interrupts for register ENCLEARX are determined through the following formula: INTENCLEAR[31] -> (32*X)+31

TZIC_ENCLEAR*n* field descriptions (continued)

Field	Description
	INTENCLEAR[0] -> (32*X)+0 If ARPROT[1] == 1'b0 (secure) 0 Interrupt is disabled 1 Interrupt is enabled If ARPROT[1] == 1'b1 (nonsecure) 0 Interrupt is disabled or configured as secure 1 Interrupt is enabled If AWPROT[1] == 1'b0 (secure) 0 no effect 1 Interrupt enable bit cleared If AWPROT[1] == 1'b1 (nonsecure) 0 no effect 1 Interrupt enable bit cleared if bit is configured as non-secure

75.6.9 Source Set Register: Irq 0 to 31 (TZIC_SRCSET*n*)

The Source Set Registers (SRCSET) are used to determine which interrupts are currently asserted and to force interrupts to an asserted state regardless of activity on the interrupt lines. Each bit in the register corresponds to an interrupt source available in the system. For each interrupt, when read, if its source bit is set to 1'b1, the interrupt is asserted and pending action from the ARM platform. If its bit is set to 1'b0, the interrupt is not asserted.

Writing a bit to 1'b1 in this register will override the interrupt line and force the corresponding interrupt to assert. Writing a bit to 1'b0 will have no effect. To clear a source bit, the Source Clear Registers or SWINT must be used.

If a source bit is set which corresponds to an interrupt not enabled in the ENSET/ ENCLEAR registers, the source bit will be 1'b1 but will not be considered pending and will not cause an interrupt to the core.

The values read in the SRCSET register represent the currently asserted interrupts.

These registers can only be accessed by 32-bit supervisor accesses.

Addresses: TZIC_SRCSET0 is FFFC000h base + 200h offset = 0FFF_C200h
TZIC_SRCSET1 is FFFC000h base + 204h offset = 0FFF_C204h
TZIC_SRCSET2 is FFFC000h base + 208h offset = 0FFF_C208h
TZIC_SRCSET3 is FFFC000h base + 20Ch offset = 0FFF_C20Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SRCSET																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TZIC_SRCSET*n* field descriptions

Field	Description
31–0 SRCSET	Interrupt Source Set. When written, this register forces the source bits of the associated interrupts. When read, this register returns the source bit values of the associated interrupts. The corresponding interrupts for register SRCSETX are determined through the following formula: SRCSET[31] -> (32*X)+31 SRCSET[0] -> (32*X)+0 If ARPROT[1] == 1'b0 (secure) 0 Interrupt is not asserted 1 Interrupt is asserted If ARPROT[1] == 1'b1 (nonsecure) 0 Interrupt is not asserted or configured as secure 1 Interrupt is asserted If AWPROT[1] == 1'b0 (secure) 0 no effect 1 Interrupt source bit set If AWPROT[1] == 1'b1 (nonsecure) 0 no effect 1 Interrupt source bit set if bit is configured as non-secure

75.6.10 Source Clear Register: Irq 0 to 31 (TZIC_SRCCLEAR*n*)

The Source Clear Registers (SRCCLEAR) are used to determine which interrupts are currently asserted and to clear interrupts from the asserted state that were triggered by software, either through the SRCSET or SWINT registers. Each bit in the register corresponds to an interrupt source available in the system. For each interrupt, when read, if its source bit is set to 1'b1, the interrupt is asserted and is pending action from the ARM platform. If its bit is set to 1'b0, the interrupt is not asserted.

Writing a bit to 1'b1 in this register will clear an asserted interrupt that was triggered by software. Writing a bit to 1'b0 will have no effect. To set a source bit, the Source Set Registers or Software Triggered Interrupt Register must be used.

The values read in the SRCCLEAR register represent the currently asserted interrupts.

These registers can only be accessed by 32-bit supervisor accesses.

Addresses: TZIC_SRCCLEAR0 is FFFC000h base + 280h offset = 0FFF_C280h
TZIC_SRCCLEAR1 is FFFC000h base + 284h offset = 0FFF_C284h
TZIC_SRCCLEAR2 is FFFC000h base + 288h offset = 0FFF_C288h
TZIC_SRCCLEAR3 is FFFC000h base + 28Ch offset = 0FFF_C28Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SRCCLEAR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TZIC_SRCCLEAR*n* field descriptions

Field	Description
31–0 SRCCLEAR	Interrupt Source Clear. When written, this register forces the source bits of the associated interrupts. When read, this register returns the source bit values of the associated interrupts. The corresponding interrupts for register SRCCLEARX are determined through the following formula: SRCCLEAR[31] -> (32*X)+31 SRCCLEAR[0] -> (32*X)+0 If ARPROT[1] == 1'b0 (secure) 0 Interrupt is not asserted 1 Interrupt is asserted If ARPROT[1] == 1'b1 (nonsecure) 0 Interrupt is not asserted or configured as secure 1 Interrupt is asserted If AWPROT[1] == 1'b0 (secure) 0 no effect 1 Interrupt source bit cleared If AWPROT[1] == 1'b1 (nonsecure) 0 no effect 1 Interrupt source bit cleared if bit is configured as non-secure

75.6.11 Priority Register: Irq 0 to 3 (TZIC_PRIORITY*n*)

The Priority Registers (PRIORITY) are used to configure the priority of each interrupt for masking purposes.

Each eight bit priority field is affected by an integration parameter which determines the available resolution of priority levels. The number of bits available can vary from four to eight, with the lowest resolution only implementing bits [7:4], and the highest resolution implementing bits [7:0]. The implemented number of bits can be found in a manner similar to that described in Priority Mask Register (TZIC_PRIOMASK).

The priority scheme used is such that an interrupt assigned to 0 has the highest priority.

These registers can only be accessed by 32-bit or 8-bit supervisor transactions.

Addresses: FFFC000h base + 400h offset + (4d × *n*), where *n* = 0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PRIO3								PRIO2								PRIO1								PRIO0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TZIC_PRIORITY*n* field descriptions

Field	Description
31–24 PRIO3	Interrupt Priority. This field stores the priority value for its corresponding interrupt. The interrupts for register PRIORITYX are determined through the following formula: PRIO3 -> (4*X)+3 PRIO2 -> (4*X)+2 PRIO1 -> (4*X)+1 PRIO0 -> (4*X)+0 If ARPROT[1] == 1'b0 (secure) 00000000 Interrupt priority is set to 0 00000001 Interrupt priority is set to 1 11111111 Interrupt priority is set to 255 If ARPROT[1] == 1'b1 (nonsecure) 00000000 Interrupt priority is set to 128 or interrupt is configured as secure 00000010 Interrupt priority is set to 129 11111110 Interrupt priority is set to 255 If AWPROT[1] == 1'b0 (secure) 00000000 Interrupt priority is set to 0 00000001 Interrupt priority is set to 1 11111111 Interrupt priority is set to 255 If AWPROT[1] == 1'b1 (nonsecure) 00000000 Interrupt priority is set to 128 if interrupt is configured as non-secure 00000001 Interrupt priority is set to 128 if interrupt is configured as non-secure 00000010 Interrupt priority is set to 129 if interrupt is configured as non-secure 11111110 Interrupt priority is set to 255 if interrupt is configured as non-secure 11111111 Interrupt priority is set to 255 if interrupt is configured as non-secure

Table continues on the next page...

TZIC_PRIORITY*n* field descriptions (continued)

Field	Description
23–16 PRIO2	Description and settings as for PRIO3.
15–8 PRIO1	Description and settings as for PRIO3.
7–0 PRIO0	Description and settings as for PRIO3.

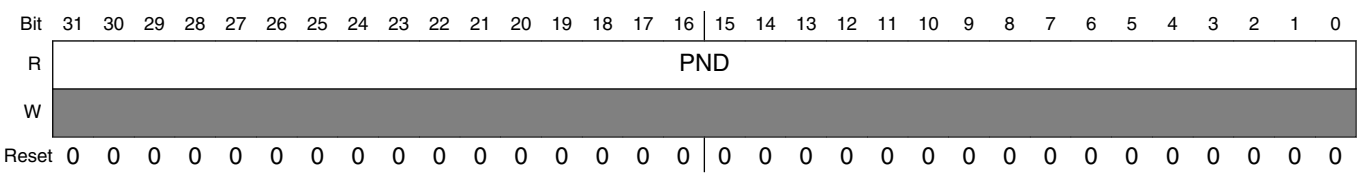
75.6.12 Pending Register: Irq 0 to 31 (TZIC_PND*n*)

The Pending Registers (PND) are used to determine which interrupts are currently pending action from the ARM platform. Each bit in the register corresponds to an interrupt source available in the system. For each interrupt, if its pending bit is set to 1'b1, the interrupt is enabled in the ENSET/ENCLEAR registers and asserted either on the interrupt inputs or in the SRCSET/SRCCLEAR registers, and therefore is pending action from the ARM platform. Priority values have no effect on whether an interrupt is pending.

This is a read-only register, writes will be ignored.

These registers can only be accessed by 32-bit supervisor transactions.

Addresses: TZIC_PND0 is FFFC000h base + D00h offset = 0FFF_CD00h
TZIC_PND1 is FFFC000h base + D04h offset = 0FFF_CD04h
TZIC_PND2 is FFFC000h base + D08h offset = 0FFF_CD08h
TZIC_PND3 is FFFC000h base + D0Ch offset = 0FFF_CD0Ch



TZIC_PND*n* field descriptions

Field	Description
31–0 PND	Interrupt Pending Status. When read, this register returns the pending bit values of the associated interrupts, indicating which interrupts are enabled and asserted. The corresponding interrupts for register PNDX are determined through the following formula: PND[31] -> (32*X)+31 PND[0] -> (32*X)+0 If ARPROT[1] == 1'b0 (secure) 0 Interrupt is not pending 1 Interrupt is pending

TZIC_PND*n* field descriptions (continued)

Field	Description
	If ARPROT[1] == 1'b1 (nonsecure) 0 Interrupt is not pending or configured as secure 1 Interrupt is pending

75.6.13 High Priority Pending Register: Irq 0 to 31 (TZIC_HIPND*n*)

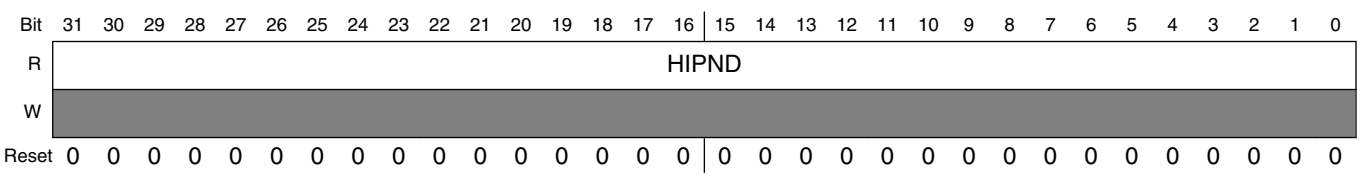
The High Priority Pending Registers (HIPND) are used to determine which interrupts are currently pending at the highest active priority. Each bit in the register corresponds to an interrupt source available in the system. For each interrupt, if its pending bit is set to 1'b1, the interrupt is pending action from the ARM platform and its priority is the highest level amongst all pending enabled interrupts. See Pending Register: Irq 0 to 31 (TZIC_PND*n*) for more information about pending interrupts.

The highest active priority is determined based on the security state of the read transaction. If the register is read by a non-secure transaction, the pending interrupts returned are those non-secure interrupts that are pending and set to the highest priority of any currently pending non-secure interrupts. If the register is read by a secure transaction, the pending interrupts returned are those interrupts, secure or non-secure, that are pending and set to the highest priority of any currently pending interrupts regardless of security configuration.

This is a read-only register, writes will be ignored.

These registers can only be accessed by 32-bit supervisor transactions.

Addresses: TZIC_HIPND0 is FFFC000h base + D80h offset = 0FFF_CD80h
TZIC_HIPND1 is FFFC000h base + D84h offset = 0FFF_CD84h
TZIC_HIPND2 is FFFC000h base + D88h offset = 0FFF_CD88h
TZIC_HIPND3 is FFFC000h base + D8Ch offset = 0FFF_CD8Ch



TZIC_HIPND*n* field descriptions

Field	Description
31–0 HIPND	High Priority Interrupt Pending Status. When read, this register returns the pending bit values of the associated interrupts which are set to the highest currently active priority level. The corresponding interrupts for register HIPNDX are determined through the following formula: HIPND[31] -> (32*X)+31

TZIC_HIPND*n* field descriptions (continued)

Field	Description
	HIPND[0] -> (32*X)+0 If ARPROT[1] == 1'b0 (secure) 0 Interrupt is not pending at the higest priority level 1 Interrupt is pending at the higest priority level If ARPROT[1] == 1'b1 (nonsecure) 0 Interrupt is not pending at the higest priority level or configured as secure 1 Interrupt is pending at the higest priority level

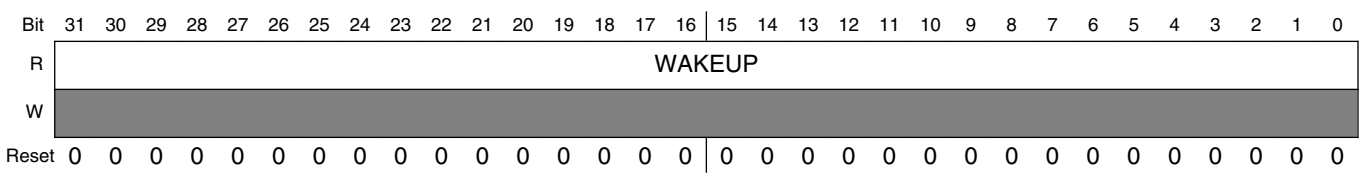
75.6.14 Wakeup Config Register: Irq 0 to 31 (TZIC_WAKEUP*n*)

The Wakeup Conguration Registers (WAKEUP) are used to set which interrupts will assert the tzic_wakeup_request signal. Each bit in the register corresponds to an interrupt source available in the system. For each interrupt, if its wakeup bit is set to 1'b1 assertion of the interrupt will cause the wakeup request signal to assert combinationally. When its wakeup bit is set to 1'b0, the interrupt will not assert the wakeup request signal.

The values read in the WAKEUP register represent interrupts currently set to wake up the system. Out of reset, all interrupts are configured not to wake up the system. Non-secure transactions can only enable or disable wakeup configuration of interrupts configured as non-secure. Secure transactions can enable or disable wakeup configuration of any interrupts.

These registers can only be accessed by 32-bit supervisor transactions.

Addresses: TZIC_WAKEUP0 is FFFC000h base + E00h offset = 0FFF_CE00h
TZIC_WAKEUP1 is FFFC000h base + E04h offset = 0FFF_CE04h
TZIC_WAKEUP2 is FFFC000h base + E08h offset = 0FFF_CE08h
TZIC_WAKEUP3 is FFFC000h base + E0Ch offset = 0FFF_CE0Ch



TZIC_WAKEUP*n* field descriptions

Field	Description
31–0 WAKEUP	Wakeup Configuration. When read, this register returns the wakeup bit values of the associated interrupts. When written, this register sets the configuration for the associated interrupts. The corresponding interrupts for register WAKEUPX are determined through the following formula: WAKEUP[31] -> (32*X)+31

TZIC_WAKEUP*n* field descriptions (continued)

Field	Description
	WAKEUP[0] -> (32*X)+0 If AxPROT[1] == 1'b0 0 Interrupt will not assert tzic_wakeup_request 1 Interrupt will assert tzic_wakeup_request If AxPROT[1] == 1'b1 0 Interrupt will not assert tzic_wakeup_request or is configured as secure and not accessible 1 Interrupt will assert tzic_wakeup_request

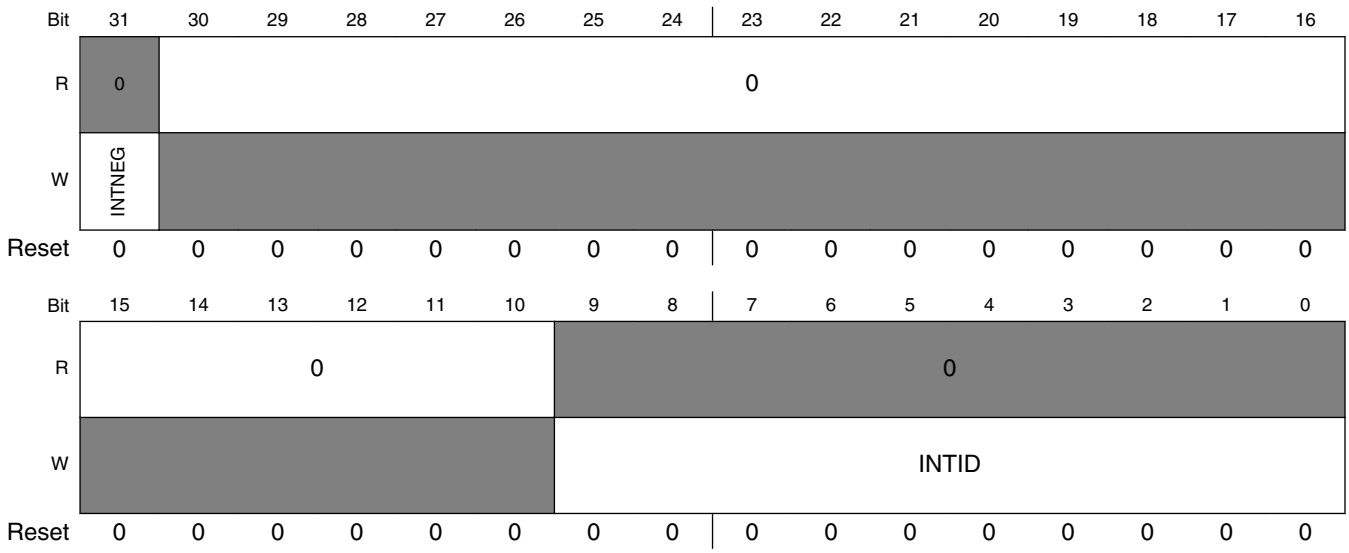
75.6.15 Software Interrupt Trigger Register (TZIC_SWINT)

The interrupt controller software interrupt register (SWINT) can be used by software to assert or negate an interrupt at a specific interrupt number. Writing the number of the desired interrupt will either trigger an interrupt to assert if not already asserted by hardware, or negate an interrupt that was originally asserted by software. An interrupt asserted by hardware cannot be negated by this register. Triggering an interrupt with the SWINT is equivalent to asserting the associated interrupt input pin, except regarding the tzic_wakeup_request output. It will be asserted in the SRCSET/SRCCLEAR regitsers, but will not be pending unless the interrupt is enabled. A software interrupt will not cause a wakeup request.

A secure write can trigger any enabled interrupt. A non-secure write can only trigger non-secure interrupts, an attempt to trigger a secure interrupt with a non-secure write will be ignored. If wbstrb[3] is not asserted to indicate whether asserting or negating an interrupt, the write will be ignored.

This register can only be accessed by 32-bit supervisor transactions.

Address: TZIC_SWINT is FFFC000h base + F00h offset = 0FFF_CF00h



TZIC_SWINT field descriptions

Field	Description
31 INTNEG	Interrupt Negate. This bit indicates whether the interrupt ID in the INTID field is being asserted or negated. Assuming the interrupt is not being asserted by hardware, when 1'b0, an interrupt is asserted, when 1'b1, an interrupt is negated.
30–10 Reserved	This read-only field is reserved and always has the value zero. Reserved
9–0 INTID	Interrupt ID. This field indicates the number of the interrupt to be triggered (asserted or negated). Triggering an interrupt outside of the maximum number of interrupts will result in UPREDICTABLE behavior. Triggering a secure interrupt with a non-secure write will have no effect.

Chapter 76
Universal Asynchronous Receiver/Transmitter (UART)

76.1 Overview

Universal Asynchronous Receiver/Transmitter (UART) provides serial communication capability with external devices through an RS-232 cable or through use of external circuitry that converts infrared signals to electrical signals (for reception) or transforms electrical signals to signals that drive an infrared LED (for transmission) to provide low speed IrDA compatibility. UART supports NRZ encoding format and IrDA-compatible infrared slow data rate (SIR) format.

The following figure is the UART block diagram.

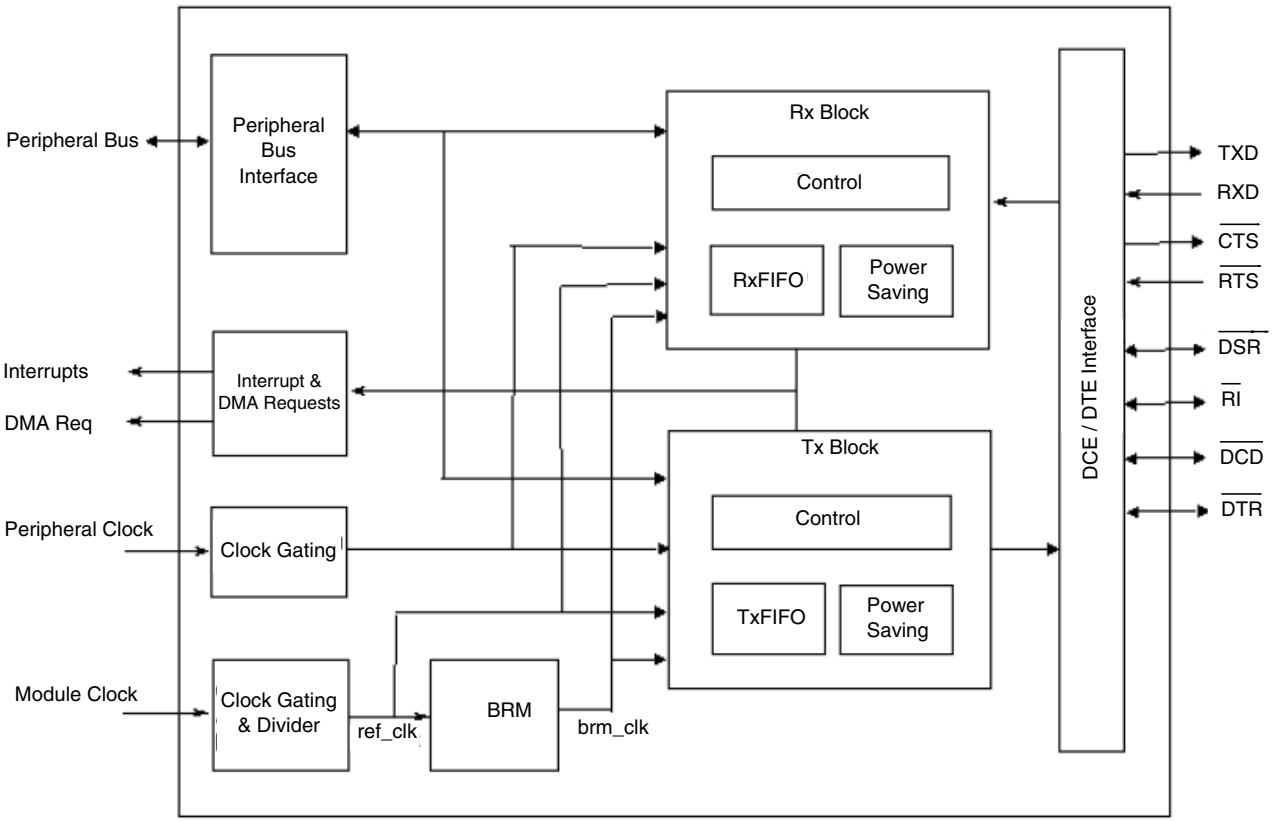


Figure 76-1. UART Block Diagram

76.1.1 Features

The UART includes the following features:

- High-speed TIA/EIA-232-F compatible, up to 4.0 Mbit/s
- Serial IR interface low-speed, IrDA-compatible (up to 115.2 Kbit/s)
- 7 or 8 data bits
- 1 or 2 stop bits
- Programmable parity (even, odd, and no parity)
- Hardware flow control support for request to send ($\overline{RTS}$) and clear to send ($\overline{CTS}$) signals
- Edge-selectable $\overline{RTS}$ and edge-detect interrupts
- Status flags for various flow control and FIFO states
- Voting logic for improved noise immunity (16x oversampling)
- Transmitter FIFO empty interrupt suppression
- UART internal clocks enable/disable
- Auto baud rate detection (up to 115.2 Kbit/s)
- Receiver and transmitter enable/disable for power saving

- DCE/DTE capability
- $\overline{\text{RTS}}$, IrDA asynchronous wake (AIRINT), receive asynchronous wake (AWAKE), $\overline{\text{RI}}$ (DTE only), $\overline{\text{DCD}}$ (DTE only), $\overline{\text{DTR}}$ (DCE only) and $\overline{\text{DSR}}$ (DTE only) interrupts wake the processor from STOP mode
- Maskable interrupts
- Two DMA Requests (TxFIFO DMA Request and RxFIFO DMA Request)
- Escape character sequence detection
- Software reset ($\overline{\text{SRST}}$)
- Two independent, 32-entry FIFOs for transmit and receive
- The peripheral clock can be totally asynchronous with the module clock. The module clock determines baud rate. This allows frequency scaling on peripheral clock (such as during DVFS mode) while remaining the module clock frequency and baud rate.

76.1.2 Modes of Operation

- Serial RS-232NRZ mode
- IrDA mode

76.1.3 UART I/O Configuration in DTE and DCE Modes

The i.MX53 UART interfaces can serve both as DTE or DCE device. This can be configured by the DCEDTE control bit (default 0 – DCE mode). Table 76-1 shows the UART I/O configuration based on the operation mode.

Table 76-1. UART I/O Configuration

Port	DTE Mode		DCE Mode	
	Direction	Description	Direction	Description
RTS	Output	RTS from DTE to DCE	Input	RTS from DTE to DCE
CTS	Input	CTS from DCE to DTE	Output	CTS from DCE to DTE
DTR	Output	DTR from DTE to DCE	Input	DTR from DTE to DCE
DSR	Input	DSR from DCE to DTE	Output	DSR from DCE to DTE
DCD	Input	DCD from DCE to DTE	Output	DCD from DCE to DTE
RI	Input	RING from DCE to DTE	Output	RING from DCE to DTE
TXD_MUX	Input	Serial data from DCE to DTE	Output	Serial data from DCE to DTE
RXD_MUX	Output	Serial data from DTE to DCE	Input	Serial data from DTE to DCE

The UART IP Signals definitions along the chapter are following DCE viewpoint (for which the CTS is output and RTS is input).

The UART pins (at IC boundary) change direction according to UART functionality: data communication equipment (DCE mode) or as data terminal equipment (DTE mode). For example, in DCE mode the TxD pad is set as output and RxD is set as input. In DTE mode the TxD pad is set as input and RxD pad is set as output.

This additional signal multiplexing occurs between the UART IP and the IC pad (at the IOMUX) and sets the pad functionality for the different modes. For information regarding UART I/O routing to IC pads, please refer to Table 4-3.

76.2 External Signals

The following table lists conventions for representing signals.

Table 76-2. Module Signal Conventions

Category	Convention	Example(s
Off-chip signal	Uppercase (all capital letters)	TXD
Internal signal[1]	Lowercase italics	*core_int*
Active low signal	_B (_b) suffix or overbar	RESET_EN_B or $\overline{\text{RESET\_EN}}$
Range of bussed or commonly named signals	Beginning and end points of the range are: • Separated by a colon. • Surrounded by square brackets.	ADDR[31:0] CSE_B[7:0] or $\overline{\text{CSE}}$[7:0]
Individual signal in a range of bussed or commonly named signals	Individual number in the range appears without a colon or square brackets	ADDR31 CSE0_B or $\overline{\text{CSE0}}$

1. Internal signals are for reference only in descriptions of internal module or SoC functionality.

The table below describes all UART signals that connect off-chip.

Table 76-3. Off-Chip Module Signals

Signal name	I/O	Active state	Description	Reset state
Serial / IrDA Signals				
RXD	I		Serial / infrared data receive	
TXD	O		Serial/infrared data transmit	High
Modem Control Signals				
$\overline{\text{CTS}}$	O	Low	Clear to send	High
$\overline{\text{RTS}}$	I	Low	Request to send	
$\overline{\text{DSR}}$	I/O	Low	Data set ready	High

Table continues on the next page...

Table 76-3. Off-Chip Module Signals (continued)

Signal name	I/O	Active state	Description	Reset state
$\overline{DCD}$	I/O	Low	Data carrier detected	High
$\overline{DTR}$	I/O	Low	Data terminal ready	
$\overline{RI}$	I/O	Low	Ring indicator	High
Interrupts				
interrupt_uart	O	Low	UART interrupt	High
DMA Requests				
dma_req_rx	O	Low	Receiver DMA request	High
dma_req_tx	O	Low	Transmitter DMA request	High
Clocks				
peripheral_clock	I		Peripheral clock	
module_clock	I		Clock source for the module's logic	
Special Signals				
stop_req	I	High	Module stop mode	
doze_req	I	High	Module doze mode	
debug_req	I	High	Module debug	

76.2.1 Detailed Signal Descriptions

76.2.1.1 Serial/IrDA Signals

76.2.1.1.1 RXD - Data Receive

Input asynchronous data receive in Serial and IrDA modes.

76.2.1.1.2 TXD - Data Transmit

Output asynchronous data transmit in Serial and IrDA modes.

76.2.1.2 Modem Control Signals

76.2.1.2.1 $\overline{\text{CTS}}$ - Clear To Send

Output in DCE and DTE mode. This signal informs the remote modem that UART is ready to receive data.

76.2.1.2.2 $\overline{\text{RTS}}$ - Request To Send

Input in DCE and DTE mode. This signal informs UART that remote modem is ready to receive data.

76.2.1.2.3 $\overline{\text{DSR}}$ - Data Set Ready

Input in DTE mode. Indicates to UART that remote modem is operational.

Output in DCE mode. Indicates to remote modem that UART is operational.

76.2.1.2.4 $\overline{\text{DCD}}$ - Data Carrier Detected

Input in DTE mode. Indicates to UART that a good carrier is being received from the remote modem.

Output in DCE mode. Indicates to remote device that a good carrier is being received from the UART.

76.2.1.2.5 $\overline{\text{DTR}}$ - Data Terminal Ready

Input in DCE mode. Indicates to UART (in DCE mode) that remote device (in DTE mode) is operational.

Output in DTE mode. Indicates to remote modem (in DCE mode) that UART (in DTE mode) is operational.

76.2.1.2.6 $\overline{\text{RI}}$ - Ring Indicator

Input in DTE mode. Indicates to UART that remote modem is detecting a ringing tone.

Output in DCE mode. Indicates to remote device that UART is detecting a ringing tone.

76.2.1.3 Interrupt Signals

76.2.1.3.1 *interrupt_uart* - UART Interrupt

Output interrupt request.

76.2.1.4 DMA Request Signals

76.2.1.4.1 *dma_req_rx* - Receiver DMA Request

Output DMA Request signal for receiver interface.

76.2.1.4.2 *dma_req_tx* - Transmitter DMA Request

Output DMA Request signal for transmitter interface. Set at 0 when TXDMAEN (UCR1[3]) is at 1 and TRDY (USR1[13]) is also at 1.

76.2.1.5 Clock Signals

76.2.1.5.1 *peripheral_clock* - Peripheral Clock

See Clocks for more information about *peripheral_clock* .

76.2.1.5.2 *module_clock* - Module Clock

See Clocks for more information about *module_clock* .

76.2.1.6 Special Signals

76.2.1.6.1 *stop_req* - Stop Mode

Input stop mode. Indicates to UART that ARM platform is going to enter in Stop Mode and clocks are going to stop running.

See Low Power Modes for more information about Stop Mode.

76.2.1.6.2 *doze_req* - Doze Mode

Input doze mode. ARM platform requests UART to switch in doze mode (power saving mode).

See Low Power Modes for more information about Doze Mode.

76.2.1.6.3 *debug_req* - Debug Mode

Input debug mode. Indicates UART it has to enter in debug mode.

See UART Operation in System Debug State, for more information about Debug Mode.

76.3 Functional Description

This section provides a complete functional description of the block.

76.3.1 Interrupts and DMA Requests

See the following table for the lists of all interrupt and DMA signals and associated interrupt and DMA sources of the UART. See register description section for explanation of interrupt/DMA enable and status.

Table 76-4. Interrupts and DMA

Interrupt/DMA Output	Interrupt/DMA Enable	Enable Register Location	Interrupt/DMA Flag	Flag Register Location
interrupt_uart	RRDYEN	UCR1 (bit 9)	RRDY	USR1 (bit 9)
	IDEN	UCR1 (bit 12)	IDLE	USR2 (bit 12)
	DREN	UCR4 (bit 0)	RDR	USR2 (bit 0)
	RXDSEN	UCR3 (bit 6)	RXDS	USR1 (bit 6)
	ATEN	UCR2 (bit 3)	AGTIM	USR1 (bit 8)
interrupt_uart	TXMPTYEN	UCR1 (bit 6)	TXFE	USR2 (bit 14)
	TRDYEN	UCR1 (bit 13)	TRDY	USR1 (bit 13)
	TCEN	UCR4 (bit 3)	TXDC	USR2 (bit 3)

Table continues on the next page...

Table 76-4. Interrupts and DMA (continued)

Interrupt/DMA Output	Interrupt/DMA Enable	Enable Register Location	Interrupt/DMA Flag	Flag Register Location
interrupt_uart	OREN	UCR4 (bit 1)	ORE	USR2 (bit 1)
	BKEN	UCR4 (bit 2)	BRCD	USR2 (bit 2)
	WKEN	UCR4 (bit 7)	WAKE	USR2 (bit 7)
	ADEN	UCR1 (bit 15)	ADET	USR2 (bit 15)
	ACIEN	UCR3 (bit 0)	ACST	USR2 (bit 11)
	ESCI	UCR2 (bit 15)	ESCF	USR1 (bit 11)
	ENIRI	UCR4 (bit 8)	IRINT	USR2 (bit 8)
	AIRINTEN	UCR3 (bit 5)	AIRINT	USR1 (bit 5)
	AWAKEN	UCR3 (bit 4)	AWAKE	USR1 (bit 4)
	FRAERREN	UCR3 (bit 11)	FRAERR	USR1 (bit 10)
	PARERREN	UCR3 (bit 12)	PARITYERR	USR1 (bit 15)
	RTSDEN	UCR1 (bit 5)	RTSD	USR1 (bit 12)
	RTSEN	UCR2 (bit 4)	RTSF	USR2 (bit 4)
	DTREN (DCE)	UCR3 (bit 13)	DTRF	USR2 (bit 13)
	RI (DTE)	UCR3 (bit 8)	RIDELT	USR2 (bit 10)
	DCD (DTE)	UCR3 (bit 9)	DCDDELT	USR2 (bit 6)
	DTRDEN	UCR3 (bit 3)	DTRD	USR1 (bit 7)
dma_req_rx	RXDMAEN	UCR1 (bit 8)	RRDY	USR1 (bit 9)
	ATDMAEN	UCR1 (bit 2)	AGTIM	USR1 (bit 8)
	IDDMAEN	UCR4 (bit 6)	IDLE	USR2 (bit 12)
dma_req_tx	TXDMAEN	UCR1 (bit 3)	TRDY	USR1 (bit 13)

76.3.2 Clocks

This section describes clocks and special clocking requirements of the UART.

76.3.2.1 Clock requirements

UART module receives 2 clocks, *peripheral_clock* and *module_clock*. The *peripheral_clock* is used as write clock of the TxFIFO, read clock of the RxFIFO and synchronization of the modem control input pins. It must always be running when UART is enabled. There is an exception in stop mode (see Clocking in Low-Power Modes).

The *module_clock* is for all the state machines, writing RxFIFO, reading TxFIFO, etc. It must always be running when UART is sending or receiving characters.This clock is used in order to allow frequency scaling on *peripheral_clock* without changing configuration of baud rate (*module_clock* staying at a fixed frequency).

The constraints on *peripheral_clock* and *module_clock* are as follows:

- *peripheral_clock* and *module_clock* can totally be asynchronous. Off course, they can also be synchronous.
- Due to the 16x oversampling of the incoming characters, *module_clock* frequency must always be greater or equal to 16x the maximum baud rate. For example, if max baud rate is 4 Mbit/s, *module_clock* must be greater or equal to 4 M x 16 = 64MHz.

NOTE

The restriction that *peripheral_clock* frequency must be higher or equal to 16x baud rate has been removed. There is no limitation on *peripheral_clock* frequencyto baud rate.

76.3.2.2 Maximum Baud Rate

The max baud rate the UART can support is determined by the max frequency of the *module_clock* and logic synthesis results.

For example, if the SoC can provide the fastest *module_clock* 66.5 MHz and the UART synthesis timing is acceptable under this constraint, the UART can transmit and receive serial data with the maximum baud rate 66.5M/16 = 4.15 Mbit/s.

The UART supports serial IR interface low speed. In the low speed IrDA mode, the max baud rate is 115.2Kbit/s. To support the 115.2Kbit/s, *module_clock* frequency must be higher or equal to 1.8432MHz.

76.3.2.3 Clocking in Low-Power Modes

The UART supports 2 low-power modes: DOZE and STOP.

In STOP mode (input pin *stop_req* is at '1'), the UART doesn't need any clock. In this mode the UART can wake-up the ARM platform with the asynchronous interrupts (refer to Low Power Modes). An application of this feature is when the system must be waken-up by the arrival of a frame of characters.

- If before entering in Stop mode the software has enabled RTSDEN interrupt, then when RTS will change state (put at '0' by external device started to send), the

asynchronous interrupt will wake-up the system, *peripheral_clock* and *module_clock* will be provided to the UART before first start bit, so that no data will be lost.

- If RTS doesn't change state (already at '0' before entering in Stop mode), then wake-up interrupt (AWAKE) will be sent at the arrival of first Start bit (on falling edge). In this case, the UART must receive the *peripheral_clock* and *module_clock* during the first half of start bit to correctly receive this character (for example, at 115.2 Kbit/s, UART must receive *peripheral_clock* and *module_clock* at maximum 4.3 microseconds after falling edge of Start bit). If the UART receives *peripheral_clock* and *module_clock* too late, first character will be lost, and so should be dropped. Also, if autobaud detection is enabled, the first character won't be correctly received and another autobaud detection will need to be initiated.

In Doze mode, UART behavior is programmable through DOZE bit (UCR1[1]). If DOZE bit is set to '1', then UART is disabled in Doze mode, and in consequence, UART clocks can be switched-off (after being sure UART is not transmitting nor receiving). On the contrary, if DOZE bit is set to '0', UART is enabled and it must receive *peripheral_clock* and *module_clock* .

76.3.3 General UART Definitions

Definitions of terms that occur the following discussions are given in this section.

- Bit Time-The period of time required to serially transmit or receive 1 bit of data (1 cycle of the baud rate frequency).
- Start bit-The bit time of a logic 0 that indicates the beginning of a data frame. A start bit begins with a 1-to-0 transition, and is preceded by at least 1 bit time of logic 1.
- Stop bit-1 bit time of logic 1 that indicates the end of a data frame.
- BREAK-A frame in which all of the data bits, including the stop bit, are logic 0. This type of frame is usually sent to signal the end of a message or the beginning of a new message.
- Mark - When no data is being sent, the serial port's transmit pin's voltage is 1 and is said to be in a MARK state.
- Space - The serial port can also be forced to keep the transmit pin at a 0 and is said to be the SPACE or BREAK state.
- Frame-A start bit followed by a specified number of data or information bits and terminated by a stop bit. The number of data or information bits depends on the format specified and must be the same for the transmitting device and the receiving device. The most common frame format is 1 start bit followed by 8 data bits (least significant bit first) and terminated by 1 stop bit. An additional stop bit and a parity bit also can be included.

- Framing Error-An error condition that occurs when the stop bit of a received frame is missing, usually when the frame boundaries in the received bit stream are not synchronized with the receiver bit counter. Framing errors can go undetected if a data bit in the expected stop bit time happens to be a logic 1. A framing error is always present on the receiver side when the transmitter is sending BREAKs. However, when the UART is programmed to expect 2 stop bits and only the first stop bit is received, this is not a framing error by definition.
- Parity Error-An error condition that occurs when the calculated parity of the received data bits in a frame does not match the parity bit received on the RXD input. Parity error is calculated only after an entire frame is received.
- Idle-One in NRZ encoding format and selectable polarity in IrDA mode.
- Overrun Error-An error condition that occurs when the latest character received is ignored to prevent overwriting a character already present in the UART receive buffer (RxFIFO). An overrun error indicates that the software reading the buffer (RxFIFO) is not keeping up with the actual reception of characters on the RXD input.

76.3.3.1 $\overline{\text{RTS}}$ - UART Request To Send

The UART Request To Send input controls the transmitter. The modem or other terminal equipment signals the UART when it is ready to receive by setting '0' on the RTS pin.

Normally, the transmitter waits until this signal is active (low) before transmitting a character, however when the Ignore RTS (IRTS) bit is set, the transmitter sends a character as soon as it is ready to transmit. An interrupt (RTSD) can be posted on any transition of this pin and can wake the ARM platform from STOP mode on its assertion. When $\overline{\text{RTS}}$ is set to '1' during a transmission, the UART transmitter finishes transmitting the current character and shuts off. The contents of the TxFIFO (characters to be transmitted) remain undisturbed. The operation of this input is the same regardless of whether the UART is in DTE or DCE mode.

76.3.3.2 RTS Edge Triggered Interrupt

The input to the $\overline{\text{RTS}}$ pin can be programmed to generate an interrupt on a selectable edge.

See the table below for summary of the operation of the RTS edge triggered interrupt (RTSF).

To enable the $\overline{\text{RTS}}$ pin to generate an interrupt, set the request to send interrupt enable (RTSEN) bit (UCR2[4]) to 1. Writing 1 to the $\overline{\text{RTS}}$ edge triggered interrupt flag (RTSF) bit (USR2[4]) clears the interrupt flag. The interrupt can occur on the rising edge, falling

edge, or either edge of the RTS input. The request to send edge control (RTEC) field (UCR2[10:9]) programs the edge that generates the interrupt. When RTEC is set to 0x00 and RTSEN = 1, the interrupt occurs on the rising edge (default). When RTEC is set to 0x01 and RTSEN = 1, the interrupt occurs on the falling edge. When RTEC is set to 0x1X and RTSEN = 1, the interrupt occurs on either edge. This is a synchronous interrupt. The RTSF bit is cleared by writing 1 to it. Writing 0 to RTSF has no effect.

Table 76-5. RTS Edge Triggered Interrupt Truth Table

RTS	RTSEN	RTEC [1]	RTEC [0]	RTSF	Interrupt Occurs On...	interrupt_uart
X	0	X	X	0	Interrupt disabled	1
1->0	1	0	0	0	Rising edge	1
0->1	1	0	0	1	Rising edge	0
1->0	1	0	1	1	Falling edge	0
0->1	1	0	1	0	Falling edge	1
1->0	1	1	X	1	Either edge	0
0->1	1	1	X	1	Either edge	0

There is another RTS interrupt that is not programmable. The status bit RTSD asserts the *interrupt_uart* interrupt when the RTS delta interrupt enable = 1. This is an asynchronous interrupt. The RTSD bit is cleared by writing 1 to it. Writing 0 to the RTSD bit has no effect.

76.3.3.3 DTR - Data Terminal Ready

This signal indicates the general readiness of the Data Terminal Equipment (DTE). This signal is an input in DCE mode and an output in DTE mode. If the connection between the DCE and the DTE is established once, the DTR signal must remain active throughout the whole connection time.

In general the DTR and DSR signals are responsible for establishing the connection. RTS and CTS are responsible for the data transfer and the transfer direction in the case of a half-duplex configuration. The DTR signal is like a "main switch". If the DTR signal is inactive the RTS and CTS signals have no effect. In DCE mode, an interrupt (DTRD) can be posted on any transition of this pin and can wake the ARM platform from STOP mode on its assertion.

76.3.3.4 $\overline{\text{DSR}}$ - Data Set Ready

This signal indicates the general readiness of the DCE. This signal is an output in DCE mode and an input in DTE mode. The DCE uses this signal to inform the DTE that it is switched on, has completed all preparations and can communicate with the DTE.

In DTE mode, an interrupt (DTRD) can be posted on any transition of this pin and can wake the ARM platform from STOP mode on its assertion.

76.3.3.5 $\overline{\text{DTR/DSR}}$ Edge Triggered Interrupt

The $\overline{\text{DTR}}$ input pin (DCE mode) or $\overline{\text{DSR}}$ input pin (DTE mode) can be configured to cause an interrupt on a selectable edge.

See the table below for summary of the operation of the DTR/DSR edge triggered interrupt. To enable the interrupt, set the DTREN bit (UCR3[13]) to '1'. Write a "one" to the DTRF bit (USR2[13]) to clear the interrupt flag.

The interrupt can be configured to occur on either the rising, falling, or either edge of the $\overline{\text{DTR/DSR}}$ input. Write to the DPEC[1:0] bits (UCR3[15:14]) to program which edge will cause an interrupt. If the bits are set to 00b and DTREN = 1, the interrupt will occur on the rising edge (default). If the bits are set to 01b and DTREN = 1, the interrupt will occur on the falling edge. If the bits are set to 1Xb and DTREN = 1, the interrupt will occur on either edge.

Table 76-6. $\overline{\text{DTR/DSR}}$ Edge Triggered Interrupt Truth Table

$\overline{\text{DTR}}$ / $\overline{\text{DSR}}$	DTREN	DPEC[1]	DPEC[0]	DTRF	Interrupt occurs on:	interrupt_uart
X	0	X	X	0	turned off	1
1->0	1	0	0	0	rising edge	1
0->1	1	0	0	1	rising edge	0
1->0	1	0	1	1	falling edge	0
0->1	1	0	1	0	falling edge	1
1->0	1	1	X	1	either edge	0
0->1	1	1	X	1	either edge	0

76.3.3.6 $\overline{\text{DCD}}$ - Data Carrier Detect

This signal is an output in DCE mode and an input in DTE mode. If used, the DCE device uses this signal to inform the DTE it has detected the carrier signal and the connection will be set up. This signal remains active while the connection remains established.

In DTE mode this input can trigger an interrupt on changing state. This is achieved by setting to '1' the interrupt enable bit (DCD, UCR3[9]). The change state is reflected in DCDDELT (USR2[6]). Also, the state of the Data Carrier Detect input is mirrored in the status register DCDIN (USR2[5]).

76.3.3.7 $\overline{\text{RI}}$ - Ring Indicator

This signal is an output in DCE mode and an input in DTE mode. If used, the DCE device uses this signal to inform the DTE that a ring just occurred.

In DTE mode this input can trigger an interrupt on changing state. This is achieved by setting to '1' the interrupt enable bit (RI, UCR3[8]). The change state is reflected in RIDELT (USR2[10]). Also, the state of the Ring Indicator input is mirrored in the status register RIIN (USR2[9]).

76.3.3.8 $\overline{\text{CTS}}$ - Clear To Send

This output pin serves two purposes. Normally, the receiver indicates that it is ready to receive data by asserting this pin (low). When the $\overline{\text{CTS}}$ trigger level is programmed to trigger at 32 characters received and the receiver detects the valid start bit of the 33 character, it de-asserts this pin. The operation of this output is the same regardless of whether the UART is in DTE or DCE mode.

76.3.3.9 Programmable $\overline{\text{CTS}}$ Deassertion

The $\overline{\text{CTS}}$ output can also be programmed to deassert when the RxFIFO reaches a certain level. Setting the CTS trigger level (UCR4[15:10]) at any value less than 32 deasserts the $\overline{\text{CTS}}$ pin on detection of the valid start bit of the N + 1 character (where N is the trigger level setting). However, the receiver continues to receive characters until the RxFIFO is full.

76.3.3.10 TXD - UART Transmit

This is the transmitter serial output. When operating in RS-232 mode, NRZ encoded data is . When operating in infrared mode, a 3/16 bit-period pulse is output for each 0 bit transmitted, and no pulse is output for each 1 bit transmitted.

For RS-232 applications, this pin must be connected to an RS-232 transmitter. The operation of this output is the same regardless of whether the UART is in DTE or DCE mode. See Figure 76-2.

76.3.3.11 RXD - UART Receive

This is the receiver serial input. When operating in RS-232 mode, NRZ encoded data is expected. When operating in infrared mode, a narrow pulse is expected for each 0 bit received and no pulse is expected for each 1 bit received.

External circuitry must convert the IR signal to an electrical signal. RS-232 applications require an external RS-232 receiver to convert voltage levels. The operation of this input is the same regardless of whether the UART is in DTE or DCE mode. See the figure below.

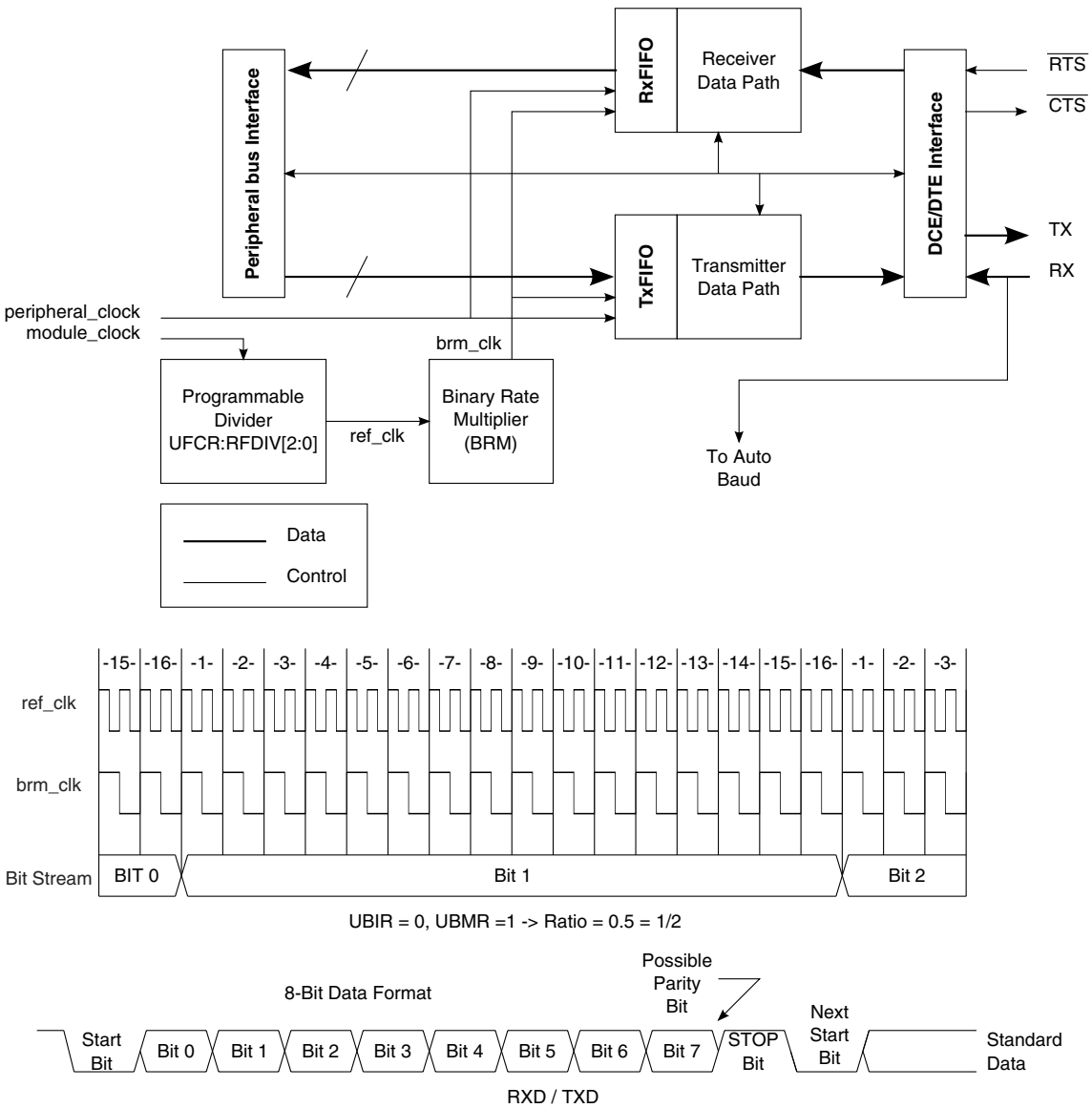


Figure 76-2. UART Simplified Block and Clock Generation Diagrams

76.3.4 Transmitter

The transmitter accepts a parallel character from the ARM platform and transmits it serially. The start, stop, and parity (when enabled) bits are added to the character.

When the ignore RTS bit (IRTS) is set, the transmitter sends a character as soon as it is ready to transmit. $\overline{RTS}$ can be used to provide flow-control of the serial data. When $\overline{RTS}$ is set to '1', the transmitter finishes sending the character in progress (if any), stops, and waits for $\overline{RTS}$ to be set to '0' again. Generation of BREAK characters and parity errors (for debugging purposes) is supported. The transmitter operates from the clock provided by the BRM. Normal NRZ encoded data is transmitted when the IR interface is disabled.

The transmitter FIFO (TxFIFO) contains 32 bytes. The data is written to TxFIFO by writing to the UTXD register with the byte data to the [7:0] bits. The data is written consecutively if the TxFIFO is not full. It is read (internally) consecutively if the TxFIFO is not empty. TXFULL bit (UTS[4]) can be used to control whether TXFIFO is full or not. The TxFIFO can be written regardless of the transmitter is disabled or enabled. If the UART is disabled, user can still write data into the TxFIFO correctly. But in this case the write access will yield to a transfer error.

76.3.4.1 Transmitter FIFO Empty Interrupt Suppression

The transmitter FIFO empty interrupt suppression logic suppresses the TXFE interrupt between writes to the TxFIFO.

When TxFIFO is empty, the software can either send one or several characters. If the software sends one character, it would write the character into the UTXD register, then that character is immediately transferred to the transmitter shift register, assuming the transmitter is already enabled. Without interrupt suppression logic, the TXFE interrupt flag would be set immediately. But, with this logic, the interrupt flag is set when the last bit of the character has been transmitted, for example, before the transmission of the parity bit (if exists) and the stop bit(s).

So, the suppression logic doesn't immediately send the TXFE interrupt flag. It allows the software to write another character to the TxFIFO before the interrupt flag is asserted.

When the transmitter shift register empties before another character is written to the TxFIFO, the interrupt flag is asserted. Writing data to the TxFIFO would release the interrupt flag. The interrupt flag is asserted on the following conditions:

- System Reset
- UART software reset

- When a single character has been written to Transmitter FIFO and then the Transmitter FIFO and the Transmitter Shift Register become empty until another character is written to the Transmitter FIFO
- The last character in the TxFIFO is transferred to the shift register, when TxFIFO contains two or more characters. See the figure below.

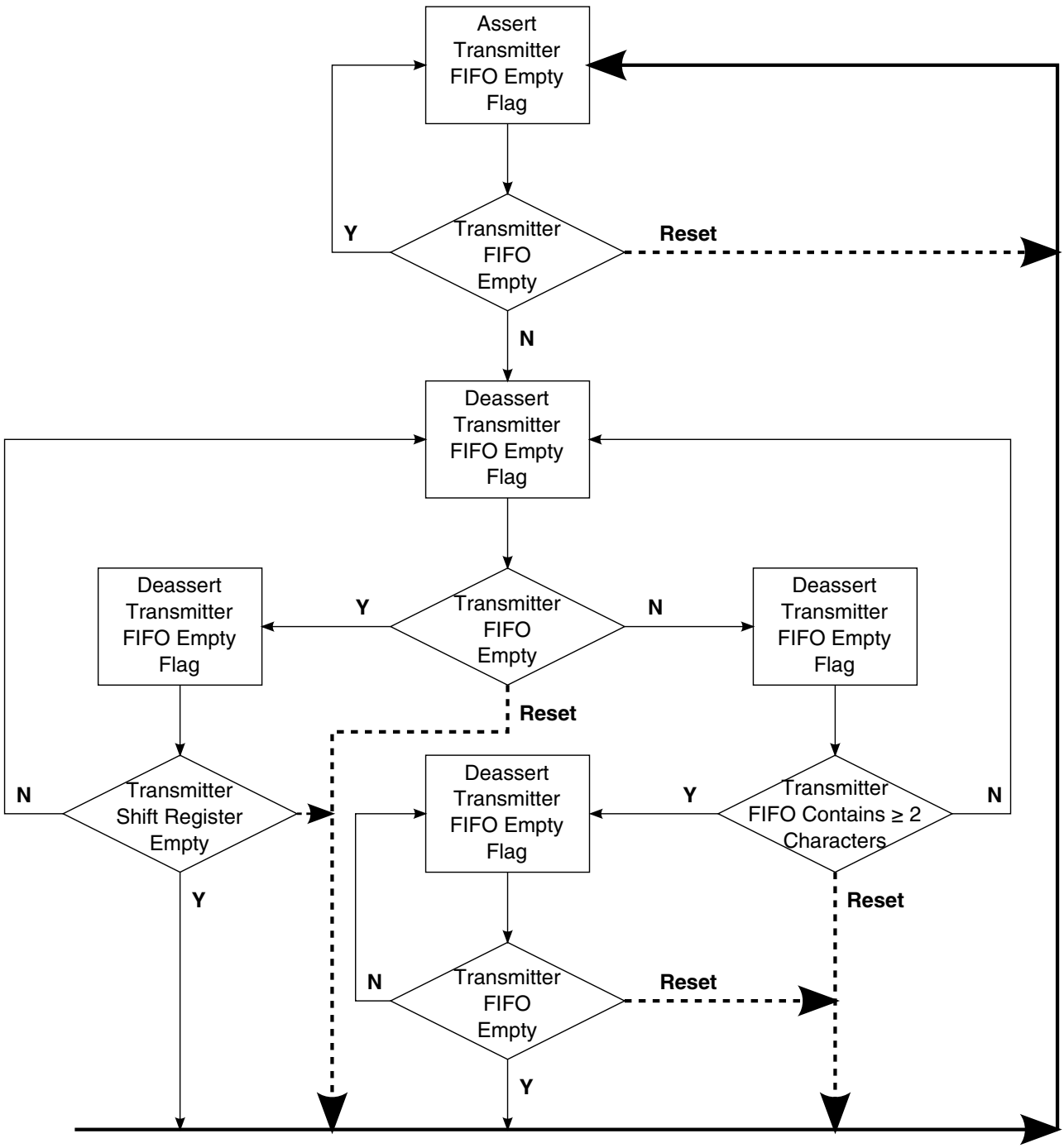


Figure 76-3. Transmitter FIFO Empty Interrupt Suppression Flow Chart

76.3.4.2 Transmitting a Break Condition

Asserting SNDBRK bit of the UCR1 Register forces the transmitter to send a break character (continuous zeros). The transmitter will finish sending the character in progress (if any) before sending break until this bit is reset.

The user is responsible to ensure that this bit is high for long enough to generate a valid BREAK. The transmitter samples SNDBRK after every bit is transmitted. Following completion of the BREAK transmission, the UART will transmit two mark bits. The user can continue to fill the FIFO and any character remaining will be transmitted when the break is terminated.

76.3.5 Receiver

See the figure below for the receiver flow chart.

The receiver accepts a serial data stream and converts it into parallel characters. When enabled, it searches for a start bit, qualifies it, and samples the following data bits at the bit-center. Jitter tolerance and noise immunity are provided by sampling at a 16x rate and using voting techniques to clean up the samples.

Once the start bit is found, the data bits, parity bit (if enabled), and stop bits (either 1 or 2 depending on user selection) are shifted in. Parity is checked and its status reported in the URXD register when parity is enabled. Frame errors and BREAKs are also checked and reported. When a new character is ready to be read by the ARM platform from the RxFIFO, the receive data ready (RDR = USR2[0]) bit is asserted and an interrupt is posted (if DREN = UCR4[0] = 1). If the receiver trigger level is set to 2 (RXTL[5:0] = UFCR[5:0] = 2), and 2 chars have been received into RxFIFO, the receiver ready interrupt flag (RRDY = USR1[9]) is asserted and an interrupt is posted if the receiver ready interrupt enable bit is set (RRDYEN = UCR1[9] = 1). If the UART Receiver Register (URXD) is read once, and in consequence there is only 1 character in the RxFIFO, the interrupt generated by the RDR bit is automatically cleared. The RRDY bit is cleared when the data in the RxFIFO falls below the programmed trigger level.

Normal NRZ encoded data is expected when the IR interface is disabled.The RxFIFO contains 32 half-word entries. Characters received are written consecutively into this FIFO. If the FIFO is full and a 33rd characters is received, this character will be ignored and the USR2[ORE] bit will be set.

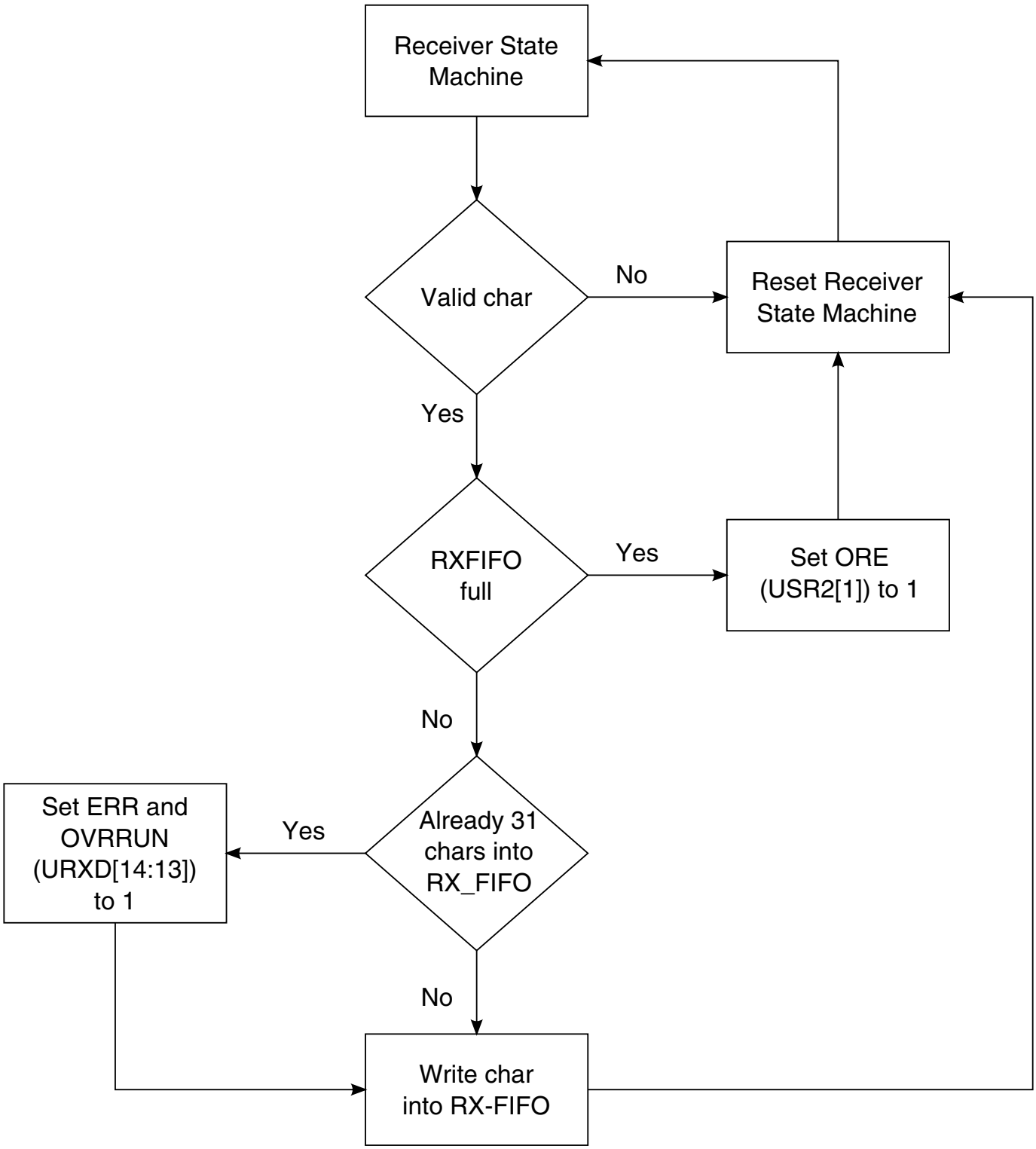


Figure 76-4. Receiver Flow Chart

76.3.5.1 Idle Line Detect

The receiver logic block includes the ability to detect an idle line. Idle lines indicate the end or the beginning of a message.

For an idle condition to occur:

- RxFIFO must be empty and
- RXD pin must be idle for more than a configured number of frames (ICD[1:0] = UCR1[11:10]).

When the idle condition detected interrupt enable (IDEN = UCR1[12]) is set and the line is idle for 4 (default), 8, 16, or 32 (maximum) frames, the detection of an idle condition flags an interrupt (see the table below). When an idle condition is detected, the IDLE (USR2[12]) bit is set. Clear the IDLE bit by writing 1 to it. Writing 0 to the IDLE bit has no effect.

Table 76-7. Detection Truth Table

IDEN	ICD [1]	ICD [0]	IDLE	*interrupt_uart*
0	X	X	0	1
1	0	0	asserted after 4 idle frames	asserted after 4 idle frames
1	0	1	asserted after 8 idle frames	asserted after 8 idle frames
1	1	0	asserted after 16 idle frames	asserted after 16 idle frames
1	1	1	asserted after 32 idle frames	asserted after 32 idle frames
NOTE: This table assumes that no other interrupt is set at the same time this interrupt is set for the *interrupt_uart* signal. This table shows how this interrupt affects the *interrupt_uart* signal.				

During a normal message there is no idle time between frames. When all of the information bits in a frame are logic 1s, the start bit ensures that at least one logic 0 bit time occurs for each frame so that the IDLE bit is not asserted.

76.3.5.2 Aging Character Detect

The receiver block also includes the possibility to detect when at least one character has been sitting into the RxFIFO for a time corresponding to 8 characters. This aging character capability allows the UART to inform the ARM platform that there is less character into the RxFIFO than the Rx trigger and, no new character has been detected on the RXD line.

The aging capability is a timer which starts to count as soon as the RxFIFO is not empty and its trigger level is not reached (RRDY=0). This counter is reset when either a RxFIFO read is performed or another character starts to present on the RXD line. If none of those two events occurs, the bit AGTIM (USR1[8]) is set when the counter has

measured a time corresponding to 8 characters. AGTIM is cleared by writing a 1 to it. AGTIM can flag an interrupt to ARM platform on *interrupt_uart* if ATEN (UCR2[3]) has been set.

To summarize, AGTIM is set when:

- There is at least one character into RxFIFO.
- No read has occurred on RxFIFO and RXD line has stayed high, for a time corresponding to 8 characters.
- The RxFIFO trigger is not reached (RRDY=0)

76.3.5.3 Receiver Wake

The WAKE bit (USR2[7]) is set when the receiver detects a qualified Start bit. For this, two conditions must be fulfilled, firstly a falling edge on RXD line must be detected and secondly the RXD line must stays at low level for more than a half-bit duration.

When the wake interrupt enable WKEN (UCR4[7]) bit is enabled, the receiver flags an interrupt (*interrupt_uart*) if the WAKE status bit is set. The WAKE bit is cleared by writing 1 to it. Writing 0 to the WAKE bit has no effect. The WAKE status bit can be asserted in either serial RS-232 mode or IR mode. The generation of the WAKE interrupt needs the clock *module_clock* .

When the asynchronous wake interrupt (AWAKE) is enabled (AWAKEN = UCR3[4] = 1), and the ARM platform is in STOP mode, and UART clocks have been shut-off, then a falling edge detected on the receive pin (RXD) asserts the AWAKE bit (USR1[4]) and the *interrupt_uart* interrupt to wake the ARM platform from STOP mode. Re-enable UART clocks and clear the AWAKE bit by writing 1 to it. Writing 0 to the AWAKE bit has no effect. When IR interface is enabled (UCR1[7]=1), the AWAKE bit is always not asserted. The generation of the asynchronous AWAKE interrupt does not need any clocks.

In IR mode, if the asynchronous IR WAKE interrupt is enabled (AIRINTEN = UCR3[5] = 1), and if the ARM platform is in STOP mode (UART clocks are off when ARM platform in STOP mode), then the detection of a falling edge on the receive pin (RXD_IR), asserts the AIRINT bit (USR1[5]), and the *interrupt_uart* interrupt. This interrupt wakes the ARM platform from STOP mode. Software re-enables UART clocks and clear the AIRINT bit by writing 1 to it. Writing 0 to the AIRINT bit has no effect. When IR interface is disabled (UCR1[7]=0), the AIRINT bit is always not asserted. The generation of the asynchronous AIRINT interrupt does not need any clocks.

Recommended procedure for programming the asynchronous interrupts is to first clear them by writing 1 to the appropriate bit in the UART Status Register 1 (USR1). Poll or enable the interrupt for the Receiver IDLE Interrupt Flag (RXDS) in the USR1. When asserted, the RXDS bit indicates to the software that the receiver state machine is in the idle state, the next state is idle, and the RXD pin is idle (high). After following this procedure, enable the asynchronous interrupt and enter STOP mode.

76.3.5.4 Receiving a BREAK Condition

A BREAK condition is received when the receiver detects all 0s (including a 0 during the bit time of the stop bit) in a frame. The BREAK condition asserts the BRCD bit (USR2[2]) and writes only the first BREAK character to the RxFIFO. Clear the BRCD bit by writing 1 to it. Writing 0 to the BRCD bit has no effect.

Asserting BRCD would generate an interrupt on *interrupt_uart*. The interrupt generation can be masked using the control bit BKEN (UCR4[2]). Receiving a break condition will also effect the following bits in the receiver register URXD:

URXD(11) = BRK. While high this bit indicates that the current char was detected as a break.

URXD(12) = FRMERR. The frame error bit will always be set when BRK is set.

URXD(10) = PRERR. If odd parity was selected the parity error bit will also be set when BRK is set.

URXD(14) = ERR. The error detect bit indicates that the character present in the rx data field has an error status. This can be asserted by a break.

76.3.5.5 Vote Logic

The vote logic block provides jitter tolerance and noise immunity by sampling with respect to a 16x clock (*brm_clk*) and using voting techniques to clean up the samples. The voting is implemented by sampling the incoming signal constantly on the rising edge of the *brm_clk*.

See Figure 76-5. The receiver is provided with the majority vote value, which is 2 out of the 3 samples. For examples of the majority vote results of the vote logic, see the following table.

Table 76-8. Majority Vote Results

Samples	Vote
000	0
101	1
001	0
111	1

The vote logic captures a sample on every rising edge of *brm_clk*, however the receiver uses 16x oversampling to take its value in the middle of the sample character.

The receiver starts to count when the Start bit is set however it does not capture the contents of the RxFIFO at the time the Start bit is set. The start bit is validated when 0s are received for 7 consecutive 1/16 of bit times following the 1-to-0 transition. Once the counter reaches 0xF, it starts counting on the next bit and captures it in the middle of the sampling frame (see Table 76-8). All data bits are captured in the same manner. Once the stop bit is detected, the receiver shift register (SIPO_OUT) data is parallel shifted to the RxFIFO.

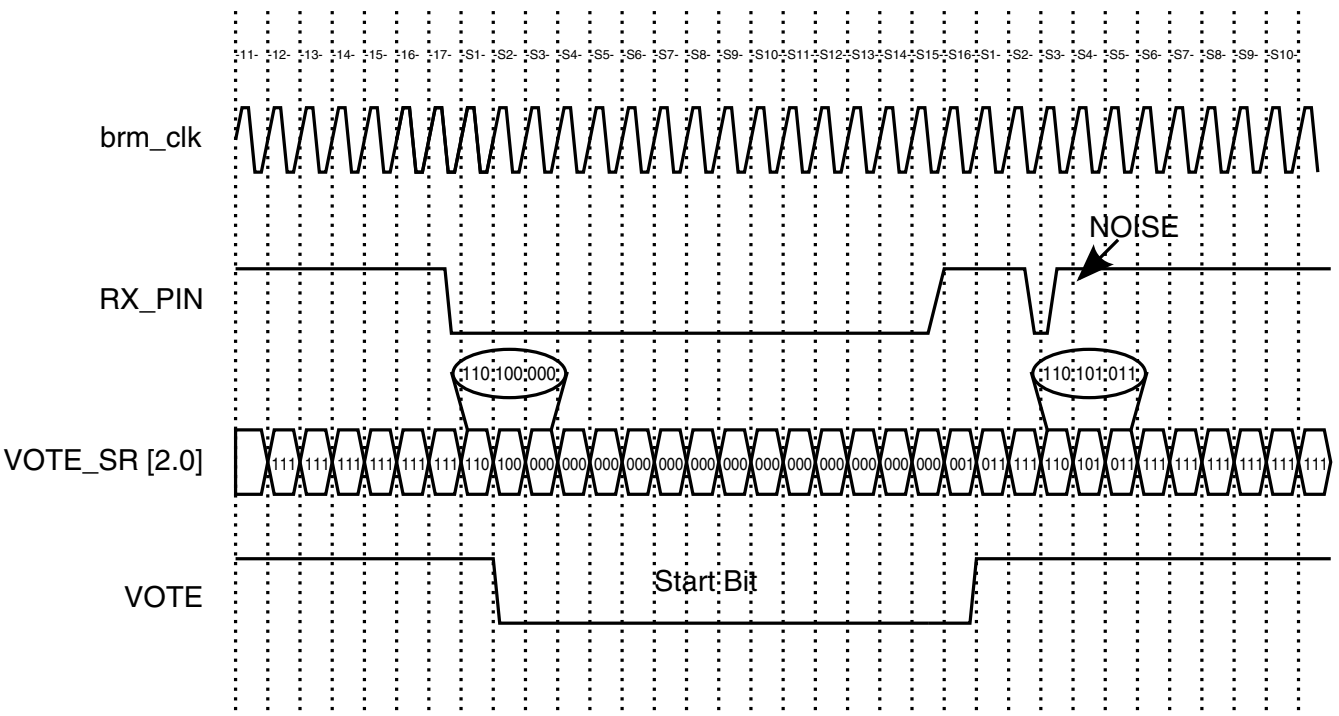


Figure 76-5. Majority Vote Results

A new feature has been recently implemented, it allows to re-synchronize the counter on each edge of RXD line. This is automatic and allows to improve the immunity of UART against signal distortion.

There is a special case when the *brm_clk* frequency is too low and is unable to capture a 0 pulse in IrDA. In this case, the software must set the IRSC (UCR4[5]) bit so that the reference clock (after internal divider) is used for the voting logic. The pulse is validated by counting the length of the pulse.

Refer to Infrared Interface for more details.

76.3.5.6 Baud Rate Automatic Detection Logic

When the baud rate automatic detection logic is enabled, the UART locks onto the incoming baud rate. To enable this feature, set the automatic detection of baud rate bit (ADBR = UCR1[14] = 1) and write 1 to the ADET bit (USR2[15]) to clear it.

When ADET=0 and ADBR =1, the detection starts. Then, once the beginning of start bit (transition from 1-to-0 of RXD) has been detected, UART start a counter (UBRC) working at reference frequency. Once the end of start bit is detected (transition from 0-to-1 of RXD), the value of UBRC - 1 is directly copied into UBMR register. UBIR register is filled with 0x000F.

So, at the end of start bit, registers gets following values:

```
UBRC = number of reference clock periods (after divider) during Start bit.
UBIR = 0x000F
UBMR = UBRC - 1
```

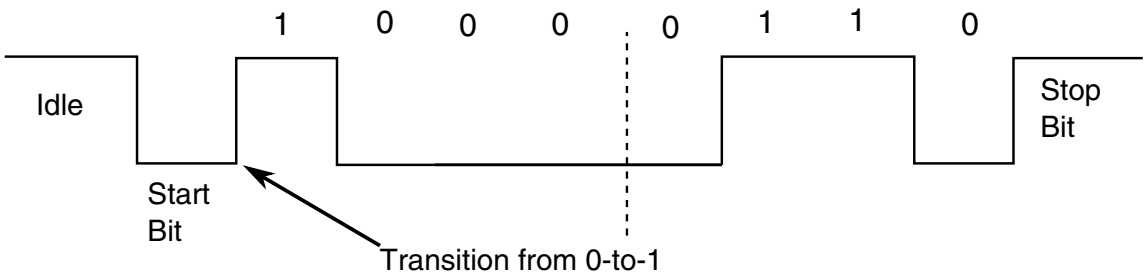
The updated values of the 3 registers can be read.

See Table 76-9 for list of parameters for baud rate detection and Figure 76-6 for baud rate detection protocol diagram.

If any of the UART BRM registers are simultaneously written by the baud rate automatic detection logic and by the peripheral data bus, the peripheral data bus would have lower priority.

Table 76-9. Baud Rate Automatic Detection

ADBR	ADET	Baud Rate Detection	*interrupt_uart*
0	X	Manual Configuration	1
1	0	Auto Detection Started	1
1	1	Auto Detection Complete	0
NOTE: This table assumes that no other interrupt is set at the same time this interrupt is set for the *interrupt_uart* signal.			



Note: LSB Transmitted first.

Figure 76-6. Baud Rate Detection Protocol Diagram

76.3.5.6.1 Baud Rate Automatic Detection Protocol

The receiver must receive an ASCII character "A" or "a" to verify proper detection of the incoming baud rate. When an ASCII character "A" (0x41) or "a" (0x61) is received and no error occurs, the Automatic Detect baud rate bit is set (ADET=1) and if the interrupt is enabled (ADEN=UCR1[15]=1), an interrupt *interrupt_uart* is generated.

When an ASCII character "A" or "a" is not received (because of a bit error or the reception of another character), the auto detection sequence restarts and waits for another 1-to-0 transition.

As long as ADET = 0 and ADBR = 1, the UART continues to try to lock onto the incoming baud rate. Once the ASCII character "A" or "a" is detected and the ADET bit is set, the receiver ignores the ADBR bit and continues normal operation with the calculated baud rate.

The UART interrupt is active (*interrupt_uart* = 0) as long as ADET = 1 and ADBR = 1. This can be disabled by clearing the automatic baud rate detection interrupt enable bit (ADEN = 0). Before starting an automatic baud rate detection sequence, set ADET = 0 and ADBR = 1.

The RxFIFO must contain the ASCII character "A" or "a" following the automatic baud rate detection interrupt.

The 16-bit UART Baud Rate Count Register (UBRC) is reset to 4 and stays at 0xFFFF when an overflow occurs. The UBRC register counts (measures) the duration of start bit. When the start bit is detected and counted, the UART Baud Rate Count Register retains its value until the next automatic baud rate detection sequence is initiated.

The Baud Rate Count Register counts only when auto detection is enabled.

76.3.5.6.2 New Baud Rate Determination

In order to fight against the problems caused by the distortion and the noise on the RXD line, the duration of the baud rate measurement has been extended.

Previously, as described above, this determination was based on the measurement of the START bit duration. Now, this measurement is based on the duration of START bit + bit0. Bit0 is the first bit following the START bit. In fact, the counter which is started at the falling edge of START bit is no longer stopped at next rising edge (end of START bit), but it is stopped at the next falling edge (end of bit0). As the character sent is always a "A" (41h) or a "a" (61h), this second falling edge will always be present and it will indicate the end of bit0. Once this counter is stopped, the result is divided by 2 and used by the BRM to determine the incoming baud rate.

NOTE

UBRC register contains the result of this division by two, in consequence it reflects the measurement of the duration of one bit.

76.3.5.6.2.1 New Autobaud Counter Stopped bit and Interrupt

A new bit has been added in USR2 register: ACST (USR2[11]). This bit is set immediately after the determination of the baud rate.

So,

- if ADNIMP is not set (default), ACST is set to 1 after the end of bit0,
- If ADNIMP is set to 1, ACST is set to 1 at the end of START bit.

If ACIEN (UCR3[0]) is set to 1, ACST will flag an interrupt on *interrupt_uart* signal. This interrupt informs the ARM platform that the BRM has just been set with the result of the bit length measurement. If needed, the ARM platform can perform a read of UBMR (or UBRC) register and determine by itself the baud rate measured. Then the ARM platform has the possibility to correct the BRM registers with the nearest standardized baud rate.

NOTE

ACST is set only if ADBR is set to 1, for example, the UART is autobauding.

Clear the ACST bit by writing 1 to it. Writing 0 to the ACST bit has no effect.

76.3.6 Escape Sequence Detection

An escape sequence typically consists of 3 characters entered in rapid succession (such as +++). Because these are valid characters by themselves, the time between characters determines if it is a valid escape sequence. Too much time between two of the "+" characters is interpreted as two "+" characters, and not part of an escape sequence.

The software chooses the escape character and writes its value to the UART Escape Character Register (UESC). The software must also enable escape detection feature by setting ESCEN (UCR2[11]) to 1. The hardware compares this value to incoming characters in the RxFIFO. When an escape character is detected, the internal escape timer starts to count. The software specifies a time-out value for the maximum allowable time between 2 successive escape characters (see the table below). The escape timer is programmable in intervals of 2 ms to a maximum interval of 8.192 seconds.

Table 76-10. Escape Timer Scaling

UTIM Register	Maximum Time Between Specified Escape Characters
0x000	2 ms
0x001	4 ms
0x002	6 ms
0x003	8 ms
0x004	10 ms
...	...
0F8	498 ms
0F9	500 ms
...	...
9C3	5 s
...	...
FFD	8.188 s
FFE	8.190 s
FFF	8.192 s
NOTE: To calculate the time interval: (UTIM_Value + 1) x 0.002 = Time_Interval Example: (09C3 + 1) x 0.002 = 5 s.	

The escape sequence detection feature is available for all the reference frequencies. Before using Escape Sequence Detection, the user must fill the ONEMS register. This 24-bit register must contain the value of the UART internal frequency divided by 1000. The internal frequency is obtained after the UART internal divider which is applied on *module_clock* clock.

Example I:

- If the input clock *module_clock* frequency is 66.5 MHz.
- And if the input clock *module_clock* is divided by 2 with the internal divider: UFCR[9:7] = 3'b100

$$ONEMS = \frac{66.5 \times 10^6}{2 \times 1000} = 33250 = 81E2h$$

Figure 76-7. Calculation of Frequency for ONEMS Register

Example II:

- If the input clock *module_clock* frequency is 66.5 MHz.
- And if the input clock *module_clock* is divided by 1with the internal divider: UFCR[9:7] = 3'b101

$$ONEMS = \frac{66.5 \times 10^6}{1000} = 66500 = 103C4h$$

Figure 76-8. Calculation of Frequency for ONEMS Register

The escape sequence detection interrupt is asserted when the escape sequence interrupt enable (ESCI) bit is set and an escape sequence is detected (ESCF set). Clear the ESCF bit by writing 1 to it. Writing 0 to the ESCF bit has no effect.

76.4 Binary Rate Multiplier (BRM)

The BRM sub-block receives *ref_clk* (*module_clock* clock after divider). From this clock, and with integer and non-integer division, BRM generates a 16x baud rate clock whose frequency is 16 times of baud rate.

The uart transmitter will shift data out based on this 16x baud rate clock. The uart receiver will sample the serial data line based on this 16x baud rate clock. The input and output frequency ratio is programmed in the UART BRM Incremental Register (UBIR) and UART BRM MOD Register (UBMR). The output frequency is divided by the input frequency to produce this ratio. For integer division, set the UBIR = 0x000F and write the divisor to the UBMR register. All values written to these registers must be one less than the actual value to eliminate division by 0 (undefined), and to increase the maximum range of the registers.

Updating the BRM registers requires writing to both registers. The UBIR register must be written before writing to the UBMR register. If only one register is written to by the software, the BRM continues to use the previous values.

The following examples show how to determine what values are to be programmed into UBIR and UBMR for a given reference frequency and desired baud rate. The following equation can be used to help determine these values:

$$\text{BaudRate} = \frac{\text{Ref Freq}}{\left(16 \times \frac{\text{UBMR} + 1}{\text{UBIR} + 1}\right)}$$

Figure 76-9. Frequency and Baud Rate for UBIR and UBMR

With:

Reference Frequency (Hz): UART Reference Frequency (*module_clock* after RFDIV divider)

Baud Rate (bit/s): Desired baud rate.

Integer Division ÷ 21

Reference Frequency = 19.44 MHz

UBIR = 0x000F

UBMR = 0x0014

Baud Rate = 925.7 kbit/s

NOTE

Observe that each value written to the registers is one less than the actual value.

Non-Integer Division

```
Reference Frequency = 16 MHz
Desired Baud Rate = 920 Kbits/s
```

$$\frac{UBMR + 1}{UBIR + 1} = \frac{RefFreq}{16 \times BaudRate} = \frac{16 \times 10^6}{16 \times 920 \times 10^3} = 1.087$$

```
Ratio = 1.087 = 1087 / 1000
UBIR = 999 (decimal)= 0x3E7
UBMR = 1086 (decimal)= 0x43E
Non-Integer Division
Reference Frequency = 25 MHz
Desired Baud Rate = 920 kbit/s
Ratio = 1.69837 = 625 / 368
UBIR = 367 (decimal)= 0x16F
UBMR = 624 (decimal)= 0x270
```

Non-Integer Division

```
Reference Frequency: 30 MHz
Desired Baud Rate = 115.2 kbit/s
Ratio = 16.276043 = 65153 / 4003
UBIR = 4002 (decimal) = 0x0FA2
UBMR = 65152 (decimal) = 0xFE80
```

76.5 Infrared Interface

76.5.1 Generalities-Infrared

The Infrared interface is selected when IREN (UCR1[7]) is set to 1.

The Infrared Interface is compatible with IrDA Serial Infrared Physical Layer Specification. In this specification, a "zero" is represented by a positive pulse, and a "one" is represented by no pulse (line remains low).

In the UART:

In TX: For each "zero" to be transmitted, a narrow positive pulse which is 3/16 of a bit time is generated. For each "one" to be transmitted no pulse is generated (output is low). External circuitry has to be provided to drive an Infrared LED.

In RX: When receiving, a narrow negative pulse is expected for each "zero" transmitted while no pulse is expected for each "one" transmitted (input is high).

NOTE

Rx part of IR block expects to receive an inverted signal compared to IrDA specification. Circuitry external to the IC transforms the Infrared signal to an electrical signal.

The IR interface has an edge triggered interrupt (IRINT). This interrupt validates a zero bit being received. This interrupt is enabled by writing a "one" to ENIRI bit.

The behavior of Infrared Interface is determined by 3 bits INVT (UCR3[1]), INVR (UCR4[9]) and IRSC (UCR4[5]).

76.5.2 Inverted Transmission and Reception bits (INVT & INVR)

The values of INVT and INVR depend of the IrDA transceiver connected on the TXD_IR and RXD_IR pins of the UART. If this transceiver is not inverting on both paths Tx and Rx, a Zero is represented by a positive pulse and a One is represented by no pulse (line remains low). In this case, the bit INVT must be set to 0 and the bit INVR must be set to 1 (because Rx IR block expects an inverted signal).

On the contrary user must set INVT=1 and INVR=0 if both paths of the transceiver are inverting, that is, a Zero is represented as a negative pulse and a One is represented by no pulse (line remains high). The transceiver can also be inverting on only one path (Tx or Rx), in this case INVT and INVR must be together equal to 1 or to 0, depending on which path is inverted.

76.5.3 InfraRed Special Case (IRSC) Bit

The value to apply to IRSC bit is based on 2 parameters: the baud rate and the Minimum Pulse Duration (MPD) of the transceiver.

According to IrDA Standard Specification, for SIR (Serial IR) baud rates from 2.4 Kbit/s to 115.2 Kbit/s this nominal pulse duration is equal to 3/16 of a bit duration (at the selected baud rate). But, for all the baud rates a Minimum Pulse Duration is also specified. According to IrDA Standard, a Zero is represented by a light pulse, so the IrDA transceiver can't emit a light pulse shorter than the MPD. For SIR, the MPD is constant and equal to 1.41 us.

But user must take into account the electrical MPD associated to the transceiver on the receiver path. Typically this value is 2.0 us, but for some manufacturers MPD can go down to 1.0 us.

In order to understand the meaning of IRSC bit, one must understand how the RX path work in IrDA mode.

When UART is in IrDA mode, a Zero is not only detected by the state of the RXD_IR line, but also with the duration of the pulse. This pulse duration can be measured with 2 different clocks. In this case, clock is selected with the IRSC bit.

- If IRSC = 0, the clock used is the BRM clock.
- If IRSC = 1, the clock used is the UART internal clock (UART clock after the divider (RFDIV)).

In normal operation, IRSC=0. This means at any time, the user must ensure that the frequency of BRM_clock is high enough to measure the pulse. In the UART and for IRSC=0, the pulse must last at least 2 BRM clock cycles.

If this condition is not fulfilled, IRSC must be set to 1.

Let's take 2 examples, with the Minimum Pulse Duration equals to the MPD of the IrDA specification (in SIR).

Calculation of BRM Clock Period (Clock Period < 1.41 µs)

The user wants to receive IrDA data at 115.2 Kbit/s. The UBIR and UBMR registers are set in order to create the BRM_clock with a frequency of 16*baud rate = 16 * 115.2 = 1.843 MHz. But at the same time, in order to correctly detect the pulse, the user must be sure that 2* BRM_clock period is lower than 1.41 us. Lets check:

BRM_clock period = 1/1843000 = 542 ns

So 2*BRM_clock period = 1.09 us < 1.41 us. It is fine.

Calculation of BRM Clock Period (Clock Period > 1.41 µs)

This time the user wants to receive at 19.2 Kbit/s. So, the BRM_clock is set to 16*19200 = 307.2 kHz. Let's check if 2* BRM_clock period < 1.41 us:

1. BRM_clock period =1/307200 = 3.25 us

   So 2*BRM_clock period = 6.50 us >> 1.41 us. It doesn't work.

   So, in this case, the BRM clock can't be used to measure the pulse duration and the user must select the UART internal clock by setting IRSC =1.

NOTE

Like for Escape character detection, when IR Special Case is enabled (IRSC=1), the UART must measure a duration. In order to do that, the user must fill the ONEMS register. Refer to Escape Sequence Detection.

76.5.4 IrDA interrupt

Serial infrared mode (SIR) uses an edge triggered interrupt flag IRINT (USR2[8]). When INVR =0, detection of a falling edge on the RXD pin asserts the IRINT bit. When INVR=1, detection of a rising edge on the RXD pin asserts the IRINT bit. When IRINT and ENIRI bits are both asserted, the *interrupt_uart* interrupt is asserted. Clear the IRINT bit by writing 1 to it. Writing 0 to the IRINT bit has no effect.

76.5.5 Conclusion about IrDA

Before using the UART in IrDA, the baud rate limit must be calculated. This baud rate limit will inform the user if IRSC bit has to be set or not.

Let's determine this limit:

As already described, if IRSC = 0, the following condition must always be fulfilled

$$2 \times BRMClockPeriod < MinPulseDuration$$

Figure 76-10. Calculation of Baud Rate

So,

$$BRMClockFrequency > \frac{2}{MPD}$$

So, knowing BRM_clock frequency = 16 * Baud Rate, we get:

$$BaudRate > \frac{1}{8 \times MinPulseDuration}$$

So, the user needs to set IRSC = 0 when:

- If Minimum Pulse Duration = 2.5 us and Baud Rate > 50 Kbit/s.
- If Minimum Pulse Duration = 2.0 us and Baud Rate > 62.5 Kbit/s.
- If Minimum Pulse Duration = 1.41 us and Baud Rate > 88.6 Kbit/s.

NOTE

For baud rates lower than the limit, IRSC must be set to1.

76.5.6 Programming IrDA Interface

76.5.6.1 High Speed

As an example, the following sequence can be used to program the IrDA interface in order to send and receive characters at 115.2 Kbit/s.

Assumptions:

- Input UART clock = 90 MHz
- Internal clock divider = 3 (divide Input UART clock by 3)
- Baud rate = 115.2 Kbit/s
- IrDA transceiver is not inverting on both channels: for Tx and Rx, a Zero is represented by a positive pulse, and a One is represented by no pulse (line stays low).
- Interrupt: Sent to ARM platform when 1 char is received into the Rx FIFO (RDR)

Registers values and Programming orders:

```
UCR1 = 0x0085
UCR1[7] = IREN = 1: Enable IR interface
UCR1[0] = UARTEN = 1: Enable UART
UTS = 0x0000
UFCR = 0x0981
TXTL[5:0] = 0x02: Default value
RFDIV[2:0] = 0x3: Divide Input UART clock by 3 (resulting internal clock  is 30 MHz)
RXTL[5:0] = 0x01: Default value
UBIR = 0x0202
UBMR = 0x20BE Baud rate = 115.2 kbit/s with internal clock = 30 MHz
UCR2 = 0x4027
UCR2[14] = IRTS = 1: Ignore level of RTS input signal
UCR2[5] = WS = 1: Characters are 8-bit length
UCR2[2] = TXEN = 1: Enable Rx path
UCR2 [1] = RXEN = 1: Enable Tx path
UCR2[0] = SRST_B = 1: No software reset
UCR3 = 0x0000

         UCR4 = 0x8201
CTSTL[5:0] = 0x20: Default value
UCR4[9] = INVR = 1: Inverted Infrared Reception (because IrDA transceiver is not inverting)
UCR4[1] = DREN = 1: To enable RDR interrupt (sent when one char is received)
```

The UART is ready to send a character as soon as there is a write into UTXD register. And an interrupt will be sent to ARM platform when a character is received.

76.5.6.2 Low Speed

This time, we keep the same assumptions but the speed is now 9.6 Kbit/s. So, this baud rate is below the limit (even with a Min. Pulse Duration of 2.5 us) and thus IRSC must be set to 1.

Assumptions:

- Input UART clock = 90 MHz
- Internal clock divider = 3 (divide Input UART clock by 3)
- Baud rate = 9.6 Kbit/s
- IrDA transceiver is not inverting on both channels: for Tx and Rx, a Zero is represented by a positive pulse, and a One is represented by no pulse (line stays low).
- Interrupt: Sent to ARM platform when 1 char is received into the Rx FIFO (RDR).

Registers values and Programming orders:

```
UCR1 = 0x0085
UCR1[7] = IREN = 1: Enable IR interface
UCR1[0] = UARTEN = 1: Enable UART
UFCR = 0x0981
UFCR[15:10] = TXTL[5:0] = 0x02: Default value
RFDIV[2:0] = 0x3: Divide Input UART clock by 3 (resulting internal clock  is 30 MHz)
UFCR[5:0] = RXTL[5:0] = 0x01: Default value
UBIR = 0x00FF
UBMR = 0xC354 Baud rate = 9.6 kbit/s with internal clock = 30 MHz
UCR2 = 0x4027
UCR2[14] = IRTS = 1: Ignore level of RTS input signal
UCR2[5] = WS = 1: Characters are 8-bit length
UCR2[2] = TXEN = 1: Enable Rx path
UCR2 [1] = RXEN = 1: Enable Tx path
UCR2[0] = SRST_B = 1: No software reset
UCR3 = 0x0000
UCR3[1] = INVT = 0: Positive pulse represents 0.
UCR4 = 0x8221
UCR4[15:10] = CTSTL[5:0] = 0x20: Default value
UCR4[9] = INVR = 1: Inverted Infrared Reception (because IrDA transceiver is not inverting)
UCR4[5] = IRSC = 1: Because data rate is below the limit and thus the UART internal clock is
used to measure the pulse duration.
UCR4[1] = DREN = 1: To enable RDR interrupt (sent when one char is received)
```

The UART is now ready to send a character as soon as there is a write into UTXD register. An interrupt will be sent to ARM platform when a character is received.

76.6 Low Power Modes

These modes are controlled by the signals *doze_req* and *stop_req*. The control/status/data registers won't change when getting into/out of low power modes.

Table 76-11. UART Low Power State Operation

	Normal State (*doze_req* = 1'b0 & *stop_req* = 1'b0)	Doze State (*doze_req* = 1'b1)		Stop State (*stop_req* = 1'b1)
		DOZE bit = 0	DOZE bit = 1	
UART-Clock	ON	ON	ON	OFF
UART Serial / IrDA	ON	ON	OFF	OFF

76.6.1 UART Operation in System Doze Mode

While in Doze State (when *doze_req* input pin is set to 1'b1), the UART behavior depends on the DOZE (UCR1[1]) control bit.

While the DOZE bit is negated, the UART serial interface is enabled. While the system is in the Doze State, and the DOZE bit is asserted, the UART is disabled. If the Doze State is entered with the DOZE bit asserted while the UART serial interface was receiving or transmitting data, it will complete the receive/transmit of the current character and signal to the far-end transmitter/receiver to stop sending/receiving.

76.6.2 UART Operation in System Stop Mode

The internal baud rate clocks of the transmitter and receiver are gated off if the *stop_req* signal to UART is asserted. Even though the clocks at the input of the UART continue to run during system Stop mode, the UART will not do any transmission or reception.

The following UART interrupts wake the ARM platform processor from STOP mode:

- RTS (RTSD)
- IrDA Asynchronous WAKE (AIRINT)
- Asynchronous WAKE (AWAKE)
- RI (RIDELT in DTE mode only)
- DCD (DCDDELT in DTE mode only)
- DTR (DTRD in DCE mode only)
- DSR (DTRD in DTE mode only)

When an asynchronous WAKE (awake) interrupt exits the ARM platform from STOP mode, make sure that a dummy character is sent first because the first character may not be received correctly.

76.6.3 Power Saving Method in UART

The RXEN (UCR2[1]), TXEN (UCR2[2]) and UARTEN (UCR1[0]) bits are set by the user and provide software control of low-power modes.

Setting the UARTEN (UCR1[0]) bit to 0 shuts off the receiver and transmitter logic and the associated clocks.

If the UART is used only in transmit mode, UARTEN and TXEN must be set to 1. If the UART is used only in receive mode, UARTEN and RXEN must be set to 1. Setting TXEN or RXEN to 0 allows to save a lot of power.

76.7 UART Operation in System Debug State

The bit UTS [11] controls whether the UART will respond to the input signal *debug_req*, or whether it will continue to run as normal.

If the UART is programmed to respond to *debug_req*:

1. The UART will halt all operations upon detecting the *debug_req* input.
2. A transfer in progress, either to/from a core (using the IP Bus interface) or to/from an external device, will be completed before halting. This means a single byte/word transfer, not an entire FIFO. Reception of any further data from an external device will be disabled.
3. Internal registers will continue to be writable and readable using the IP Bus interface. A read will leave the contents unaffected.
4. The RX FIFO is affected in debug mode in the following way:
   - All writes into the RX FIFO are prevented.
   - The bit RXDBG (UTS[9]) is used to select the readability of the RX FIFO during debug mode:

     RXDBG = 0: hold the read pointer at the location it had upon entering debug mode, and URXD register returns only the data value at that location, no matter how many reads attempted.

     RXDBG = 1, selectable at any time: Allow to read the characters received in Rx FIFO. It will not be possible to re-read previously read locations, nor will it be possible to readjust the read pointer to the value it had prior to entering debug mode.

76.8 Reset

This section describes how to reset the block and explains special requirements related to reset.

76.8.1 Hardware reset

All of registers, FIFOs, state machines and sequential elements can be reset to their initial values by hardware reset or power on reset.

76.8.2 Software reset

The status registers USR1 and USR2, BRM registers UBIR and UBMR, TxFIFO and RxFIFO, and transmitter and receiver state machines can be reset by software reset. Internal logic will remain the software reset signal at active for about 4 *module_clock* cycles.

Programmer can follow the following software reset sequence:

1. Clear the $\overline{\text{SRST}}$ bit (UCR2[0])
2. Wait for software reset complete: poll SOFTRST bit (UTS[0]) until it is 0.
3. Re-program baud rate registers: Re-write UBIR and UBMR.

76.9 Transfer Error

The UART can generate a transfer error on the peripheral bus in the following cases:

- Core is writing into a read-only register.
- Core is accessing (read or write) an unused location within the assigned address space reserved to UART.
- Core is writing into UTXD register with transmit interface disabled (TXEN=0 or UARTEN=0)
- Core is reading URXD register with receive interface disabled (RXEN=0 or UARTEN=0)

76.10 Functional Timing

This section includes timing diagrams for functional signaling.

76.10.1 RS-232/RS-485 Mode

When transmitting a byte, the UART first sends a Start Bit which is logic 0, followed by the data (general 8 bits, but could be 7 bits) followed by parity bit (optional in RS-232 mode, replaced by the ninth bit in RS-485 mode) and one or two Stop Bits which is logic 1. The sequence is repeated for each byte sent. The receiver also sample data according to this format.

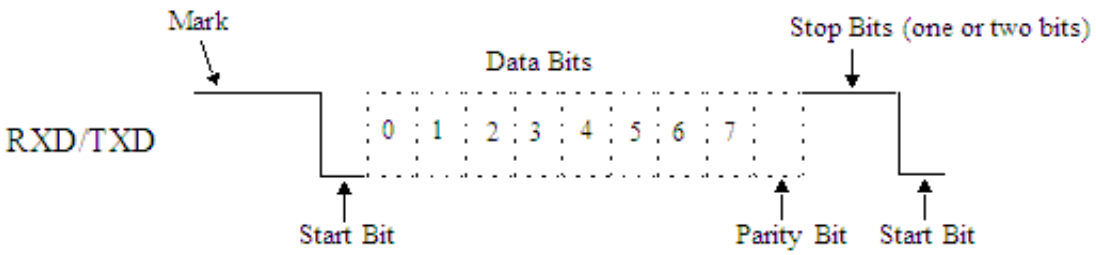


Figure 76-11. Timing Diagram of RS-232 Serial Data Line

76.10.2 IrDA Mode

According to IrDA specification, the low speed (115.2Kbit/s and below) IR frame format is compatible with UART frame.

In this figure, an example data 0x65 is used.

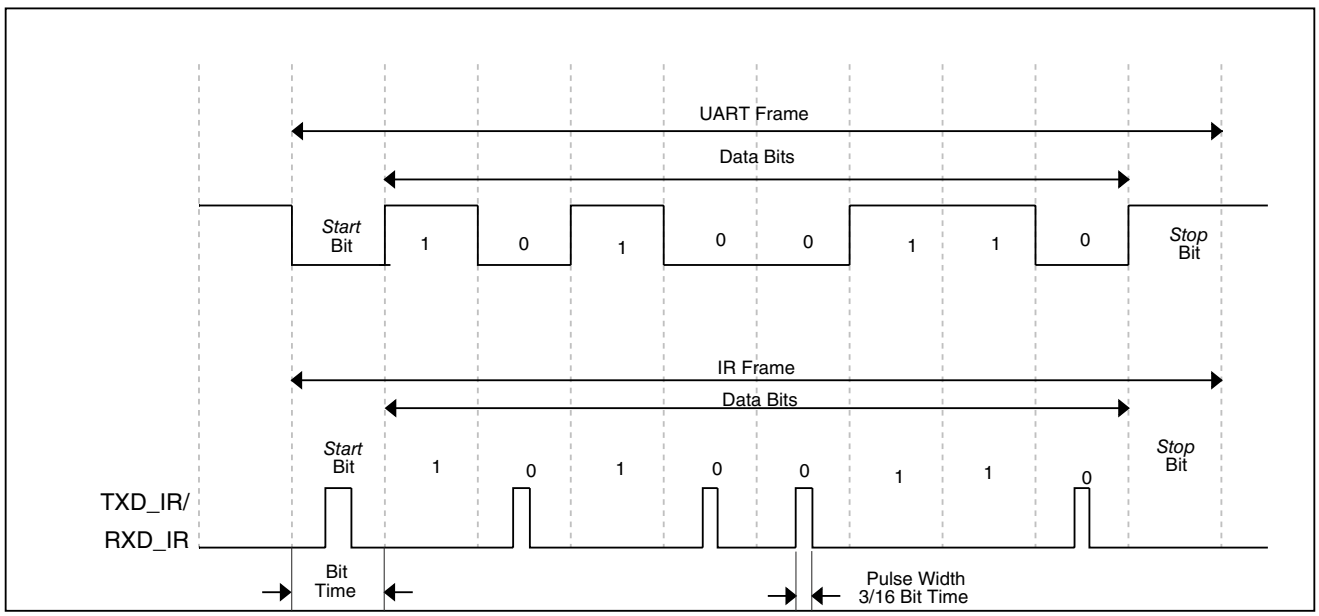


Figure 76-12. Timing diagram of Low Speed IR (<=115.2 Kbit/s) Data Line

76.11 Initialization

76.11.1 Programming the UART in RS-232 mode

As an example, the following sequence can be used to program the UART in order to send and receive characters in RS-232 mode.

Assumptions:

- Input uart clock = 100 MHz
- Baud rate = 921.6Kbps
- Data bits = 8 bits
- Parity = Even
- Stop bits = 1 bit
- Flow control = Hardware

Main program:

1. UCR1 = 0x0001

   Enable the UART.

2. UCR2 = 0x2127

   Set hardware flow control, data format and enable transmitter and receiver.

3. UCR3 = 0x0704

   Set UCR3[RXDMUXSEL] = 1.

4. UCR4 = 0x7C00

   Set CTS trigger level to 31,

5. UFCR = 0x089E

   Set internal clock divider = 5 (divide input uart clock by 5). So the reference clock is 100MHz/5 = 20MHz.

   Set TXTL = 2 and RXTL = 30.

6. UBIR = 0x08FF
7. UBMR = 0x0C34

   In the above two steps, set baud rate to 921.6Kbps based on the 20MHz reference clock.

8. UCR1 = 0x2201>

   Enable the TRDY and RRDY interrupts.

Interrupt service routine for the transmitter:

- Write characters into UTXD

The TRDY interrupt will be automatically de-asserted when the data level of the TxFIFO exceeds the TXTL=2. Note: For the first time the interrupt may be de-asserted after 4 characters are written into the TxFIFO because of the shift register.

Interrupt service routine for the receiver:

- Read characters from URXD

The RRDY interrupt will be automatically de-asserted when the data level of the RxFIFO is below the RXTL=30.

76.12 References

- EIA/TIA-232-F Interface Standard

  http://www.eia.org, http://www.tiaonline.org/standards

- IrDA Standard

  http://www.irda.org

76.13 UART Memory Map/Register Definition

UART supports 8-bit, 16-bit and 32-bit accesses to 32-bit memory-mapped addresses. Any access to unmapped memory location will yield a transfer error.

All registers except the ONEMS described in this section are 16-bit registers. The ONEMS register is a 24-bit register.

- For 32-bit write accesses, the upper two bytes will not be taken into account.
- For 32-bit read accesses the upper two bytes will return 0.

The ONEMS register is expanded from 16 bits to 24 bits in order to support the high frequency of the BRM internal clock *ref_clk* (*module_clock* after divider). The ONEMS register can be accessed as 8 bits, 16 bits or 32 bits.

- For 32-bit write accesses, the most significant byte of the ONEMS will be discarded.
- For 32-bit read accesses, the most significant byte of the ONEMS will be read as 0.

UART memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FB_C000	UART Receiver Register (UART-1_URXD)	32	R	0000_0000h	76.13.1/ 4673
53FB_C040	UART Transmitter Register (UART-1_UTXD)	32	W	0000_0000h	76.13.2/ 4675
53FB_C080	UART Control Register 1 (UART-1_UCR1)	32	R/W	0000_0000h	76.13.3/ 4676
53FB_C084	UART Control Register 2 (UART-1_UCR2)	32	R/W	0000_0001h	76.13.4/ 4678

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FB_C088	UART Control Register 3 (UART-1_UCR3)	32	R/W	0000_0700h	76.13.5/ 4681
53FB_C08C	UART Control Register 4 (UART-1_UCR4)	32	R/W	0000_8000h	76.13.6/ 4683
53FB_C090	UART FIFO Control Register (UART-1_UFCR)	32	R/W	0000_0801h	76.13.7/ 4685
53FB_C094	UART Status Register 1 (UART-1_USR1)	32	R/W	0000_2040h	76.13.8/ 4687
53FB_C098	UART Status Register 2 (UART-1_USR2)	32	R/W	0000_4028h	76.13.9/ 4689
53FB_C09C	UART Escape Character Register (UART-1_UESC)	32	R/W	0000_002Bh	76.13.10/ 4692
53FB_C0A0	UART Escape Timer Register (UART-1_UTIM)	32	R/W	0000_0000h	76.13.11/ 4692
53FB_C0A4	UART BRM Incremental Register (UART-1_UBIR)	32	R/W	0000_0000h	76.13.12/ 4693
53FB_C0A8	UART BRM Modulator Register (UART-1_UBMR)	32	R/W	0000_0000h	76.13.13/ 4693
53FB_C0AC	UART Baud Rate Count Register (UART-1_UBRC)	32	R	0000_0004h	76.13.14/ 4694
53FB_C0B0	UART One Millisecond Register (UART-1_ONEMS)	32	R/W	0000_0000h	76.13.15/ 4695
53FB_C0B4	UART Test Register (UART-1_UTS)	32	R/W	0000_0060h	76.13.16/ 4696
53FC_0000	UART Receiver Register (UART-2_URXD)	32	R	0000_0000h	76.13.1/ 4673
53FC_0040	UART Transmitter Register (UART-2_UTXD)	32	W	0000_0000h	76.13.2/ 4675
53FC_0080	UART Control Register 1 (UART-2_UCR1)	32	R/W	0000_0000h	76.13.3/ 4676
53FC_0084	UART Control Register 2 (UART-2_UCR2)	32	R/W	0000_0001h	76.13.4/ 4678
53FC_0088	UART Control Register 3 (UART-2_UCR3)	32	R/W	0000_0700h	76.13.5/ 4681
53FC_008C	UART Control Register 4 (UART-2_UCR4)	32	R/W	0000_8000h	76.13.6/ 4683
53FC_0090	UART FIFO Control Register (UART-2_UFCR)	32	R/W	0000_0801h	76.13.7/ 4685
53FC_0094	UART Status Register 1 (UART-2_USR1)	32	R/W	0000_2040h	76.13.8/ 4687

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FC_0098	UART Status Register 2 (UART-2_USR2)	32	R/W	0000_4028h	76.13.9/ 4689
53FC_009C	UART Escape Character Register (UART-2_UESC)	32	R/W	0000_002Bh	76.13.10/ 4692
53FC_00A0	UART Escape Timer Register (UART-2_UTIM)	32	R/W	0000_0000h	76.13.11/ 4692
53FC_00A4	UART BRM Incremental Register (UART-2_UBIR)	32	R/W	0000_0000h	76.13.12/ 4693
53FC_00A8	UART BRM Modulator Register (UART-2_UBMR)	32	R/W	0000_0000h	76.13.13/ 4693
53FC_00AC	UART Baud Rate Count Register (UART-2_UBRC)	32	R	0000_0004h	76.13.14/ 4694
53FC_00B0	UART One Millisecond Register (UART-2_ONEMS)	32	R/W	0000_0000h	76.13.15/ 4695
53FC_00B4	UART Test Register (UART-2_UTS)	32	R/W	0000_0060h	76.13.16/ 4696
53FC_4000	UART Receiver Register (UART-3_URXD)	32	R	0000_0000h	76.13.1/ 4673
53FC_4040	UART Transmitter Register (UART-3_UTXD)	32	W	0000_0000h	76.13.2/ 4675
53FC_4080	UART Control Register 1 (UART-3_UCR1)	32	R/W	0000_0000h	76.13.3/ 4676
53FC_4084	UART Control Register 2 (UART-3_UCR2)	32	R/W	0000_0001h	76.13.4/ 4678
53FC_4088	UART Control Register 3 (UART-3_UCR3)	32	R/W	0000_0700h	76.13.5/ 4681
53FC_408C	UART Control Register 4 (UART-3_UCR4)	32	R/W	0000_8000h	76.13.6/ 4683
53FC_4090	UART FIFO Control Register (UART-3_UFCR)	32	R/W	0000_0801h	76.13.7/ 4685
53FC_4094	UART Status Register 1 (UART-3_USR1)	32	R/W	0000_2040h	76.13.8/ 4687
53FC_4098	UART Status Register 2 (UART-3_USR2)	32	R/W	0000_4028h	76.13.9/ 4689
53FC_409C	UART Escape Character Register (UART-3_UESC)	32	R/W	0000_002Bh	76.13.10/ 4692
53FC_40A0	UART Escape Timer Register (UART-3_UTIM)	32	R/W	0000_0000h	76.13.11/ 4692
53FC_40A4	UART BRM Incremental Register (UART-3_UBIR)	32	R/W	0000_0000h	76.13.12/ 4693

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53FC_40A8	UART BRM Modulator Register (UART-3_UBMR)	32	R/W	0000_0000h	76.13.13/ 4693
53FC_40AC	UART Baud Rate Count Register (UART-3_UBRC)	32	R	0000_0004h	76.13.14/ 4694
53FC_40B0	UART One Millisecond Register (UART-3_ONEMS)	32	R/W	0000_0000h	76.13.15/ 4695
53FC_40B4	UART Test Register (UART-3_UTS)	32	R/W	0000_0060h	76.13.16/ 4696
53FF_0000	UART Receiver Register (UART-4_URXD)	32	R	0000_0000h	76.13.1/ 4673
53FF_0040	UART Transmitter Register (UART-4_UTXD)	32	W	0000_0000h	76.13.2/ 4675
53FF_0080	UART Control Register 1 (UART-4_UCR1)	32	R/W	0000_0000h	76.13.3/ 4676
53FF_0084	UART Control Register 2 (UART-4_UCR2)	32	R/W	0000_0001h	76.13.4/ 4678
53FF_0088	UART Control Register 3 (UART-4_UCR3)	32	R/W	0000_0700h	76.13.5/ 4681
53FF_008C	UART Control Register 4 (UART-4_UCR4)	32	R/W	0000_8000h	76.13.6/ 4683
53FF_0090	UART FIFO Control Register (UART-4_UFCR)	32	R/W	0000_0801h	76.13.7/ 4685
53FF_0094	UART Status Register 1 (UART-4_USR1)	32	R/W	0000_2040h	76.13.8/ 4687
53FF_0098	UART Status Register 2 (UART-4_USR2)	32	R/W	0000_4028h	76.13.9/ 4689
53FF_009C	UART Escape Character Register (UART-4_UESC)	32	R/W	0000_002Bh	76.13.10/ 4692
53FF_00A0	UART Escape Timer Register (UART-4_UTIM)	32	R/W	0000_0000h	76.13.11/ 4692
53FF_00A4	UART BRM Incremental Register (UART-4_UBIR)	32	R/W	0000_0000h	76.13.12/ 4693
53FF_00A8	UART BRM Modulator Register (UART-4_UBMR)	32	R/W	0000_0000h	76.13.13/ 4693
53FF_00AC	UART Baud Rate Count Register (UART-4_UBRC)	32	R	0000_0004h	76.13.14/ 4694
53FF_00B0	UART One Millisecond Register (UART-4_ONEMS)	32	R/W	0000_0000h	76.13.15/ 4695
53FF_00B4	UART Test Register (UART-4_UTS)	32	R/W	0000_0060h	76.13.16/ 4696

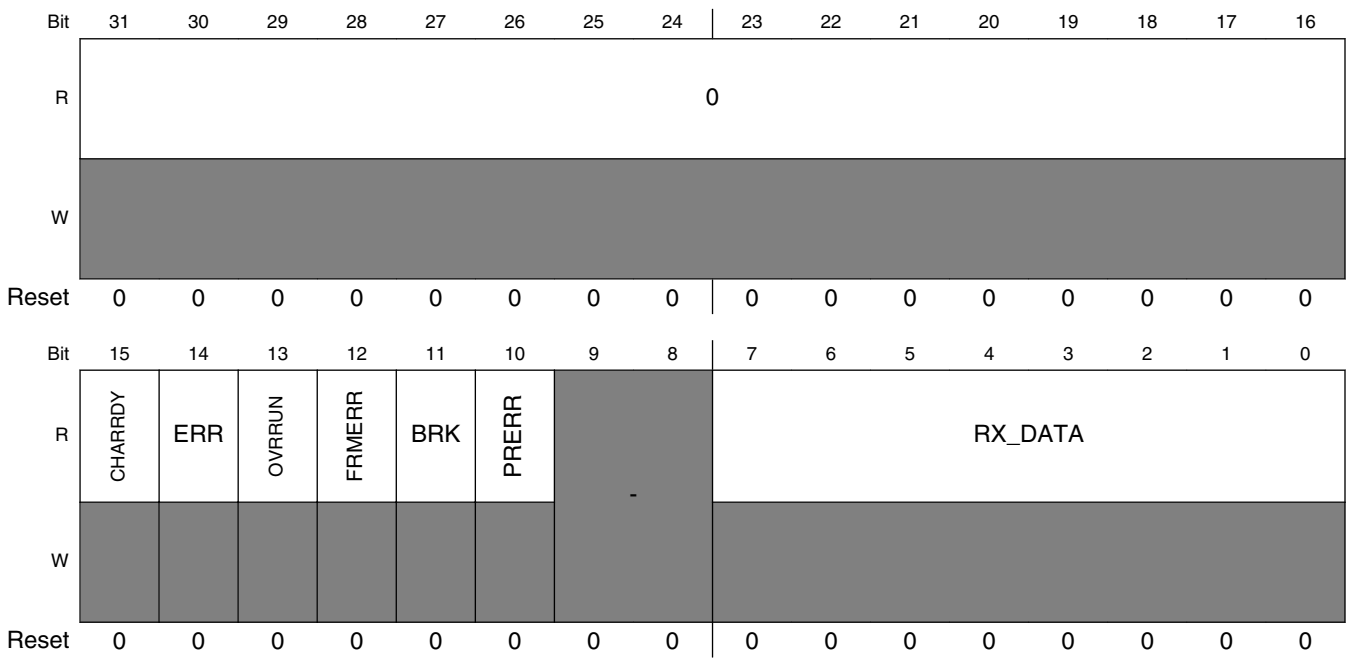
UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63F9_0000	UART Receiver Register (UART-5_URXD)	32	R	0000_0000h	76.13.1/ 4673
63F9_0040	UART Transmitter Register (UART-5_UTXD)	32	W	0000_0000h	76.13.2/ 4675
63F9_0080	UART Control Register 1 (UART-5_UCR1)	32	R/W	0000_0000h	76.13.3/ 4676
63F9_0084	UART Control Register 2 (UART-5_UCR2)	32	R/W	0000_0001h	76.13.4/ 4678
63F9_0088	UART Control Register 3 (UART-5_UCR3)	32	R/W	0000_0700h	76.13.5/ 4681
63F9_008C	UART Control Register 4 (UART-5_UCR4)	32	R/W	0000_8000h	76.13.6/ 4683
63F9_0090	UART FIFO Control Register (UART-5_UFCR)	32	R/W	0000_0801h	76.13.7/ 4685
63F9_0094	UART Status Register 1 (UART-5_USR1)	32	R/W	0000_2040h	76.13.8/ 4687
63F9_0098	UART Status Register 2 (UART-5_USR2)	32	R/W	0000_4028h	76.13.9/ 4689
63F9_009C	UART Escape Character Register (UART-5_UESC)	32	R/W	0000_002Bh	76.13.10/ 4692
63F9_00A0	UART Escape Timer Register (UART-5_UTIM)	32	R/W	0000_0000h	76.13.11/ 4692
63F9_00A4	UART BRM Incremental Register (UART-5_UBIR)	32	R/W	0000_0000h	76.13.12/ 4693
63F9_00A8	UART BRM Modulator Register (UART-5_UBMR)	32	R/W	0000_0000h	76.13.13/ 4693
63F9_00AC	UART Baud Rate Count Register (UART-5_UBRC)	32	R	0000_0004h	76.13.14/ 4694
63F9_00B0	UART One Millisecond Register (UART-5_ONEMS)	32	R/W	0000_0000h	76.13.15/ 4695
63F9_00B4	UART Test Register (UART-5_UTS)	32	R/W	0000_0060h	76.13.16/ 4696

76.13.1 UART Receiver Register (UARTx_URXD)

The UART will yield a transfer error on the peripheral bus when core is reading URXD register with receive interface disabled (RXEN=0 or UARTEN=0).

Addresses: UART-1_URXD is 53FB_C000h base + 0h offset = 53FB_C000h
UART-2_URXD is 53FC_0000h base + 0h offset = 53FC_0000h
UART-3_URXD is 53FC_4000h base + 0h offset = 53FC_4000h
UART-4_URXD is 53FF_0000h base + 0h offset = 53FF_0000h
UART-5_URXD is 63F9_0000h base + 0h offset = 63F9_0000h



UARTx_URXD field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15 CHARRDY	Character Ready. This read-only bit indicates an invalid read when the FIFO becomes empty and software tries to read the same old data. This bit should not be used for polling for data written to the RX FIFO. 0 Character in RX_DATA field and associated flags are invalid. 1 Character in RX_DATA field and associated flags valid and ready for reading.
14 ERR	**Error Detect.** Indicates whether the character present in the RX_DATA field has an error (OVRRUN, FRMERR, BRK or PRERR) status. The ERR bit is updated and valid for each received character. 0 No error status was detected 1 An error status was detected

Table continues on the next page...

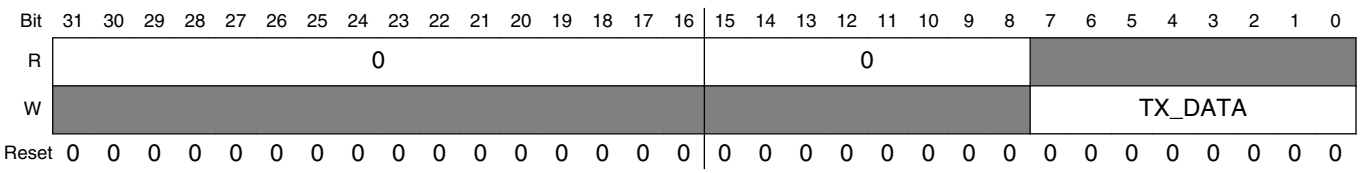
UARTx_URXD field descriptions (continued)

Field	Description
13 OVRRUN	**Receiver Overrun.** This read-only bit, when HIGH, indicates that the corresponding character was stored in the last position (32nd) of the Rx FIFO. Even if a 33rd character has not been detected, this bit will be set to '1' for the 32nd character. 0 No RxFIFO overrun was detected 1 A RxFIFO overrun was detected
12 FRMERR	**Frame Error.** Indicates whether the current character had a framing error (a missing stop bit) and is possibly corrupted. FRMERR is updated for each character read from the RxFIFO. 0 The current character has no framing error 1 The current character has a framing error
11 BRK	**BREAK Detect.** Indicates whether the current character was detected as a BREAK character. The data bits and the stop bit are all 0. The FRMERR bit is set when BRK is set. When odd parity is selected, PRERR is also set when BRK is set. BRK is valid for each character read from the RxFIFO. 0 The current character is not a BREAK character 1 The current character is a BREAK character
10 PRERR	**Parity Error flag**. Indicates whether the current character was detected with a parity error and is possibly corrupted. PRERR is updated for each character read from the RxFIFO. When parity is disabled, PRERR always reads as 0. 0 = No parity error was detected for data in the RX_DATA field 1 = A parity error was detected for data in the RX_DATA field
9–8 -	**Reserved**
7–0 RX_DATA	**Received Data**. Holds the received character. In 7-bit mode, the most significant bit (MSB) is forced to 0. In 8-bit mode, all bits are active.

76.13.2 UART Transmitter Register (UARTx_UTXD)

The UART will yield a transfer error on the peripheral bus when core is writing into UART_URXD register with transmit interface disabled (TXEN=0 or UARTEN=0). Memory space between UART_URXD and UART_UTXD registers is reserved. Any read or write access to this space will be considered as an invalid access and yield a transfer error.

Addresses: UART-1_UTXD is 53FB_C000h base + 40h offset = 53FB_C040h
UART-2_UTXD is 53FC_0000h base + 40h offset = 53FC_0040h
UART-3_UTXD is 53FC_4000h base + 40h offset = 53FC_4040h
UART-4_UTXD is 53FF_0000h base + 40h offset = 53FF_0040h
UART-5_UTXD is 63F9_0000h base + 40h offset = 63F9_0040h



UARTx_UTXD field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–0 TX_DATA	**Transmit Data**. Holds the parallel transmit data inputs. In 7-bit mode, D7 is ignored. In 8-bit mode, all bits are used. Data is transmitted least significant bit (LSB) first. A new character is transmitted when the TX_DATA field is written. The TX_DATA field must be written only when the TRDY bit is high to ensure that corrupted data is not sent.

76.13.3 UART Control Register 1 (UARTx_UCR1)

Addresses: UART-1_UCR1 is 53FB_C000h base + 80h offset = 53FB_C080h
UART-2_UCR1 is 53FC_0000h base + 80h offset = 53FC_0080h
UART-3_UCR1 is 53FC_4000h base + 80h offset = 53FC_4080h
UART-4_UCR1 is 53FF_0000h base + 80h offset = 53FF_0080h
UART-5_UCR1 is 63F9_0000h base + 80h offset = 63F9_0080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADEN	ADBR	TRDYEN	IDEN	ICD		RRDYEN	RXDMAEN	IREN	TXMPTYEN	RTSDEN	SNDBRK	TXDMAEN	ATDMAEN	DOZE	UARTEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_UCR1 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15 ADEN	**Automatic Baud Rate Detection Interrupt Enable**. Enables/Disables the automatic baud rate detect complete (ADET) bit to generate an interrupt (*interrupt_uart* = 0). 0 Disable the automatic baud rate detection interrupt 1 Enable the automatic baud rate detection interrupt
14 ADBR	**Automatic Detection of Baud Rate**. Enables/Disables automatic baud rate detection. When the ADBR bit is set and the ADET bit is cleared, the receiver detects the incoming baud rate automatically. The ADET flag is set when the receiver verifies that the incoming baud rate is detected properly by detecting an ASCII character "A" or "a" (0x61 or 0x41). 0 Disable automatic detection of baud rate 1 Enable automatic detection of baud rate
13 TRDYEN	**Transmitter Ready Interrupt Enable**. Enables/Disables the transmitter Ready Interrupt (TRDY) when the transmitter has one or more slots available in the TxFIFO. The fill level in the TXFIFO at which an interrupt is generated is controlled by TxTL bits. When TRDYEN is negated, the transmitter ready interrupt is disabled. **NOTE:** An interrupt will be issued as long as TRDYEN and TRDY are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the TRDY interrupt. 0 Disable the transmitter ready interrupt 1 Enable the transmitter ready interrupt

Table continues on the next page...

UARTx_UCR1 field descriptions (continued)

Field	Description
12 IDEN	**Idle Condition Detected Interrupt Enable**. Enables/Disables the IDLE bit to generate an interrupt (*interrupt_uart* = 0). 0 Disable the IDLE interrupt 1 Enable the IDLE interrupt
11–10 ICD	**Idle Condition Detect**. Controls the number of frames RXD is allowed to be idle before an idle condition is reported. 00 Idle for more than 4 frames 01 Idle for more than 8 frames 10 Idle for more than 16 frames 11 Idle for more than 32 frames
9 RRDYEN	**Receiver Ready Interrupt Enable**. Enables/Disables the RRDY interrupt when the RxFIFO contains data. The fill level in the RxFIFO at which an interrupt is generated is controlled by the RXTL bits. When RRDYEN is negated, the receiver ready interrupt is disabled. 0 Disables the RRDY interrupt 1 Enables the RRDY interrupt
8 RXDMAEN	**Receive Ready DMA Enable**. Enables/Disables the receive DMA request *dma_req_rx* when the receiver has data in the RxFIFO. The fill level in the RxFIFO at which a DMA request is generated is controlled by the RXTL bits. When negated, the receive DMA request is disabled. 0 Disable DMA request 1 Enable DMA request
7 IREN	**Infrared Interface Enable**. Enables/Disables the IR interface. See the IR interface description in Infrared Interface, for more information. 0 Disable the IR interface 1 Enable the IR interface
6 TXMPTYEN	**Transmitter Empty Interrupt Enable**. Enables/Disables the transmitter FIFO empty (TXFE) interrupt. *interrupt_uart*. When negated, the TXFE interrupt is disabled. **NOTE:** An interrupt will be issued as long as TXMPTYEN and TXFE are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the TXFE interrupt. 0 Disable the transmitter FIFO empty interrupt 1 Enable the transmitter FIFO empty interrupt
5 RTSDEN	**RTS Delta Interrupt Enable**. Enables/Disables the RTSD interrupt. The current status of the $\overline{\text{RTS}}$ pin is read in the RTSS bit. 0 Disable RTSD interrupt 1 Enable RTSD interrupt
4 SNDBRK	**Send BREAK**. Forces the transmitter to send a BREAK character. The transmitter finishes sending the character in progress (if any) and sends BREAK characters until SNDBRK is reset. Because the transmitter samples SNDBRK after every bit is transmitted, it is important that SNDBRK is asserted high for a sufficient period of time to generate a valid BREAK. After the BREAK transmission completes, the UART transmits 2 mark bits. The user can continue to fill the TxFIFO and any characters remaining are transmitted when the BREAK is terminated.

Table continues on the next page...

UARTx_UCR1 field descriptions (continued)

Field	Description
	0 Do not send a BREAK character 1 Send a BREAK character (continuous 0s)
3 TXDMAEN	**Transmitter Ready DMA Enable**. Enables/Disables the transmit DMA request *dma_req_tx* when the transmitter has one or more slots available in the TxFIFO. The fill level in the TxFIFO that generates the *dma_req_tx* is controlled by the TXTL bits. **NOTE:** A DMA request will be issued as long as TXDMAEN and TRDY are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the transmit DMA request. 0 Disable transmit DMA request 1 Enable transmit DMA request
2 ATDMAEN	**Aging DMA Timer Enable**. Enables/Disables the receive DMA request *dma_req_rx* for the aging timer interrupt (triggered with AGTIM flag in USR1[8]). 0 Disable AGTIM DMA request 1 Enable AGTIM DMA request
1 DOZE	**DOZE**. Determines the UART enable condition in the DOZE state. When *doze_req* input pin is at '1', (the ARM Platform executes a doze instruction and the system is placed in the Doze State), the DOZE bit affects operation of the UART. While in the Doze State, if this bit is asserted, the UART is disabled. See the description in Low Power Modes. 0 The UART is enabled when in DOZE state 1 The UART is disabled when in DOZE state
0 UARTEN	**UART Enable**. Enables/Disables the UART. If UARTEN is negated in the middle of a transmission, the transmitter stops and pulls the TXD line to a logic 1. UARTEN must be set to 1 before any access to UTXD and URXD registers, otherwise a transfer error is returned. This bit can be set to 1 along with other bits in this register. There is no restriction to the sequence of programing this bit and other control registers. 0 Disable the UART 1 Enable the UART

76.13.4 UART Control Register 2 (UARTx_UCR2)

Addresses: UART-1_UCR2 is 53FB_C000h base + 84h offset = 53FB_C084h
UART-2_UCR2 is 53FC_0000h base + 84h offset = 53FC_0084h
UART-3_UCR2 is 53FC_4000h base + 84h offset = 53FC_4084h
UART-4_UCR2 is 53FF_0000h base + 84h offset = 53FF_0084h
UART-5_UCR2 is 63F9_0000h base + 84h offset = 63F9_0084h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																ESCI	IRTS	CTSC	CTS	ESCEN	RTEC		PREN	PROE	STPB	WS	RTSEN	ATEN	TXEN	RXEN	SRST
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

UARTx_UCR2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15 ESCI	**Escape Sequence Interrupt Enable**. Enables/Disables the ESCF bit to generate an interrupt. 0 Disable the escape sequence interrupt 1 Enable the escape sequence interrupt
14 IRTS	**Ignore RTS Pin**. Forces the RTS input signal presented to the transmitter to always be asserted (set to low), effectively ignoring the external pin. When in this mode, the RTS pin serves as a general purpose input. 0 Transmit only when the RTS pin is asserted 1 Ignore the RTS pin
13 CTSC	CTS **Pin Control**. Controls the operation of the $\overline{CTS}$ output pin. When CTSC is asserted, the $\overline{CTS}$ output pin is controlled by the receiver. When the RxFIFO is filled to the level of the programmed trigger level and the start bit of the overflowing character (TRIGGER LEVEL + 1) is validated, the $\overline{CTS}$ output pin is negated to indicate to the far-end transmitter to stop transmitting. When the trigger level is programmed for less than 32, the receiver continues to receive data until the RxFIFO is full. When the CTSC bit is negated, the $\overline{CTS}$ output pin is controlled by the CTS bit. On reset, because CTSC is cleared to 0, the $\overline{CTS}$ pin is controlled by the CTS bit, which again is cleared to 0 on reset. This means that on reset the $\overline{CTS}$ signal is negated. 0 The $\overline{CTS}$ pin is controlled by the CTS bit 1 The $\overline{CTS}$ pin is controlled by the receiver
12 CTS	**Clear to Send**. Controls the $\overline{CTS}$ pin when the CTSC bit is negated. CTS has no function when CTSC is asserted. 0 The $\overline{CTS}$ pin is high (inactive) 1 The $\overline{CTS}$ pin is low (active)
11 ESCEN	**Escape Enable**. Enables/Disables the escape sequence detection logic. 0 Disable escape sequence detection 1 Enable escape sequence detection
10–9 RTEC	**Request to Send Edge Control**. Selects the edge that triggers the RTS interrupt. This has no effect on the RTS delta interrupt. RTEC has an effect only when RTSEN = 1 (see Table 76-5). 00 Trigger interrupt on a rising edge 01 Trigger interrupt on a falling edge 1X Trigger interrupt on any edge
8 PREN	**Parity Enable**. Enables/Disables the parity generator in the transmitter and parity checker in the receiver. When PREN is asserted, the parity generator and checker are enabled, and disabled when PREN is negated. 0 Disable parity generator and checker 1 Enable parity generator and checker
7 PROE	**Parity Odd/Even**. Controls the sense of the parity generator and checker. When PROE is high, odd parity is generated and expected. When PROE is low, even parity is generated and expected. PROE has no function if PREN is low.

Table continues on the next page...

UARTx_UCR2 field descriptions (continued)

Field	Description
	0 Even parity 1 Odd parity
6 STPB	**Stop**. Controls the number of stop bits after a character. When STPB is low, 1 stop bit is sent. When STPB is high, 2 stop bits are sent. STPB also affects the receiver. 0 The transmitter sends 1 stop bit. The receiver expects 1 or more stop bits. 1 The transmitter sends 2 stop bits. The receiver expects 2 or more stop bits.
5 WS	**Word Size**. Controls the character length. When WS is high, the transmitter and receiver are in 8-bit mode. When WS is low, they are in 7-bit mode. The transmitter ignores bit 7 and the receiver sets bit 7 to 0. WS can be changed in-between transmission (reception) of characters, however not when a transmission (reception) is in progress, in which case the length of the current character being transmitted (received) is unpredictable. 0 7-bit transmit and receive character length (not including START, STOP or PARITY bits) 1 8-bit transmit and receive character length (not including START, STOP or PARITY bits)
4 RTSEN	**Request to Send Interrupt Enable.** Controls the RTS edge sensitive interrupt. When RTSEN is asserted and the programmed edge is detected on the $\overline{\text{RTS}}$ pin (the RTSF bit is asserted), an interrupt will be generated on the *interrupt_uart* pin. (See Table 76-5.) 0 Disable request to send interrupt 1 Enable request to send interrupt
3 ATEN	**Aging Timer Enable**. This bit is used to enable the aging timer interrupt (triggered with AGTIM) 0 AGTIM interrupt disabled 1 AGTIM interrupt enabled
2 TXEN	**Transmitter Enable**. Enables/Disables the transmitter. When TXEN is negated the transmitter is disabled and idle. When the UARTEN and TXEN bits are set the transmitter is enabled. If TXEN is negated in the middle of a transmission, the UART disables the transmitter immediately, and starts marking 1s. The transmitter FIFO cannot be written when this bit is cleared. 0 Disable the transmitter 1 Enable the transmitter
1 RXEN	**Receiver Enable**. Enables/Disables the receiver. When the receiver is enabled, if the RXD input is already low, the receiver does not recognize BREAK characters, because it requires a valid 1-to-0 transition before it can accept any character. 0 Disable the receiver 1 Enable the receiver
0 SRST	**Software Reset**. Once the software writes 0 to $\overline{\text{SRST}}$, the software reset remains active for 4 *module_clock* cycles before the hardware deasserts $\overline{\text{SRST}}$. The software can only write 0 to $\overline{\text{SRST}}$. Writing 1 to $\overline{\text{SRST}}$ is ignored. 0 Reset the transmit and receive state machines, all FIFOs and register USR1, USR2, UBIR, UBMR, UBRC , URXD, UTXD and UTS[6-3]. 1 No reset

76.13.5 UART Control Register 3 (UARTx_UCR3)

Addresses: UART-1_UCR3 is 53FB_C000h base + 88h offset = 53FB_C088h
UART-2_UCR3 is 53FC_0000h base + 88h offset = 53FC_0088h
UART-3_UCR3 is 53FC_4000h base + 88h offset = 53FC_4088h
UART-4_UCR3 is 53FF_0000h base + 88h offset = 53FF_0088h
UART-5_UCR3 is 63F9_0000h base + 88h offset = 63F9_0088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DPEC		DTREN	PARERREN	FRAERREN	DSR	DCD	RI	ADNIMP	RXDSEN	AIRINTEN	AWAKEN	DTRDEN	RXDMUXSEL	INVT	ACIEN
W																
Reset	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0

UARTx_UCR3 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–14 DPEC	**DTR/DSR Interrupt Edge Control**. These bits control the edge of $\overline{\text{DTR}}$ (DCE) or $\overline{\text{DSR}}$ (DTE) on which an interrupt will be generated. An interrupt will only be generated if the DTREN bit is set. 00 interrupt generated on rising edge 01 interrupt generated on falling edge 1X interrupt generated on either edge
13 DTREN	**Data Terminal Ready Interrupt Enable**. When this bit is set, it will enable the status bit DTRF (USR2 [13]) (DTR/DSR edge sensitive interrupt) to cause an interrupt. 0 Data Terminal Ready Interrupt Disabled 1 Data Terminal Ready Interrupt Enabled
12 PARERREN	**Parity Error Interrupt Enable. Enables/Disables the interrupt. When asserted, PARERREN causes the PARITYERR bit to generate an interrupt.** 0 Disable the parity error interrupt 1 Enable the parity error interrupt
11 FRAERREN	**Frame Error Interrupt Enable. Enables/Disables the interrupt. When asserted, FRAERREN causes the FRAMERR bit to generate an interrupt.** 0 Disable the frame error interrupt 1 Enable the frame error interrupt

Table continues on the next page...

UARTx_UCR3 field descriptions (continued)

Field	Description
10 DSR	**Data Set Ready**. This bit is used by software to control the DSR/DTR output pin for the modem interface. In DCE mode it applies to $\overline{DSR}$ and in DTE mode it applies to $\overline{DTR}$. 0 DSR/ DTR pin is logic zero 1 DSR/ DTR pin is logic one
9 DCD	**Data Carrier Detect**. In DCE mode this bit is used by software to control the $\overline{DCD}$ output pin for the modem interface. In DTE mode, when this bit is set, it will enable the status bit DCDDELT (USR2 (6)) to cause an interrupt. 0 $\overline{DCD}$ pin is logic zero (DCE mode) 1 $\overline{DCD}$ pin is logic one (DCE mode) 0 DCDDELT interrupt disabled (DTE mode) 1 DCDDELT interrupt enabled (DTE mode)
8 RI	**Ring Indicator**. In DCE mode this bit is used by software to control the $\overline{RI}$ output pin for the modem interface. In DTE mode, when this bit is set, it will enable the status bit RIDELT (USR2 (10)) to cause an interrupt. 0 $\overline{RI}$ pin is logic zero (DCE mode) 1 $\overline{RI}$ pin is logic one (DCE mode) 0 RIDELT interrupt disabled (DTE mode) 1 RIDELT interrupt enabled (DTE mode)
7 ADNIMP	**Autobaud Detection Not Improved-. Disables new features of autobaud detection (See Baud Rate Automatic Detection Protocol, for more details).** 0 Autobaud detection new features selected 1 Keep old autobaud detection mechanism
6 RXDSEN	Receive Status Interrupt Enable. Controls the receive status interrupt (*interrupt_uart*). When this bit is enabled and RXDS status bit is set, the interrupt *interrupt_uart* will be generated. 0 Disable the RXDS interrupt 1 Enable the RXDS interrupt
5 AIRINTEN	Asynchronous IR WAKE Interrupt Enable. Controls the asynchronous IR WAKE interrupt. An interrupt is generated when AIRINTEN is asserted and a pulse is detected on the RXD pin. 0 Disable the AIRINT interrupt 1 Enable the AIRINT interrupt
4 AWAKEN	Asynchronous WAKE Interrupt Enable. Controls the asynchronous WAKE interrupt. An interrupt is generated when AWAKEN is asserted and a falling edge is detected on the RXD pin. 0 Disable the AWAKE interrupt 1 Enable the AWAKE interrupt
3 DTRDEN	**Data Terminal Ready Delta Enable**. Enables / Disables the asynchronous DTRD interrupt. When DTRDEN is asserted and an edge (rising or falling) is detected on $\overline{DTR}$ (in DCE mode) or on $\overline{DSR}$ (in DTE mode), then an interrupt is generated. 0 Disable DTRD interrupt 1 Enable DTRD interrupt
2 RXDMUXSEL	RXD Muxed Input Selected. Selects proper input pins for serial and Infrared input signal.

Table continues on the next page...

UARTx_UCR3 field descriptions (continued)

Field	Description
	NOTE: In this chip, UARTs are used in MUXED mode, so that this bit should always be set.
1 INVT	In **IrDA mode**, when INVT is cleared, the infrared logic block transmits a positive IR 3/16 pulse for all 0s and 0s are transmitted for 1s. When INVT is set (INVT = 1), the infrared logic block transmits an active low or negative infrared 3/16 pulse for all 0s and 1s are transmitted for 1s. 0 TXDActive low transmission 1 **TXD** Active high transmission
0 ACIEN	**Autobaud Counter Interrupt Enable.** This bit is used to enable the autobaud counter stopped interrupt (triggered with ACST (USR2[11]). 0 ACST interrupt disabled 1 ACST interrupt enabled

76.13.6 UART Control Register 4 (UARTx_UCR4)

Addresses: UART-1_UCR4 is 53FB_C000h base + 8Ch offset = 53FB_C08Ch
UART-2_UCR4 is 53FC_0000h base + 8Ch offset = 53FC_008Ch
UART-3_UCR4 is 53FC_4000h base + 8Ch offset = 53FC_408Ch
UART-4_UCR4 is 53FF_0000h base + 8Ch offset = 53FF_008Ch
UART-5_UCR4 is 63F9_0000h base + 8Ch offset = 63F9_008Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CTSTL						INVR	ENIRI	WKEN	IDDMAEN	IRSC	LPBYP	TCEN	BKEN	OREN	DREN
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_UCR4 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–10 CTSTL	**CTS Trigger Level**. Controls the threshold at which the $\overline{\text{CTS}}$ pin is deasserted by the RxFIFO. After the trigger level is reached and the $\overline{\text{CTS}}$ pin is deasserted, the RxFIFO continues to receive data until it is full. The CTSTL bits are encoded as shown in the Settings column. Settings 0 to 32 are in use. All other settings are Reserved.

Table continues on the next page...

UARTx_UCR4 field descriptions (continued)

Field	Description
	000000 0 characters received 000001 1 characters in the RxFIFO ... — ... — 100000 32 characters in the RxFIFO (maximum)
9 INVR	**In IrDA mode**,when cleared, the infrared logic block expects an active low or negative IR 3/16 pulse for 0s and 1s are expected for 1s. When INVR is set (INVR 1), the infrared logic block expects an active high or positive IR 3/16 pulse for 0s and 0s are expected for 1s. 0 **RXD** active low detection 1 **RXD** active high detection
8 ENIRI	**Serial Infrared Interrupt Enable**. Enables/Disables the serial infrared interrupt. 0 Serial infrared Interrupt disabled 1 Serial infrared Interrupt enabled
7 WKEN	**WAKE Interrupt Enable**. Enables/Disables the WAKE bit to generate an interrupt. The WAKE bit is set at the detection of a start bit by the receiver. 0 Disable the WAKE interrupt 1 Enable the WAKE interrupt
6 IDDMAEN	**DMA IDLE Condition Detected Interrupt Enable** Enables/Disables the receive DMA request *dma_req_rx* for the IDLE interrupt (triggered with IDLE flag in USR2[12]). 0 DMA IDLE interrupt disabled 1 DMA IDLE interrupt enabled
5 IRSC	**IR Special Case**. Selects the clock for the vote logic. When set, IRSC switches the vote logic clock from the sampling clock to the UART reference clock. The IR pulses are counted a predetermined amount of time depending on the reference frequency. See InfraRed Special Case (IRSC) Bit. 0 The vote logic uses the sampling clock (16x baud rate) for normal operation 1 The vote logic uses the UART reference clock
4 LPBYP	**Low Power Bypass**. Allows to bypass the low power new features in UART. To use during debug phase. 0 Low power features enabled 1 Low power features disabled
3 TCEN	**TransmitComplete Interrupt Enable**. Enables/Disables the TXDC bit to generate an interrupt (*interrupt_uart* = 0) **NOTE:** An interrupt will be issued as long as TCEN and TXDC are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the TXDC interrupt. 0 Disable TXDC interrupt 1 Enable TXDC interrupt
2 BKEN	**BREAK Condition Detected Interrupt Enable**. Enables/Disables the BRCD bit to generate an interrupt. 0 Disable the BRCD interrupt 1 Enable the BRCD interrupt
1 OREN	**Receiver Overrun Interrupt Enable**. Enables/Disables the ORE bit to generate an interrupt.

Table continues on the next page...

UARTx_UCR4 field descriptions (continued)

Field	Description
	0 Disable ORE interrupt 1 Enable ORE interrupt
0 DREN	**Receive Data Ready Interrupt Enable**. Enables/Disables the RDR bit to generate an interrupt. 0 Disable RDR interrupt 1 Enable RDR interrupt

76.13.7 UART FIFO Control Register (UARTx_UFCR)

Addresses: UART-1_UFCR is 53FB_C000h base + 90h offset = 53FB_C090h
UART-2_UFCR is 53FC_0000h base + 90h offset = 53FC_0090h
UART-3_UFCR is 53FC_4000h base + 90h offset = 53FC_4090h
UART-4_UFCR is 53FF_0000h base + 90h offset = 53FF_0090h
UART-5_UFCR is 63F9_0000h base + 90h offset = 63F9_0090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXTL						RFDIV			DCEDTE	RXTL					
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

UARTx_UFCR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–10 TXTL	**Transmitter Trigger Level**. Controls the threshold at which a maskable interrupt is generated by the TxFIFO. A maskable interrupt is generated whenever the data level in the TxFIFO falls below the selected threshold. The bits are encoded as shown in the Settings column. Settings 0 to 32 are in use. All other settings are Reserved. 000000 Reserved 000001 Reserved 000010 TxFIFO has 2 or fewer characters ... — ... —

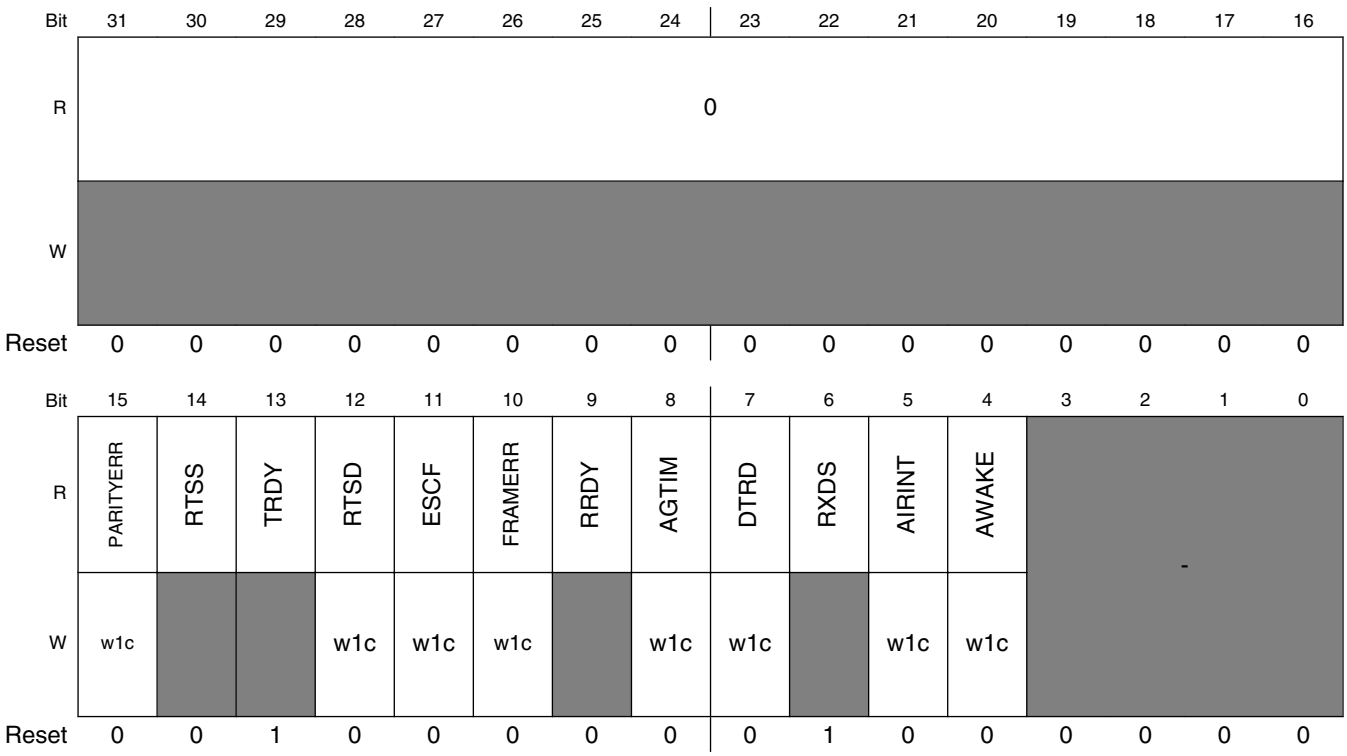
Table continues on the next page...

UARTx_UFCR field descriptions (continued)

Field	Description
	011111 TxFIFO has 31 or fewer characters 100000 TxFIFO has 32 characters (maximum)
9–7 RFDIV	Reference Frequency Divider. Controls the divide ratio for the reference clock. The input clock is *module_clock*. The output from the divider is *ref_clk* which is used by BRM to create the 16x baud rate oversampling clock (*brm_clk*). 000 Divide input clock by 6 001 Divide input clock by 5 010 Divide input clock by 4 011 Divide input clock by 3 100 Divide input clock by 2 101 Divide input clock by 1 110 Divide input clock by 7 111 Reserved
6 DCEDTE	**DCE/DTE mode select**. Select UART as data communication equipment (DCE mode) or as data terminal equipment (DTE mode). 0 DCE mode selected 1 DTE mode selected
5–0 RXTL	**Receiver Trigger Level**. Controls the threshold at which a maskable interrupt is generated by the RxFIFO. A maskable interrupt is generated whenever the data level in the RxFIFO reaches the selected threshold. The RXTL bits are encoded as shown in the Settings column. Setting 0 to 32 are in use. All other settings are Reserved. 000000 0 characters received 000001 RxFIFO has 1 character ... — ... — 011111 RxFIFO has 31 characters 100000 RxFIFO has 32 characters (maximum)

76.13.8 UART Status Register 1 (UARTx_USR1)

Addresses: UART-1_USR1 is 53FB_C000h base + 94h offset = 53FB_C094h
UART-2_USR1 is 53FC_0000h base + 94h offset = 53FC_0094h
UART-3_USR1 is 53FC_4000h base + 94h offset = 53FC_4094h
UART-4_USR1 is 53FF_0000h base + 94h offset = 53FF_0094h
UART-5_USR1 is 63F9_0000h base + 94h offset = 63F9_0094h



UARTx_USR1 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15 PARITYERR	**Parity Error Interrupt Flag**. Indicates a parity error is detected. PARITYERR is cleared by writing 1 to it. Writing 0 to PARITYERR has no effect. When parity is disabled, PARITYERR always reads 0. At reset, PARITYERR is set to 0. 0 No parity error detected 1 Parity error detected (write 1 to clear)
14 RTSS	$\overline{\text{RTS}}$ **Pin Status**. Indicates the current status of the $\overline{\text{RTS}}$ pin. A "snapshot" of the pin is taken immediately before RTSS is presented to the data bus. RTSS cannot be cleared because all writes to RTSS are ignored. At reset, RTSS is set to 0. 0 The $\overline{\text{RTS}}$ pin is high (inactive) 1 The $\overline{\text{RTS}}$ pin is low (active)

Table continues on the next page...

UARTx_USR1 field descriptions (continued)

Field	Description
13 TRDY	**Transmitter Ready Interrupt / DMA Flag**. Indicates that the TxFIFO emptied below its target threshold and requires data. TRDY is automatically cleared when the data level in the TxFIFO exceeds the threshold set by TXTL bits. At reset, TRDY is set to 1. 0 The transmitter does not require data 1 The transmitter requires data (interrupt posted)
12 RTSD	**RTS Delta.** Indicates whether the $\overline{\text{RTS}}$ pin changed state. It (RTSD) generates a maskable interrupt. When in STOP mode, RTS assertion sets RTSD and can be used to wake the processor. The current state of the $\overline{\text{RTS}}$ pin is available on the RTSS bit. Clear RTSD by writing 1 to it. Writing 0 to RTSD has no effect. At reset, RTSD is set to 0. 0 $\overline{\text{RTS}}$ pin did not change state since last cleared 1 $\overline{\text{RTS}}$ pin changed state (write 1 to clear)
11 ESCF	**Escape Sequence Interrupt Flag**. Indicates if an escape sequence was detected. ESCF is asserted when the ESCEN bit is set and an escape sequence is detected in the RxFIFO. Clear ESCF by writing 1 to it. Writing 0 to ESCF has no effect. 0 No escape sequence detected 1 Escape sequence detected (write 1 to clear).
10 FRAMERR	**Frame Error Interrupt Flag**. Indicates that a frame error is detected. The *interrupt_uart* interrupt will be generated if a frame error is detected and the interrupt is enabled. Clear FRAMERR by writing 1 to it. Writing 0 to FRAMERR has no effect. 0 No frame error detected 1 Frame error detected (write 1 to clear)
9 RRDY	**Receiver Ready Interrupt / DMA Flag**. Indicates that the RxFIFO data level is above the threshold set by the RXTL bits. (See the RXTL bits description in UART FIFO Control Register (UART_UFCR) for setting the interrupt threshold.) When asserted, RRDY generates a maskable interrupt or DMA request. RRDY is automatically cleared when data level in the RxFIFO goes below the set threshold level. At reset, RRDY is set to 0. 0 No character ready 1 Character(s) ready (interrupt posted)
8 AGTIM	Ageing Timer Interrupt Flag. Indicates that data in the RxFIFO has been idle for a time of 8 character lengths (where a character length consists of 7 or 8 bits, depending on the setting of the WS bit in UCR2, with the bit time corresponding to the baud rate setting) and FIFO data level is less than RxFIFO threshold level (RXTL in the UFCR). Clear by writing a 1 to it. 0 AGTIM is not active 1 AGTIM is active (write 1 to clear)
7 DTRD	**DTR Delta.** Indicates whether $\overline{\text{DTR}}$ (in DCE mode) or $\overline{\text{DSR}}$ (in DTE mode) pins changed state. DTRD generates a maskable interrupt if DTRDEN (UCR3[3]) is set. Clear DTRD by writing 1 to it. Writing 0 to DTRD has no effect. 0 $\overline{\text{DTR}}$ (DCE) or $\overline{\text{DSR}}$ (DTE) pin did not change state since last cleared 1 $\overline{\text{DTR}}$ (DCE) or $\overline{\text{DSR}}$ (DTE) pin changed state (write 1 to clear)
6 RXDS	Receiver IDLE Interrupt Flag. Indicates that the receiver state machine is in an IDLE state, the next state is IDLE, and the receive pin is high. RXDS is automatically cleared when a character is received. RXDS is active only when the receiver is enabled.

Table continues on the next page...

UARTx_USR1 field descriptions (continued)

Field	Description
	0 Receive in progress 1 Receiver is IDLE
5 AIRINT	Asynchronous IR WAKE Interrupt Flag. Indicates that the IR WAKE pulse was detected on the RXD pin. Clear AIRINT by writing 1 to it. Writing 0 to AIRINT has no effect. 0 No pulse was detected on the RXD IrDA pin 1 A pulse was detected on the RXD IrDA pin
4 AWAKE	Asynchronous WAKE Interrupt Flag. Indicates that a falling edge was detected on the RXD pin. Clear AWAKE by writing 1 to it. Writing 0 to AWAKE has no effect. 0 No falling edge was detected on the RXD Serial pin 1 A falling edge was detected on the RXD Serial pin
3–0 -	Reserved

76.13.9 UART Status Register 2 (UARTx_USR2)

Addresses: UART-1_USR2 is 53FB_C000h base + 98h offset = 53FB_C098h
UART-2_USR2 is 53FC_0000h base + 98h offset = 53FC_0098h
UART-3_USR2 is 53FC_4000h base + 98h offset = 53FC_4098h
UART-4_USR2 is 53FF_0000h base + 98h offset = 53FF_0098h
UART-5_USR2 is 63F9_0000h base + 98h offset = 63F9_0098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADET	TXFE	DTRF	IDLE	ACST	RIDELT	RIIN	IRINT	WAKE	DCDDELT	DCDIN	RTSF	TXDC	BRCD	ORE	RDR
W	w1c		w1c	w1c	w1c	w1c		w1c	w1c	w1c		w1c		w1c	w1c	
Reset	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0

UARTx_USR2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15 ADET	**Automatic Baud Rate Detect Complete**. Indicates that an "A" or "a" was received and that the receiver detected and verified the incoming baud rate. Clear ADET by writing 1 to it. Writing 0 to ADET has no effect. 0 ASCII "A" or "a" was not received 1 ASCII "A" or "a" was received (write 1 to clear)
14 TXFE	**Transmit Buffer FIFO Empty**. Indicates that the transmit buffer (TxFIFO) is empty. TXFE is cleared automatically when data is written to the TxFIFO. Even though TXFE is high, the transmission might still be in progress. 0 The transmit buffer (TxFIFO) is not empty 1 The transmit buffer (TxFIFO) is empty
13 DTRF	**DTR edge triggered interrupt flag**. This bit is asserted, when the programmed edge is detected on the $\overline{\text{DTR}}$ pin (DCE mode) or on $\overline{\text{DSR}}$ (DTE mode). This flag can cause an interrupt if DTREN (UCR3[13]) is enabled. 0 Programmed edge not detected on DTR/DSR 1 Programmed edge detected on DTR/DSR (write 1 to clear)
12 IDLE	**Idle Condition**. Indicates that an idle condition has existed for more than a programmed amount frame (see Idle Line Detect. An interrupt can be generated by this IDLE bit if IDEN (UCR1[12]) is enabled. IDLE is cleared by writing 1 to it. Writing 0 to IDLE has no effect. 0 No idle condition detected 1 Idle condition detected (write 1 to clear)
11 ACST	**Autobaud Counter Stopped**. In autobaud detection (ADBR=1), indicates the counter which determines the baud rate was running and is now stopped. This means either START bit is finished (if ADNIMP=1), or Bit 0 is finished (if ADNIMP=0). See New Autobaud Counter Stopped bit and Interrupt, for more details. An interrupt can be flagged on *interrupt_uart* if ACIEN=1. 0 Measurement of bit length not finished (in autobaud) 1 Measurement of bit length finished (in autobaud). (write 1 to clear)
10 RIDELT	**Ring Indicator Delta**. This bit is used in DTE mode to indicate that the Ring Indicator input ($\overline{\text{RI}}$) has changed state. This flag can generate an interrupt if RI (UCR3[8]) is enabled. RIDELT is cleared by writing 1 to it. Writing 0 to RIDELT has no effect. 0 Ring Indicator input has not changed state 1 Ring Indicator input has changed state (write 1 to clear)
9 RIIN	**Ring Indicator Input**. This bit is used in DTE mode to reflect the status if the Ring Indicator input ($\overline{\text{RI}}$). The Ring Indicator input is used to indicate that a ring has occurred. In DCE mode this bit is always zero. 0 Ring Detected 1 No Ring Detected
8 IRINT	**Serial Infrared Interrupt Flag.** When an edge is detected on the RXD pin during SIR Mode, this flag will be asserted. This flag can cause an interrupt which can be masked using the control bit ENIRI: UCR4 [8]. 0 no edge detected 1 valid edge detected (write 1 to clear)

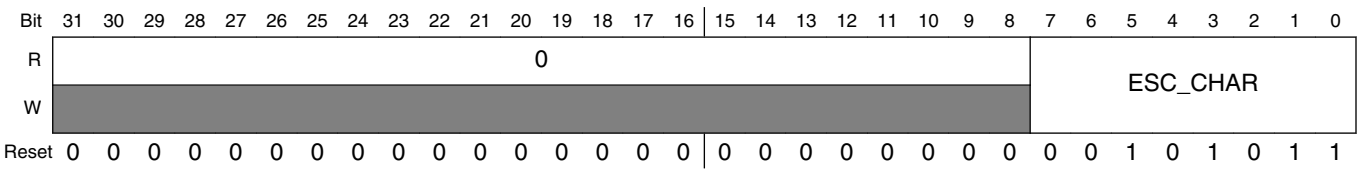
Table continues on the next page...

UARTx_USR2 field descriptions (continued)

Field	Description
7 WAKE	**Wake**. Indicates the start bit is detected. WAKE can generate an interrupt that can be masked using the WKEN bit. Clear WAKE by writing 1 to it. Writing 0 to WAKE has no effect. 0 start bit not detected 1 start bit detected (write 1 to clear)
6 DCDDELT	**Data Carrier Detect Delta**. This bit is used in DTE mode to indicate that the Data Carrier Detect input ($\overline{DCD}$) has changed state. This flag can cause an interrupt if DCD (UCR3[9]) is enabled. When in STOP mode, this bit can be used to wake the processor. In DCE mode this bit is always zero. 0 Data Carrier Detect input has not changed state 1 Data Carrier Detect input has changed state (write 1 to clear)
5 DCDIN	**Data Carrier Detect Input**. This bit is used in DTE mode reflect the status of the Data Carrier Detect input ($\overline{DCD}$). The Data Carrier Detect input is used to indicate that a carrier signal has been detected. In DCE mode this bit is always zero. 0 Carrier signal Detected 1 No Carrier signal Detected
4 RTSF	**RTS Edge Triggered Interrupt Flag. Indicates if** a programmed edge is detected on the $\overline{RTS}$ pin. The RTEC bits select the edge that generates an interrupt (see Table 76-5). RTSF can generate an interrupt that can be masked using the RTSEN bit. Clear RTSF by writing 1 to it. Writing 0 to RTSF has no effect. 0 Programmed edge not detected on $\overline{RTS}$ 1 Programmed edge detected on $\overline{RTS}$ (write 1 to clear)
3 TXDC	**Transmitter Complete**. Indicates that the transmit buffer (TxFIFO) and Shift Register is empty; therefore the transmission is complete. TXDC is cleared automatically when data is written to the TxFIFO. 0 Transmit is incomplete 1 Transmit is complete
2 BRCD	**BREAK Condition Detected**. Indicates that a BREAK condition was detected by the receiver. Clear BRCD by writing 1 to it. Writing 0 to BRCD has no effect. 0 No BREAK condition was detected 1 A BREAK condition was detected (write 1 to clear)
1 ORE	**Overrun Error**. When set to 1, ORE indicates that the receive buffer (RxFIFO) was full (32 chars inside), and a 33rd character has been fully received. This 33rd character has been discarded. Clear ORE by writing 1 to it. Writing 0 to ORE has no effect. 0 No overrun error 1 Overrun error (write 1 to clear)
0 RDR	**Receive Data Ready**-Indicates that at least 1 character is received and written to the RxFIFO. If the URXD register is read and there is only 1 character in the RxFIFO, RDR is automatically cleared. 0 No receive data ready 1 Receive data ready

76.13.10 UART Escape Character Register (UARTx_UESC)

Addresses: UART-1_UESC is 53FB_C000h base + 9Ch offset = 53FB_C09Ch
UART-2_UESC is 53FC_0000h base + 9Ch offset = 53FC_009Ch
UART-3_UESC is 53FC_4000h base + 9Ch offset = 53FC_409Ch
UART-4_UESC is 53FF_0000h base + 9Ch offset = 53FF_009Ch
UART-5_UESC is 63F9_0000h base + 9Ch offset = 63F9_009Ch

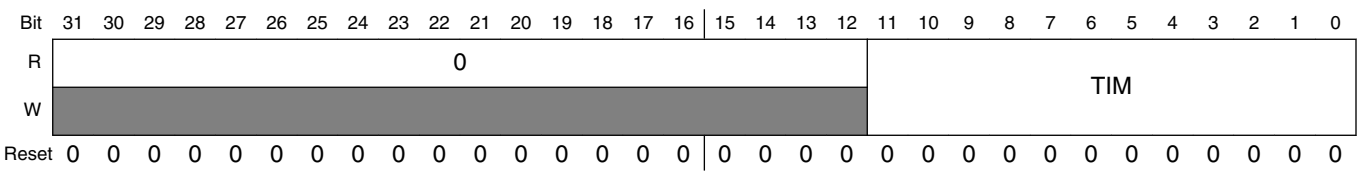


UARTx_UESC field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero. Reserved
7–0 ESC_CHAR	**UART Escape Character**. Holds the selected escape character that all received characters are compared against to detect an escape sequence.

76.13.11 UART Escape Timer Register (UARTx_UTIM)

Addresses: UART-1_UTIM is 53FB_C000h base + A0h offset = 53FB_C0A0h
UART-2_UTIM is 53FC_0000h base + A0h offset = 53FC_00A0h
UART-3_UTIM is 53FC_4000h base + A0h offset = 53FC_40A0h
UART-4_UTIM is 53FF_0000h base + A0h offset = 53FF_00A0h
UART-5_UTIM is 63F9_0000h base + A0h offset = 63F9_00A0h



UARTx_UTIM field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero. Reserved
11–0 TIM	UART Escape Timer. Holds the maximum time interval (in ms) allowed between escape characters. The escape timer register is programmable in intervals of 2 ms. See Escape Sequence Detection and Table 76-10 for more information on the UART escape sequence detection. Reset value 0x000 = 2 ms up to 0xFFF = 8.192 s.

76.13.12 UART BRM Incremental Register (UARTx_UBIR)

This register can be written by both software and hardware. When enabling the automatic baud rate detection feature hardware can write 0x000F value into the UBIR after finishing detecting baud rate. Hardware has higher priority when both software and hardware try to write it at the same cycle[1].

Please note software reset will reset the register to its reset value.

Addresses: UART-1_UBIR is 53FB_C000h base + A4h offset = 53FB_C0A4h
UART-2_UBIR is 53FC_0000h base + A4h offset = 53FC_00A4h
UART-3_UBIR is 53FC_4000h base + A4h offset = 53FC_40A4h
UART-4_UBIR is 53FF_0000h base + A4h offset = 53FF_00A4h
UART-5_UBIR is 63F9_0000h base + A4h offset = 63F9_00A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																INC															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_UBIR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–0 INC	Incremental Numerator. Holds the numerator value minus one of the BRM ratio (see Binary Rate Multiplier (BRM)). The UBIR register MUST be updated before the UBMR register for the baud rate to be updated correctly. If only one register is written to by software, the BRM will ignore this data until the other register is written to by software. Updating this field using byte accesses is not recommended and is undefined.

1. Note: The write priority in the new design is not same as the original UART. In the orginal design, software has higher priotiry than hardware when writing this register at the same time.

76.13.13 UART BRM Modulator Register (UARTx_UBMR)

This register can be written by both software and hardware. When enabling the automatic baud rate detection feature hardware can write a proper value into the UBMR based on detected baud rate. Hardware has higher priority when both software and hardware try to write it at the same cycle[1].

Please note software reset will reset the register to its reset value.

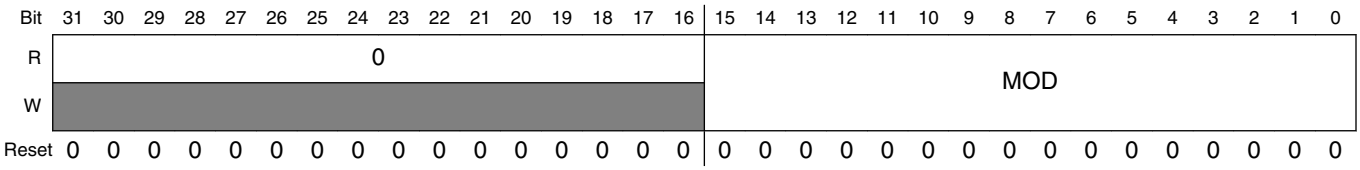
Addresses: UART-1_UBMR is 53FB_C000h base + A8h offset = 53FB_C0A8h

UART-2_UBMR is 53FC_0000h base + A8h offset = 53FC_00A8h

UART-3_UBMR is 53FC_4000h base + A8h offset = 53FC_40A8h

UART-4_UBMR is 53FF_0000h base + A8h offset = 53FF_00A8h

UART-5_UBMR is 63F9_0000h base + A8h offset = 63F9_00A8h



UARTx_UBMR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–0 MOD	**Modulator Denominator.** Holds the value of the denominator minus one of the BRM ratio (see Binary Rate Multiplier (BRM)). The UBIR register MUST be updated before the UBMR register for the baud rate to be updated correctly. If only one register is written to by software, the BRM will ignore this data until the other register is written to by software. Updating this register using byte accesses is not recommended and undefined.

1. Note: The write priority in the new design is not same as the original UART. In the orginal design, software has higher priotiry than hardware when writing this register at the same time.

76.13.14 UART Baud Rate Count Register (UARTx_UBRC)

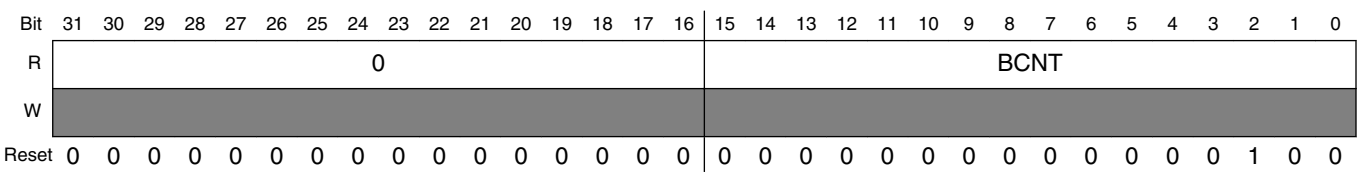
Addresses: UART-1_UBRC is 53FB_C000h base + ACh offset = 53FB_C0ACh

UART-2_UBRC is 53FC_0000h base + ACh offset = 53FC_00ACh

UART-3_UBRC is 53FC_4000h base + ACh offset = 53FC_40ACh

UART-4_UBRC is 53FF_0000h base + ACh offset = 53FF_00ACh

UART-5_UBRC is 63F9_0000h base + ACh offset = 63F9_00ACh



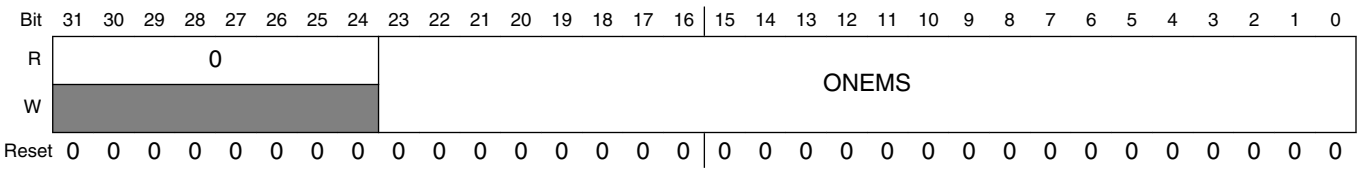
UARTx_UBRC field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero. Reserved
15–0 BCNT	**Baud Rate Count Register.** This read only register is used to count the start bit of the incoming baud rate (if ADNIMP=1), or start bit + bit0 (if ADNIMP=0). When the measurement is done, the Baud Rate Count Register contains the number of UART internal clock cycles (clock after divider) present in an incoming bit. BCNT retains its value until the next Automatic Baud Rate Detection sequence has been initiated. The 16 bit Baud Rate Count register is reset to 4 and stays at hex FFFF in the case of an overflow.

76.13.15 UART One Millisecond Register (UARTx_ONEMS)

This register has been expanded from 16 bits to 24 bits. In previous versions, the 16-bit ONEMS can only support the maximum 65.535MHz (0xFFFFx1000) ref_clk. To support 4Mbps Bluetooth application with 66.5MHz module_clock, the value 0x103C4 (66.5M/1000) should be written into this register. In this case, the 16 bits are not enough to contain the 0x103C4. So this register was expanded to 24 bits to support high frequency of the ref_clk.

Addresses: UART-1_ONEMS is 53FB_C000h base + B0h offset = 53FB_C0B0h
UART-2_ONEMS is 53FC_0000h base + B0h offset = 53FC_00B0h
UART-3_ONEMS is 53FC_4000h base + B0h offset = 53FC_40B0h
UART-4_ONEMS is 53FF_0000h base + B0h offset = 53FF_00B0h
UART-5_ONEMS is 63F9_0000h base + B0h offset = 63F9_00B0h

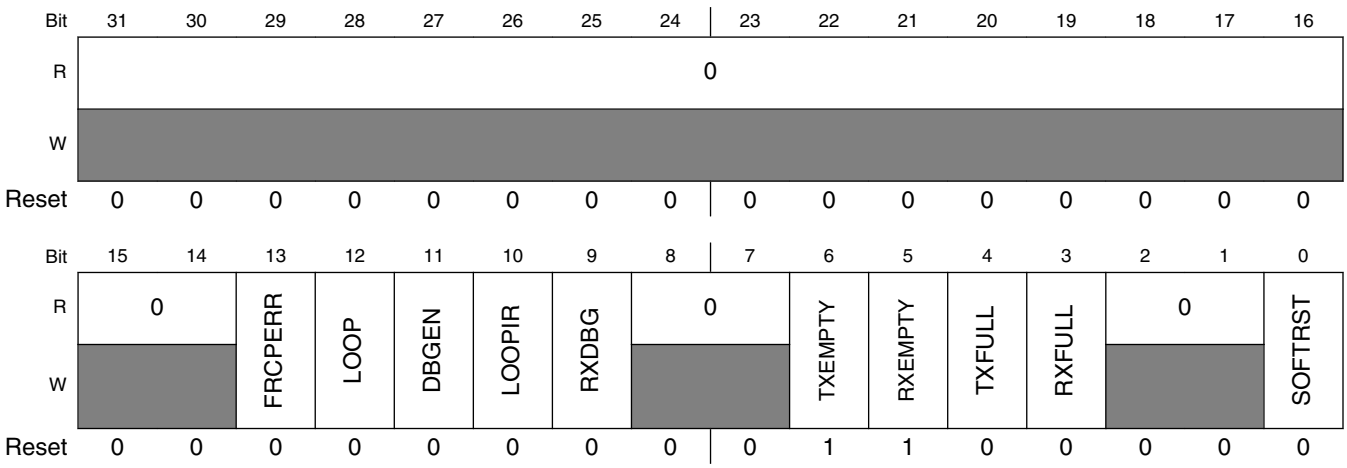


UARTx_ONEMS field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero. Reserved
23–0 ONEMS	**One Millisecond Register.** This 24-bit register must contain the value of the UART internal frequency (*ref_clk* in Figure 76-1) divided by 1000. The internal frequency is obtained after the UART BRM internal divider (F (*ref_clk*) = F(*module_clock*) / RFDIV). In fact this register contains the value corresponding to the number of UART BRM internal clock cycles present in one millisecond. The ONEMS (and UTIM) registers value are used in the escape character detection feature (Escape Sequence Detection) to count the number of clock cycles left between two escape characters. The ONEMS register is also used in infrared special case mode (IRSC = UCR4[5] = 1'b1), see InfraRed Special Case (IRSC) Bit.

76.13.16 UART Test Register (UARTx_UTS)

Addresses: UART-1_UTS is 53FB_C000h base + B4h offset = 53FB_C0B4h
UART-2_UTS is 53FC_0000h base + B4h offset = 53FC_00B4h
UART-3_UTS is 53FC_4000h base + B4h offset = 53FC_40B4h
UART-4_UTS is 53FF_0000h base + B4h offset = 53FF_00B4h
UART-5_UTS is 63F9_0000h base + B4h offset = 63F9_00B4h



UARTx_UTS field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value zero. Reserved
13 FRCPERR	Force Parity Error. Forces the transmitter to generate a parity error if parity is enabled. FRCPERR is provided for system debugging. 0 Generate normal parity 1 Generate inverted parity (error)
12 LOOP	Loop TX and RX for Test. Controls loopback for test purposes. When LOOP is high, the receiver input is internally connected to the transmitter and ignores the RXD pin. The transmitter is unaffected by LOOP. If RXDMUXSEL (UCR3[2]) is set to 1, the loopback is applied on serial and IrDA signals. If RXDMUXSEL is set to 0, the loopback is only applied on serial signals. 0 Normal receiver operation 1 Internally connect the transmitter output to the receiver input
11 DBGEN	$\overline{\text{debug\_enable}}$. This bit controls whether to respond to the *debug_req* input signal. 0 UART will go into debug mode when *debug_req* is HIGH 1 UART will not go into debug mode even if *debug_req* is HIGH
10 LOOPIR	**Loop TX and RX for IR Test (LOOPIR).** This bit controls loopback from transmitter to receiver in the InfraRed interface.

Table continues on the next page...

UARTx_UTS field descriptions (continued)

Field	Description
	0 No IR loop 1 Connect IR transmitter to IR receiver
9 RXDBG	**RX_fifo_debug_mode.** This bit controls the operation of the RX fifo read counter when in debug mode. 0 rx fifo read pointer does not increment 1 rx_fifo read pointer increments as normal
8–7 Reserved	This read-only field is reserved and always has the value zero. Reserved
6 TXEMPTY	TxFIFO Empty. Indicates that the TxFIFO is empty. 0 The TxFIFO is not empty 1 The TxFIFO is empty
5 RXEMPTY	RxFIFO Empty. Indicates the RxFIFO is empty. 0 The RxFIFO is not empty 1 The RxFIFO is empty
4 TXFULL	TxFIFO FULL. Indicates the TxFIFO is full. 0 The TxFIFO is not full 1 The TxFIFO is full
3 RXFULL	RxFIFO FULL. Indicates the RxFIFO is full. 0 The RxFIFO is not full 1 The RxFIFO is full
2–1 Reserved	This read-only field is reserved and always has the value zero. Reserved
0 SOFTRST	Software Reset. Indicates the status of the software reset ($\overline{\text{SRST}}$ bit of UCR2). 0 Software reset inactive 1 Software reset active

Chapter 77
Universal Serial Bus Controller (USB)

77.1 Overview

The USB controller block provides high performance USB functionality that conforms to the USB 2.0 specification, and the OTG supplement.

The USB controller consists of four independent USB controller cores: one OTG controller core, and three Host-only controller cores. Each controller core can support ULPI, Serial, UTMI, IC-USB or HSIC interface according to its feature. All four controller cores are single-port cores. For the OTG core, there is only one port. It is used as both a downstream and upstream port. For the Host-only core, there is also only one port which is used as a downstream port.

The following figure is a block diagram of USB.

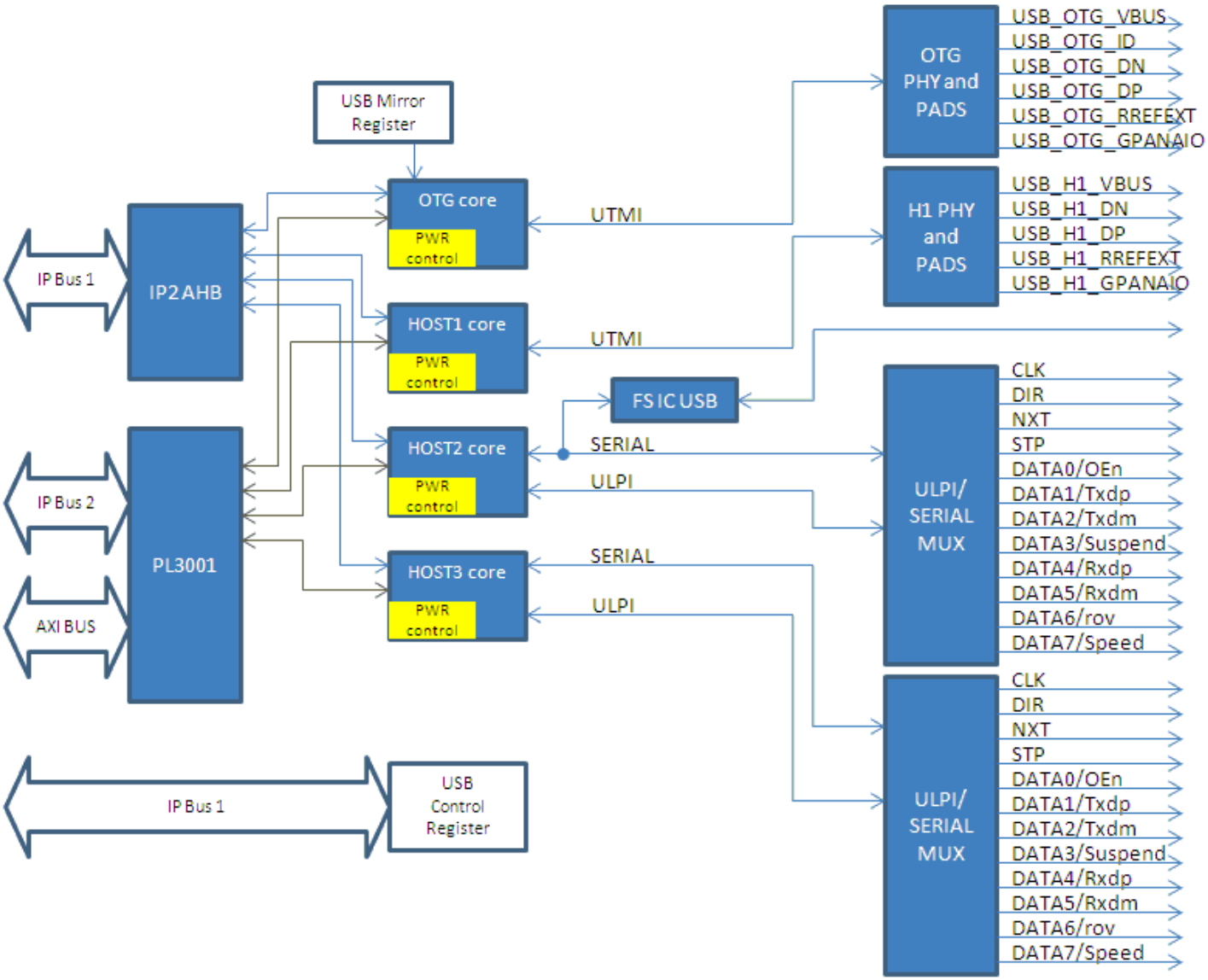


Figure 77-1. USBOH3 Block Diagram

77.2 Features

The USB includes the following features:

- High-speed/full-speed/low-speed host only core (Host1)
  - HS/FS/LS UTMI compliant interface
- High-speed/full-speed/low-speed host only core (Host2)
  - HS/FS/LS ULPI compliant interface
  - Software configurable for full speed/low speed interface for Serial transceiver
  - Full Speed Inter-Chip USB compliant interface (IC-USB)

- High-speed/full-speed/low-speed Host-only core (Host3)
  - HS/FS/LS ULPI compliant interface
  - Software configurable for full-speed/low-speed interface for Serial transceiver
- High-speed/full-speed/low-speed OTG core
  - HS/FS/LS UTMI compliant interface
  - High Speed, Full Speed and Low Speed operation in Host mode (with UTMI transceiver)
  - High Speed, and Full Speed operation in Peripheral mode (with UTMI transceiver)
  - Hardware support for OTG signaling, session request protocol, and host negotiation protocol
  - Up to 8 bidirectional endpoints
- Low-power mode with local and remote wake-up capability
- Serial PHY interfaces configurable for bidirectional/unidirectional and differential/ single ended
- Embedded DMA controller

77.2.1 Modes of Operation

The USB has two main modes of operation: normal mode and low power mode.

Each USB controller core can be configured for High Speed operation (480 Mbps), Full Speed operation (12Mbps) and Low Speed operation (1.5 Mbps).

This chapter explains the operation modes.

77.2.1.1 Normal Mode

For OTG controller core, it could work in Host mode, or Device (Peripheral) mode; for Host-only controller core, it could work in Host mode only.

Each USB controller core has its corresponding port, which could work in one or more interface modes.

- Host1 Port
  - This port supports on-chip UTMI transceivers only.
- Host2 Port
  - This port supports ULPI, Serial and IC-USB interface.
  - Serial interface mode

    For connections to external Serial transceivers.

- ULPI interface mode

  For connections to external ULPI compatible transceivers.

- IC-USB interface mode

  For connection to USB peripherals with IC-USB interface.

- Host3 Port
  - This port supports ULPI and Serial interface.
  - Serial interface mode

    For connections to external Serial transceivers.

  - ULPI interface mode

    For connections to external ULPI compatible transceivers.

- OTG port
  - This port supports on-chip UTMI transceiver only.

NOTE

As the ULPI and Serial mode do not work at the same time, the ULPI_DATA[7:0] signals are shared as the Serial interface signals.

The signal mapping for serial interface in 4 different serial modes are given in Table 77-1.

Table 77-1. Serial Interface Signal Mapping

	Unidirection & Differential	Unidirection & Single-End	Bidirection & Differential	Bidirection & Single-End
ULPI_DATA[0]	oen	oen	oen	oen
ULPI_DATA[1]	txdp	tx_dat	dp	dat
ULPI_DATA[2]	txdm	tx_se0	dm	se0
ULPI_DATA[3]	suspend	suspend	suspend	suspend
ULPI_DATA[4]	rxdp	rxdp	-	-
ULPI_DATA[5]	rxdm	rxdm	-	-
ULPI_DATA[6]	rcv	rcv	rcv	-
ULPI_DATA[7]	speed	speed	speed	speed

77.2.1.2 Low Power Mode

Each USB controller core has a low power mode (Suspend Mode) to save power consumption.

As described in USB 2.0 specification, devices can go into the Suspend state after they see a constant Idle state on their upstream facing bus. The OTG controller core could enter Suspend Mode after there is no bus activity for 3ms on port when it is in Device operation mode. Host1, Host2 and Host3 controller core do not have this function because they support host mode only.

Either the local ARM platform or the remote USB Host/Peripheral can initiate a wake-up sequence to resume USB communication. For details about Suspend/Resume, please see .

77.3 Functional Description

These sections describe the functionality of the various building blocks of the USB.

77.3.1 USB Host Controller 1

Host Controller 1 is an instantiation of EHCI-compatible core which supports High Speed / Full Speed / Low Speed operation.

77.3.1.1 Pins Used for Host Controller 1

USB Host Controller 1 signals do not multiplex with other pins on the i.MX53. Therefore it is not necessary to port IOMUX settings for this particular interface.

- Supply USB_H1_VDDA33 with 3.3V
- Supply USB_H1_VDDA25 with 2.5V

USB H1 PHY pins used:

- USB_H1_VBUS (D15)
- USB_H1_VDDA25 (F13)
- USB_H1_DN (B17)
- USB_H1_VDDA33 (G13)
- USB_H1_DP (A17)
- USB_H1_RREFEXT (B16)
- USB_H1_GPANAIO (A16)

77.3.2 USB Host Controller 2

Host Controller 2 is an instantiation of the EHCI-compatible core which supports High Speed / Full Speed / Low Speed operation.

The Host 2 core USB signals do not connect directly to the Host2 I/O pins. Instead, the signals pass through the Serial and ULPI Mux to allow for additional functionality on Host2 Port. Host controller 2 is configured for operation with a FS/LS serial transceiver or a parallel ULPI transceiver (HS/FS/LS). The selection between transceiver type is software programmable. The block defaults to serial mode.

77.3.3 USB Host Controller 3

Host Controller 3 is an instantiation of EHCI-compatible core which supports High Speed / Full Speed / Low Speed operation.

The Host 3 core USB signals do not connect directly to the Host3 I/O pins. Instead, the signals pass through the Serial and ULPI Mux to allow for additional functionality on Host3 Port. Host controller 3 is configured for operation with a FS/LS serial transceiver or a parallel ULPI transceiver (HS/FS/LS). The selection between transceiver type is software programmable. The block defaults to serial mode.

77.3.4 USB OTG Controller

OTG Controller 3 is an instantiation of EHCI-compatible OTG core which supports High Speed / Full Speed / Low Speed operation.

The OTG controller core supports HS/FS/LS operation in Host mode and HS/FS operation in device mode.

77.3.4.1 Host Mode

The controller supports direct connection of a HS/FS/LS device with on-chip UTMI transceiver.

Although there is no separate Transaction Translator block in the system, the transaction translator function normally associated with a USB 2.0 high speed hub has been implemented within the DMA and protocol engine blocks to support connection to full and low speed devices.

77.3.4.2 Peripheral (Device) Mode

- Up to 8 bidirectional endpoints
- High/Full speed operation
- Supports HNP and SRP
- Remote wake-up capable

77.3.4.3 Pins Used for OTG

USB OTG signals do not multiplex with other pins on the i.MX53. Therefore it is not necessary to port IOMUX settings for this particular interface.

The only required setup is to:

- supply USB_OTG_VDDA33 with 3.3V
- supply USB_OTG_VDDA25 with 2.5V

USB OTG PHY pins used:

- USB_OTG_VBUS (E15)
- USB_OTG_ID (C16)
- USB_OTG_VDDA25 (F14)
- USB_OTG_DN (A19)
- USB_OTG_VDDA33 (G14)
- USB_OTG_DP (B19)
- USB_OTG_RREFEXT (D16)
- USB_OTG_GPANAIO (F15)

77.3.5 Interrupts

77.3.5.1 USB Core Interrupts

Each USB core uses one dedicated vector in the Interrupt Table. The vector numbers associated with each of the cores can be found in the Interrupt section.

With the exception of the wake-up interrupts, all of the interrupt sources are controlled in the USB Cores. Refer to the Register Interface for details.

77.3.5.2 USB Wake-Up Interrupts

Each USB Core has an associated wake-up interrupt. The wake-up interrupts are generated outside of the USB controller cores, but use the same vector as the corresponding USB controller cores' interrupt.

These interrupts are generated by the Power Control blocks which run on the 32KHz standby clock. The wake-up interrupt is designed to work even when the USB and ARM platform clocks are disabled, such that a wake-up condition on the USB bus can re-activate the ARM platform clocks.

Because the wake-up interrupt is generated and cleared on a 32 KHz clock, this interrupt request responds very slowly to clear actions. For this reason, the software must disable the wake-up interrupt to clear the request flag. Disabling the interrupt masks the request instantaneously as this is clocked by the ARM platform clock. The software should than wait for at least three 32 KHz clock cycles before re-enabling this interrupt to allow sufficient time for the request flag to clear. Because this interrupt is only used during low power modes of the USB, it is sufficient to enable the wake-up interrupt just prior to entering the USB suspend mode.

77.3.6 USB Clock System

This section describes the clock diagram of USB.

The figure below shows the clock system of USB.

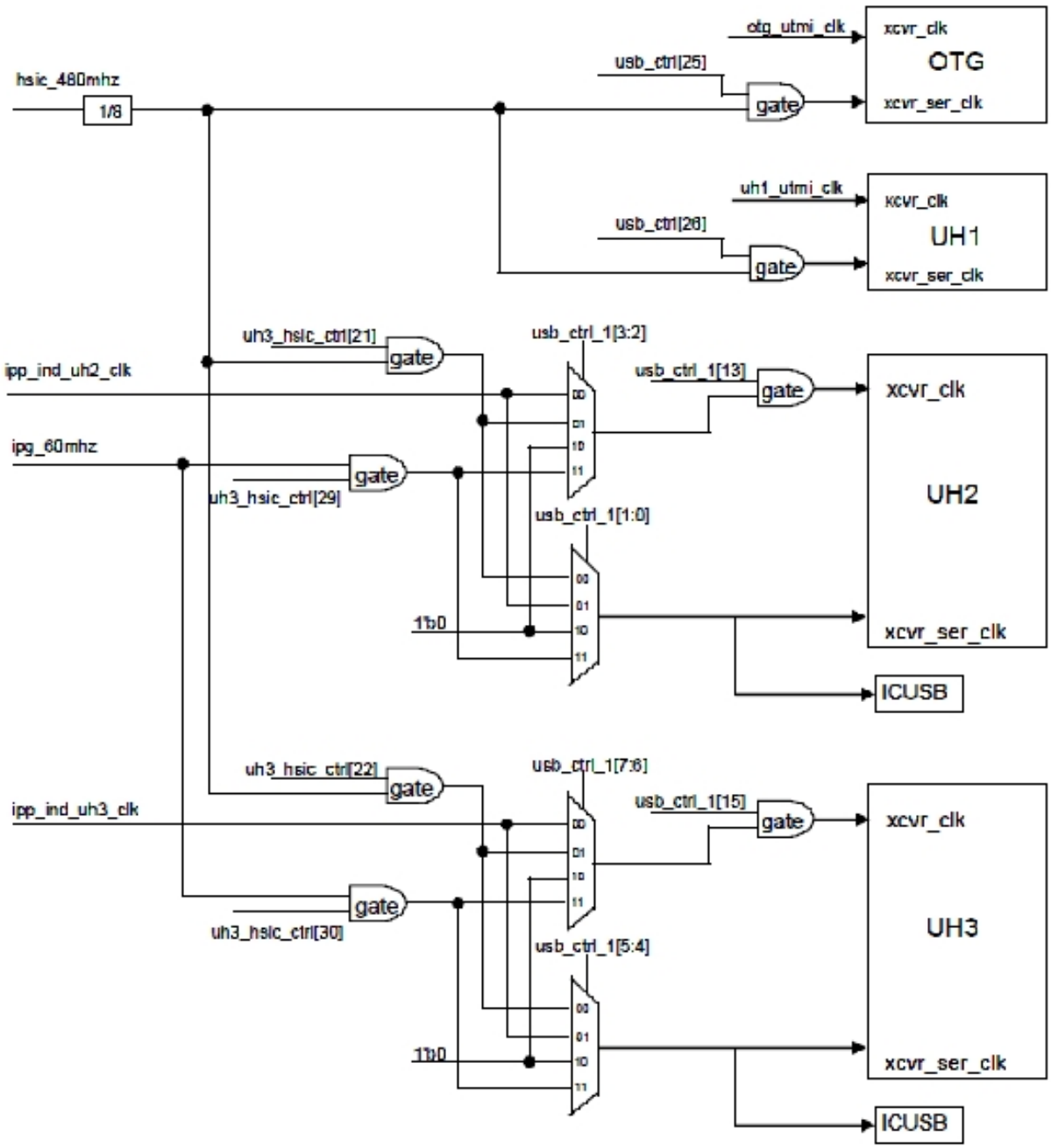


Figure 77-2. USB Clock System

- HSIC_480MHz: this clock comes from on-chip OTG UTMI PHY DPLL, its freq is 480 MHz.
- otg_utmi_clk: this clock comes from on-chip OTG UTMI PHY sieclock port, its freq is 30 MHz.
- uh1_utmi_clk: this clock comes from on-chip Host1 UTMI PHY sieclock port, its freq is 30 MHz.
- ipg_60mhz: this clock comes from system DPLL, its freq is 60MHz.
- ipp_ind_uh2_clk: this clock is from Host2 port external ULPI PHY clock pad

- ipp_ind_uh3_clk: this clock is from Host3 port external ULPI PHY clock pad.
- usb_ctrl: this register is the USB Control Register 0 (Base+0x800).
- usb_ctrl_1: this register is the USB Control Register 1 (Base+0x810).
- usb_uh2_ctrl: this register is the USB Host2 Control Register (Base+0x814).
- usb_uh3_ctrl: this register is the USB Host3 Control Register (Base+0x818).
- uh3_hsic_ctrl: this register is the USB Host3 HS_IC Control Register (Base+0x824).
- otg_ser_mode/uh1_ser_mode: these signals are used to enable the serial mode of OTG/Host1

77.4 USB Operation Model

This section describes the detailed application knowledge for Host1, Host2, Host3 and OTG ports. It can be generally divided in two parts, one is for Host and the other is for Device.

Host part applies to all three host ports, and to OTG port when operating in Host mode. Device part only applies to OTG port when operating in Device mode.

77.4.1 Register Interface

Slave accesses from the controlling processor enables access to the configuration, control, and status registers. One function of the system address map is the registers base address, which must begin on a DWord (32-bit) boundary.

Register offset definitions are listed in the table below.

Configuration, control and status registers are divided into three categories, identification, capability and operational registers.

- Identification registers are used to declare the slave interface presence along with the complete set of the hardware configuration parameters.
- Static, read only capability registers define the software limits, restrictions, and capabilities of the host/device controller.
- Operational registers are comprised of dynamic control or status registers that may be read only, read/write, or read/write to clear. The following sections define the use of these registers.

EHCI registers are listed alongside device registers to show the complementary nature of host and device control.

The following table describes the Interface register sets.

Table 77-2. Interface Register Sets

Offset	Register Set	Explanation
000h-07Ch	Identification Registers	Identification registers are used to declare the slave interface presence and include a table of the hardware configuration parameters.
100h-124h	Capability Registers	Capability registers specify the limits, restrictions, and capabilities of a host/device controller implementation. These values are used as parameters to the host/device controller driver.
080h-0FCh 140h-1FCh	Operational Registers	Operational registers are used by the system software to control and monitor the operational state of the host/device controller.

77.4.1.1 Configuration, Control and Status Register Set

The following table describes the Device/Host capability registers.

NOTE

Depending on implementation, "*x*" can have the following values: UOG, UH1, UH2 or UH3.

.

Table 77-3. Device/Host Capability Registers

Offset	Size (Bytes)	Mnemonic	Register Name	Device Mode	Host Mode
000h	4	USB_*x*_ID	Identification Register	O	O
004h	4	USB_*x*_HWGENERAL	General Hardware Parameters	O	O
008h	4	USB_*x*_HWHOST	Host Hardware Parameters	X	O
00Ch	4	USB_*x*_HWDEVICE	Device Hardware Parameters	O	X
010h	4	USB_*x*_HWTXBUF	TX Buffer Hardware Parameters	O	O
014h	4	USB_*x*_HWRXBUF	RX Buffer Hardware Parameters	O	O
018-07Fh		-	Reserved		
080h	4	USB_*x*_GPTIMER0LD	General Purpose Timer #0 Load Register	O	O
084h	4	USB_*x*_GPTIMER0CTRL	General Purpose Timer #0 Control Register	O	O
088h	4	USB_*x*_GPTIMER1LD	General Purpose Timer #1 Load Register	O	O
08Ch	4	USB_*x*_GPTIMER1CTRL	General Purpose Timer #1 Control Register	O	O
090h	4	USB_*x*_SBUSCFG	System Bus Interface Configuration Register	O	O
094-09Fh		-	Reserved		
100h	1	USB_*x*_CAPLENGTH	Capability Register Length	O	O

Table continues on the next page...

Table 77-3. Device/Host Capability Registers (continued)

Offset	Size (Bytes)	Mnemonic	Register Name	Device Mode	Host Mode
101h		-	Reserved		
102h	2	USB_*x*_HCIVERSION	Host Controller Interface Version Number	X	O
104h	4	USB_*x*_HCSPARAMS	Host Controller Structural Parameters	X	O
108h	4	USB_*x*_HCCPARAMS	Host Controller Capability Parameters	X	O
10C-11Fh		-	Reserved		
120h	2	USB_*x*_DCIVERSION	Device Controller Interface Version Number	O	X
122h	2	-	Reserved		
124h	4	USB_*x*_DCCPARAMS	Device Controller Capability Parameters	O	X
128-13Fh		-	Reserved		
140h	4	USB_*x*_USBCMD	USB Command Register	O	O
144h	4	USB_*x*_USBSTS	USB Status Register	O	O
148h	4	USB_*x*_USBINTR	USB Interrupt Enable Register	O	O
14Ch	4	USB_*x*_FRINDEX	USB Frame Index	O	O
150h	4	-	Reserved		
154h	4	USB_*x*_PERIODICLISTBASE	Frame List Base Address	X	O
		USB_*x*_DEVICEADDR	USB Device Address	O	X
158h	4	USB_*x*_ASYNCLISTADDR	Next Asynchronous List Address	X	O
		USB_*x*_ENDPOINTLISTADDR	Address at Endpoint list in memory	O	X
15Ch	4	-	Reserved		
160h	4	USB_*x*_BURSTSIZE	Programmable Burst Size	O	O
164h	4	USB_*x*_TXFILLTUNING	Host Transmit Pre-Buffer Packet Tuning	X	O
168h	4	-	Reserved		
16Ch	4	USB_*x*_IC_USB	IC_USB enable and voltage negotiation	O	O
170h	4	-	Reserved		
174h	4	USB_*x*_ENDPTNAK	Endpoint NAK register	O	X
178h	4	USB_*x*_ENDPTNAKEN	Endpoint NAK Enable register	O	X
17Ch	4	-	Reserved		
180h	4	USB_*x*_CONFIGFLAG	Configured Flag Register	X	O
184h	4	USB_*x*_PORTSC1	Port Status/Control Register 1	O	O
188-1A3h		-	Reserved		
1A4h	4	USB_*x*_OTGSC	On-The-Go Status/Control Register (OTG only)	O	O
1A8h	4	USB_*x*_USBMODE	USB Device Mode	O	O

Table continues on the next page...

Table 77-3. Device/Host Capability Registers (continued)

Offset	Size (Bytes)	Mnemonic	Register Name	Device Mode	Host Mode
1ACh	4	USB_*x*_ENDPTSETUPSTAT	Endpoint Setup Status	O	X
1B0h	4	USB_*x*_ENDPTPRIME	Endpoint Initialization	O	X
1B4h	4	USB_*x*_ENDPTFLUSH	Endpoint De-Initialization	O	X
1B8h	4	USB_*x*_ENDPTSTATUS	Endpoint Status	O	X
1BCh	4	USB_*x*_ENDPTCOMPLETE	Endpoint Complete	O	X
1C0 1C4 ... 1DCh	64	USB_*x*_ENDPTCTRL0 USB_*x*_ENDPTCTRL1 .... USB_*x*_ENDPTCTRL7	Endpoint Control Register 0-7	O	X

77.4.1.2 Identification Registers

Identification registers are used to declare the slave interface presence and include a table of the hardware configuration parameters.

77.4.1.3 OTG Operations

77.4.1.3.1 Register Bits

In the previous section, the Register interface has behaviors described for device mode and behaviors described for host mode. However, during OTG operations it is necessary to perform tasks independent of the controller mode.

Note also from Endpoint Control0 (USB_UOG_ENDPTCTRL0), that the only way to transition the controller mode out of host or device mode is with the controller reset bit. Therefore, it is also necessary for the OTG tasks to be performed independent of a controller reset as well as independent of the controller mode.

The following figure shows the controller mode.

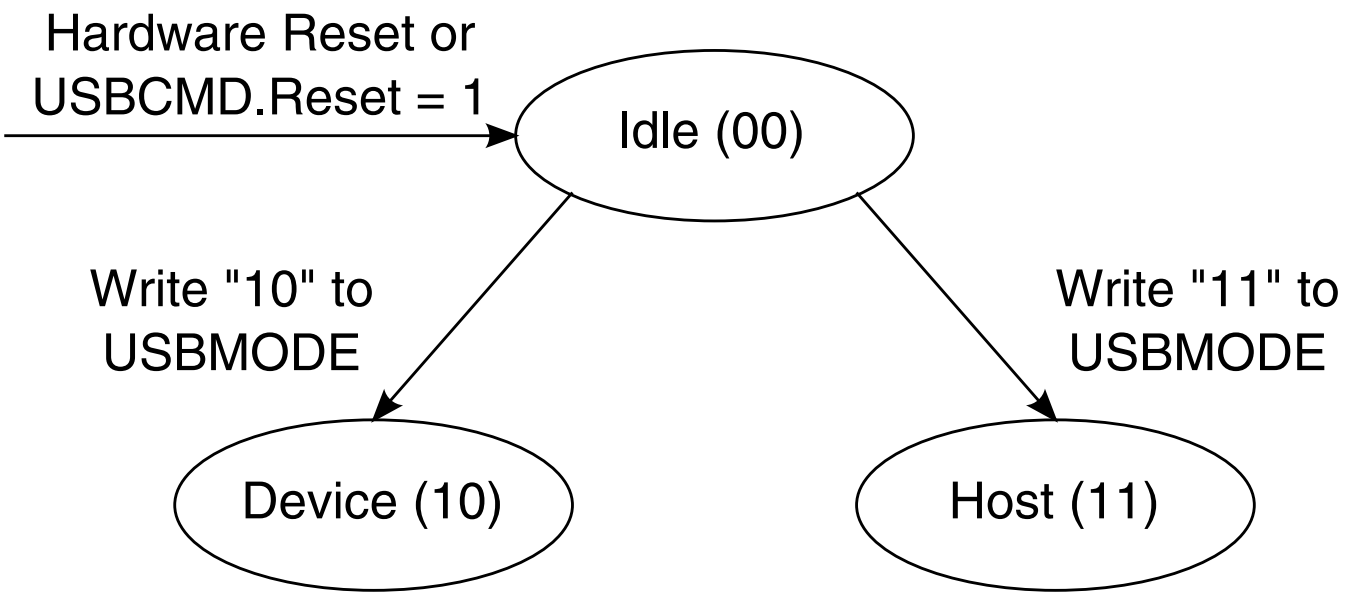


Figure 77-3. Controller Mode

To this end, the listed below are the register bits that are used for OTG operations, which are independent of the controller mode and are also not affected by a write to the reset bit in the USBCMD register:

All Identification Registers

All Device/Host Capability Registers

OTGSC: All bits

PORTSC1:

Physical Interface Select

Physical Interface Serial Select

Physical Interface Data Width

Physical Interface Low Power

Physical Interface Wake Signals

Port Indicators

Port Power

77.4.1.3.2 Hardware Assist

The hardware assist provides automated response and sequencing that may not be possible to software with significant interrupt latency response times.

The use of this additional circuitry is optional and can be used to assist the 3 sequences below.

77.4.1.3.2.1 Auto-Reset

When the HAAR is set to one, the host automatically starts a reset after a connect event. This shortcuts the normal process where software is notified of the connect event and starts the reset.

Software still receives notification of the connect event but should not write the reset bit when the HAAR is set. Software is notified again after the reset is complete via the enable change bit in the USB_PORTSC1 register which cause a port change interrupt.

This assist ensures the OTG parameter TB_ACON_BSE0_MAX = 1ms is met.

77.4.1.3.2.2 Data-Pulse

Writing one to HADP starts a data pulse of approximately 7ms in duration and then automatically cease the data pulsing. During the data pulse, the DP is set and then cleared. This automation relieves software from accurately controlling the data-pulse duration.

During the data pulse, the HCD can poll to see that the HADP and DP bit have returned low to recognize the completion or simply launch the data pulse and wait to see if a VBUS Valid interrupt occurs when the A-side supplies bus power.

This assist ensures data pulsing meets the OTG requirement of > 5ms and < 10ms.

77.4.1.3.2.3 B-Disconnect to A-Connect

During HNP, the B-disconnect occurs from the OTG A_suspend state and within 3 ms, the A device must enable the pullup on the DP leg in the A-peripheral state.

When HABA is set, the Host Controller port is in suspend mode, and the device disconnects, then this hardware assist begins.

1. Reset the OTG core.
2. Write the OTG core into device mode.
3. Write the device run bit to 1 and enable necessary interrupts including:

USB Reset Enable (URE); enables interrupt on usb bus reset to device

Sleep Enable (SLE); enables interrupt on device suspend

Port Change Detect Enable (PCE); enables interrupt on device connect

When software has enabled this hardware assist, it must not interfere during the transition and should not write any register in the core until it gets an interrupt from the device controller signifying that a reset interrupt has occurred or at least first verify that the core has entered device mode. HCD/DCD must not activate the core soft reset at any time because this action is performed by hardware. During the transition, the software may see an interrupt from the disconnect and/or other spurious interrupts (that is SOF, and so on) that may or may not cascade and my be cleared by the soft reset depending on the software response time.

After the core has entered device mode by the hardware assist, the DCD must ensure that the USB_ENDPTLISTADDR is programmed properly before the host sends a setup packet. Because the end of the reset duration, which may be initiated quickly (a few microseconds) after connect, will require at a minimum 50 ms, this is the time for which the DCD must be ready to accept setup packets after having received notification that the reset has been detected or simply that the OTG is in device mode which ever occurs first.

In the case where the A-peripheral fails to see a reset after the controller enters device mode and engages the DP-pullup, the device controller interrupt the DCD signifying that a suspend has occurred.

This assist ensures the parameter TA_BDIS_ACON_MAX = 3ms is met.

77.4.2 Host Data Structures

This section defines the interface data structures used to communicate control, status, and data between HCD (software) and the Enhanced Host Controller (hardware). The data structure definitions in this chapter support a 32-bit memory buffer address space.

The interface consists of a Periodic Schedule, Periodic Frame List, Asynchronous Schedule, Isochronous Transaction Descriptors, Split-transaction Isochronous Transfer Descriptors, Queue Heads, and Queue Element Transfer Descriptors.

The periodic frame list is the root of all periodic (isochronous and interrupt transfer type) support for the host controller interface. The asynchronous list is the root for all the bulk and control transfer type support. Isochronous data streams are managed using Isochronous Transaction Descriptors. Isochronous split-transaction data streams are managed with Split-transaction Isochronous Transfer Descriptors. All Interrupt, Control, and Bulk data streams are managed via queue heads and Queue Element Transfer

Descriptors. These data structures are optimized to reduce the total memory footprint of the schedule and to reduce (on average) the number of memory accesses needed to execute a USB transaction.

Note that software must ensure that no interface data structure reachable by the EHCI host controller spans a 4 K-page boundary.

The data structures defined in this section are (from the host controller's perspective) a mix of read-only and read/writeable fields. The host controller must preserve the read-only fields on all data structure writes.

77.4.2.1 Periodic Frame List

This schedule is for all periodic transfers (isochronous and interrupt). The periodic schedule is referenced from the operational registers space using the USB_PERIODICLISTBASEaddress register and the USB_FRINDEX register.

The periodic schedule is based on an array of pointers called the Periodic Frame List. The USB_PERIODICLISTBASE address register is combined with the USB_FRINDEX register to produce a memory pointer into the frame list. The Periodic Frame List implements a sliding window of work over time.

The following figure shows the organization of periodic schedule.

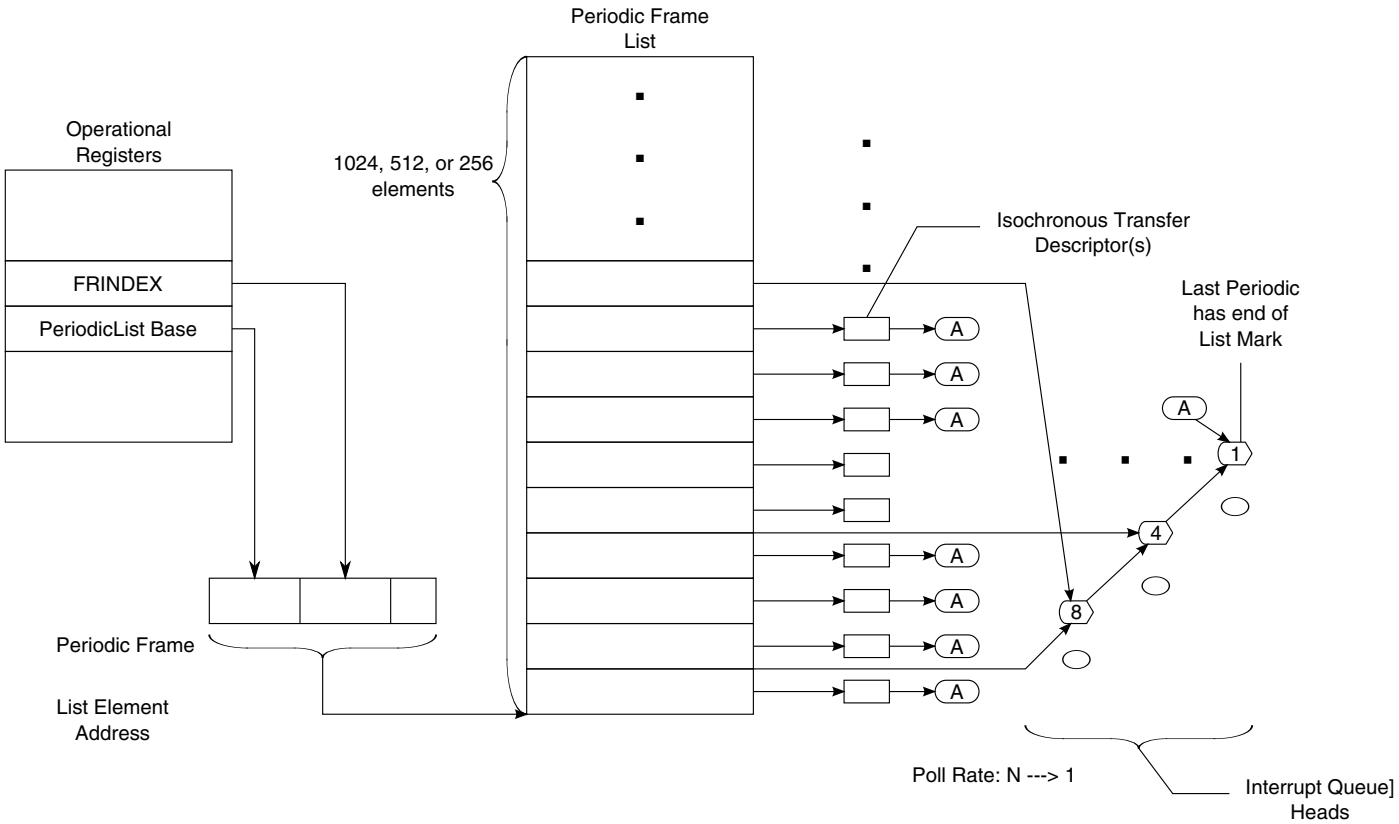


Figure 77-4. Periodic Schedule Organization

Split transaction Interrupt, Bulk and Control are also managed using queue heads and queue element transfer descriptors.

The periodic frame list is a 4 K-page aligned array of Frame List Link pointers. The length of the frame list may be programmable. The programmability of the periodic frame list is exported to system software via the USB_HCCPARAMS register. If non-programmable, the length is 1024 elements. If programmable, the length can be selected by system software as one of 256, 512, or 1024 elements. An implementation must support all three sizes. Programming the size (that is, the number of elements) is accomplished by system software writing the appropriate value into Frame List Size field in the USB_USBCMD register.

Frame List Link pointers direct the host controller to the first work item in the frame's periodic schedule for the current micro-frame. The link pointers are aligned on DWord boundaries within the Frame List.

The table below illustrates the format of the Frame list element pointer.

Table 77-4. Format of Frame List Element Pointer

Table continues on the next page...

Table 77-4. Format of Frame List Element Pointer (continued)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Frame List Link Pointer																											0		Typ			03-00 H

Frame List Link pointers always reference memory objects that are 32-byte aligned. The referenced object may be an isochronous transfer descriptor for high-speed devices, a split-transaction isochronous transfer descriptor (for full-speed isochronous endpoints), or a queue head (used to support high-, full- and low-speed interrupt). System software should not place non-periodic schedule items into the periodic schedule. The least significant bits in a frame list pointer are used to key the host controller as to the type of object the pointer is referencing.

The least significant bit is the T-Bit (bit 0). When this bit is set to a one, the host controller never uses the value of the frame list pointer as a physical memory pointer. The Typ field is used to indicate the exact type of data structure being referenced by this pointer. The value encodings are.

Table 77-5. Typ Field Value Definitions

Value	Meaning
00b	Isochronous Transfer Descriptor
01b	Queue Head
10b	Split Transaction Isochronous Transfer Descriptor.
11b	Frame Span Traversal Node.

77.4.2.2 Asynchronous List Queue Head Pointer

The Asynchronous Transfer List (based at the USB_ASYNCLISTADDR register) is where all of the control and bulk transfers are managed. Host controllers use this list only when it reaches the end of the periodic list, the periodic list is disabled, or the periodic list is empty.

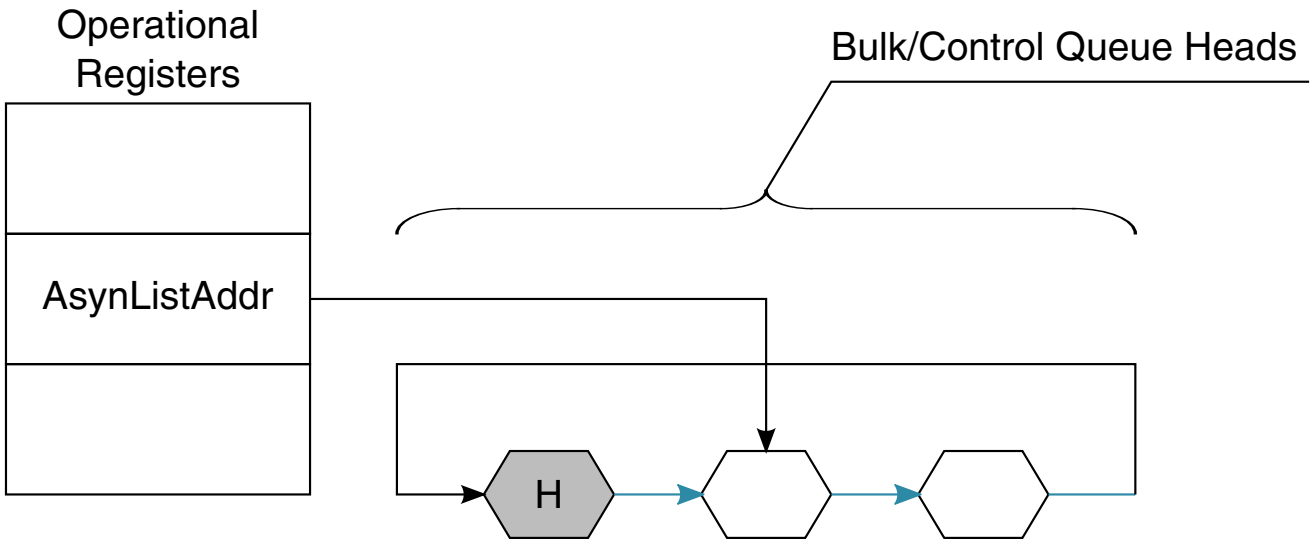


Figure 77-5. Asynchronous Schedule Organization

The Asynchronous list is a simple circular list of queue heads. The USB_ASYNCLISTADDR register is simply a pointer to the next queue head. This implements a pure round-robin service for all queue heads linked into the asynchronous list.

77.4.2.3 Isochronous (High-Speed) Transfer Descriptor (iTD)

The format of an isochronous transfer descriptor is shown in the table below.

This structure is used only for high-speed isochronous endpoints. All other transfer types should use queue structures. Isochronous TDs must be aligned on a 32-byte boundary.

Table 77-6. Isochronous Transaction Descriptor (iTD)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Next Link Pointer																											0		Typ		T	03-00H
Status				Transaction 0 Length												ioc	PG*			Transaction 0 Offset*												07-04H
Status				Transaction 1 Length												ioc	PG*			Transaction 1 Offset*												0B-08H
Status				Transaction 2 Length												ioc	PG*			Transaction 2 Offset*												0F-0CH

Table continues on the next page...

Table 77-6. Isochronous Transaction Descriptor (iTD) (continued)

Status	Transaction 3 Length	ioc	PG*	Transaction 3 Offset*	13-10H
Status	Transaction 4 Length	ioc	PG*	Transaction 4 Offset*	17-14H
Status	Transaction 5 Length	ioc	PG*	Transaction 5 Offset*	1B-18H
Status	Transaction 6 Length	ioc	PG*	Transaction 6 Offset*	1F-1CH
Status	Transaction 7 Length	ioc	PG*	Transaction 7 Offset*	23-20H
Buffer Pointer (Page 0)				EndPt / R / Device Address	27-24H
Buffer Pointer (Page 1)				I/O / Maximum Packet Size	2B-28H
Buffer Pointer (Page 2)				- / Mult	2F-2CH
Buffer Pointer (Page 3)				-	33-30H
Buffer Pointer (Page 4)				-	37-34H
Buffer Pointer (Page 5)				-	3B-38H
Buffer Pointer (Page 6)				-	3F-3CH

	Host Controller Read/Write		Host Controller Read Only.

These fields may be modified by the host controller if the I/O field indicates an OUT.

77.4.2.3.1 Next Link Pointer-Host Data Structures

The first DWord of an iTD is a pointer to the next schedule data structure.

The following table describes the Next Schedule Element pointer field.

Table 77-7. Next Schedule Element Pointer

Bit	Description

Table continues on the next page...

Table 77-7. Next Schedule Element Pointer (continued)

31-5	Link Pointer (LP). These bits correspond to memory address signals [31:5], respectively. This field points to another Isochronous Transaction Descriptor (iTD/siTD) or Queue Head (QH).
4-3	Reserved. These bits are reserved and their value has no effect on operation. Software should initialize this field to zero.
2-1	QH/(s)iTD Select (Typ). This field indicates to the Host Controller whether the item referenced is an iTD, siTD or a QH. This allows the Host Controller to perform the proper type of processing on the item after it is fetched. Value encodings are: Value Meaning 00b iTD (isochronous transfer descriptor) 01b QH (queue head) 10b siTD (split transaction isochronous transfer descriptor 11b FSTN (frame span traversal node)
0	Terminate (T). 1= Link Pointer field is not valid. 0= Link Pointer field is valid.

77.4.2.3.2 iTD Transaction Status and Control List

DWords 1 through 8 are eight slots of transaction control and status.

Each transaction description includes:

- Status results field
- Transaction length (bytes to send for OUT transactions and bytes received for IN transactions).
- Buffer offset. The PG and Transaction X Offset fields are used with the buffer pointer list to construct the starting buffer address for the transaction.

The host controller uses the information in each transaction description plus the endpoint information contained in the first three DWords of the Buffer Page Pointer list, to execute a transaction on the

The following table describes iTD Transaction Status and Control fields.

Table 77-8. iTD Transaction Status and Control

Bit	Description	
31-28	Status. This field records the status of the transaction executed by the host controller for this slot. This field is a bit vector with the following encoding:	
	Bit	Definition
	31	Active. Set to one by software to enable the execution of an isochronous transaction by the Host Controller. When the transaction associated with this descriptor is completed, the Host Controller sets this bit to zero indicating that a transaction for this element should not be executed when it is next encountered in the schedule.
	30	Data Buffer Error. Set to a one by the Host Controller during status update to indicate that the Host Controller is unable to keep up with the reception of incoming data (overrun) or is unable to supply data fast enough during transmission (under run). If an overrun condition occurs, no action is necessary.
	29	Babble Detected. Set to one by the Host Controller during status update when" babble" is detected during the transaction generated by this descriptor.

Table continues on the next page...

Table 77-8. iTD Transaction Status and Control (continued)

Bit	Description	
	28	Transaction Error (XactErr). Set to one by the Host Controller during status update in the case where the host did not receive a valid response from the device (Timeout, CRC, Bad PID, etc.). This bit may only be set for isochronous IN transactions.
27-16	Transaction X Length. For an OUT, this field is the number of data bytes the host controller sends during the transaction. The host controller is not required to update this field to reflect the actual number of bytes transferred during the transfer. For an IN, the initial value of the endpoint to deliver. During the status update, the host controller writes back the field is the number of bytes the host expects the number of bytes successfully received. The value in this register is the actual byte count (0‡zero length data, 1‡one byte, 2‡two bytes, etc.). The maximum value this field may contain is 0xC00 (3072).	
15	Interrupt On Complete (IOC). If this bit is set to one, it specifies that when this transaction completes, the Host Controller should issue an interrupt at the next interrupt threshold.	
14-12	Page Select (PG). These bits are set by software to indicate which of the buffer page pointers the offset field in this slot should be concatenated to produce the starting memory address for this transaction. The valid range of values for this field is 0 to 6.	
11-0	Transaction X Offset. This field is a value that is an offset, expressed in bytes, from the beginning of a buffer. This field is concatenated onto the buffer page pointer indicated in the adjacent *PG* field to produce the starting buffer address for this transaction.	

77.4.2.3.3 iTD Buffer Page Pointer List (Plus)

DWords 9-15 of an isochronous transaction descriptor are nominally page pointers (4 K aligned) to the data buffer for this transfer descriptor. This data structure requires the associated data buffer to be contiguous (relative to virtual memory), but allows the physical memory pages to be non-contiguous.

Seven page pointers are provided to support the expression of eight isochronous transfers. The seven pointers allow for 3 (transactions) * 1024 (maximum packet size) * 8 (transaction records) (24576 bytes) to be moved with this data structure, regardless of the alignment offset of the first page.

Because each pointer is a 4 K aligned page pointer, the least significant 12 bits in several of the page pointers are used for other purposes.

The tables below illustrate the field descriptions.

Table 77-9. iTD Buffer Pointer Page 0 (Plus)

Bit	Description
31-12	Buffer Pointer (Page 0). This is a 4 K aligned pointer to physical memory. Corresponds to memory address bits [31:12].
11-8	Endpoint Number (Endpt). This 4-bit field selects the particular endpoint number on the device serving as the data source or sink.
7	Reserved. Bit reserved for future use and should be initialized by software to zero.
6-0	Device Address. This field selects the specific device serving as the data source or sink.

Table 77-10. iTD Buffer Pointer Page 1 (Plus)

Bit	Description
31-12	Buffer Pointer (Page 1). This is a 4K aligned pointer to physical memory. Corresponds to memory address bits [31:12].
11	Direction (I/O). 0 = OUT; 1 = IN. This field encodes whether the high-speed transaction should use an IN or OUT PID.
10-0	Maximum Packet Size. This directly corresponds to the maximum packet size of the associated endpoint *(wMaxPacketSize)*. This field is used for high-bandwidth endpoints where more than one transaction is issued per transaction description (per micro-frame). This field is used with the *Multi* field to support high-bandwidth pipes. This field is also used for all IN transfers to detect packet babble. Software should not set a value larger than 1024 (400h). Any value larger yields undefined results.

Table 77-11. iTD Buffer Pointer Page 2 (Plus)

Bit	Description
31-12	Buffer Pointer. This is a 4K aligned pointer to physical memory. Corresponds to memory address bits [31:12].
11-2	Reserved. This bit reserved for future use and should be set to zero.
1-0	Multi. This field is used to indicate to the host controller the number of transactions that should be executed per transaction description (per micro-frame). The valid values are: Value Meaning 00b Reserved. A zero in this field yields undefined results. 01b One transaction to be issued for this endpoint per micro- frame 10b Two transactions to be issued for this endpoint per micro-frame 11b Three transactions to be issued for this endpoint per micro- frame

Table 77-12. iTD Buffer Pointer Page 3-6

Bit	Description
31-12	Buffer Pointer. This is a 4 K aligned pointer to physical memory. Corresponds to memory address bits [31:12].
11-0	Reserved. These bits reserved for future use and should be set to zero.

77.4.2.4 Split Transaction Isochronous Transfer Descriptor (siTD)

All Full-speed isochronous transfers through the internal transaction translator are managed using the siTD data structure. This data structure satisfies the operational requirements for managing the split transaction protocol.

The table below shows the Split Transaction Isochronous Transfer Descriptor (siTD).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Next Link Pointer																											0		Typ		T	03-00H

Table continues on the next page...

I/O	Port Number						-	Hub Addr						-					EndPt	-	Device Address											07-04H
Reserved														µFrame C-mask						µFrame S-mask												0B-08H
ioc	P	-					Total Bytes to Transfer							µFrame C-prog-mask						Status												0F-0CH
Buffer Pointer (Page 0)																			Current Offset													13-10H
Buffer Pointer (Page 1)																			-							TP	T-count					17-14H
Back Pointer																										0					T	1B-18H

Table 77-13. Split-transaction Isochronous Transaction Descriptor (siTD)

	Host Controller Read/Write		Host Controller Read Only.

77.4.2.4.1 Next Link Pointer

DWord0 of a siTD is a pointer to the next schedule data structure.

The following table describes the Next Link Pointer fields.

Table 77-14. Next Link Pointer

Bit	Description
31-5	Next Link Pointer (LP). This field contains the address of the next data object to be processed in the periodic list and corresponds to memory address signals [31:5], respectively.
4-3	Reserved. These bits must be written as zeros.
2-1	QH/(s)iTD Select (Typ). This field indicates to the Host Controller whether the item referenced is an iTD/siTD or a QH. This allows the Host Controller to perform the proper type of processing on the item after it is fetched. Value encodings are: Value Meaning 00b iTD (isochronous transfer descriptor) 01b QH (queue head) 10b siTD (split transaction isochronous transfer descriptor 11b FSTN (frame span traversal node)
0	Terminate (T). 1=Link Pointer field is not valid. 0=Link Pointer is valid.

77.4.2.4.2 siTD Endpoint Capabilities/Characteristics

DWords 1 and 2 specify static information about the full-speed endpoint, the addressing of the parent Companion Controller, and micro-frame scheduling control.

The tables below describe the Endpoint and transaction translator characteristics and micro-frame schedule control fields.

Table 77-15. Endpoint and Transaction Translator Characteristics

Bit	Description
31	Direction (I/O).0 = OUT; 1 = IN. This field encodes whether the full-speed transaction should be an IN or OUT.
30-24	Port Number. This field is the port number of the recipient Transaction Translator.
23	Reserved. Bit reserved and should be set to zero.
22-16	Hub Address. This field holds the device address of the Companion Controllers' hub.
15-12	Reserved. Field reserved and should be set to zero.
11-8	Endpoint Number (Endpt). This 4-bit field selects the particular endpoint number on the device serving as the data source or sink.
7	Reserved. Bit is reserved for future use. It should be set to zero.
6-0	Device Address. This field selects the specific device serving as the data source or sink.

Table 77-16. Micro-frame Schedule Control

Bit	Description
31-16	Reserved. This field reserved for future use. It should be set to zero.
15-8	Split Completion Mask (mFrame C-Mask). This field (along with the *Active* and *SplitX- state* fields in the *Status* byte) is used to determine during which micro-frames the host controller should execute complete-split transactions. When the criteria for using this field is met, an all zeros value has undefined behavior. The host controller uses the value of the three low-order bits of the FRINDEX register to index into this bit field. If the FRINDEX register value indexes to a position where the m*Frame C-Mask* field is a one, then this siTD is a candidate for transaction execution. There may be more than one bit in this mask set.
7-0	Split Start Mask (mFrame S-mask). This field (along with the *Active* and *SplitX-state* fields in the *Status* byte) is used to determine during which micro-frames the host controller should execute start-split transactions. The host controller uses the value of the three low-order bits of the FRINDEX register to index into this bit field. If the FRINDEX register value indexes to a position where the m*Frame S-mask* field is a one, then this siTD is a candidate for transaction execution. An all zeros value in this field, in combination with existing periodic frame list has undefined results.

77.4.2.4.3 siTD Transfer State

DWords 3-6 are used to manage the state of the transfer.

The following table describes siTD transfer state fields.

Table 77-17. siTD Transfer Status and Control

Bit	Description
31	Interrupt On Complete (ioc). 0 = Do not interrupt when transaction is complete. 1 = Do interrupt when transaction is complete. When the host controller determines that the split transaction has completed it asserts a hardware interrupt at the next interrupt threshold.
30	Page Select (P). Used to indicate which data page pointer should be concatenated with the *CurrentOffset* field to construct a data buffer pointer (0 selects *Page 0* pointer and 1 selects *Page 1*). The host controller is not required to write this field back when the siTD is retired (*Active* bit transitioned from a one to a zero).
29-26	Reserved. This field reserved for future use and should be set to zero.

Table continues on the next page...

Table 77-17. siTD Transfer Status and Control (continued)

Bit	Description	
25-16	Total Bytes To Transfer. This field is initialized by software to the total number of bytes expected in this transfer. Maximum value is 1023 (3FFh)	
15-8	µFrame Complete-split Progress Mask (C-prog-Mask). This field is used by the host controller to record which split-completes has been executed.	
7-0	Status. This field records the status of the transaction executed by the host controller for this slot. This field is a bit vector with the following encoding:	
	Bit	Definition
	7	Active. Set to one by software to enable the execution of an isochronous split transaction by the Host Controller.
	6	ERR. Set to a one by the Host Controller when an ERR response is received from the Companion Controller.
	5	Data Buffer Error. Set to a one by the Host Controller during status update to indicate that the Host Controller is unable to keep up with the reception of incoming data (overrun) or is unable to supply data fast enough during transmission (under run). In the case of an under run, the Host Controller transmits an incorrect CRC (thus invalidating the data at the endpoint). If an overrun condition occurs, no action is necessary.
	4	Babble Detected. Set to a one by the Host Controller during status update when" babble" is detected during the transaction generated by this descriptor.
	3	Transaction Error (XactErr). Set to a one by the Host Controller during status update in the case where the host did not receive a valid response from the device (Timeout, CRC, Bad PID, etc.). This bit is set only for IN transactions.
	2	Missed Micro-Frame. The host controller detected that a host-induced hold- off caused the host controller to miss a required complete-split transaction.
	1	Split Transaction State (SplitXstate). The bit encodings are: Value Meaning 00b Do Start Split. This value directs the host controller to issue a Start split transaction to the endpoint when a match is encountered in the S-mask. 01b Do Complete Split. This value directs the host controller to issue a Complete split transaction to the endpoint when a match is encountered in the C-mask.
	0	Reserved. Bit reserved for future use and should be set to zero.

77.4.2.4.4 siTD Buffer Pointer List (plus)

DWords 4 and 5 are the data buffer page pointers for the transfer. This structure supports one physical page cross. The most significant 20 bits of each DWord in this section are the 4 K (page) aligned buffer pointers.

The least significant 12 bits of each DWord are used as additional transfer state. The following table describes the siTD buffer pointer fields.

Table 77-18. Buffer Page Pointer List (plus)

Bit	Description	
31-12	Buffer Pointer List. Bits [31:12] of DWords 4 and 5 are 4 K paged aligned, physical memory addresses. These bits correspond to physical address bits [31:12] respectively. The lower 12 bits in each pointer are defined and used as specified below. The field *P* specifies the *current* active pointer	
11-0	Page 0: Current Offset. The 12 least significant bits of the Page 0 pointer is the current byte offset for the current page pointer (as selected with the page indicator bit (*P* field)). The host controller is not required to write this field back when the siTD is retired (*Active* bit transitioned from a one to a zero). The least significant bits of Page 1 pointer is split into three sub-fields Page 1:	
	Bits	Description
	11-5	Reserved
	4-3	Transaction position (TP). This field is used with T-count to determine whether to send *all*, *first*, *middle*, or *last* with each outbound transaction payload. System software must initialize this field with the appropriate starting value. The host controller must correctly manage this state during the lifetime of the transfer. The bit encodings are: Value Meaning 00b All. The entire full-speed transaction data payload is in this transaction (that is, less than or equal to 188 bytes). 01b Begin. This is the first data payload for a full- speed that is greater than 188 bytes.transaction 10B Mid. This is the *middle* payload for a full-speed OUT transaction that is larger than 188 bytes. 11b End. This is the *last* payload for a full-speed OUT transaction that was larger than 188 bytes.
	2-0	Transaction count (T-Count). Software initializes this field with the number of OUT start-splits this transfer requires. Any value larger than 6 is undefined.

77.4.2.4.5 siTD Back Link Pointer

DWord 6 of a siTD is simply another schedule link pointer. This pointer is always zero, or references a siTD, and it cannot reference any other schedule data structure.

The following table describes the siTD back link pointer fields

.

Table 77-19. siTD Back Link Pointer

Bit	Description
31-5	siTD Back Pointer. This field is a physical memory pointer to a siTD.
4-1	Reserved. This field is reserved for future use. It should be set to zero.
0	Terminate (T). 1 = siTD Back Pointer field is not valid. 0 = siTD Back Pointer field is valid.

77.4.2.5 Queue Element Transfer Descriptor (qTD)

This data structure is only used with a queue head. It is used for one or more USB transactions, as well as to transfer up to 20480 (5*4096) bytes.

The structure contains two structure pointers used for queue advancement, a DWord of transfer state, and a five-element array of data buffer pointers. It is 32 bytes (or one 32-byte cache line). Also, this data structure must be physically contiguous.

The buffer associated with this transfer must be virtually contiguous. The buffer may start on any byte boundary. A separate buffer pointer list element must be used for each physical page in the buffer, regardless of whether the buffer is physically contiguous.

Host controller updates (host controller writes) to stand-alone qTDs only occur during transfer retirement. References in the following bit field definitions of updates to the qTD are to the qTD portion of a queue head.

The following table shows the Queue head data structure.

Table 77-20. Queue Head Data Structure

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Next qTD Pointer																											0				T	03-00H
Alternate Next qTD Pointer																											0				T	07-04H
dt	Total Bytes to Transfer															ioc	C_Page			Cerr		PID Code		Status								0B-08H
Buffer Pointer (page 0)																				Current Offset												0F-0CH
Buffer Pointer (page 0)																				-												13-10H

Table continues on the next page...

Table 77-20. Queue Head Data Structure (continued)

Buffer Pointer (page 0)	-	17-14H
Buffer Pointer (page 0)	-	1B-18H
Buffer Pointer (page 0)	-	1F-1CH

	Host Controller Read/Write		Host Controller Read Only.

Queue Element Transfer Descriptors must be aligned on 32-byte boundaries.

77.4.2.5.1 Next qTD Pointer

The first DWord of an element transfer descriptor is a pointer to another transfer element descriptor.

The following table describes Next qTD pointer fields.

Table 77-21. qTD Next Element Transfer Pointer (DWord 0)

Bit	Description
31-5	Next Transfer Element Pointer. This field contains the physical memory address of the next qTD to be processed. The field corresponds to memory address signals[31:5], respectively.
4-1	Reserved
0	Terminate (T). 1= pointer is invalid. 0=Pointer is valid (points to a valid Transfer Element Descriptor). This bit indicates to the Host Controller that there are no more valid entries in the queue.

77.4.2.5.2 Alternate Next qTD Pointer

The second DWord of a queue element transfer descriptor is used to support hardware-only advance of the data stream to the next client buffer on short packet. To be more explicit the host controller always uses this pointer when the current qTD is retired due to short packet.

The following table describes the TD Alternate Next Element Transfer Pointer field descriptions.

Table 77-22. TD Alternate Next Element Transfer Pointer (DWord 1)

Bit	Description
31-5	Alternate Next Transfer Element Pointer. This field contains the physical memory address of the next qTD to be processed in the event that the current qTD execution encounters a short packet (for an IN transaction). The field corresponds to memory address signals [31:5], respectively.
4-1	Reserved
0	Terminate (T). 1= pointer is invalid. 0=Pointer is valid (points to a valid Transfer Element Descriptor). This bit indicates to the Host Controller that there are no more valid entries in the queue.

77.4.2.5.3 qTD Token

The third DWord of a queue element transfer descriptor contains most of the information the host controller requires to execute a USB transaction (the remaining endpoint-addressing information is specified in the queue head).

NOTE

The field descriptions forward reference fields defined in the queue head. Where necessary, these forward references are preceded with a QH notation.

The following table describes the TD Token fields.

Table 77-23. TD Token (DWord 2)

Bit	Description
31	Data Toggle. This is the data toggle sequence bit. The use of this bit depends on the setting of the *Data Toggle Control* bit in the queue head.
30-16	Total Bytes to Transfer. This field specifies the total number of bytes to be moved with this transfer descriptor. This field is decremented by the number of bytes actually moved during the transaction, only on the successful completion of the transaction. The maximum value software may store in this field is 5 * 4K (5000H). This is the maximum number of bytes 5 page pointers can access. If the value of this field is zero when the host controller fetches this transfer descriptor (and the active bit is set), the host controller executes a zero-length transaction and retires the transfer descriptor. It is not a requirement for OUT transfers that *Total Bytes To Transfer* be an even multiple of QHD.Maximum Packet Length. If software builds such a transfer descriptor for an OUT transfer, the last transaction is always less than QHD.Maximum Packet Length. Although it is possible to create a transfer up to 20K this assumes the 1st offset into the first page is 0. When the offset cannot be predetermined, crossing past the 5th page can be guaranteed by limiting the total bytes to 16K**. Therefore, the maximum recommended transfer is 16 K(4000H).
15	Interrupt On Complete (IOC). If this bit is set to a one, it specifies that when this qTD is completed, the Host Controller should issue an interrupt at the next interrupt threshold.
14-12	Current Page (C_Page). This field is used as an index into the qTD buffer pointer list. Valid values are in the range 0H to 4H. The host controller is not required to write this field back when the qTD is retired.

Table continues on the next page...

Table 77-23. TD Token (DWord 2) (continued)

Bit	Description	
11-10	Error Counter (CERR). This field is a 2-bit down counter that keeps track of the number of consecutive Errors detected while executing this qTD. If this field is programmed with a non-zero value during set-up, the Host Controller decrements the count and writes it back to the qTD if the transaction fails. If the counter counts from one to zero, the Host Controller marks the qTD inactive, sets the *Halted* bit to a one, and error status bit for the error that caused *CERR* to decrement to zero. An interrupt is generated if the *USB Error Interrupt Enable* bit in the USBINTR register is set to a one. If HCD programs this field to zero during set-up, the Host Controller does not count errors for this qTD and there is no limit on the retries of this qTD. Note that write-backs of intermediate execution state are to the queue head overlay area, not the qTD. Error Decrement Counter Transaction Error Yes Data Buffer Error No[3] Stalled No[1] Babble Detected No[1] No Error No[2]	
	Error	Decrement Counter Error Decrement Counter
	1	Detection of Babble or Stall automatically halts the queue head. Thus, count is not decremented
	2	If the EPS field indicates a HS device or the queue head is in the Asynchronous Schedule (and *PIDCode* indicates an IN or OUT) and a bus transaction completes and the host controller does not detect a transaction error, then the host controller should reset *CERR* to extend the total number of errors for this transaction. For example, *CERR* should be reset with maximum value (3) on each successful completion of a transaction. The host controller must never reset this field if the value at the start of the transaction is 00b. See Split Transaction Interrupt for CERR adjustment rules when the EPS field indicates a FS or LS device and the queue head is in the Periodic Schedule. See Asynchronous - Do Complete Split for CERR adjustment rules when the EPS field indicates a FS or LS device, the queue head is in the Asynchronous schedule and the *PIDCode* indicates a SETUP.
	3	Data buffer errors are host problems. They don't count against the device's retries.
	NOTE: Software must not program CERR to a value of zero when the EPS field is programmed with a value indicating a Full- or Low-speed device. This combination could result in undefined behavior.	
9-8	PID Code. This field is an encoding of the token, which should be used for transactions associated with this transfer descriptor. Encodings are:	
	00b	OUT Token generates token (E1H)
	01b	IN Token generates token (69H)
	10b	SETUP Token generates token (2DH) (undefined if endpoint is an interrupt, the queue head is non-zero) transfer type, for example, µ*Frame S-mask* field in
	11b	Reserved
7-0	Status. This field is used by the Host Controller to communicate individual command execution states back to HCD. This field contains the status of the last transaction performed on this qTD. The bit encodings are:	
	Bit	Status Field Description
	7	Active. Set to one by software to enable the execution of transactions by the Host Controller.
	6	Halted. Set to one by the Host Controller during status updates to indicate that a serious error has occurred at the device/endpoint addressed by this qTD. This can be caused by babble, the error counter counting down to zero, or reception of the STALL handshake from the device during a transaction. Any time that a transaction results in the Halted bit being set to a one, the Active bit is also set to zero.

Table continues on the next page...

Table 77-23. TD Token (DWord 2) (continued)

Bit		Description
	5	Data Buffer Error. Set to a one by the Host Controller during status update to indicate that the Host Controller is unable to keep up with the reception of incoming data (overrun) or is unable to supply data fast enough during transmission (under run). If an overrun condition occurs, the Host Controller forces a timeout condition on the USB, invalidating the transaction at the source. If the host controller sets this bit to a one, then it remains a one for the duration of the transfer.
	4	Babble Detected. Set to a one by the Host Controller during status update when" babble" is detected during the transaction. In addition to setting this bit, the Host Controller also sets the *Halted* bit to a one. Because "babble" is considered a fatal error for the transfer, setting the Halted bit to a one insures that no more transactions occur because of this descriptor.
	3	Transaction Error (XactErr). Set to a one by the Host Controller during status update in the case where the host did not receive a valid response from the device (Timeout, CRC, Bad PID, etc.). If the host controller sets this bit to a one, then it remains a one for the duration of the transfer.
	2	Missed Micro-Frame. This bit is ignored unless the *QH.EPS* field indicates a full- or low-speed endpoint and the queue head is in the periodic list. This bit is set when the host controller detected that a host-induced hold-off caused the host controller to miss a required complete-split transaction. If the host controller sets this bit to a one, then it remains a one for the duration of the transfer.
	1	Split Transaction State (SplitXstate). This bit is ignored by the host controller unless the *QH.EPS* field indicates a full- or low-speed endpoint. When a Full- or Low-speed device, the host controller uses this bit to track the state of the split- transaction. The functional requirements of the host controller for managing this state bit and the split transaction protocol depends on whether the endpoint is in the periodic or asynchronous schedule. The bit encodings are: Value Meaning 0b Do Start Split. This value directs the host controller to issue a Start split transaction to the endpoint. 1b Do Complete Split. This value directs the host controller to issue a Complete split transaction to the endpoint.
	0	Ping State (P)/ERR. If the *QH.EPS* field indicates a High-speed device and the *PID_Code* indicates an OUT endpoint, then this is the state bit for the Ping protocol. The bit encodings are: Value Meaning 0b Do OUT. This value directs the host controller to issue an OUT PID to the endpoint. 1b Do Ping. This value directs the host controller to issue a PING PID to the endpoint. If the *QH.EPS* field does not indicate a High-speed device, then this field is used as an error indicator bit. It is set to a one by the host controller whenever a periodic split-transaction receives an ERR handshake.

77.4.2.5.4 qTD Buffer Page Pointer List

The last five DWords of a queue element transfer descriptor is an array of physical memory address pointers. These pointers reference the individual pages of a data buffer.

System software initializes Current Offset field to the starting offset into the current page, where current page is selected through the value in the C_Page field.

The following table describes the qTD Buffer Pointer(s) (DWords 3-7) fields.

Table 77-24. qTD Buffer Pointer(s) (DWords 3-7)

Bit	Description
31-12	Buffer Pointer List. Each element in the list is a 4 K page aligned physical memory address. The lower 12 bits in each pointer are reserved (except for the first one), as each memory pointer must reference the start of a 4 K page. The field C_Page specifies the current active pointer. When the transfer element descriptor is fetched, the starting buffer address is selected using C_Page (similar to an array index to select an array element). If a transaction spans a 4K buffer boundary, the host controller must detect the page-span boundary in the data stream, increment C_Page and advance to the next buffer pointer in the list, and conclude the transaction through the new buffer pointer.
11-0	Current Offset (Reserved). This field is reserved in all pointers except the first one (for example Page 0). The host controller should ignore all reserved bits. For the page 0 current offset interpretation, this field is the byte offset into the current page (as selected by C_Page). The host controller is not required to write this field back when the qTD is retired. Software should ensure the Reserved fields are initialized to zero.

77.4.2.6 Queue Head

The following table shows the Queue head structure layout.

Table 77-25. Queue Head Structure Layout

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Queue Head Horizontal Link Pointer																											0		Typ		T	03-00H
RL			C	Maximum Packet Length											H	dtc	EP		EndPt			I	Device Address									07-04H
Mult		Port Number*								Hub Addr*						μFrame C-mask*							μFrame S-mask*									0B-08H
Current qTD Pointer																													0			0F-0CH
Next qTD Pointer																													0		T	13-10H
Alternate Next qTD pointer																													NakCnt		T	17-14H
dt	Total Bytes to Transfer															ioc	C_Page			Cerr		PID Code		Status								1B-18H
Buffer Pointer (Page 0)																				Current Offset												1F-1CH
Buffer Pointer (Page 1)																				Reserved			C-prog-mask*								23-20H

Table continues on the next page...

Table 77-25. Queue Head Structure Layout (continued)

Buffer Pointer (Page 2)	S-bytes*	FrameTag*	27-24H
Buffer Pointer (Page 3)	-		2B-28H
Buffer Pointer (Page 4)	-		2F-2CH

Transfer Overlay Transfer Results

Static Endpoint State

These fields are used exclusively to support split transactions to USB 2.0 Hubs

	Host Controller Read/Write		Host Controller Read Only.

77.4.2.6.1 Queue Head Horizontal Link Pointer

The first DWord of a Queue Head contains a link pointer to the next data object to be processed after any required processing in this queue has been completed, as well as the control bits defined below.

This pointer may reference a queue head or one of the isochronous transfer descriptors. It must not reference a queue element transfer descriptor.

The following table describes the Queue head DWord 0 fields.

Table 77-26. Queue Head DWord 0

Bit	Description
31-5	Queue Head Horizontal Link Pointer (QHLP). This field contains the address of the next data object to be processed in the horizontal list and corresponds to memory address signals [31:5], respectively.
4-3	Reserved
2-1	QH/(s)iTD Select (Typ). This field indicates to the hardware whether the item referenced by the link pointer is an iTD, siTD or a QH. This allows the Host Controller to perform the proper type of processing on the item after it is fetched. Value encodings are: Value Meaning 00b iTD (isochronous transfer descriptor) 01b QH (queue head) 10b siTD (split transaction isochronous transfer descriptor) 11b FSTN (frame span traversal node)
0	Terminate (T). 1=Last QH (pointer is invalid). 0=Pointer is valid. If the queue head is in the context of the periodic list, a one bit in this field indicates to the host controller that this is the end of the periodic list. This bit is ignored by the host controller when the queue head is in the Asynchronous schedule. Software must ensure that queue heads reachable by the host controller always have valid horizontal link pointers.

77.4.2.6.2 Queue Head Endpoint Capabilities/Characteristics

The second and third DWords of a Queue Head specifies static information about the endpoint. This information does not change over the lifetime of the endpoint.

There are three types of information in this region:

- Endpoint Characteristics. These are the USB endpoint characteristics including addressing, maximum packet size, and endpoint speed.
- Endpoint Capabilities. These are adjustable parameters of the endpoint. They effect how the endpoint data stream is managed by the host controller.
- Split Transaction Characteristics. This data structure is used to manage full- and low-speed data streams for bulk, control, and interrupt via split transactions to USB2.0 Hub Transaction Translator. There are additional fields used for addressing the hub and scheduling the protocol transactions (for periodic).

The host controller must not modify the bits in this region.

The following table describes the Endpoint characteristics: Queue head DWord 1 fields.

Table 77-27. Endpoint Characteristics: Queue Head DWord 1

Bit	Description	
31-28	Nak Count Reload (RL). This field contains a value, which is used by the host controller to reload the Nak Counter field.	
27	Control Endpoint Flag (C). If the *QH.EPS* field indicates the endpoint is not a high-speed device, and the endpoint is a control endpoint, then software must set this bit to a one. Otherwise, it should always set this bit to zero.	
26-16	Maximum Packet Length. This directly corresponds to the maximum packet size of the associated endpoint (*wMaxPacketSize*). The maximum value this field may contain is 0x400 (1024).	
15	Head of Reclamation List Flag (H). This bit is set by System Software to mark a queue head as being the head of the reclamation list.	
14	Data Toggle Control (DTC). This bit specifies where the host controller should get the initial data toggle on an overlay transition. 0b Ignore DT bit from incoming qTD. Host controller preserves DT bit in the queue head. 1b Initial data toggle comes from incoming qTD DT bit. Host controller replaces DT bit in the queue head from the DT bit in the qTD.	
13-12	Endpoint Speed (EPS). This is the speed of the associated endpoint. Bit combinations are:	
	Value	Meaning
	00b	Full-Speed (12 Mbits/sec)
	01b	Low-Speed (1.5 Mbits/sec)
	10b	High-Speed (480 Mbits/sec)
	11b	Reserved
	This field must not be modified by the host controller.	

Table continues on the next page...

Table 77-27. Endpoint Characteristics: Queue Head DWord 1 (continued)

11-8	Endpoint Number (Endpt). This 4-bit field selects the particular endpoint number on the device serving as the data source or sink.
7	Inactivate on Next Transaction (I). This bit is used by system software to request that the host controller set the Active bit to zero. See Rebalancing the Periodic Schedule , for full operational details. This field is only valid when the queue head is in the Periodic Schedule and the *EPS* field indicates a Full or Low-speed endpoint. Setting this bit to one when the queue head is in the Asynchronous Schedule or the *EPS* field indicates a high-speed device yields undefined results.
6-0	Device Address. This field selects the specific device serving as the data source or sink.

The table below describes the Endpoint capabilities: Queue head DWord 2 field descriptions.

Table 77-28. Endpoint Capabilities: Queue Head DWord 2

Bit	Description
31-30	High-Bandwidth Pipe Multiplier (Mult). This field is a multiplier used to key the host controller as the number of successive packets the host controller may submit to the endpoint in the current execution. The host controller makes the simplifying assumption that software properly initializes this field (regardless of location of queue head in the schedules or other run time parameters). The valid values are: Value Meaning 00b Reserved. A zero in this field yields undefined results. 01b One transaction to be issued for this endpoint per micro-frame 10b Two transactions to be issued for this endpoint per micro-frame 11b Three transactions to be issued for this endpoint per micro-frame
29-23	Port Number. This field is ignored by the host controller unless the *EPS* field indicates a full- or low-speed device. The value is the port number identifier on the USB 2.0 Hub (for hub at device address *Hub Addr* below), below which the full- or low-speed device associated with this endpoint is attached. This information is used in the split-transaction protocol.
22-16	Hub Addr. This field is ignored by the host controller unless the *EPS* field indicates a full-or low-speed device. The value is the USB device address of the USB 2.0 Hub below which the full- or low-speed device associated with this endpoint is attached. This field is used in the split-transaction protocol.
15-8	Split Completion Mask (µFrame C-Mask). This field is ignored by the host controller unless the *EPS* field indicates this device is a low- or full-speed device and this queue head is in the periodic list. This field (along with the *Active* and *SplitX-state* fields) is used to determine during which micro-frames the host controller should execute a complete-split transaction. When the criteria for using this field are met, a zero value in this field has undefined behavior. This field is used by the host controller to match against the three low-order bits of the FRINDEX register. If the FRINDEX register bits decode to a position where the µ*Frame C- Mask* field is a one, then this queue head is a candidate for transaction execution. There may be more than one bit in this mask set.
7-0	Interrupt Schedule Mask (µFrame S-mask). This field is used for all endpoint speeds. Software should set this field to a zero when the queue head is on the asynchronous schedule. A non-zero value in this field indicates an interrupt endpoint. The host controller uses the value of the three low-order bits of the FRINDEX register as an index into a bit position in this bit vector. If the µ*Frame S-mask* field has a one at the indexed bit position then this queue head is a candidate for transaction execution. If the *EPS* field indicates the endpoint is a high-speed endpoint, then the transaction executed is determined by the *PID_Code* field contained in the execution area. This field is also used to support split transaction types: Interrupt (IN/OUT). This condition is true when this field is non-zero and the *EPS* field indicates this is either a full- or low-speed device. A zero value in this field, in combination with existing in the periodic frame list has undefined results.

77.4.2.6.3 Transfer Overlay-Queue Head

The nine DWords in this area represent a transaction working space for the host controller. The general operational model is that the host controller can detect whether the overlay area contains a description of an active transfer. If it does not contain an active transfer, then it follows the Queue Head Horizontal Link Pointer to the next queue head. The host controller will never follow the Next Transfer Queue Element or Alternate Queue Element pointers unless it is actively attempting to advance the queue. For the duration of the transfer, the host controller keeps the incremental status of the transfer in the overlay area. When the transfer is complete, the results are written back to the original queue element.

The DWord3 of a Queue Head contains a pointer to the source qTD currently associated with the overlay. The host controller uses this pointer to write back the overlay area into the source qTD after the transfer is complete.

The following table describes the current qTD link pointer field descriptions.

Table 77-29. Current qTD Link Pointer

Bit	Description
31-5	Current Element Transaction Descriptor Link Pointer. This field contains the address Of the current transaction being processed in this queue and corresponds to memory address signals [31:5], respectively.
4-0	Reserved (R). These bits are ignored by the host controller when using the value as an address to write data. The actual value may vary depending on the usage.

The DWords 4-11 of a queue head are the transaction overlay area. This area has the same base structure as a Queue Element Transfer Descriptor. The queue head utilizes the reserved fields of the page pointers to implement tracking the state of split transactions.

This area is characterized as an overlay because when the queue is advanced to the next queue element, the source queue element is merged onto this area. This area serves an execution cache for the transfer.

The table below describes the Host-controller rules for bits in overlay.

Table 77-30. Host-Controller Rules for Bits in Overlay (DWords 5, 6, 8 and 9)

DWord	Bit	Description
5	4-1	Nak Counter (NakCnt)µRW. This field is a counter the host controller decrements whenever a transaction for the endpoint associated with this queue head results in a Nak or Nyet response. This counter is reloaded from *RL* before a transaction is executed during the first pass of the reclamation list (relative to an Asynchronous List Restart condition). It is also loaded from RL during an overlay.
6	31	Data Toggle. The *Data Toggle Control* controls whether the host controller preserves this bit when an overlay operation is performed.

Table continues on the next page...

Table 77-30. Host-Controller Rules for Bits in Overlay (DWords 5, 6, 8 and 9) (continued)

6	15	Interrupt On Complete (IOC). The IOC control bit is always inherited from the source qTD when the overlay operation is performed.
6	11-10	Error Counter (C_ERR). This two-bit field is copied from the qTD during the overlay and written back during queue advancement.
6	0	Ping State (P)/ERR. If the *EPS* field indicates a high-speed endpoint, then this field should be preserved during the overlay operation.
8	7-0	Split-transaction Complete-split Progress (C-prog-mask). This field is initialized to zero during any overlay. This field is used to track the progress of an interrupt split-transaction.
9	4-0	Split-transaction Frame Tag (Frame Tag). This field is initialized to zero during any overlay. This field is used to track the progress of an interrupt split-transaction.
9	11-5	S-bytes. Software must ensure that the *S-bytes* field in a *qTD* is zero before activating the *qTD*. This field is used to keep track of the number of bytes sent or received during an IN or OUT split transaction.

77.4.2.7 Periodic Frame Span Traversal Node (FSTN)

This data structure is to be used only for managing Full- and Low-speed transactions that span a Host-frame boundary.

See Host Controller Operational Model for FSTNs for full operational details. Software must not use an FSTN in the Asynchronous Schedule. An FSTN in the Asynchronous schedule results in undefined behavior. Software must not use the FSTN feature with a host controller whose USB_HCIVERSION register indicates a revision implementation below 0096h. FSTNs are not defined for implementations before 0.96 and their use yields undefined results.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Normal Path Link Pointer																											0		Typ		T	03-00H
	Back Path Link Pointer																											0		Typ		T	07-04H

Table 77-31. Frame Span Traversal Node Structure Layout

	Host Controller Read/Write		Host Controller Read Only.

77.4.2.7.1 FSTN Normal Path Pointer

The first DWord of an FSTN contains a link pointer to the next schedule object. This object can be of any valid periodic schedule data type.

The following table describes the FSTN normal path pointer fields.

Table 77-32. FSTN Normal Path Pointer Field Descriptions

Bit	Description
31-5	Normal Path Link Pointer (NPLP). This field contains the address of the next data object to be processed in the periodic list and corresponds to memory address signals [31:5], respectively.
4-3	Reserved
2-1	QH/(s)iTD/FSTN Select (Typ). This field indicates to the Host Controller whether the item referenced is a iTD/siTD, a QH or an FSTN. This allows the Host Controller to perform the proper type of processing on the item after it is fetched. Value encodings are: Value Meaning 00b iTD (isochronous transfer descriptor) 01b QH (queue head) 10b siTD (split transaction isochronous transfer descriptor) 11b FSTN (Frame Span Traversal Node)
0	Terminate (T). 1=Link Pointer field is not valid. 0=Link Pointer is valid.

77.4.2.7.2 FSTN Back Path Link Pointer

The second DWord of an FTSN node contains a link pointer to a queue head.

If the T-bit in this pointer is zero, then this FSTN is a Save-Place indicator. Its Typ field must be set by software to indicate the target data structure is a queue head. If the T-bit in this pointer is set to one, then this FSTN is the Restore indicator. When the T-bit is one, the host controller ignores the Typ field.

The following table describes the FSTN back path link pointer fields.

Table 77-33. FSTN Back Path Link Pointer Field Descriptions

Bit	Description
31-5	Back Path Link Pointer (BPLP). This field contains the address of a Queue Head. This field corresponds to memory address signals [31:5], respectively.
4-3	Reserved
2-1	Typ. Software must ensure this field is set to indicate the target data structure is a Queue Head. Any other value in this field yields undefined results.
0	Terminate (T). 1=Link Pointer field is not valid (that is the host controller must not use bits [31:5] as a valid memory address). This value also indicates that this FSTN is a Restore indicator. 0=Link Pointer is valid (that is the host controller may use bits [31:5] (in combination with the CTRLDSSEGMENT register if applicable) as a valid memory address). This value also indicates that this FSTN is a Save-Place indicator.

77.4.3 Host Operational Model

The general operational model is for the enhanced interface host controller hardware and enhanced interface host controller driver (generally referred to as system software).

Each significant operational feature of the EHCI host controller is discussed in a separate section. Each section presents the operational model requirements for the host controller hardware. Where appropriate, recommended system software operational models for features are also presented.

77.4.3.1 Host Controller Initialization

When the system boots, the host controller is enumerated, assigned a base address for the register space and BIOS sets the USB_FLADJ register to a system-specific value.

After initial power-on or HCReset (hardware or through HCReset bit in the USB_USBCMD register), all of the operational registers are at their default values. After a hardware reset, only the operational registers not contained in the Auxiliary power well are at their default values.

The following table describes the default values of operational registers.

.

Table 77-34. Default Values of Operational Register Space

Operational Register	Default Value (after Reset)
USB_USBCMD	00080000h (00080B00h, if *Asynchronous Schedule Park Capability is one*)
USB_USBSTS	00001000h
USB_USBINTR	00000000h
USB_FRINDEX	00000000h
USB_CTRLDSSEGMENT	00000000h
USB_PERIODICLISTBASE	Undefined
USB_ASYNCLISTADDR	Undefined
USB_CONFIGFLAG	00000000h
USB_PORTSC1	00002000h (w/*PPC* set to one); 00003000h (w/*PPC* set to zero)

To initialize the host controller, software should perform the following steps:

- Program the USB_CTRLDSSEGMENT register with 4-Gbyte segment where all of the interface data structures are allocated.

- Write the appropriate value to the USB_USBINTR register to enable the appropriate interrupts.
- Write the base address of the Periodic Frame List to the USB_PERIODICLIST BASE register. If no work items are in the periodic schedule, all elements of the Periodic Frame List should have their T-Bits set to one.
- Write the USB_USBCMD register to set the desired interrupt threshold, frame list size (if applicable) and turn the host controller ON through setting the Run/Stop bit.
- Write 1 to USB_CONFIGFLAG register to route all the ports to the EHCI controller.

At this point, the host controller is up and running and the port registers begin reporting device connects, and so on. System software can enumerate a port through the reset process (where the port is in the enabled state). At this point, the port is active with SOFs occurring down the enabled port enabled High-speed ports, but the schedules have not enabled. The EHCI Host controller do not transmit SOFs to enabled Full- or Low-speed ports. To communicate with devices through the asynchronous schedule, system software must write the USB_ASYNDLISTADDR register with the address of a control or bulk queue head. Software must then enable the asynchronous schedule by writing one to the Asynchronous Schedule Enable bit in the USB_USBCMD register. To communicate with devices through the periodic schedule, system software must enable the periodic schedule by writing one to the Periodic Schedule Enable bit in the USB_USBCMD register.

NOTE

The schedules can be turned on before the first port is reset (and enabled).

When the USB_USBCMD register is written, system software must ensure the appropriate bits are preserved, depending on the intended operation.

77.4.3.2 Port Routing and Control

A USB 2.0 Host controller is comprised of one high-speed host controller, which implements the EHCI programming interface and 0 to N USB 1.1 companion host controllers. Companion host controllers (cHCs) may be implementations of either Universal or Open host controller specifications.

This configuration is used to deliver the required full USB 2.0-defined port capability; for example, Low-, Full-, and High-speed capability for every port.

NOTE

The USB controller on i.MX parts do not require nor support companion controllers to support Full Speed/Low Speed device.Therefore, no port routing is present in the controller.

Please refer to Embedded Transaction Translator Function for detail!

The following figure illustrates a simple block diagram of the port routing logic and its relationship to the high-speed and companion host controllers within a USB 2.0 host controller.

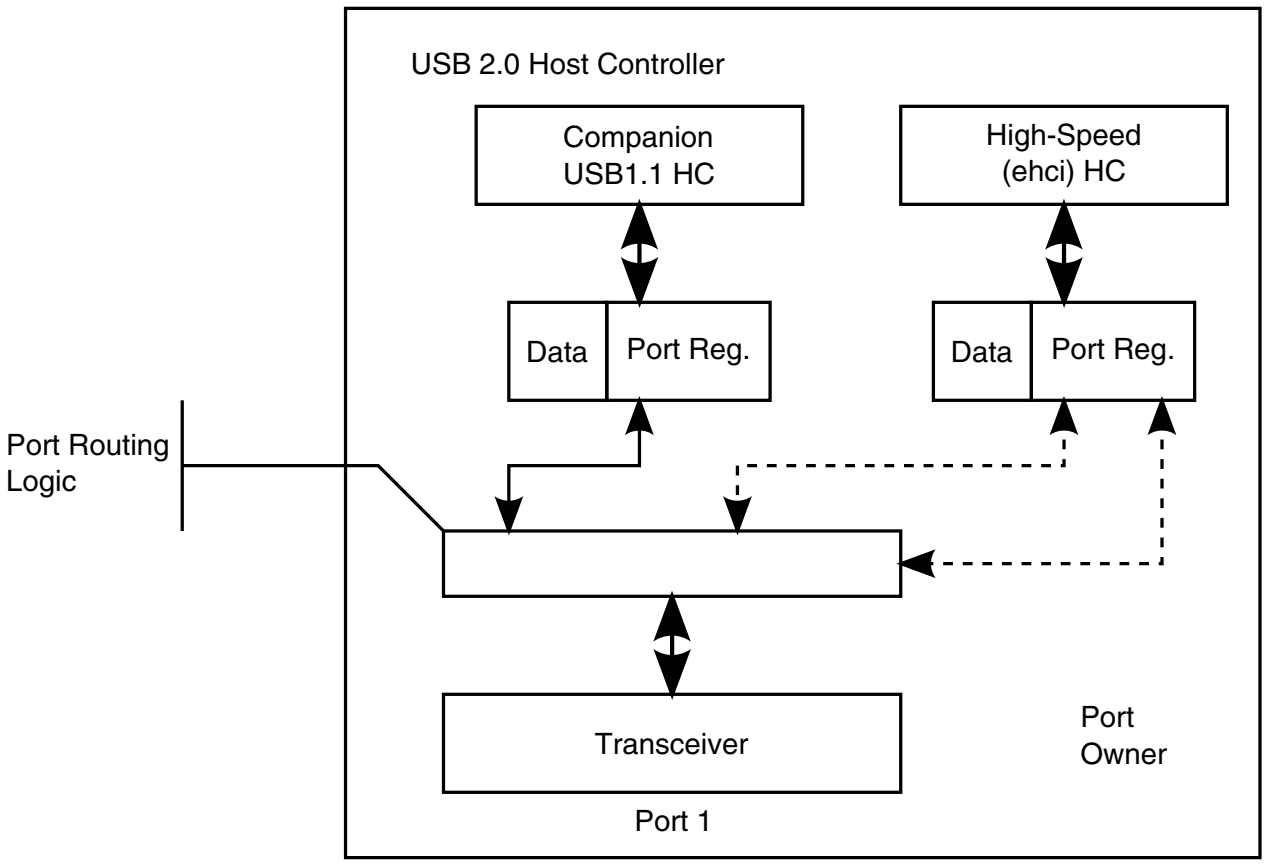


Figure 77-6. Example USB 2.0 Host Controller Port Routing Block Diagram

There exists one transceiver per physical port and each host controller block has its own port status and control registers. The EHCI controller has port status and control registers for every port. Each companion host controller has only the port control and status registers it is required to operate. Either the EHCI host controller or one companion host controller controls each transceiver. Routing logic lies between the transceiver, the port status and control registers.[1]

The port routing logic is controlled from signals originating in the EHCI host controller. The EHCI host controller has a global routing policy control field and per-port ownership control fields. The Configured Flag (CF) bit is the global routing policy control. At power-on or reset, the default routing policy is to the companion controllers (if they

1. The routing logic should not be implemented in the 480 MHz clock domain of the transceiver.

exist). If the system does not include a driver for the EHCI host controller and the host controller includes Companion Controllers, then the ports still work in Full- and Low-speed mode (assuming the system includes a driver for the companion controllers). In general, when the EHCI owns the ports, the companion host controllers' port registers do not see a connect indication from the transceiver. Similarly, when a companion host controller owns a port, the EHCI controller's port registers do not see a connect indication from the transceiver. The details on the rules for the port routing logic are described in the following sections. The USB 2.0 host controller must be implemented as a multi-function PCI device if the implementation includes companion controllers. The companion host controllers' function numbers must be less than the EHCI host controller function number. The EHCI host controller must be a larger function number with respect to the companion host controllers associated with this EHCI host controller. If a PCI device implementation contains only an EHCI controller (that is no companion controllers or other PCI functions), then the EHCI host controller must be function zero, in accordance with the PCI Specification. The N_CC field in the Structural Parameter register (HCSPARAMS) indicates whether the controller implementation includes companion host controllers. When N_CC has a non-zero value there exists companion host controllers. If N_CC has a value of zero, then the host controller implementation does not include companion host controllers. If the host controller root ports are exposed to attachment of full- or low-speed devices, the ports always fails the high-speed chirp during reset and the ports are not enabled. System software can notify the user of the illegal condition. This type of implementation requires a USB 2.0 hub be connected to a root port to provide full and low-speed device connectivity.

System software uses information in the host controller capability registers to determine how the ports are routed to the companion host controllers. See Host Controller Structural Parameters (USB__HCSPARAMS).

77.4.3.2.1 Port Routing Control through EHCI Configured (CF) Bit

Each port in the USB 2.0 host controller are routed either to a single companion host controller or to the EHCI host controller.

The port routing logic is controlled by two mechanisms in the EHCI HC: a host controller global flag and per-port control. The Configured Flag (CF) bit, is used to globally set the policy of the routing logic. Each port register has a Port Owner control bit which allows the EHCI Driver to explicitly control the routing of individual ports. Whenever the CF bit transitions from zero to one (this transition is only available under program control) the port routing unconditionally routes all of the port registers to the EHCI HC (all Port Owner bits go to zero). While the CF-bit is one, the EHCI Driver controls individual ports' routing through the Port Owner control bit. Likewise, whenever the CF bit transitions from one to zero (as a result of Aux power application, HCRESET, or

software writing zero to CF-bit), the port routing unconditionally routes all of the port registers to the appropriate companion HC. The default value for the EHCI HC's CF bit (after Aux power application or HCRESET) is zero.

The *view* of the port depends on the current owner. A Universal or Open companion host controller will see port register bits consistent with the appropriate specification. Port bit definitions that are required for EHCI host controllers are not visible to companion host controllers.

The following table summarizes the default routing for all the ports, based on the value of the EHCI HC's CF bit.

Table 77-35. Default Port Routing Depending on EHCI HC CF Bit

HS CF Bit	Default Port Ownership	Explanation
0B	Companion HCs	The companion host controllers own the ports and only Full- and Low-speed devices are supported in the system. The exact port assignments are implementation dependent. The ports behave only as Full- and Low-speed ports in this configuration
1B	EHCI HC	The EHCI host controller has default ownership over all of the ports. The routing logic inhibits device connect events from reaching the companion HCs' port status and control registers when the port owner is the EHCI HC.The EHCI HC has access to the additional port status and control bits defined in this specification (see Port Status & Control (USB__PORTSC1)). The EHCI HC can temporarily release control of the port to a companion HC by setting the *PortOwner* bit in the PORTSC1 register to one.

77.4.3.2.2 Port Routing Control through PortOwner and Disconnect Event

Manipulating the port routing through the CF-bit is an extreme process and not intended to be used during normal operation.

The normal mode of port ownership transferal is on the granularity of individual ports using the Port Owner bit in the EHCI HC's USB_PORTSC1 register (for hand-offs from EHCI to companion host controllers). Individual port ownership is returned to the EHCI controller when the port registers a device disconnect. When the disconnect is detected, the port routing logic immediately returns the port ownership to the EHCI controller. The companion host controller port register detects the device disconnect and operates normally.

Under normal operating conditions (assuming all HC drivers loaded and operational and the EHCI *CF-bit* is set to one), the typical port enumeration sequence proceeds as illustrated below:

- Initial condition is that EHCI is port owner. A device is connected causing the port to detect a connect, set the port connect change bit and issue a port-change interrupt (if enabled).
- EHCI Driver identifies the port with the new connect change bit asserted and sends a change report to the hub driver. Hub driver issues a GetPortStatus() request and identifies the connect change. It then issues a request to clear the connect change, followed by a request to reset and enable the port.
- When the EHCI Driver receives the request to reset and enable the port, it first checks the value reported by the LineStatus bits in the USB_PORTSC1 register. If they indicate the attached device is a full-speed device (for example, D+ is asserted), then the EHCI Driver sets the PortReset control bit to one (and sets the PortEnable bit to zero) which begins the reset-process. Software times the duration of the reset, then terminates reset signaling by writing zero to the port reset bit. The reset process is actually complete when software reads zero in the PortReset bit. The EHCI Driver checks the PortOwner bit in the USB_PORTSC1 register. If set to one, the connected device is a high-speed device and EHCI Driver (root hub emulator) issues a change report to the hub driver and the hub driver continues to enumerate the attached device.
- At the time the EHCI Driver receives the port reset and enable request the LineStatus bits might indicate a low-speed device. Additionally, when the port reset process is complete, the PortEnable field may indicate that a full-speed device is attached. In either case the EHCI driver sets the PortOwner bit in the USB_PORTSC1 register to one to release port ownership to a companion host controller.
- When the EHCI Driver sets PortOwner bit to one, the port routing logic makes the connection state of the transceiver available to the companion host controller port register and removes the connection state from the EHCI HC port. The EHCI USB_PORTSC1 register observes and reports a disconnect event through the disconnect change bit. The EHCI Driver detects the connection status change (either by polling or by port change interrupt) and then sends a change report to the hub driver. When the hub driver requests that port-state, the EHCI Driver responds with a reset complete change set to one, a connect change set to one and a connect status set to zero. This information is derived directly from the EHCI port register. This allows the hub driver to assume the device was disconnected during reset. It acknowledges the change bits and wait for the next change event. While the EHCI controller does not own the port, it simply remains in a state where the port reports no device connected. The device-connect evaluation circuitry of the companion HC activates and detects the device, the companion Driver detects the connection and enumerates the port.

When a port is routed to a companion HC, it remains under the control of the companion HC until the device is disconnected from the root port (ignoring for now the scenario where EHCI's CF-bit transitions from1b to 0b). When a disconnect occurs, the disconnect

event is detected by both the companion HC port control and the EHCI port ownership control. On the event, the port ownership is returned immediately to the EHCI controller. The companion HC stack detects the disconnect and acknowledges as it would in an ordinary standalone implementation. Subsequent connects is detected by the EHCI port register and the process repeats.

77.4.3.2.3 Example Port Routing State Machine

The following figure illustrates an example of how the port ownership should be managed. The following sections describe the entry conditions to each state.

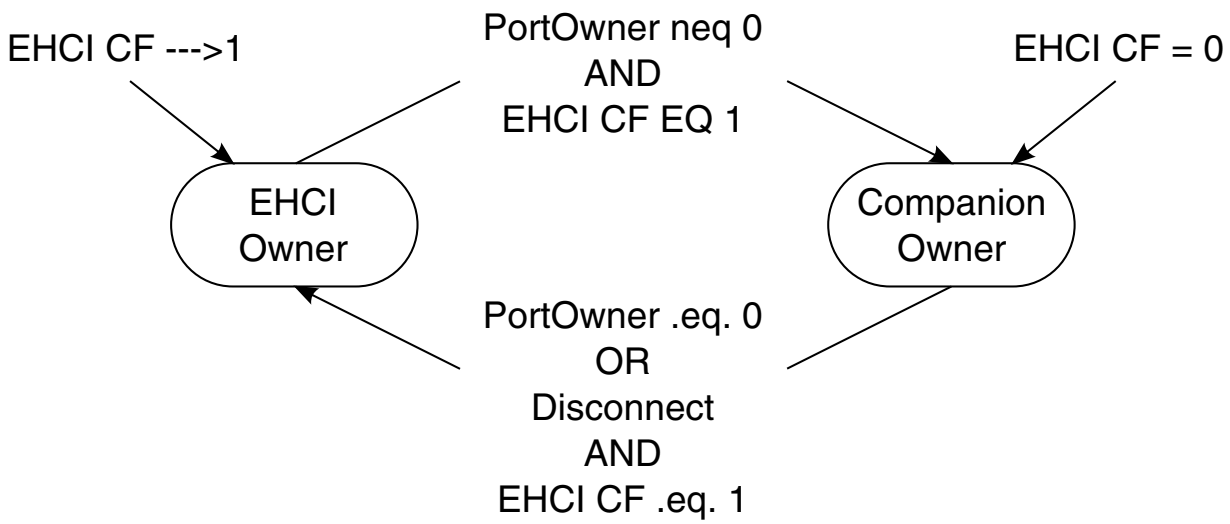


Figure 77-7. Port Owner Handoff State Machine

77.4.3.2.3.1 EHCI HC Owner

Entry to this state occurs when one of the following events occur:

- When the EHCI HC's Configure Flag (CF) bit in the USB_CONFIGFLAG register transitions from zero to one. This signals the fact that the system has a host controller driver for the EHCI HC and that all ports in the USB 2.0 host controller must default route to the EHCI controller.
- When the port is owned by a companion HC and the device is disconnected from the port. The EHCI port routing control logic is notified of the disconnect, and returns port routing to the EHCI controller. The connection state of the companion HC goes immediately to the disconnected state (with appropriate side effect to connect change, enable and enable change). The companion HC driver acknowledges the disconnect by setting the connect status change bit to zero. This allows the

companion HC's driver to interact with the port completely through the disconnect process.

- When system software writes zero to the PortOwner bit in the USB_PORTSC1 register. This allows software to take ownership of a port from a companion host controller. When this occurs, the routing logic to the companion HC effectively signals a disconnect to the companion HC's port status and control register.

77.4.3.2.3.2 Companion HC Owner

Entry to this state occurs whenever one of the following events occur:

- When the PortOwner field transitions from zero to one.
- When the HS-mode HC's Configure Flag (CF) is equal to zero.

On entry to this state, the routing logic allows the companion HC port register to detect a device connect. Normal port enumeration proceeds.

77.4.3.2.4 Port Power

The Port Power Control (PPC) bit in the USB_HCSPARAMS register indicates whether the USB 2.0 host controller has port power control (see Host Controller Structural Parameters (USB__HCSPARAMS)).

When this bit is zero, then the host controller does not support software control of port power switches. When in this configuration, the port power is always available and the companion host controllers must implement functionality consistent with port power always on. When the *PPC* bit is one, then the host controller implementation includes port power switches. Each available switch has an output enable, which is referred to in this discussion as PortPowerOutputEnable (PPE). PPE is controlled based on the state of the combination bits PPC bit, EHCI Configured (CF)-bit and individual Port Power (PP) bits.

The following table describes the summary behavioral model.

Table 77-36. Port Power Enable Control Rules

CF	CHC[1] (PP)	EHC[2] (PP)	Owner	PPE[3]	Description
0	0	X	CHC	0	When the EHCI controller is not configured, the port is owned by the companion host controller. When the companion HC's port power select is off, then the port power is off.
0	1	X	CHC	1	Similar to previous entry. When the companion HC's port power select is on, then the port power is on.

Table continues on the next page...

Table 77-36. Port Power Enable Control Rules (continued)

1	0	0	CHC	0	Port owner has port power turned off, the power to port is off.
1	0	0	EHC	0	Port owner has port power turned off, the power to port is off.
1	0	1	EHC	1	Port owner has port power on, so power to port is on.
1	0	1	CHC	1	If either HC has port power turned on, the power to the port is on.
1	1	0	EHC	1	If either HC has port power turned on, the power to the port is on.
1	1	0	CHC	1	Port owner has port power on, so power to port is on.
1	1	1	CHC	1	Port owner has port power on, so power to port is on.
1	1	1	EHC	1	Port owner has port power on, so power to port is on.

1. CHC (Companion Host Controller).
2. EHC (EHCI Host Controller).
3. PPE (Port Power Enable). This bit actually turns on the port power switch (if one exists).

77.4.3.2.5 Port Reporting Over-Current

Host controllers are by definition power providers on USB. Whether the ports are considered high- or low-powered is a platform implementation issue. Each EHCI USB_PORTSC1 register has an over-current status and over-current change bit.

The functionality of these bits are specified in the USB Specification Revision 2.0.

The over current detection and limiting logic usually resides outside the host controller logic. This logic may be associated with one or more ports. When this logic detects an over-current condition it is made available to both the companion and EHCI ports. The effect of an over-current status on a companion host controller port is beyond the scope of this document.

The over-current condition effects the following bits in the USB_PORTSC1 register on the EHCI port:

- Over-current Active bits are set to one. When the over-current condition goes away, the Over-current Active bit transitions from one to zero.
- Over-current Change bits are set to one. On every transition of the Over-current Active bit the host controller sets the Over-current Change bit to one. Software sets the Over-current Change bit to zero by writing one to this bit.

- Port Enabled/Disabled bit is set to zero. When this change bit gets set to one, then the Port Change Detect bit in the USB_USBSTS register is set to one.
- Port Power (PP) bits may optionally be set to zero. There is no requirement in USB that a power provider shut off power in an over current condition. It is sufficient to limit the current and leave power applied. When the Over-current Change bit transitions from zero to one, the host controller also sets the Port Change Detect bit in the USB_USBSTS register to one. In addition, if the Port Change Interrupt Enable bit in the USB_USBINTR register is one, then the host controller issues an interrupt to the system. Refer to Table 77-37 for summary behavior for over-current detection when the host controller is halted (suspended from a device component point of view).

77.4.3.3 Suspend/Resume-Host Operational Model

The EHCI host controller provides an equivalent suspend and resume model as that defined for individual ports in a USB 2.0 Hub.

Control mechanisms are provided to allow system software to suspend and resume individual ports. The mechanisms allow the individual ports to be resumed completely through software initiation. Other control mechanisms are provided to parameterize the host controller's response (or sensitivity) to external resume events. In this discussion, host-initiated, or software initiated resumes are called Resume Events/Actions. Bus-initiated resume events are called wake-up events. The classes of wake-up events are:

- Remote-wake-up enabled device asserts resume signaling. In similar kind to USB 2.0 Hubs, EHCI controllers must always respond to explicit device resume signaling and wake-up the system (if necessary).
- Port connect and disconnect and over-current events. Sensitivity to these events can be turned on or off by using the per-port control bits in the USB_PORTSC1 registers.

Selective suspend is a feature supported by every USB_PORTSC1 register. It is used to place specific ports into a suspend mode. This feature is used as a functional component for implementing the appropriate power management policy implemented in a particular operating system. When system software intends to suspend the entire bus, it should selectively suspend all enabled ports, then shut off the host controller by setting the Run/Stop bit in the USB_USBCMD register to zero. The EHCI sub-block can then be placed into a lower device state through the PCI power management interface (see Appendix A, Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 0.95, November 2000, Intel Corporation. http://www.intel.com).

When a wake event occurs, the system resumes operation and system software eventually set the Run/Stop bit to one and resume the suspended ports. Software must not set the Run/Stop bit to one until it is confirmed that the clock to the host controller is stable. This is usually confirmed in a system implementation in that all of the clocks in the system are stable before the ARM platform is restarted. So, by definition, if software is running, clocks in the system are stable and the Run/Stop bit in the USB_USBCMD register can be set to one. Minimum system software delays are also defined in the PCI Power Management Specification. Refer to PCI Power Management Specification for more information.

77.4.3.3.1 Port Suspend/Resume

System software places individual ports into suspend mode by writing one into the appropriate USB_PORTSC1 Suspend bit. Software must only set the Suspend bit when the port is in the enabled state (Port Enabled bit is one) and the EHCI is the port owner (PortOwner bit is zero).

The host controller may evaluate the Suspend bit immediately or wait until a micro-frame or frame boundary occurs. If evaluated immediately, the port is not suspended until the current transaction (if one is executing) completes. Therefore, there may be several micro-frames of activity on the port until the host controller evaluates the Suspend bit. The host controller must evaluate the Suspend bit at least every frame boundary.

System software can initiate a resume on a selectively suspended port by writing one to the Force Port Resume bit. Software should not attempt to resume a port unless the port reports that it is in the suspended state (see Port Status & Control (USB__PORTSC1)). If system software sets Force Port Resume bit to one when the port is not in the suspended state, the resulting behavior is undefined. In order to assure proper USB device operation, software must wait for at least 10 ms after a port indicates that it is suspended (Suspend bit is one) before initiating a port resume through the Force Port Resume bit. When Force Port Resume bit is one, the host controller sends resume signaling down the port. System software times the duration of the resume (nominally 20 ms) then sets the Force Port Resume bit to zero. When the host controller receives the write to transition Force Port Resume to zero, it completes the resume sequence as defined in the USB specification, and sets both the Force Port Resume and Suspend bits to zero. Software-initiated port resumes do not affect the Port Change Detect bit in the USB_USBSTS register nor do they cause an interrupt if the Port Change Interrupt Enable bit in the USB_USBINTR register is one. An external USB event may also initiate a resume. The wake events are defined above. When a wake event occurs on a suspended port, the resume signaling is detected by the port and the resume is reflected downstream within 100 µsec. The port's

Force Port Resume bit is set to one and the Port Change Detect bit in the USB_USBSTS register is set to one. If the Port Change Interrupt Enable bit in the USB_USBINTR register is one the host controller issues a hardware interrupt.

System software observes the resume event on the port, delays a port resume time (nominally 20 ms), then terminates the resume sequence by writing zero to the Force Port Resume bit in the port. The host controller receives the write of zero to Force Port Resume, terminates the resume sequence and sets Force Port Resume and Suspend port bits to zero. Software can determine that the port is enabled (not suspended) by sampling the USB_PORTSC1 register and observing that the Suspend and Force Port Resume bits are zero. Software must ensure that the host controller is running (that is HCHalted bit in the USB_USBSTS register is zero), before terminating a resume by writing zero to a port's Force Port Resume bit. If HCHalted is one when Force Port Resume is set to zero, then SOFs do not occur down the enabled port and the device returns to suspend mode in a maximum of 10 msec.

The table below summarizes the wake-up events. Whenever a resume event is detected, the Port Change Detect bit in the USB_USBSTS register is set to one. If the Port Change Interrupt Enable bit is one in the USB_USBINTR register, the host controller generates an interrupt on the resume event. Software acknowledges the resume event interrupt by clearing the Port Change Detect status bit in the USB_USBSTS register.

Table 77-37. Behavior During Wake-up Events

Port Status and Signaling Type	Signaled Port Response	Device State	
		D0	Not D0
Port disabled, resume K-State received	No Effect	N/A	N/A
Port suspended, resume K-State received	Resume reflected downstream on signaled port. Force Port Resume status bit in USB_PORTSC1 register is set to one. Port Change Detect bit in USB_USBSTS register set to one.	[1], [2]	[2]
Port is enabled, disabled or suspended, and the port's WKDSCNNT_E bit is one. A disconnect is detected.	Depending in the initial port state, theUSB_ PORTSC1 Connected Enable status bits are set to zero, and the Connect Change status bit is set to one. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [2]	[2]
Port is enabled, disabled or suspended, and the port's WKDSCNNT_E bit is zero. A disconnect is detected.	Depending on the initial port state, the USB_PORTSC1 Connect and Enable status bits are set to zero, and the Connect Change status bit is set to one. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [3]	[3]
Port is not connected and the port's WKCNNT_E bit is one. A connect is detected.	USB_PORTSC1 Connect Status and Connect Status Change bits are set to one. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [2]	[2]
Port is not connected and the port's WKCNNT_E bit is zero. A connect is detected.	USB_PORTSC1 Connect Status and Connect Status Change bits are set to one. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [3]	[3]

Table continues on the next page...

Table 77-37. Behavior During Wake-up Events (continued)

Port is connected and the port's WKOC_E bit is one. An over-current condition occurs.	USB_PORTSC1 Over-current Active, Over-current Change bits are set to one. If Port Enable/Disable bit is one, it is set to zero. Port Change Detect bit in the USB_USBSTS register is set to one	[1], [2]	[2]
Port is connected and the port's WKOC_E bit is zero. An over-current condition occurs.	USB_PORTSC1 Over-current Active, Over-current Change bits are set to one. If Port Enable/Disable bit is one, it is set to zero. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [3]	[3]

[1] Hardware interrupt issued if Port Change Interrupt Enable bit in the USB_USBINTR register is one.

[2] PME# asserted if enabled (Note: PME Status must always be set to one).

[3] PME# not asserted.

77.4.3.4 Schedule Traversal Rules

The host controller executes transactions for devices using a simple, shared-memory schedule. The schedule is comprised of a few data structures, organized into two distinct lists. The data structures are designed to provide the maximum flexibility required by USB, minimize memory traffic and hardware / software complexity.

System software maintains two schedules for the host controller: a periodic schedule and an asynchronous schedule. The root of the periodic schedule is the USB_PERIODICLISTBASE register (see Frame List Base Address (USB__PERIODICLISTBASE))/Device Address (USB_UOG_DEVICEADDR). The USB_PERIODICLISTBASE register is the physical memory base address of the periodic frame list. The periodic frame list is an array of physical memory pointers. The objects referenced from the frame list must be valid schedule data structures as defined in Host Data Structures. In each micro-frame, if the periodic schedule is enabled (see Periodic Scheduling Threshold) then the host controller must execute from the periodic schedule before executing from the asynchronous schedule. It only executes from the asynchronous schedule after it encounters the end of the periodic schedule. The host controller traverses the periodic schedule by constructing an array offset reference from the USB_PERIODICLISTBASE and the USB_FRINDEX registers (see the following figure). It fetches the element and begins traversing the graph of linked schedule data structures.

The end of the periodic schedule is identified by a next link pointer of a schedule data structure having its T-bit set to one. When the host controller encounters a T-Bit set to one during a horizontal traversal of the periodic list, it interprets this as an End-Of-Periodic-List mark. This causes the host controller to cease working on the periodic

schedule and transitions immediately to traversing the asynchronous schedule. After the transition, the host controller executes from the asynchronous schedule until the end of the micro-frame.

The following figure illustrates the derivation of pointer into frame list array.

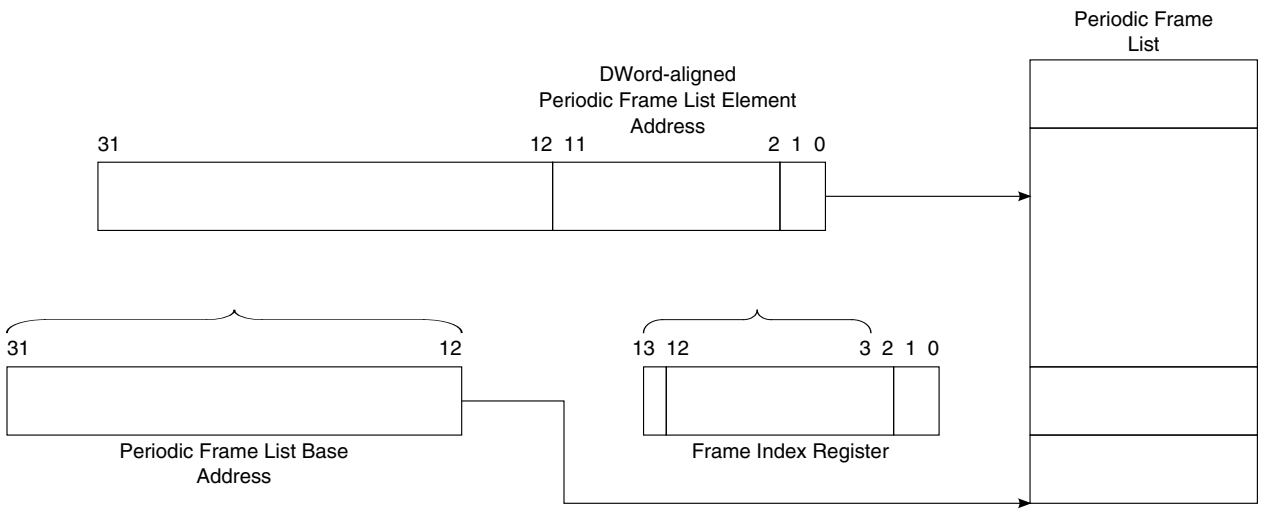


Figure 77-8. Derivation of Pointer into Frame List Array

When the host controller determines that it is the time to execute from the asynchronous list, it uses the operational register USB_ASYNCLISTADDR to access the asynchronous schedule, see the figure below.

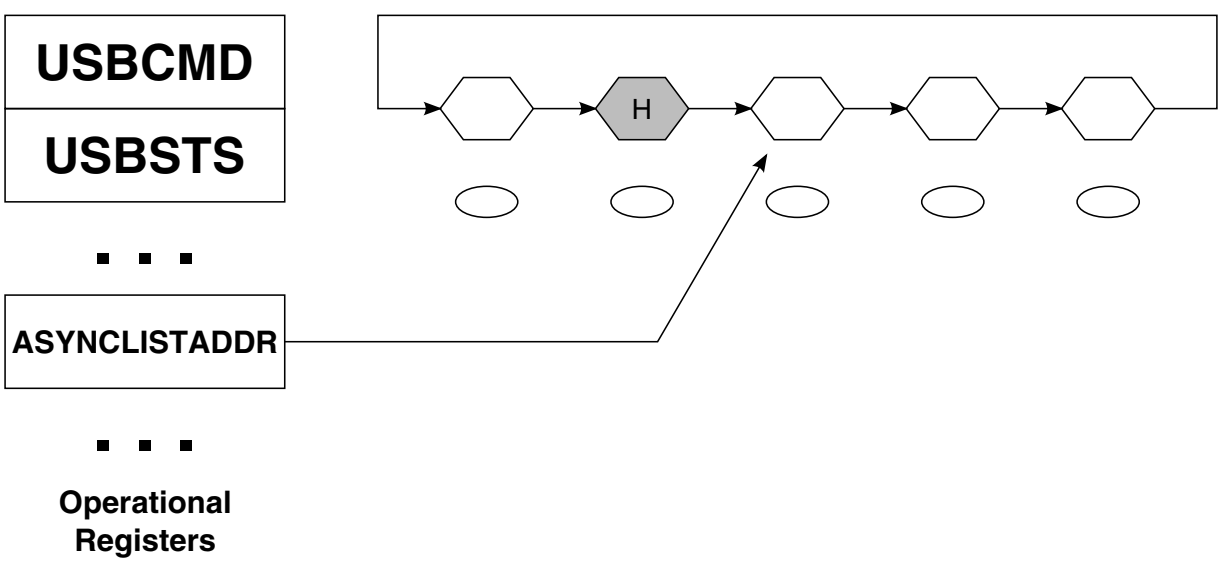


Figure 77-9. General Format of Asynchronous Schedule List

The USB_ASYNCLISTADDR register contains a physical memory pointer to the next queue head. When the host controller makes a transition to executing the asynchronous schedule, it begins by reading the queue head referenced by the USB_ASYNCLISTADDR register. Software must set queue head horizontal pointer T-bits to zero for queue heads in the asynchronous schedule. See Asynchronous Schedule for complete operational details.

77.4.3.4.1 Example - Preserving Micro-Frame Integrity

One of the requirements of a USB host controller is to maintain Frame Integrity. This means that the HC must preserve the micro-frame boundaries.

For example, SOF packets must be generated on time (within the specified allowable jitter), and High-speed EOF1,2 thresholds must be enforced. The end of micro-frame timing points EOF1 and EOF2 are clearly defined in the USB Specification Revision 2.0. One implication of this responsibility is that the HC must ensure that it does not start transactions that do not complete before the end of the micro-frame. More precisely, no transactions should be started by the host controller, which do not complete in their entirety before the EOF1 point. In order to enforce this rule, the host controller must check each transaction before it starts to ensure that it completes before the end of the micro-frame.

So, what exactly needs to be involved in this check? Fundamentally, the transaction data payload, plus bit stuffing, plus transaction overhead must be taken into consideration. It is possible to be extremely accurate on how much time the next transaction takes. Take OUTs for an example. The host controller must fetch all of the OUT data from memory in order to send it onto the USB bus. A host controller implementation could pre-fetch all of the OUT data, and pre-compute the actual number of bits in the token and data packets. In addition, the system knows the depth of the target endpoint, so it could closely estimate turnaround time for handshake. In addition, the host controller knows the size of a handshake packet. Pre-computing effects of bit stuffing and summing up the other overhead numbers can allow the host controller to know exactly whether there is enough bus time, before EOF1 to complete the OUT transaction. To accomplish this particular approach takes an inordinate amount of time and hardware complexity.

The alternative is to make a reasonable guess whether the next transaction can be started. An example approximation algorithm is described below. This example algorithm relies on the EHCI policy that periodic transactions are scheduled first in the micro-frame. It is a reasonable assumption that software never over-commits the micro-frame to periodic transactions greater than the specification allowable 80%. In the available remaining 20% bandwidth, the host controller has some ability (in this example) to decide whether or not to execute a transaction. The result of this algorithm is that sometimes, under some

circumstances a transaction is not executed that could have been executed. However, under all circumstances, a transaction is never started unless there is enough time in the frame to complete the transaction.

77.4.3.4.1.1 Transaction Fit - A Best-Fit Approximation Algorithm

A curve is calculated which represents the latest start time for every packet size, at which software schedules the start of a periodic transaction.

This curve is the 80% bandwidth curve. Another curve is calculated which is the absolute, latest permitted start time for every packet size. This curve represents the absolute latest time, that a transaction of each packet size can be started and completed, in the micro-frame. A plot of these two curves are illustrated in the following figure. The plot Y-axis represents the number of byte-times left in a frame.

The space between the 80% and the Last Start plots is bandwidth reclamation area. In this algorithm the host controller may skip transactions during this time if it is prudent.

The Best-Fit Approximation method plots a function ($f(x)$) between the 80% and Last Start curves. The function $f(x)$ adds a constant to every transaction's maximum packet size and the result compared with the number of bytes left in the frame. The constant represents an approximation of the effects of bit stuffing and protocol overhead. The host controller starts transactions whose results land above the function curve. The host controller will not start transactions whose results land below the function curve.

The following figure illustrates the Best-Fit Approximation.

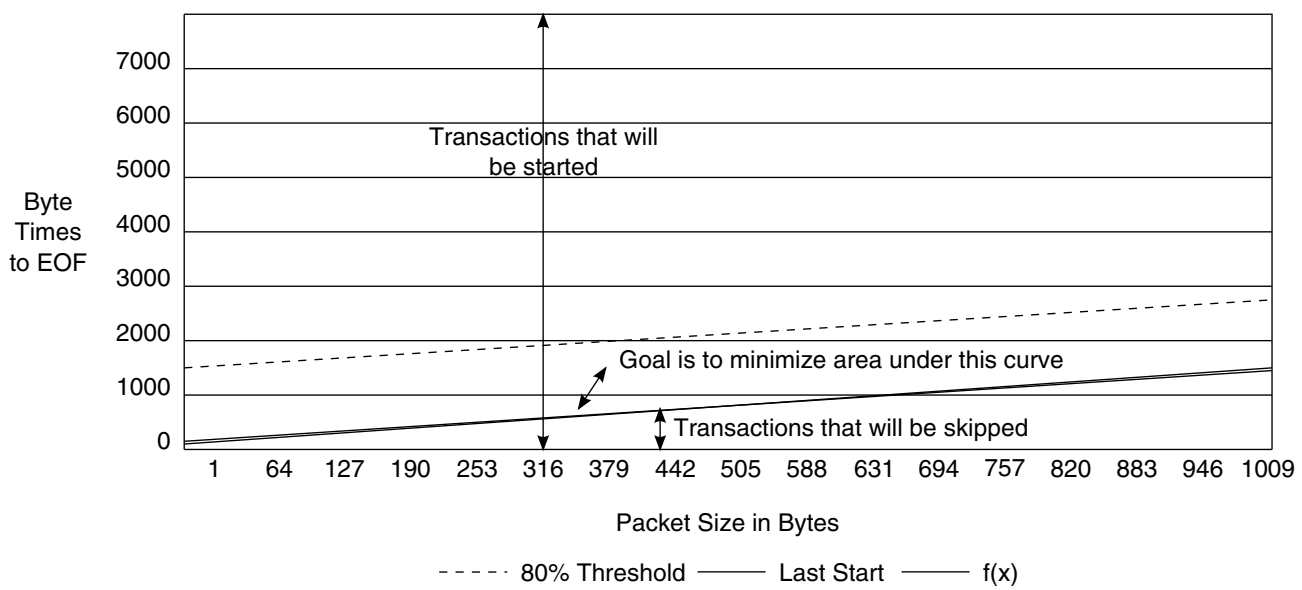


Figure 77-10. Best Fit Approximation

The LastStart line was calculated in this example to assume the absolute worst-case bus overhead per transaction. The particular transaction used is a start-split, zero-length OUT transaction with a handshake. Summaries of the component parts are listed in the table below. The component times were derived from the protocol timings defined in the USB Specification Revision 2.0.

Table 77-38. Example Worse-case Transaction Timing Components

Component	Bit time	Byte Time	Explanation
Split Token	76	9.5	Split token as defined in USB core specification. Includes sync, token, eop, and so on.
Host 2 Host IPG	88	11	Number of bit times required between consecutive host packets.
Token	67	8.375	Token as defined in USB core specification. Includes sync, token, eop, and so on.
Host 2 Host IPG	88	11	Token as defined in USB core specification. Includessync, token, eop, and so on.
Data Packet (0 data bytes)	66.7	8.34	Zero-length data packet. Includes sync, PID, crc16, eop, and so on.
Turnaround time	721	90.125	Time for packet initiator (Host) to see the beginning of a response to a transmitted packet.
Handshake packet	48	6	Handshake packet as defined in USB core specification. Includes sync, PID, eop, and so on.
		144	Total

The exact details of the function (f(x)) are up to the particular implementation. However, it should be obvious that the goal is to minimize the area under the curve between the approximation function and the Last Start curve, without dipping below the LastStart line, while at the same time keeping the check as simple as possible for hardware implementation. The f(x) in Figure 77-10 was constructed using the following pseudo-code test on each transaction size data point. This algorithm assumes that the host controller keeps track of the remaining bits in the frame.

```
Alorithm CheckTransactionWillFit (MaximumPacketSize, HC_BytesLeftInFrame)
Begin
Local Temp = MaximumPacketSize + 192
Local rvalue = TRUE
If MaximumPacketSize >= 128 then
     Temp += 128
End If
If Temp > HC_BytesLeftInFrame then
     Rvalue = FALSE
End If
Return rvalue
End
```

This algorithm takes two inputs, the current maximum packet size of the transaction and the hardware counter of the number of bytes left in the current micro-frame. It unconditionally adds a simple constant of 192 to the maximum packet size to account for a first-order effect of transaction overhead and bit stuffing. If the transaction size is

greater than or equal to 128 bytes, then an additional constant of 128 is added to the running sum to account for the additional worst-case bit stuffing of payloads larger than 128. An inflection point was inserted at 128 because the $f(x)$ plot was getting close to the LastStart line.

77.4.3.5 Periodic Schedule Frame Boundaries vs Bus Frame Boundaries

The USB Specification Revision 2.0 requires that the frame boundaries (SOF frame number changes) of the high-speed bus and the full- and low-speed bus(s) below USB 2.0 Hubs be strictly aligned.

Super-imposed on this requirement is that USB 2.0 Hubs manage full- and low-speed transactions through a micro-frame pipeline (see start- (SS) and complete- (CS) splits illustrated in the following figure). A simple, direct projection of the frame boundary model into the host controller interface schedule architecture creates tension (complexity for both hardware and software) between the frame boundaries and the scheduling mechanisms required to service the full- and low-speed transaction translator periodic pipelines.

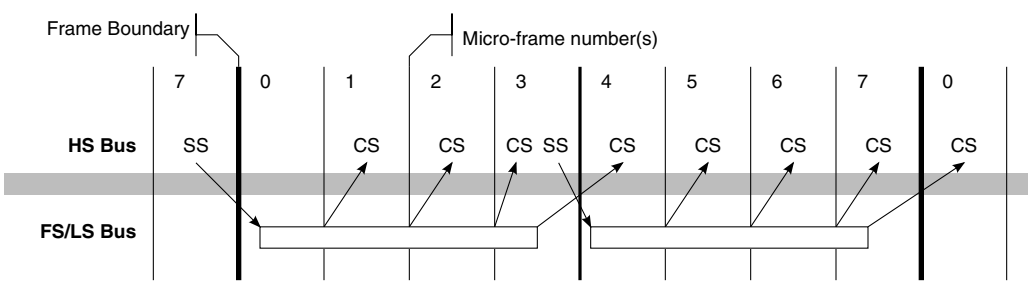


Figure 77-11. Frame Boundary Relationship between HS bus and FS/LS Bus

The simple projection, as the above figure illustrates, introduces frame-boundary wrap conditions for scheduling on both the beginning and end of a frame. In order to reduce the complexity for hardware and software, the host controller is required to implement one micro-frame phase shift for its view of frame boundaries. The phase shift eliminates the beginning of frame and frame-wrap scheduling boundary conditions.

The implementation of this phase shift requires that the host controller use one register value for accessing the periodic frame list and another value for the frame number value included in the SOF token. These two values are separate, but tightly coupled. The periodic frame list is accessed through the Frame List Index Register (USB_FRINDEX) documented in USB Frame Index (USB__FRINDEX) and initially illustrated in Schedule Traversal Rules . Bits FRINDEX[2:0], represent the micro-frame number. The SOF value is coupled to the value of FRINDEX[13:3]. Both FRINDEX[13:3] and

increment based on FRINDEX[2:0]. It is required that the SOF value be delayed from the FRINDEX value by one micro-frame. The one micro-frame delay yields host controller periodic schedule and bus frame boundary relationship as illustrated in the following figure. This adjustment allows software to trivially schedule the periodic start and complete-split transactions for full-and low-speed periodic endpoints, using the natural alignment of the periodic schedule interface. The reasons for selecting this phase-shift are beyond the scope of this specification.

The following figure illustrates how periodic schedule data structures relate to schedule frame boundaries and bus frame boundaries. To aid the presentation, two terms are defined: The host controller's view of the 1 msec boundaries is called H-Frames. The high-speed bus's view of the 1 msec boundaries is called B-Frames.

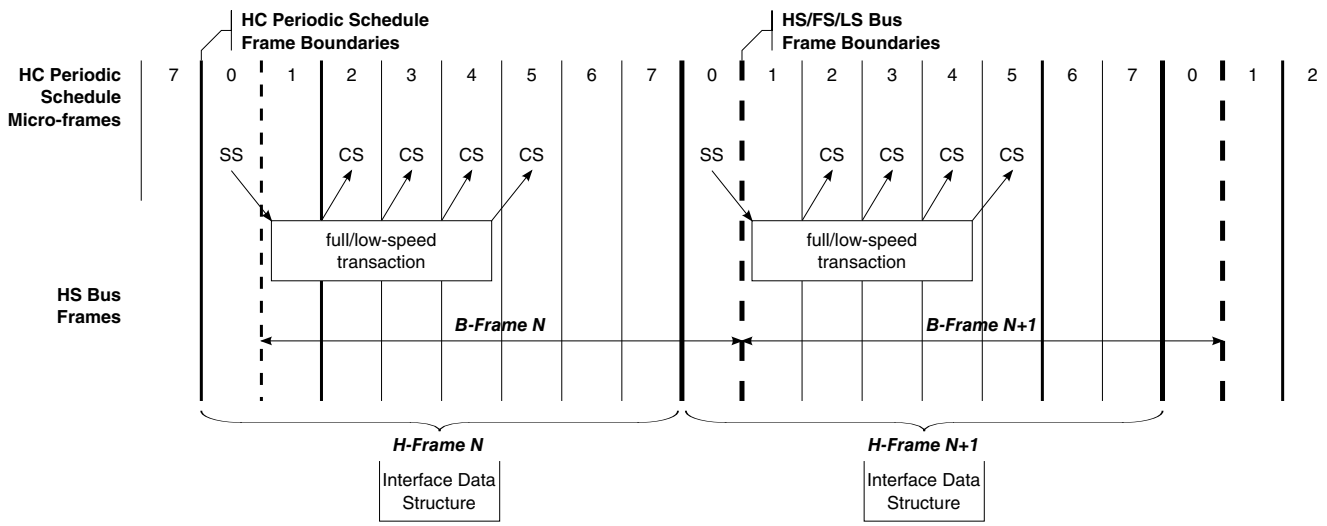


Figure 77-12. Relationship of Periodic Schedule Frame Boundaries to Bus Frame Boundaries

H-Frame boundaries for the host controller correspond to increments of FRINDEX[13:3]. Micro-frame numbers for the H-Frame are tracked by FRINDEX[2:0]. B-Frame boundaries are visible on the high-speed bus through changes in the SOF token's frame number. Micro-frame numbers on the high-speed bus are only derived from the SOF token's frame number (that is the high-speed bus sees eight SOFs with the same frame number value). H-Frames and B-Frames have the fixed relationship (that is B-Frames lag H-Frames by one micro-frame time) illustrated in the figure above. The host controller's periodic schedule is naturally aligned to H-Frames. Software schedules transactions for full- and low-speed periodic endpoints relative the H-Frames. The result is these transactions execute on the high-speed bus at exactly the right time for the USB 2.0 Hub periodic pipeline. As described in USB Frame Index (USB__FRINDEX), the SOF Value can be implemented as a shadow register (in this example, called SOFV), which lags the FRINDEX register bits [13:3] by one micro-frame count. This lag behavior can be

accomplished by incrementing FRINDEX[13:3] based on carry-out on the 7 to 0 increment of FRINDEX[2:0] and incrementing SOFV based on the transition of 0 to 1 of FRINDEX[2:0].

Software is allowed to write to FRINDEX. USB Frame Index (USB__FRINDEX) provides the requirements that software should adhere when writing a new value in FRINDEX.

The table below illustrates the required relationship between the value of FRINDEX and the value of SOFV.

Table 77-39. Operation of FRINDEX and SOFV (SOF Value Register)

Current			Next		
FRINDEX[F]	SOFV	FRINDEX[mF]	FRINDEX[F]	SOFV	FRINDEX[mF]
N	N	111b	N+1	N	000b
N+1	N	000b	N+1	N+1	001b
N+1	N+1	001b	N+1	N+1	010b
N+1	N+1	010b	N+1	N+1	011b
N+1	N+1	011b	N+1	N+1	100b
N+1	N+1	100b	N+1	N+1	101b
N+1	N+1	101b	N+1	N+1	110b
N+1	N+1	110b	N+1	N+1	111b

NOTE

Where [F] = [13:3]; [μF] = [2:0]

77.4.3.6 Periodic Schedule

The periodic schedule traversal is enabled or disabled through the Periodic Schedule Enable bit in the USB_USBCMD register.

If the Periodic Schedule Enable bit is set to zero, then the host controller simply does not try to access the periodic frame list through the USB_PERIODICLISTBASE register. Likewise, when the Periodic Schedule Enable bit is one, then the host controller does use the USB_PERIODICLISTBASE register to traverse the periodic schedule. The host controller will not react to modifications to the Periodic Schedule Enable immediately. In order to eliminate conflicts with split transactions, the host controller evaluates the Periodic Schedule Enable bit only when FRINDEX[2:0] is zero. System software must not disable the periodic schedule if the schedule contains an active split transaction work item that spans the 000b micro-frame. These work items must be removed from the schedule before the Periodic Schedule Enable bit is written to zero. The Periodic

Schedule Status bit in the USB_USBSTS register indicates status of the periodic schedule. System software enables (or disables) the periodic schedule by writing one (or zero) to the Periodic Schedule Enable bit in the USB_USBCMD register. Software then can poll the Periodic Schedule Status bit to determine when the periodic schedule has made the desired transition. Software must not modify the Periodic Schedule Enable bit unless the value of the Periodic Schedule Enable bit equals that of the Periodic Schedule Status bit.

The periodic schedule is used to manage all isochronous and interrupt transfer streams. The base of the periodic schedule is the periodic frame list. Software links schedule data structures to the periodic frame list to produce a graph of scheduled data structures. The graph represents an appropriate sequence of transactions on the

The following figure illustrates isochronous transfers (using iTDs and siTDs) with a period of one are linked directly to the periodic frame list. Interrupt transfers (are managed with queue heads) and isochronous streams with periods other than one are linked following the period-one iTD/siTDs. Interrupt queue heads are linked into the frame list ordered by poll rate. Longer poll rates are linked first (for example, closest to the periodic frame list), followed by shorter poll rates, with queue heads with a poll rate of one, on the very end.

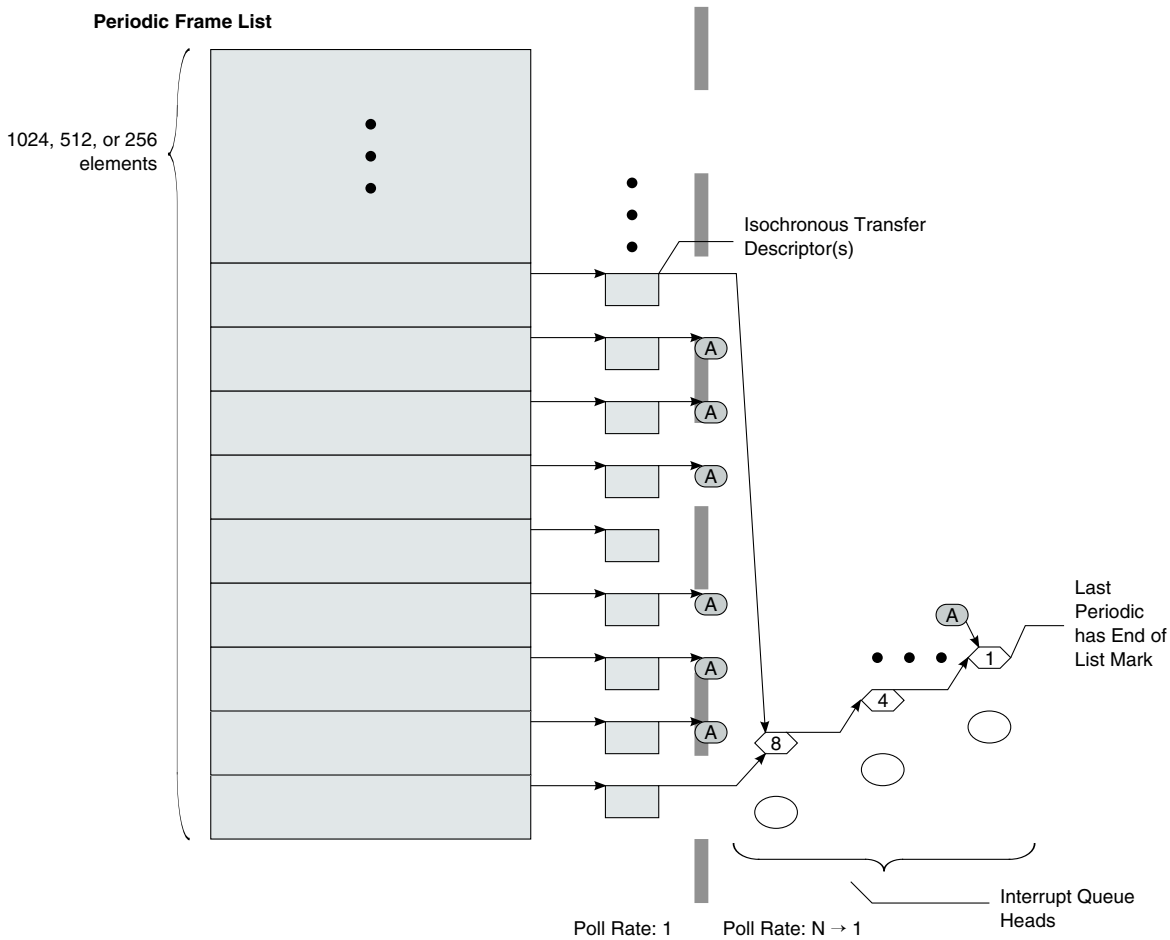


Figure 77-13. Example Periodic Schedule

77.4.3.7 Managing Isochronous Transfers Using iTDs

The structure of an iTD is presented in Isochronous (High-Speed) Transfer Descriptor (iTD). The four distinct sections to an iTD:

- The first field is the Next Link Pointer. This field is for schedule linkage purposes only.
- Transaction description array. This area is an eight-element array. Each element represents control and status information for one micro-frame's worth of transactions for a single high-speed isochronous endpoint.
- The buffer page pointer array is a 7-element array of physical memory pointers to data buffers. These are 4 K aligned pointers to physical memory.
- Endpoint capabilities. This area utilizes the unused low-order 12 bits of the buffer page pointer array. The fields in this area are used across all transactions executed for this iTD, including endpoint addressing, transfer direction, maximum packet size and high-bandwidth multiplier.

77.4.3.7.1 Host Controller Operational Model for iTDs

The host controller uses FRINDEX register bits [12:3] to index into the periodic frame list. This means that the host controller visits each frame list element eight consecutive times before incrementing to the next periodic frame list element. Each iTD contains eight transaction descriptions, which map directly to FRINDEX register bits [2:0]. Each iTD can span 8 micro-frames worth of transactions.

When the host controller fetches an iTD, it uses FRINDEX register bits [2:0] to index into the transaction description array.

If the active bit in the Status field of the indexed transaction description is set to zero, the host controller ignores the iTD and follows the Next pointer to the next schedule data structure.

When the indexed active bit is one, the host controller continues to parse the iTD. It stores the indexed transaction description and the general endpoint information (device address, endpoint number, maximum packet size, and so on.). It also uses the Page Select (PG) field to index the buffer pointer array, storing the selected buffer pointer and the next sequential buffer pointer. For example, if PG field is 0, then the host controller stores Page 0 and Page 1.

The host controller constructs a physical data buffer address by concatenating the current buffer pointer (as selected using the current transaction description's PG field) and the transaction description's Transaction Offset field. The host controller uses the endpoint addressing information and I/O-bit to execute a transaction to the appropriate endpoint. When the transaction is complete, the host controller clears the active bit and writes back any additional status information to the Status field in the currently selected transaction description.

The data buffer associated with the iTD must be virtually contiguous memory. Seven page pointers are provided to support eight high-bandwidth transactions regardless of the starting packet's offset alignment into the first page. A starting buffer pointer (physical memory address) is constructed by concatenating the page pointer (for example, page 0 pointer) selected by the active transaction descriptions' PG (for example, value: 00B) field with the transaction offset field. As the transaction moves data, the host controller must detect when an increment of the current buffer pointer crosses a page boundary. When this occurs the host controller simply replaces the current buffer pointer's page portion with the next page pointer (for example, page 1 pointer) and continues to move data. The size of each bus transaction is determined by the value in the Maximum Packet Size field. An iTD supports high-bandwidth pipes through the Mult (multiplier) field. When the Mult field is 1, 2, or 3, the host controller executes the specified number of Maximum Packet sized bus transactions for the endpoint in the current micro-frame. In

other words, the Mult field represents a transaction count for the endpoint in the current micro-frame. If the Mult field is zero, the operation of the host controller is undefined. The transfer description is used to service all transactions indicated by the Mult field.

For OUT transfers, the value of the Transaction X Length field represents the total bytes to be sent during the micro-frame. The Mult field must be set by software to be consistent with Transaction X Length and Maximum Packet Sixe. The host controller sends the bytes in Maximum Packet Size'd portions. After each transaction, the host controller decrements its local copy of Transaction X Length by Maximum Packet Size. The number of bytes the host controller sends is always Maximum Packet Size or Transaction X Length, whichever is less. The host controller advances the transfer state in the transfer description, updates the appropriate record in the iTD and moves to the next schedule data structure. The maximum sized transaction supported is 3 x 1024 bytes.

For IN transfers, the host controller issues Mult transactions. It is assumed that software has properly initialized the iTD to accommodate all of the possible data. During each IN transaction, the host controller must use Maximum Packet Size to detect packet babble errors. The host controller keeps the sum of bytes received in the Transaction X Length field. After all transactions for the endpoint have completed for the micro-frame, Transaction X Length contains the total bytes received. If the final value of Transaction X Length is less than the value of Maximum Packet Size, then less data than was allowed for was received from the associated endpoint. This short packet condition does not set the USBINT bit in the USB_USBSTS register to one. The host controller will not detect this condition. If the device sends more than Transaction X Length or Maximum Packet Size bytes (whichever is less), then the host controller sets the Babble Detected bit to one and set the Active bit to zero. Note, that the host controller is not required to update the iTD field Transaction X Length in this error scenario. If the Mult field is greater than one, then the host controller automatically executes the value of Mult transactions. The host controller will not execute all Mult transactions if:

- The endpoint is an OUT and Transaction X Length goes to zero before all the Mult transactions have executed (ran out of data), or
- The endpoint is an IN and the endpoint delivers a short packet, or an error occurs on a transaction before Mult transactions have been executed. The end of micro-frame may occur before all of the transaction opportunities have been executed. When this happens, the transfer state of the transfer description is advanced to reflect the progress that was made, the result written back to the iTD and the host controller proceeds to processing the next micro-frame. Refer to Appendix D for a table summary of the host controller required behavior for all the high-bandwidth transaction cases.

77.4.3.7.2 Software Operational Model for iTDs

A client buffer request to an isochronous endpoint may span 1 to N micro-frames. When N is larger than one, system software may have to use multiple iTDs to read or write data with the buffer (if N is larger than eight, it must use more than one iTD).

The following figure illustrates the simple model of how a client buffer is mapped by system software to the periodic schedule (that is the periodic frame list and a set of iTDs). On the right is the client description of its request. The description includes a buffer base address plus additional annotations to identify which portions of the buffer should be used with each bus transaction. In the middle is the iTD data structures used by the system software to service the client request. Each iTD can be initialized to service up to 24 transactions, organized into eight groups of up to three transactions each. Each group maps to one micro-frame's worth of transactions. The EHCI controller does not provide per-transaction results within a micro-frame. It treats the per-micro-frame transactions as a single logical transfer. On the left is the host controller's frame list. System software establishes references from the appropriate locations in the frame list to each of the appropriate iTDs. If the buffer is large, then system software can use a small set of iTDs to service the entire buffer. System software can activate the transaction description records (contained in each iTD) in any pattern required for the particular data stream.

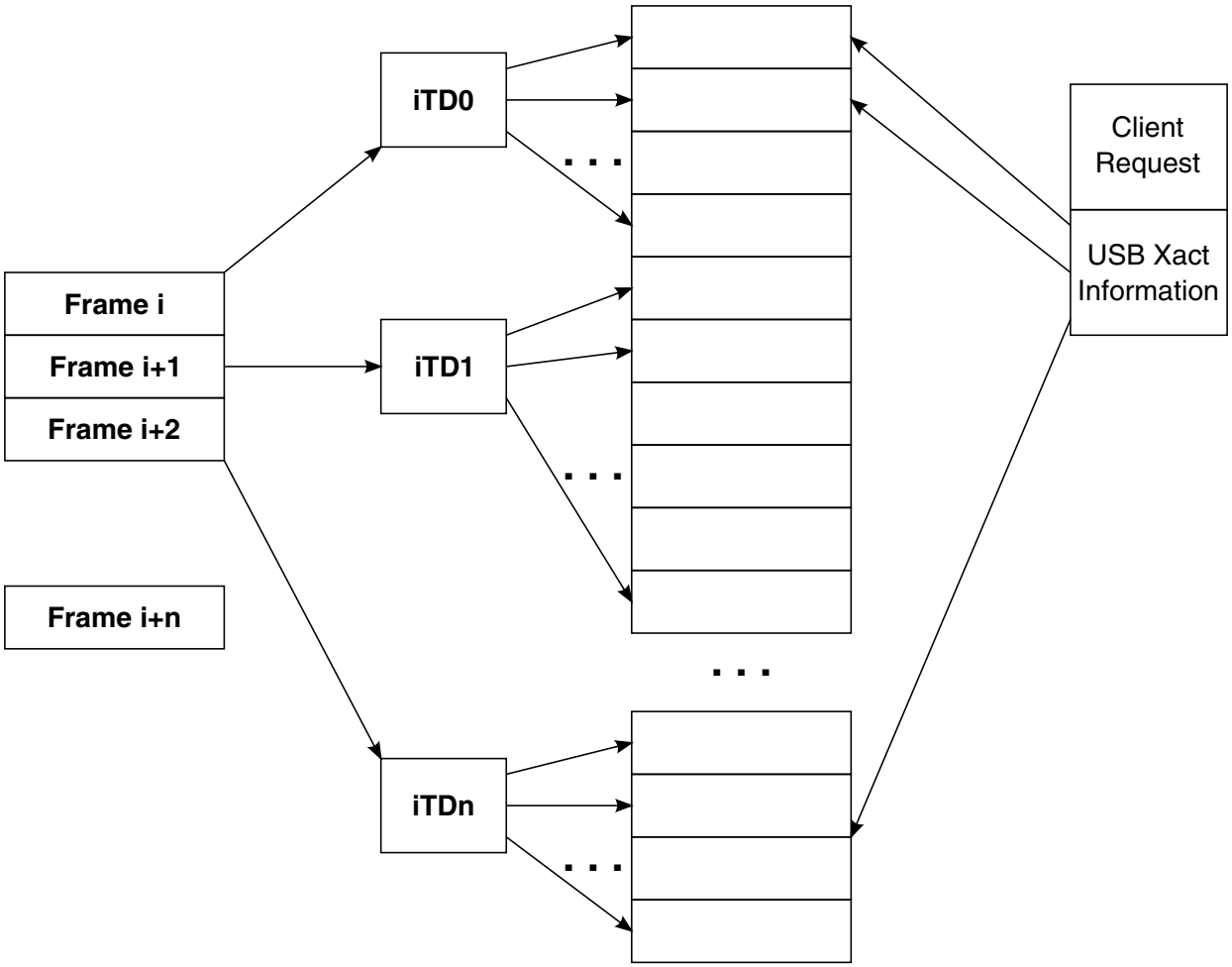


Figure 77-14. Example Association of iTDs to Client Request Buffer

As noted above, the client request includes a pointer to the base of the buffer and offsets into the buffer to annotate which buffer sections are to be used on each bus transaction that occurs on this endpoint. System software must initialize each transaction description in an iTD to ensure it uses the correct portion of the client buffer. For example, for each transaction description, the PG field is set to index the correct physical buffer page pointer and the Transaction Offset field is set relative to the correct buffer pointer page (for example, the same one referenced by the PG field). When the host controller executes a transaction it selects a transaction description record based on FRINDEX[2:0]. It then uses the current Page Buffer Pointer (as selected by the PG field) and concatenates to the transaction offset field. The result is a starting buffer address for the transaction. As the host controller moves data for the transaction, it must watch for a page wrap condition and properly advance to the next available Page Buffer Pointer. System software must not use the Page 6 buffer pointer in a transaction description where the length of the transfer wraps a page boundary. Doing so yields undefined behavior. The host controller hardware is not required to 'alias' the page selector to Page zero. USB 2.0 isochronous

endpoints can specify a period greater than one. Software can achieve the appropriate scheduling by linking iTDs into the appropriate frames (relative to the frame list) and by setting appropriate transaction description elements active bits to one.

77.4.3.7.2.1 Periodic Scheduling Threshold

The Isochronous Scheduling Threshold field in the USB_HCCPARAMS capability register is an indicator to system software as to how the host controller pre-fetches and effectively caches schedule data structures. It is used by system software when adding isochronous work items to the periodic schedule.

The value of this field indicates to system software the minimum distance it can update isochronous data (relative to the current location of the host controller execution in the periodic list) and still have the host controller process them.

The iTD and siTD data structures each describe 8 micro-frames worth of transactions. The host controller is allowed to cache one (or more) of these data structures in order to reduce memory traffic. Three basic caching models that account for the fact the isochronous data structures span 8 micro-frames. The three caching models are: no caching, micro-frame caching and frame caching.

When software is adding new isochronous transactions to the schedule, it always performs a read of the USB_FRINDEX register to determine the current frame and micro-frame the host controller is currently executing. Of course, there is no information about where in the micro-frame the host controller is, so a constant uncertainty-factor of one micro-frame has to be assumed. Combining the knowledge of where the host controller is executing with the knowledge of the caching model allows the definition of simple algorithms for how closely software can reliably work to the executing host controller.

No caching is indicated with a value of zero in the Isochronous Scheduling Threshold field. The host controller may pre-fetch data structures during a periodic schedule traversal (per micro-frame) but always dumps any accumulated schedule state at the end of the micro-frame. At the appropriate time relative to the beginning of every micro-frame, the host controller always begins schedule traversal from the frame list. Software can use the value of the USB_FRINDEX register (plus the constant 1 uncertainty-factor) to determine the approximate position of the executing host controller. When no caching is selected, software can add an isochronous transaction as near as 2 micro-frames in front of the current executing position of the host controller.

Frame caching is indicated with a non-zero value in bit [7] of the Isochronous Scheduling Threshold field. In the frame-caching model, system software assumes that the host controller caches one (or more) isochronous data structures for an entire frame (8 micro-frames). Software uses the value of the USB_FRINDEX register (plus the constant 1

uncertainty) to determine the current micro-frame/frame (assume modulo 8 arithmetic in adding the constant 1 to the micro-frame number). For any current frame N, if the current micro-frame is 0 to 6, then software can safely add isochronous transactions to Frame N + 1. If the current micro-frame is 7, then software can add isochronous transactions to Frame N + 2.

Micro-frame caching is indicated with a non-zero value in the least-significant 3 bits of the Isochronous Scheduling Threshold field. System software assumes the host controller caches one or more periodic data structures for the number of micro-frames indicated in the Isochronous Scheduling Threshold field. For example, if the count value were 2, then the host controller keeps a window of 2 micro-frames worth of state (current micro-frame, plus the next) on-chip. On each micro-frame boundary, the host controller releases the current micro-frame state and begins accumulating the next micro-frame state.

77.4.3.8 Asynchronous Schedule

The Asynchronous schedule traversal is enabled or disabled through the Asynchronous Schedule Enable bit in the USB_USBCMD register.

If the Asynchronous Schedule Enable bit is set to zero, then the host controller simply does not try to access the asynchronous schedule through the USB_ASYNCLISTADDR register. Likewise, when the Asynchronous Schedule Enable bit is one, then the host controller does use the USB_ASYNCLISTADDR register to traverse the asynchronous schedule. Modifications to the Asynchronous Schedule Enable bit are not necessarily immediate. Rather the new value of the bit is taken into consideration the next time the host controller needs to use the value of the USB_ASYNCLISTADDR register to get the next queue head.

The Asynchronous Schedule Status bit in the USB_USBSTS register indicates status of the asynchronous schedule. System software enables (or disables) the asynchronous schedule by writing one (or zero) to the Asynchronous Schedule Enable bit in the USB_USBCMD register. Software then can poll the Asynchronous Schedule Status bit to determine when the asynchronous schedule has made the desired transition. Software must not modify the Asynchronous Schedule Enable bit unless the value of the Asynchronous Schedule Enable bit equals that of the Asynchronous Schedule Status bit.

The asynchronous schedule is used to manage all Control and Bulk transfers. Control and Bulk transfers are managed using queue head data structures. The asynchronous schedule is based at the USB_ASYNCLISTADDR register. The default value of the USB_ASYNCLISTADDR register after reset is undefined and the schedule is disabled when the Asynchronous Schedule Enable bit is zero.

Software may only write this register with defined results when the schedule is disabled. For example, Asynchronous Schedule Enable bit in the USB_USBCMD and the Asynchronous Schedule Status bit in the USB_USBSTS register are zero. System software enables execution from the asynchronous schedule by writing a valid memory address (of a queue head) into this register. Then software enables the asynchronous schedule by setting the Asynchronous Schedule Enable bit is set to one. The asynchronous schedule is actually enabled when the Asynchronous Schedule Status bit is one.

When the host controller begins servicing the asynchronous schedule, it begins by using the value of the USB_ASYNCLISTADDR register. It reads the first referenced data structure and begins executing transactions and traversing the linked list as appropriate. When the host controller completes processing the asynchronous schedule, it retains the value of the last accessed queue head's horizontal pointer in the USB_ASYNCLISTADDR register. Next time the asynchronous schedule is accessed, this is the first data structure that is serviced. This provides round-robin fairness for processing the asynchronous schedule.

A host controller completes processing the asynchronous schedule when one of the following events occur:

- The end of a micro-frame occurs.
- The host controller detects an empty list condition (see Empty Asynchronous Schedule Detection)
- The schedule has been disabled through the Asynchronous Schedule Enable bit in the USB_USBCMD register.

The queue heads in the asynchronous list are linked into a simple circular list as shown in Figure 77-9. Queue head data structures are the only valid data structures that may be linked into the asynchronous schedule. An isochronous transfer descriptor (iTD or siTD) in the asynchronous schedule yields undefined results.

The maximum packet size field in a queue head is sized to accommodate the use of this data structure for all non-isochronous transfer types. The USB Specification, Revision 2.0 specifies the maximum packet sizes for all transfer types and transfer speeds. System software should always parameterize the queue head data structures according to the core specification requirements.

77.4.3.8.1 Adding Queue Heads to Asynchronous Schedule

This is a software requirement section.

There are two independent events for adding queue heads to the asynchronous schedule. The first is the initial activation of the asynchronous list. The second is inserting a new queue head into an activated asynchronous list.

Activation of the list is simple. System software writes the physical memory address of a queue head into the USB_ASYNCLISTADDR register, then enables the list by setting the Asynchronous Schedule Enable bit in the USB_USBCMD register to one.

When inserting a queue head into an active list, software must ensure that the schedule is always coherent from the host controllers' point of view. This means that the system software must ensure that all queue head pointer fields are valid. For example, qTD pointers have T-Bits set to one or reference valid qTDs and the Horizontal Pointer references a valid queue head data structure. The following algorithm represents the functional requirements:

```
InsertQueueHead (pQHeadCurrent, pQueueHeadNew)
          --
          -- Requirement: all inputs must be properly initialized.
          --
          -- pQHeadCurrent is a pointer to a queue head that is
          -- already in the active list
          -- pQHeadNew is a pointer to the queue head to be added
          --
          -- This algorithm links a new queue head into a existing
          -- list
          --
          pQueueHeadNew.HorizontalPointer  =  pQueueHeadCurrent.HorizontalPointer
          pQueueHeadCurrent.HorizontalPointer =  physicalAddressOf(pQueueHeadNew)
End InsertQueueHead
```

77.4.3.8.2 Removing Queue Heads from Asynchronous Schedule

This is a software requirement section.

There are two independent events for removing queue heads from the asynchronous schedule. The first is shutting down (deactivating) the asynchronous list. The second is extracting a single queue head from an activated list.

Software deactivates the asynchronous schedule by setting the Asynchronous Schedule Enable bit in the USB_USBCMD register to zero. Software can determine when the list is idle when the Asynchronous Schedule Status bit in the USB_USBSTS register is zero. The normal mode of operation is that software removes queue heads from the asynchronous schedule without shutting it down. Software must not remove an active queue head from the schedule. Software should first deactivate all active qTDs, wait for the queue head to go inactive, then remove the queue head from the asynchronous list. Software removes a queue head from the asynchronous list through the following algorithm. As illustrated, the unlinking is quite easy. Software merely must ensure all of the link pointers reachable by the host controller are kept consistent.

```
UnlinkQueueHead (pQHeadPrevious, pQueueHeadToUnlink, pQHeadNext)
          --
```

```
        -- Requirement: all inputs must be properly initialized.
        --
        -- pQHeadPrevious is a pointer to a queue head that
        -- references the queue head to remove
        -- pQHeadToUnlink is a pointer to the queue head to be
        -- removed
        -- pQheadNext is a pointer to a queue head still in the
        -- schedule. Software provides this pointer with the
        -- following strict rules:
        --          if the host software is one queue head, then
        --          pQHeadNext must be the same as
        --          QueueheadToUnlink.HorizontalPointer. If the host
        --          software is unlinking a consecutive series of
        --          queue heads, QHeadNext must be set by software to
        --          the  queue head remaining in the schedule.
        --
        -- This algorithm unlinks a queue head from a circular list
        --
        pQueueHeadPrevious.HorizontalPointer =  pQueueHeadToUnlink.HorizontalPointer
                            pQueueHeadToUnlink.HorizontalPointer = pQHeadNext
End UnlinkQueueHead
```

If software removes the queue head with the H-bit set to one, it must select another queue head still linked into the schedule and set its H-bit to one. This should be completed before removing the queue head. The requirement is that software keep one queue head in the asynchronous schedule, with its H-bit set to one. At the point software has removed one or more queue heads from the asynchronous schedule, it is unknown whether the host controller has a cached pointer to them. Similarly, it is unknown how long the host controller might retain the cached information, as it is implementation dependent and may be affected by the actual dynamics of the schedule load. Therefore, once software has removed a queue head from the asynchronous list, it must retain the coherency of the queue head (link pointers, and so on). It cannot disturb the removed queue heads until it knows that the host controller does not have a local copy of a pointer to any of the removed data structures.

The method software uses to determine when it is safe to modify a removed queue head is to handshake with the host controller. The handshake mechanism allows software to remove items from the asynchronous schedule, then execute a simple, lightweight handshake that is used by software as a key that it can free (or reuse) the memory associated the data structures it has removed from the asynchronous schedule.

The handshake is implemented with three bits in the host controller. The first bit is a command bit (Interrupt on Async Advance Doorbell bit in the USB_USBCMD register) that allows software to inform the host controller that something has been removed from its asynchronous schedule. The second bit is a status bit (Interrupt on Async Advance bit in the USB_USBSTS register) that the host controller sets after it has released all on-chip state that may potentially reference one of the data structures just removed. When the host controller sets this status bit to one, it also sets the command bit to zero. The third bit is an interrupt enable (Interrupt on Async Advance bit in the USB_USBINTR register) that is matched with the status bit. If the status bit is one and the interrupt enable bit is one, then the host controller asserts a hardware interrupt.

The figure below illustrates a general example. In this example, consecutive queue heads (B and C) are unlinked from the schedule using the algorithm above. Before the unlink operation, the host controller has a copy of queue head A.

The unlink algorithm requires that as software unlinks each queue head, the unlinked queue head is loaded with the address of a queue head that remains in the asynchronous schedule.

When the host controller observes that doorbell bit being set to one, it makes a note of the local reachable schedule information. In this example, the local reachable schedule information includes both queue heads (A and B). It is sufficient that the host controller can set the status bit (and clear the doorbell bit) as soon as it has traversed beyond current reachable schedule information (that is traversed beyond queue head (B) in this example). The following figure illustrates the generic queue head unlink scenario.

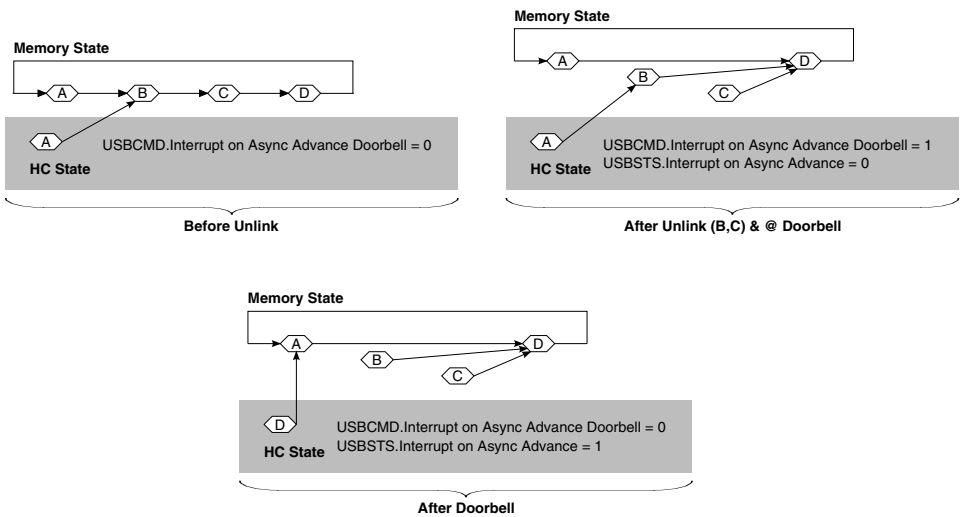


Figure 77-15. Generic Queue Head Unlink Scenario

Alternatively, a host controller implementation is allowed to traverse the entire asynchronous schedule list (for example, observed the head of the queue (twice)) before setting the Advance on Async status bit to one.

Software may re-use the memory associated with the removed queue heads after it observes the Interrupt on Async Advance status bit is set to one, following assertion of the doorbell. Software should acknowledge the Interrupt on Async Advance status as indicated in the USB_USBSTS register, before using the doorbell handshake again.

77.4.3.8.3 Empty Asynchronous Schedule Detection

The Enhanced Host Controller Interface uses two bits to detect when the asynchronous schedule is empty.

The queue head data structure (see Table 77-25) defines an *H-bit* in the queue head, which allows software to mark a queue head as being the *head* of the reclaim list. The Enhanced Host Controller Interface also keeps a 1-bit flag in the USB_USBSTS register (*Reclamation*) that is set to zero when the Enhanced Interface Host Controller observes a queue head with the H-bit set to one. The reclamation flag in the status register is set to one when any USB transaction from the asynchronous schedule is executed (or whenever the asynchronous schedule starts, see Asynchronous Schedule Traversal: *Start Event*).

If the Enhanced Host Controller Interface ever encounters an *H-bit* of one and a *Reclamation* bit of zero, the EHCI controller simply stops traversal of the asynchronous schedule.

An example illustrating the H-bit in a schedule is shown in the following figure.

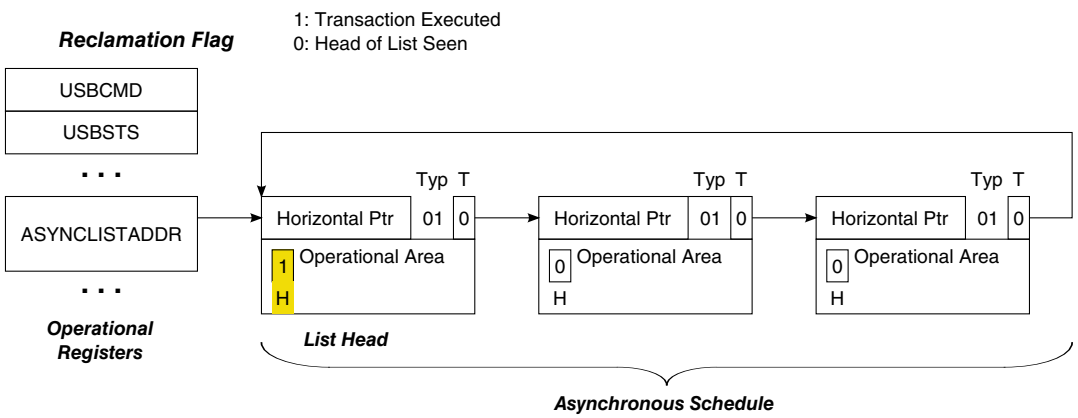


Figure 77-16. Asynchronous Schedule List w/Annotation to Mark Head of List

Software must ensure there is at most one queue head with the *H-bit* set to one, and that it is always coherent with respect to the schedule.

77.4.3.8.4 Restarting Asynchronous Schedule Before EOF

There are many situations where the host controller will detect an empty list *long* before the end of the micro-frame.

It is important to remember that under many circumstances the schedule traversal has stopped due to Nak/Nyet responses from all endpoints.

An example of particular interest is when a start-split for a bulk endpoint occurs early in the micro-frame. Given the EHCI simple traversal rules, the complete-split for that transaction may Nak/Nyet out very quickly. If it is the only item in the schedule, then the host controller ceases traversal of the Asynchronous schedule very early in the micro-frame. In order to provide reasonable service to this endpoint, the host controller should issue the complete-split before the end of the current micro-frame, instead of waiting

until the next micro-frame. When the reason for host controller idling asynchronous schedule traversal is because of empty list detection, it is mandatory the host controller implement a 'waking' method to resume traversal of the asynchronous schedule. An example method is described below.

77.4.3.8.4.1 Example Method for Restarting Asynchronous Schedule Traversal

The reason for idling the host controller when the list is empty is to keep the host controller from unnecessarily occupying too much memory bandwidth. The question is: *how long should the host controller stay idle before restarting*?

The answer in this example is based on deriving a manifest constant, which is the amount of time the host controller will stay idle before restarting traversal. In this example, the manifest constant is called *AsyncSchedSleepTime*, and has a value of 10 µsec. The value is derived based on the analysis in Example Derivation for *AsyncSchedSleepTime*, The traversal algorithm is simple:

- Traverse the Asynchronous schedule until the either an End-Of-micro-Frame event occurs, or an empty list is detected. If the event is an End-of-micro-Frame, go attempt to traverse the Periodic schedule. If the event is an empty list, then set a sleep timer and go to a *schedule sleep* state.
- When the sleep timer expires, set working context to the Asynchronous Schedule start condition and go to *schedule active* state. The start context allows the HC to reload *Nakcnt* fields, and so on. So the HC has a chance to run for more than one iteration through the schedule.

This process simply repeats itself each micro-frame. The figure below illustrates a sample state machine to manage the active and sleep states of the Asynchronous Schedule traversal policy. There are three states: Actively traversing the Asynchronous schedule, Sleeping, and Not Active. The last two are similar in terms of interaction with the Asynchronous schedule, but the Not Active state means that the host controller is busy with the Periodic schedule or the Asynchronous schedule is not enabled. The Sleeping state is specifically a special state where the host controller is just waiting for a period of time before resuming execution of the Asynchronous schedule.

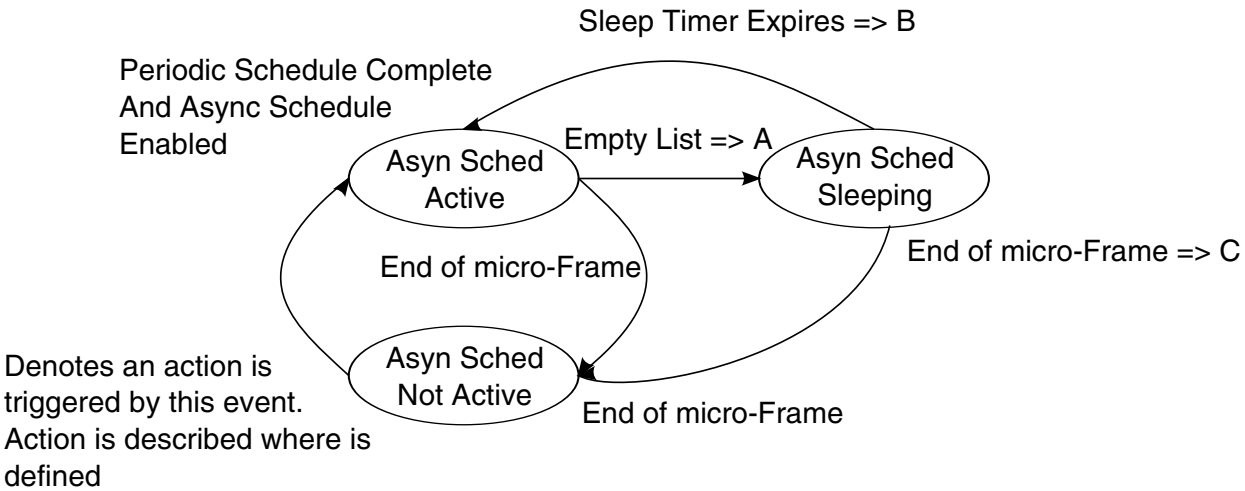


Figure 77-17. Example State Machine for Managing Asynchronous Schedule Traversal

The actions referred to in the figure above are defined in the following table.

Table 77-40. Asynchronous Schedule SM Transition Actions

Action	Action Description Label
A	On detection of the empty list, the host controller sets the *AsynchronousTraversalSleepTimer* to *AsyncSchedSleepTime*.
B	When the *AsynchronousTraversalSleepTimer* expires, the host controller sets the *Reclamation* bit in the USBSTS register to one and moves the Nak Counter reload state machine to WaitForListHead (see Nak Count Reload Control).
C	The host controller cancels the sleep timer (*AsynchronousTraversalSleepTimer*).

77.4.3.8.4.2 Async Sched Not Active

This is the initial state of the traversal state machine after a host controller reset. The traversal state machine does not leave this state when the *Asynchronous Schedule Enable* bit in the USB_USBCMD register is zero.

This state is entered from Async Sched Active or Async Sched Sleeping states when the end-of-micro-frame event is detected.

77.4.3.8.4.3 Async Sched Active

This state is entered from the Async Sched Not Active state when the periodic schedule is not active. It is also entered from the Async Sched Sleeping states when the *AsyncrhonousTraversalSleepTimer* expires. On every transition into this state, the host controller sets the *Reclamation* bit in the USB_USBSTS register to one.

While in this state, the host controller continually traverses the asynchronous schedule until either the end of micro-frame or an empty list condition is detected.

77.4.3.8.4.4 Async Sched Sleeping

The state is entered from the Async Sched Active state when a schedule empty condition is detected. On entry to this state, the host controller sets the *AsynchronousTraversalSleepTimer* to *AsyncSchedSleepTime*.

77.4.3.8.4.5 Example Derivation for *AsyncSchedSleepTime*

The derivation is based on analysis of what work the host controller could be doing next.

It assumes the host controller does not keep any state about what work is possibly pending in the asynchronous schedule. The schedule could contain any mix of the possible combinations of high- full- or low-speed control and bulk requests.

The table below summarizes some of the typical 'next transactions' that could be in the schedule, and the amount of time (for example *footprint*, or *wall clock*) the transaction takes to complete.

Table 77-41. Typical Low-/Full-speed Transaction Times

Transaction Attributes		Footprint (time)	Description
Speed	HS	11.9 ms	Maximum foot print for a worst-case, full-sized bulk data transaction.
Size	512	9.45 ms	Maximum footprint for an approximate best-case, full-sized bulk data transaction.
Type	Bulk		
Speed	FS	~50 ms	Approximate typical for full-sized bulk data. An 8-byte low-speed is about 2x, or between 90 and 100 ms.
Size	64		
Type	Bulk		
Speed	FS	~12 ms	Approximate typical for 8-byte bulk/control (that is setup)
Size	8		
Type	Cntrl		

A *AsyncSchedSleepTime* value of 10 µs provides a reasonable relaxation of the system memory load and still provides a good level of service for the various transfer types and payload sizes. For example, say we detect an empty list after issuing a start-split for a 64-byte full-speed bulk request. Assuming this is the only thing in the list, the host controller gets the results of the full-speed transaction from the hub during the fifth complete-split request. If the full-speed transaction was an IN and it nak'd, the 10 µs sleep period would allow the host controller to get the NAK results on the first complete-split.

77.4.3.8.5 Asynchronous Schedule Traversal: *Start Event*

Once the HC has *idled* itself through the empty schedule detection (Section 0), it will naturally *activate* and begin processing from the Periodic Schedule at the beginning of each micro-frame. In addition, it may have idled itself early in a micro-frame.

When this occurs (idles early in the micro-frame) the HC must occasionally *re-activate* during the micro-frame and traverse the asynchronous schedule to determine whether any progress can be made. The requirements and method for this restart are described in Restarting Asynchronous Schedule Before EOF . Asynchronous schedule *Start Event*s are defined to be:

- Whenever the host controller transitions from the periodic schedule to the asynchronous schedule. If the periodic schedule is disabled and the asynchronous schedule is enabled, then the beginning of the micro-frame is equivalent to the transition from the periodic schedule, or
- The asynchronous schedule traversal restarts from a sleeping state (see Restarting Asynchronous Schedule Before EOF).

77.4.3.8.6 Reclamation Status Bit (USBSTS Register)

The operation of the empty asynchronous schedule detection feature (see Empty Asynchronous Schedule Detection) depends on the proper management of the *Reclamation* bit in the USB_USBSTS register.

The host controller tests for an empty schedule just after it fetches a new queue head while traversing the asynchronous schedule (see Fetch Queue Head).

It is required that the host controller sets the *Reclamation* bit to one whenever an asynchronous schedule traversal *Start Event*, as documented inAsynchronous Schedule Traversal: *Start Event*, occurs. The *Reclamation* bit is also set to one whenever the host controller executes a transaction while traversing the asynchronous schedule (see Execute Transaction). The host controller sets the *Reclamation* bit to zero whenever it finds a queue head with its *H-bit* set to one. Software should only set a queue head's *H-bit* if the queue head is in the asynchronous schedule. If software sets the *H-bit* in an interrupt queue head to one, the resulting behavior is undefined. The host controller may set the *Reclamation* bit to zero when executing from the periodic schedule.

77.4.3.9 Operational Model for Nak Counter

This section describes the operational model for the *NakCnt* field defined in a queue head (see Queue Head Initialization). Software should not use this feature for interrupt queue heads. This rule is not required to be enforced by the host controller.

USB protocol has built-in flow control through the Nak response by a device. There are several scenarios, beyond the Ping feature, where an endpoint may naturally Nak or Nyet the majority of the time. An example is the host controller management of the split transaction protocol for control and bulk endpoints. All bulk endpoints (High- or Full-speed) are serviced through the same asynchronous schedule. The time between the *Start-split* transaction and the first *Complete-split* transaction could be very short (that is like when the endpoint is the only one in the asynchronous schedule). The hub NYETs (effectively Naks) the *Complete-split* transaction until the classic transaction is complete. This could result in the host controller thrashing memory, repeatedly fetching the queue head and executing the transaction to the Hub, which does not complete until after the transaction on the classic bus completes.

The two component fields in a queue head to support the throttling feature: a counter field (*NakCnt*), and a counter reload field (*RL*). *NakCnt* is used by the host controller as one of the criteria to determine whether or not to execute a transaction to the endpoint. The two operational modes associated with this counter:

- Not Used- This mode is set when the *RL* field is zero. The host controller ignores the *NakCnt* field for any execution of transactions through a queue head with an *RL* field of zero. Software must use this selection for interrupt endpoints.
- Nak Throttle Mode- This mode is selected when the *RL* field is non-zero. In this mode, the value in the *NakCnt* field represents the maximum number of Nak or Nyet responses the host controller tolerates on each endpoint. In this mode, the HC decrements the *NakCnt* field based on the token/handshake criteria listed in the table below. The host controller must reload *NakCnt* when the endpoint successfully moves data (for example, policy to reward device for moving data).

The following table describes the NakCnt field adjustment rules.

Table 77-42. NakCnt Field Adjustment Rules

Token	Handshake	
	Handshake NAK	NYET
IN/PING	decrement *NakCnt*	N/A (protocol error)
OUT	decrement *NakCnt*	No Action[1, 1] Start
Split	decrement *NakCnt*	N/A (protocol error)
Complete Split	No Action	Decrement *NakCnt*

1. Recommended behavior on this response is to reload *NakCnt*

In summary, system software enables the counter by setting the reload field (*RL*) to a non-zero value. The host controller may execute a transaction if *NakCnt* is non-zero. The host controller does not execute a transaction if *NakCnt* is zero. The reload mechanism is described in detail in Nak Count Reload Control .

NOTE

When all queue heads in the Asynchronous Schedule either exhausts all transfers or all NakCnt's go to zero, then the host controller detects an empty Asynchronous Schedule and idle schedule traversal (see Empty Asynchronous Schedule Detection).

Any time the host controller begins a new traversal of the Asynchronous Schedule, a *Start Event* is assumed, see Asynchronous Schedule Traversal: *Start Event*. Every time a Start-Event occurs, the Nak Count reload procedure is enabled.

77.4.3.9.1 Nak Count Reload Control

When the host controller reaches the *Execute Transaction* state for a queue head (meaning that it has an active operational state), it checks to determine whether the *NakCnt* field should be reloaded from *RL* (see Execute Transaction). If the answer is yes, then *RL* is copied into *NakCnt*. After the reload or if the reload is not active, the host controller evaluates whether to execute the transaction.

The host controller must reload nak counters (*NakCnt* see Table 77-25) in queue heads during the first pass through the reclamation list after an asynchronous schedule Start Event (see Asynchronous Schedule Traversal: *Start Event* for the definition of the Start Event). The Asynchronous Schedule should have at most one queue head marked as the head (see Figure 77-16).

The following figure illustrates an example state machine that satisfies the operational requirements of the host controller detecting the first pass through the Asynchronous Schedule. This state machine is maintained internal to the host controller and is only used to gate reloading of the nak counter during the queue head traversal state: Execute Transaction (see the figure below). The host controller does not perform the nak counter reload operation if the RL field (see Table 77-25) is set to zero.

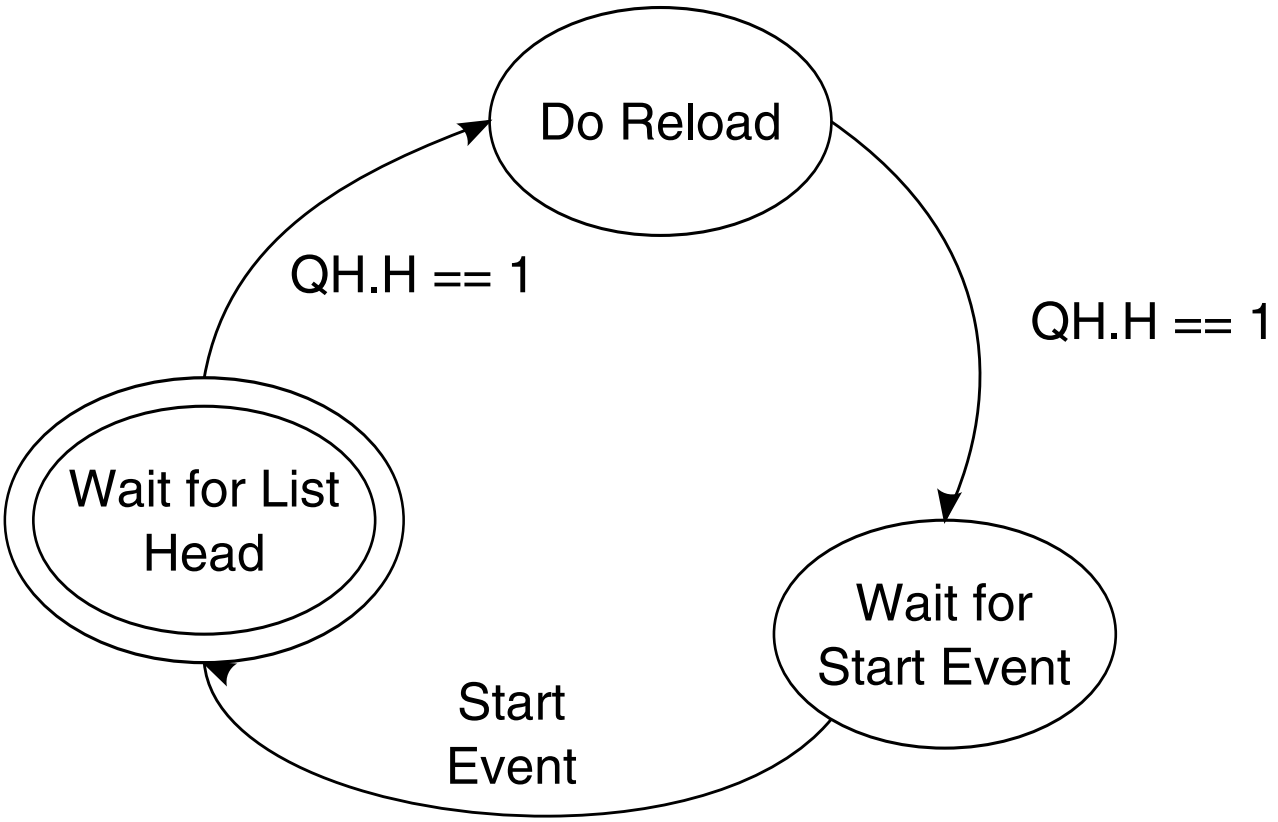


Figure 77-18. Example HC State Machine for Controlling Nak Counter Reloads

77.4.3.9.1.1 Wait for List Head

This is the initial state.

The state machine enters this state from Wait for Start Event when a start event as defined in Asynchronous Schedule Traversal: *Start Event*occurs.

The purpose of this state is to wait for the first observation of the head of the Asynchronous Schedule.

This occurs when the host controller fetches a queue head whose *H-bit* is set to one.

77.4.3.9.1.2 Do Reload

This state is entered from the Wait for List Head state when the host controller fetches a queue head with the *H-bit* set to one. While in this state, the host controller performs nak counter reloads for every queue head visited that has a non-zero nak reload value (*RL*) field.

77.4.3.9.1.3 Wait for Start Event

This state is entered from the *Do Reload* state when a queue head with the *H-bit* set to one is fetched. While in this state, the host controller does not perform nak counter reloads.

77.4.3.10 Managing Control/Bulk/Interrupt Transfers through Queue Heads

This section presents an overview of how the host controller interacts with queuing data structures.

Queue heads use the Queue Element Transfer Descriptor (qTD) structure. One queue head is used to manage the data stream for one endpoint. The queue head structure contains static endpoint characteristics and capabilities. It also contains a working area from where individual bus transactions for an endpoint are executed (see Overlay area defined in Table 77-25). Each qTD represents one or more bus transactions, which is defined in the context of this specification as a *transfer*.

The general processing model for the host controller's use of a queue head is simple:

- read a queue head,
- execute a transaction from the overlay area,
- write back the results of the transaction to the overlay area,
- move to the next queue head.

If the host controller encounters errors during a transaction, the host controller sets one (or more) of the error reporting bits in the queue head's *Status* field. The *Status* field accumulates all errors encountered during the execution of a qTD (for example, the error bits in the queue head *Status* field are 'sticky' until the transfer (qTD) has completed). This state is always written back to the source qTD when the transfer is complete. On transfer (for example, buffer or halt conditions) boundaries, the host controller must auto-advance (without software intervention) to the next qTD. Additionally, the hardware must be able to halt the queue so no additional bus transactions occurs for the endpoint and the host controller does not advance the queue.

An example host controller operational state machine of a queue head traversal is illustrated in the following figure. This state machine is a model for how a host controller should traverse a queue head. The host controller must be able to advance the queue from the *Fetch QH* state in order to avoid all hardware/software race conditions. This simple mechanism allows software to simply link qTDs to the queue head and *activate* them, then the host controller always *find* them if/when they are reachable. The figure below illustrates the Host Controller Queue Head Traversal State Machine.

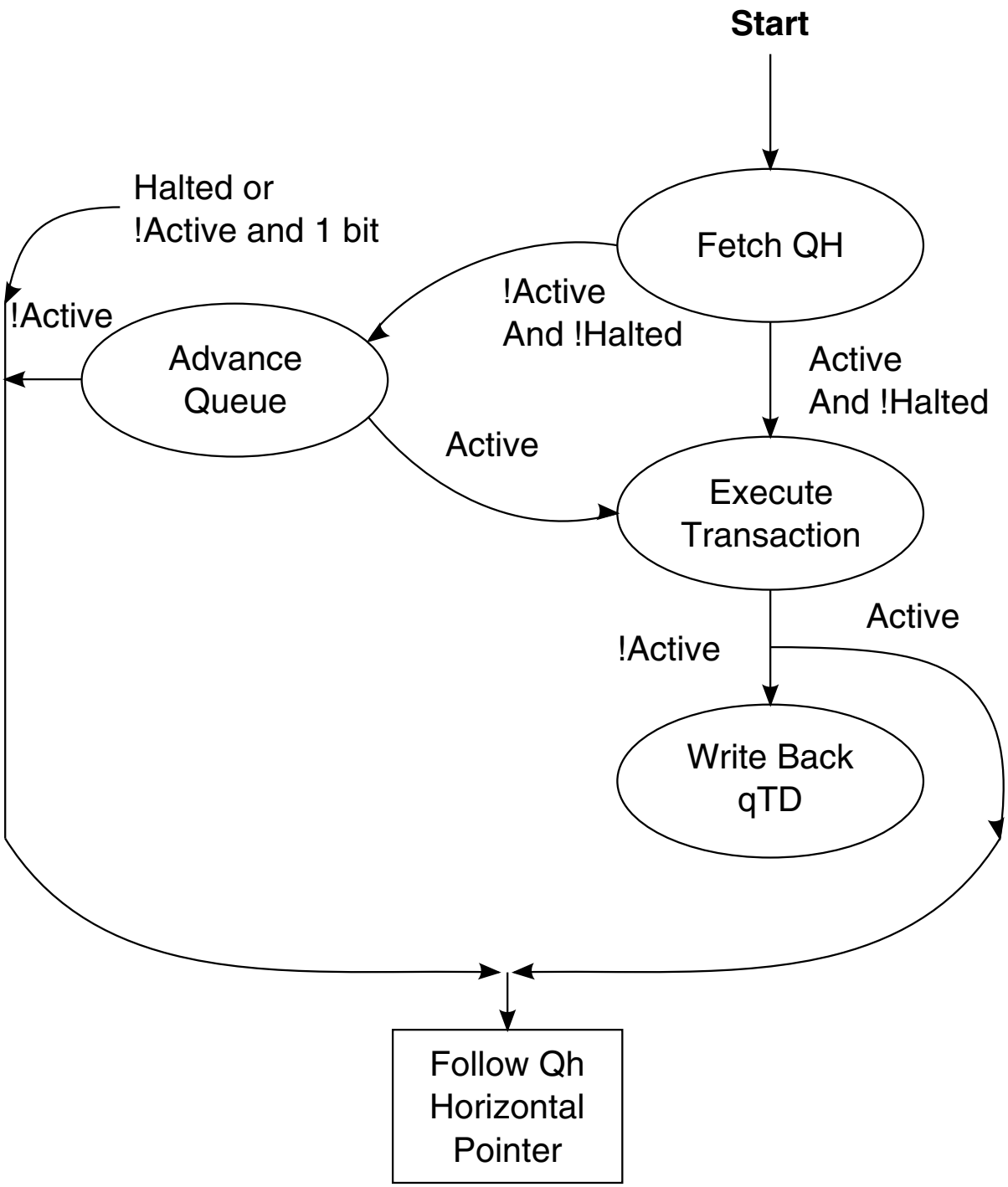


Figure 77-19. Host Controller Queue Head Traversal State Machine

This traversal state machine applies to all queue heads, regardless of transfer type or whether split transactions are required. The following sections describe each state. Each state description describes the entry criteria. The Execute Transaction state (see Execute

Transaction) describes the basic requirements for all endpoints. Split Transactions for Asynchronous Transfers and Split Transaction Interrupt describe details of the required extensions to the Execute Transaction state for endpoints requiring split transactions.

NOTE

Prior to software placing a queue head into either the periodic or asynchronous list, software must ensure the queue head is properly initialized. Minimally, the queue head should be initialized to the following (see Section Queue Head for layout of a queue head):

Valid static endpoint state.

- For the very first use of a queue head, software may zero-out the queue head transfer overlay, then set the *Next qTD Pointer* field value to reference a valid qTD.

77.4.3.10.1 Fetch Queue Head

A queue head can be referenced from the physical address stored in the ASYNCLISTADDR Register (see Next Asynch. Address (USB__ASYNCLISTADDR))/ Endpoint List Address (USB_UOG_ENDPTLISTADDR) Additionally, it may be referenced from the *Next LinkPointer* field of an iTD, siTD, FSTN or another Queue Head. If the referencing link pointer has the *Typ* field set to indicate a queue head, it is assumed to reference a queue head structure as defined in Table 77-25.

While in this state, the host controller performs operations to implement empty schedule detection (see Empty Asynchronous Schedule Detection) and Nak Counter reloads (see Operational Model for Nak Counter). After the queue head has been fetched, the host controller conducts the following queries for empty schedule detection:

- If queue head is not an interrupt queue head (that is *S-mask* is zero), and
- The *H-bit* is one, and
- The *Reclamation* bit in the USBSTS register is zero.

When these criteria are met, the host controller stops traversing the asynchronous list (as described in Empty Asynchronous Schedule Detection). When the criteria are not met, the host controller continues schedule traversal. If the queue head is not an interrupt and the *H-bit* is one and the *Reclamation* bit is one, then the host controller sets the *Reclamation* bit in the USBSTS register to zero before completing this state. The operations for reloading of the Nak Counter are described in detail in Operational Model for Nak Counter.

This state is complete when the queue head has been read on-chip.

77.4.3.10.2 Advance Queue

To advance the queue, the host controller must find the next qTD, adjust pointers, perform the overlay and write back the results to the queue head.

This state is entered from the FetchQHD state if the overlay *Active* and *Halt* bits are set to zero. On entry to this state, the host controller determines which next pointer to use to fetch a qTD, fetches a qTD and determines whether or not to perform an overlay.

NOTE

If the *I-bit* is one and the *Active* bit is zero, the host controller immediately skips processing of this queue head, exits this state and uses the horizontal pointer to the next schedule data structure. If the field *Bytes to Transfer* is not zero and the *T-bit* in the *Alternate Next qTD Pointer* is set to zero, then the host controller uses the *Alternate Next qTD Pointer*. Otherwise, the host controller uses the *NextqTD Pointer*. If *NextqTD Pointer's T-bit* is set to one, then the host controller exits this state and uses the horizontal pointer to the next schedule data structure.

Using the selected pointer the host controller fetches the referenced qTD. If the fetched qTD has its *Active* bit set to one, the host controller moves the pointer value used to reach the qTD (*Next* or *Alternate Next*) to the *Current qTD Pointer* field, then performs the overlay. If the fetched qTD has its *Active* bit set to zero, the host controller aborts the queue advance and follows the queue head's horizontal pointer to the next schedule data structure.

The host controller performs the overlay based on the following rules:

- The value of the data toggle (*dt*) field in the overlay area depends on the value of the *data toggle control* (*dtc*) bit (see Table 77-27).
- If the *EPS* field indicates the endpoint is a high-speed endpoint, the *Ping* state field is preserved by the host controller. The value of this field is not changed as a result of the overlay.
- *C-prog-mask* field is set to zero (field from incoming qTD is ignored, as is the current contents of the overlay area).
- *Frame Tag* field is set to zero (field from incoming qTD is ignored, as is the current contents of the overlay area).
- *NakCnt* field in the overlay area is loaded from the *RL* field in the queue head's Static Endpoint State.
- All other areas of the overlay are set by the incoming qTD.

The host controller exits this state when it has committed the write to the queue head.

77.4.3.10.3 Execute Transaction

The host controller enters this state from the Fetch Queue Head state only if the *Active* bit in *Status* field of the queue head is set to one.

On entry to this state, the host controller executes a few pre-operations, then checks some pre-condition criteria before committing to executing a transaction for the queue head.

The pre-operations performed and the pre-condition criteria depend on whether the queue head is an interrupt endpoint. The host controller can determine that a queue head is an interrupt queue head when the queue head's *S-mask* field contains a non-zero value. It is the responsibility of software to ensure the *S-mask* field is appropriately initialized based on the transfer type. There are other criteria that must be met if the *EPS* field indicates that the endpoint is a low- or full-speed endpoint, see Split Transactions for Asynchronous Transfers and Split Transaction Interrupt .

77.4.3.10.3.1 Interrupt Transfer Pre-condition Criteria

If the queue head is for an interrupt endpoint (for example, non-zero *S-mask* field), then the FRINDEX[2:0] field must identify a bit in the *S-mask* field that has one in it.

For example, an *S-mask* value of 00100000b would evaluate to true only when FRINDEX[2:0] is equal to 101b. If this condition is met then the host controller considers this queue head for a transaction.

77.4.3.10.3.2 Asynchronous Transfer Pre-operations and Pre-condition Criteria

If the queue head is not for an interrupt endpoint (for example, zero *S-mask* field), then the host controller performs one pre-operation and then evaluates one pre-condition criteria.

The pre-operation is:

Checks the Nak counter reload state (Operational Model for Nak Counter). It may be necessary for the host controller to reload the Nak Counter field. The reload is performed at this time.

The pre-condition evaluated is:

- Whether or not the *NakCnt* field has been reloaded, the host controller checks the value of the *NakCnt* field in the queue head. If *NakCnt* is non-zero, or if the *Reload Nak Counter* field is zero, then the host controller considers this queue head for a transaction.

77.4.3.10.3.3 Transfer Type Independent Pre-operations

Regardless of the transfer type, the host controller always performs at least one pre-operation and evaluates one pre-condition. The pre-operation is:

- A host controller internal transaction (down) counter *qHTransactionCounter* is loaded from the queue head's *Mult* field. A host controller implementation is allowed to ignore this for queue heads on the asynchronous list. It is mandatory for interrupt queue heads. Software should ensure that the *Mult* field is set appropriately for the transfer type.

The pre-conditions evaluated are:

- The host controller determines whether there is enough time in the micro-frame to complete this transaction (see Transaction Fit - A Best-Fit Approximation Algorithm for an example evaluation method). If there is not enough time to complete the transaction, the host controller exits this state.
- If the value of *qHTransactionCounter* for an interrupt endpoint is zero, then the host controller exits this state.

When the pre-operations are complete and pre-conditions are met, the host controller sets the *Reclamation* bit in the USBSTS register to one and then begins executing one or more transactions using the endpoint information in the queue head. The host controller iterates *qHTransactionCounter* times in this state executing transactions. After each transaction is executed, *qHTransactionCounter* is decremented by one. The host controller exits this state when one of the following events occurs:

- The *qHTransactionCounter* decrements to zero, or
- The endpoint responds to the transaction with any handshake other than an ACK,[4] or
- The transaction experiences a transaction error, or
- The *Active* bit in the queue head goes to zero, or
- There is not enough time in the micro-frame left to execute the next transaction(see Transaction Fit - A Best-Fit Approximation Algorithm) for example method for implementing the frame boundary test).

NOTE

For a high-bandwidth interrupt OUT endpoint, the host controller may optionally immediately retry the transaction if it fails.

The results of each transaction is recorded in the on-chip overlay area. If data was successfully moved during the transaction, the transfer state in the overlay area is advanced. To advance queue head's transfer state, the *Total Bytes to Transfer* field is decremented by the number of bytes moved in the transaction, the data toggle bit (*dt*) is toggled, the current page offset is advanced to the next appropriate value (for example,

advanced by the number of bytes successfully moved), and the *C_Page* field is updated to the appropriate value (if necessary). See Buffer Pointer List Use for Data Streaming with qTDs .

NOTE

The *Total Bytes To Transfer* field may be zero when all the other criteria for executing a transaction are met. When this occurs, the host controller executes zero-length transaction to the endpoint. If the *PID_Code* field indicates an IN transaction and the device delivers data, the host controller detects a packet babble condition, set the *babble* and *halted* bits in the *Status* field, set the *Active* bit to zero, write back the results to the source qTD, then exit this state.

In the event an IN token receives a data PID mismatch response, the host controller must ignore the received data (for example not advance the transfer state for the bytes received). Additionally, if the endpoint is an interrupt IN, then the host controller must record that the transaction occurred (for example, decrement *qHTransactionCounter*). It is recommended (but not required) the host controller continue executing transactions for this endpoint if the resultant value of *qHTransactionCounter* is greater than one.

If the response to the IN bus transaction is a Nak (or Nyet) and *RL* is non-zero, *NakCnt* is decremented by one. If *RL* is zero, then no write-back by the host controller is required (for a transaction receiving a Nak or Nyet response and the value of *CErr* did not change). Software should set the *RL* field to zero if the queue head is an interrupt endpoint. Host controller hardware is not required to enforce this rule or operation.

After the transaction has finished and the host controller has completed the post processing of the results (advancing the transfer state and possibly *NakCnt*, the host controller writes back the results of the transaction to the queue head's overlay area in main memory).

The number of bytes moved during an IN transaction depends on how much data the device endpoint delivers. The maximum number of bytes a device can send is *MaximumPacket Size*. The number of bytes moved during an OUT transaction is either *Maximum Packet Length* bytes or *Total Bytes to Transfer*, whichever is less.

If there was a transaction error during the transaction, the transfer state (as defined above) is not advanced by the host controller. The *CErr* field is decremented by one and the status field is updated to reflect the type of error observed. Transaction errors are summarized in Transaction Error .

The following events causes the host controller to clear the *Active* bit in the queue head's overlay status field. When the *Active* bit transitions from one to zero, the transfer in the overlay is considered complete. The reason for the transfer completion (clearing the *Active* bit) determines the next state.

- *CErr* field decrements to zero. When this occurs the *Halted* bit is set to one and *Active* is set to zero. This results in the hardware not advancing the queue and the pipe halts. Software must intercede to recover.
- The device responds to the transaction with a STALL PID. When this occurs, the *Halted* bit is set to one and the *Active* bit is set to zero. This results in the hardware not advancing the queue and the pipe halts. Software must intercede to recover.
- The *Total Bytes to Transfer* field is zero after the transaction completes.
  - For a zero length transaction, it was zero before the transaction was started. When this condition occurs, the *Active* bit is set to zero.
- The PID code is an IN, and the number of bytes moved during the transaction is less than the *Maximum Packet Length*. When this occurs, the *Active* bit is set to zero and a short packet condition exists. The short-packet condition is detected during the Advance Queue state. Refer to Split Transactions for additional rules for managing low- and full-speed transactions.

With the exception of a NAK response (when *RL* field is zero), the host controller always writes the results of the transaction back to the overlay area in main memory. This includes when the transfer completes. For a high-speed endpoint, the queue head information written back includes minimally the following fields: The *PID Code* field indicates an IN and the device sends more than the expected number of bytes (for example *Maximum Packet Length* or *Total Bytes to Transfer* bytes, whichever is less) (for example a packet babble). This results in the host controller setting the *Halted* bit to one.

- NakCnt, dt, Total Bytes to Transfer, C_Page, Status, CERR, and Current Offset

For a low- or full-speed device the queue head information written back also includes the fields:

- C-prog-mask, FrameTag and S-bytes.

The duration of this state depends on the time it takes to complete the transaction(s) and the status write to the overlay is committed.

77.4.3.10.3.4 Halting a Queue Head

A halted endpoint is defined only for the transfer types that are managed through queue heads (control, bulk and interrupt).

The following events indicate that the endpoint has reached a condition where no more activity can occur without intervention from the driver:

- An endpoint may return a STALL handshake during a transaction,
- A transaction had three consecutive error conditions, or
- A Packet Babble error occurs on the endpoint.

When any of these events occur (for a queue head) the Host Controller halts the queue head and set the USBERRINT status bit in the USB.USBSTS register to one. To halt the queue head, the *Active* bit is set to zero and the *Halted* bit is set to one. There may be other error status bits that are set when a queue is halted. The host controller always writes back the overlay area to the source qTD when the transfer is complete, regardless of the reason (normal completion, short packet or halt). The host controller does not advance the transfer state on a transaction that results in a *Halt* condition (for example no updates necessary for *Total Bytes to Transfer*, *C_Page*, *Current Offset,* and *dt*). The host controller must update *CErr* as appropriate. When a queue head is halted, the *USB Error Interrupt* bit in the USB.USBSTS register is set to one. If the *USB Error Interrupt Enable* bit in the USB.USBINTR register is set to one, a hardware interrupt is generated at the next interrupt threshold.

77.4.3.10.3.5 Asynchronous Schedule Park Mode

Asynchronous Schedule Park mode is a special execution mode that can be enabled by system software, where the host controller is permitted to execute more than one bus transaction from a high-speed queue head in the Asynchronous schedule before continuing horizontal traversal of the Asynchronous schedule.

This feature has no effect on queue heads or other data structures in the Periodic schedule. This feature is similar in intent as the *Mult* feature that is used in the Periodic schedule. Where-as the *Mult* feature is a characteristic that is tunable for each endpoint; park-mode is a policy that is applied to all high-speed queue heads in the asynchronous schedule. It is essentially the specification of an iterator for consecutive bus transactions to the same endpoint. All of the rules for managing bus transactions and the results of those as defined in Execute Transaction apply. This feature merely specifies how many consecutive times the host controller is permitted to execute from the same queue head before moving to the next queue head in the Asynchronous List. This feature should allow the host controller to attain better bus utilization for those devices that are capable of moving data at maximum rate, while at the same time providing a fair service to all endpoints.

A host controller exports its capability to support this feature to system software by setting the *Asynchronous Schedule Park Capability* bit in the USB.HCCPARAMs register to one. This information keys system software that the *Asynchronous Schedule Park Mode Enable* and *Asynchronous Schedule Park Mode Count* fields in the USB.USBCMD register are modifiable. System software enables the feature by writing a one to the *Asynchronous Schedule Park Mode Enable* bit.

When park-mode is not enabled (for example *Asynchronous Schedule Park Mode Enable* bit in the USB.USBCMD register is zero), the host controller must not execute more than one bus transaction per high-speed queue head, per traversal of the asynchronous schedule. When park-mode is enabled, the host controller must not apply the feature to a queue head whose *EPS* field indicates a Low/Full-speed device (for example only one bus transaction is allowed from each Low/Full-speed queue head per traversal of the asynchronous schedule). Park-mode may only be applied to queue heads in the Asynchronous schedule whose *EPS* field indicates that it is a high-speed device.

The host controller must apply park mode to queue heads whose *EPS* field indicates a high-speed endpoint. The maximum number of consecutive bus transactions a host controller may execute on a high-speed queue head is determined by the value in the *Asynchronous Schedule Park Mode Count* field in the USB.USBCMD register. Software must not set *Asynchronous Schedule Park Mode Enable* bit to one and also set *Asynchronous Schedule Park Mode Count* field to zero. The resulting behavior is not defined. An example behavioral example describes the operational requirements for the host controller implementing park-mode. This feature does not affect how the host controller handles the bus transaction as defined in Execute Transaction . It only effects how many consecutive bus transactions for the current queue head can be executed. All boundary conditions, error detection and reporting applies as usual. This feature is similar in concept to the use of the *Mult* field for high-bandwidth Interrupt for queue heads in the Periodic Schedule.

The host controller effectively loads an internal down-counter *PM-Count* from *Asynchronous Schedule Park Mode Count* when *Asyncrhonous Schedule Park Mode Enable* bit is one, and a high-speed queue head is first fetched and meets all the criteria for executing a bus transaction. After the bus transaction, *PM-Count* is decremented. The host controller may continue to execute bus transactions from the current queue head until *PM-Count* goes to zero, an error is detected, the buffer for the current transfer is exhausted or the endpoint responds with a flow-control or STALL handshake.

The following table summarizes the responses that effect whether the host controller continues with another bus transaction for the current queue head.

Table 77-43. Actions for Park Mode, based on Endpoint Response and Residual Transfer State

PID	Endpoint Response	Transfer State after Transaction		Action
		PM-Count	Bytes to Transfer	

Table continues on the next page...

Table 77-43. Actions for Park Mode, based on Endpoint Response and Residual Transfer State (continued)

IN	DATA[0,1] w/ Maximum Packet sized data	Not zero	Not Zero	Allowed to perform another bus transaction. [1],, [2]
		Not zero	Zero	Retire qTD and move to next QH
		Zero	Don't care	Move to next QH.
	DATA[0,1] w/short packet	Don't care	Don't care	Retire qTD and move to next QH.
	NAK	Don't care	Don't care	Move to next QH.
	STALL, XactErr	Don't care	Don't care	Move to next QH.
OUT	ACK	Not zero	Not Zero	Allowed to perform another bus transaction. [2]
		Not zero	Zero	Retire qTD and move to next QH
		Zero	Don't' care	Move to next QH.
	NYET, NAK	Don't care	Don't care	Move to next QH.
	STALL, XactErr	Don't care	Don't care	Move to next QH
PING	ACK	Not Zero	Not Zero	Allowed to perform another bus transaction. [2]
	NAK	Don't care	Don't care	Move to next QH
	STALL, XactErr	Don't care	Don't care	Move to next QH

1. The host controller may continue to execute bus transactions from the current high-speed queue head (if *PM-Count* is not equal to zero), if a PID mismatch is detected (for example expected DATA1 and received DATA0, or visa-versa).
2. This specification does not *require* that the host controller execute another bus transaction when *PM-Count* is non-zero. Implementations are encouraged to make appropriate complexity and performance trade-offs.

77.4.3.10.4 Write Back qTD

This state is entered from the Execute Transaction state when the *Active* bit is set to zero.

The source data for the write-back is the transfer results area of the queue head overlay area (see Table 77-43).

The host controller uses the *Current qTD Pointer* field as the target address for the qTD.

The queue head transfer result area is written back to the transfer result area of the target qTD. This state is also referred to as: qTD retirement. The fields that must be written back to the source qTD include *Total Bytes to Transfer*, *Cerr*, and *Status*.

The duration of this state depends on when the qTD write-back is committed.

77.4.3.10.5 Follow Queue Head Horizontal Pointer

The host controller must use the horizontal pointer in the queue head to the next schedule data structure when any of the following conditions exist:

- If the *Active* bit is one on exit from the Execute Transaction state, or

- When the host controller exits the Write Back qTD state, or
- If the Advance Queue state fails to advance the queue because the target qTD is not active, or
- If the *Halted* bit is one on exit from the Fetch QH state.

There is no functional requirement that the host controller wait until the current transaction is complete before using the horizontal pointer to read the next linked data structure. However, it must wait until the current transaction is complete before executing the next data structure.

77.4.3.10.6 Buffer Pointer List Use for Data Streaming with qTDs

A qTD has an array of buffer pointers, which is used to reference the data buffer for a transfer. This specification requires that the buffer associated with the transfer be *virtually contiguous*.

This means: if the buffer spans more than one physical page, it must obey the following rules (the figure below illustrates an example):

- The first portion of the buffer must begin at some offset in a page and extend through the end of the page.
- The remaining buffer cannot be allocated in small chunks scattered around memory. For each 4 K chunk beyond the first page, each buffer portion matches to a full 4 K page. The final portion, which may only be large enough to occupy a portion of a page, must start at the top of the page and be contiguous within that page.

The buffer pointer list in the qTD is long enough to support a maximum transfer size of 20 K bytes. This case occurs when all five buffer pointers are used and the first offset is zero. A qTD handles a 16 Kbyte buffer with any starting buffer alignment.

The host controller uses the field *C_Page* field as an index value to determine which buffer pointer in the list should be used to start the current transaction. The host controller uses a different buffer pointer for each physical page of the buffer. This is always true, even if the buffer is physically contiguous.

The host controller must detect when the current transaction spans a page boundary and automatically move to the next available buffer pointer in the page pointer list. The next available pointer is reached by incrementing *C_Page* and pulling the next page pointer from the list. Software must ensure there are sufficient buffer pointers to move the amount of data specified in the *Bytes to Transfer* field.

The following figure illustrates a nominal example of how System software would initialize the buffer pointers list and the *C_Page* field for a transfer size of 16383 bytes. *C_Page* is set to zero. The upper 20-bits of Page 0 references the start of the physical

page. *Current Offset* (the lower 12-bits of queue head Dword 7) holds the offset in the page for example 2049 (for example 4096-2047). The remaining page pointers are set to reference the beginning of each subsequent 4 K page.

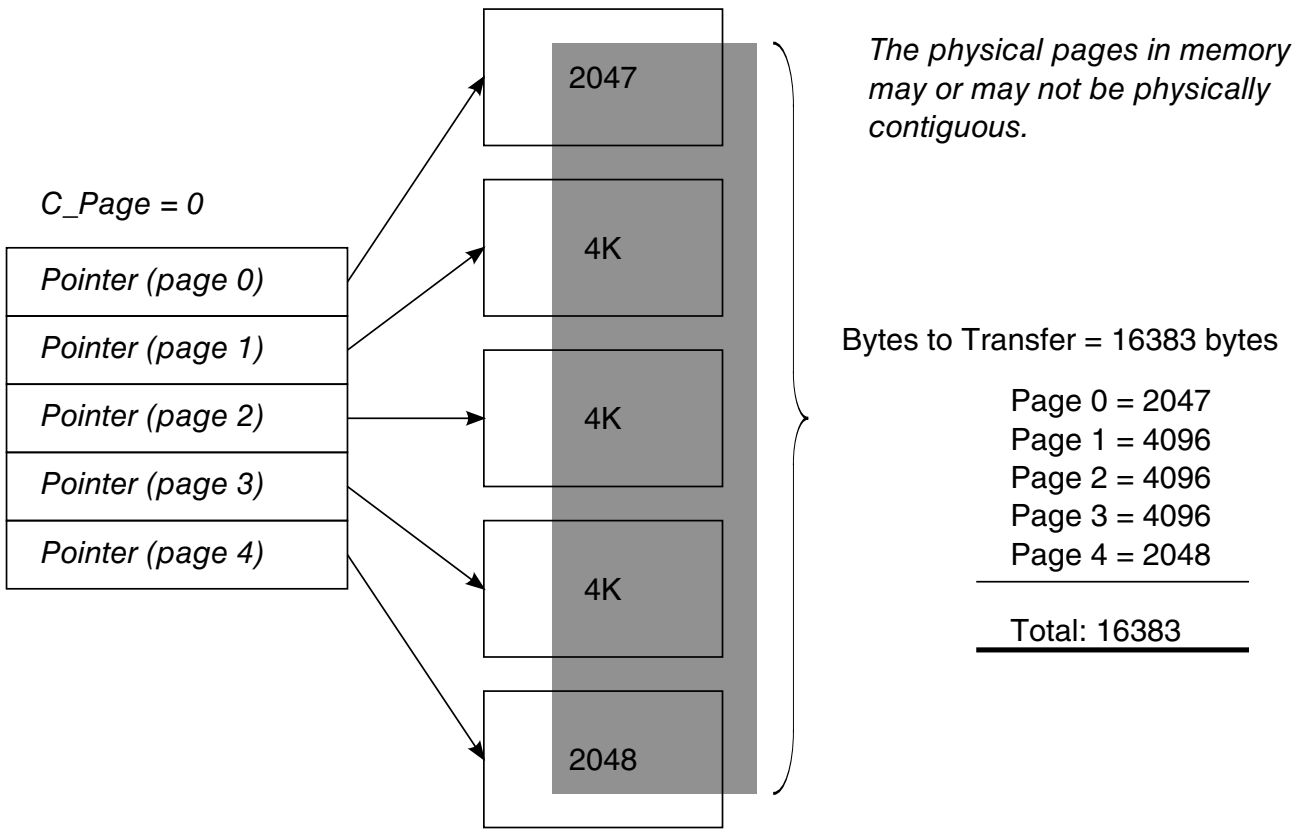


Figure 77-20. Example Mapping of qTD Buffer Pointers to Buffer Pages

For the first transaction on the qTD (assuming a 512-byte transaction), the host controller uses the first buffer pointer (page 0 because *C_Page* is set to zero) and concatenates the *Current Offset* field. The 512 bytes are moved during the transaction, the *Current Offset* and *Total Bytes to Transfer* are adjusted by 512 and written back to the queue head working area.

During the 4th transaction, the host controller needs 511 bytes in page 0 and one byte in page 1. The host controller increments *C_Page* (to 1) and use the page 1 pointer to move the final byte of the transaction. After the 4th transaction, the active page pointer is the page 1 pointer and *Current Offset* has rolled to one, and both are written back to the overlay area. The transactions continue for the rest of the buffer, with the host controller automatically moving to the next page pointer (that is *C_Page*) when necessary. The three conditions for how the host controller handles *C_Page*:

- The current transaction does not span a page boundary. The value of *C_Page* is not adjusted by the host controller.

- The current transaction does span a page boundary. The host controller must detect the page cross condition and advance to the next buffer while streaming data to/from the USB.
- The current transaction completes on a page boundary (that is the last byte moved for the current transaction is the last byte in the page for the current page pointer). The host controller must increment *C_Page* before writing back status for the transaction.

NOTE

The only valid adjustment the host controller may make to *C_Page* is to increment by one.

77.4.3.10.7 Adding Interrupt Queue Heads to the Periodic Schedule

The link path(s) from the periodic frame list to a queue head establishes in which frames a transaction can be executed for the queue head. Queue heads are linked into the periodic schedule so they are polled at the appropriate rate.

System software sets a bit in a queue head's *S-Mask* to indicate which micro-frame within 1 msec period a transaction should be executed for the queue head. Software must ensure that all queue heads in the periodic schedule have *S-Mask* set to a non-zero value. An *S-mask* with zero value in the context of the periodic schedule yields undefined results.

If the desired poll rate is greater than one frame, system software can use a combination of queue head linking and *S-Mask* values to spread interrupts of equal poll rates through the schedule so that the periodic bandwidth is allocated and managed in the most efficient manner possible. Some examples are illustrated in the following table.

Table 77-44. Example Periodic Reference Patterns for Interrupt Transfers with 2ms Poll Rate

Frame # Reference Sequence	Description
0, 2, 4, 6, 8, and so on *S-Mask* = 01h	A queue head for the *bInterval* of 2 msec (16 micro-frames) is linked into the periodic schedule so that it is reachable from the periodic frame list locations indicated in the previous column. In addition, the *S-Mask* field in the queue head is set to 01h, indicating that the transaction for the endpoint should be executed on the bus during micro-frame 0 of the frame.
0, 2, 4, 6, 8, and so on *S-Mask* = 02h	Another example of a queue head with a *bInterval* of 2 msec is linked into the periodic frame list at exactly the same interval as the previous example. However, the *S-Mask* is set to 02h indicating that the transaction for the endpoint should be executed on the bus during micro-frame 1 of the frame.

77.4.3.10.8 Managing Transfer Complete Interrupts from Queue Heads

The host controller sets an interrupt to be signaled at the next interrupt threshold when the completed transfer (qTD) has an *Interrupt on Complete* (*IOC*) bit set to one, or whenever a transfer (qTD) completes with a short packet.

If system software needs multiple qTDs to complete a client request (that is like a control transfer) the intermediate qTDs do not require interrupts. System software may only need a single interrupt to notify it that the complete buffer has been transferred. System software may set IOC's to occur more frequently. A motivation for this may be that it wants early notification so that interface data structures can be re-used in a timely manner.

77.4.3.11 Ping Control

USB 2.0 defines an addition to the protocol for high-speed devices called Ping. Ping is required for all USB 2.0 High-speed bulk and control endpoints.

Ping is not allowed for a split-transaction stream. This extension to the protocol eliminates the bad side-effects of Naking OUT endpoints. The *Status* field has a *Ping State* bit, which the host controller uses to determine the *next* actual PID it uses in the next transaction to the endpoint (see the table below).

The Ping State bit is only managed by the host controller for queue heads that meet the following criteria:

- Queue head is not an interrupt and
- *EPS* field equals High-Speed and
- *PIDCode* field equals OUT

The following table illustrates the state transition table for the host controller's responsibility for maintaining the PING protocol. Refer to Chapter 8 in the USB Specification Revision 2.0 for detailed description on the Ping protocol.

Table 77-45. Ping Control State Transition Table

Event			
Current	Host	Device	Next
Do Ping	PING	Nak	Do Ping
Do Ping	PING	Ack	Do OUT
Do Ping	PING	XactErr[1]	Do Ping
Do Ping	PING	Stall	N/C[2] Do
OUT	OUT	Nak	Do Ping

Table continues on the next page...

Table 77-45. Ping Control State Transition Table (continued)

Do OUT	OUT	Nyet	Do Ping
Do OUT	OUT	Ack	Do OUT
Do OUT	OUT	XactErr[1]	Do Ping
Do OUT	OUT	Stall	N/C[2]

1. Transaction Error (XactErr) is any time the host misses the handshake.
2. No transition change required for the Ping State bit. The Stall handshake results in the endpoint being halted (for example Active set to zero and Halt set to one). Software intervention is required to restart queue. 3 A Nyet response to an OUT means that the device has accepted the data, but cannot receive any more at this time. Host must advance the transfer state and additionally, transition the Ping State bit to Do Ping. The Ping State bit has the following encoding:

Table 77-46. Ping State bit Encoding

Value	Meaning
0B	Do OUT The host controller uses an OUT PID during the next bus transaction to this endpoint.
1B	Do Ping The host controller uses a PING PID during the next bus transaction to this endpoint.

The defined ping protocol (see USB 2.0 Specification, Chapter 8) allows the host to be *imprecise* on the initialization of the ping protocol (that is start in *Do OUT* when we don't know whether there is space on the device or not). The host controller manages the *Ping State* bit. System software sets the initial value in the queue head when it initializes a queue head. The host controller preserves the *Ping State* bit across all queue advancements. This means that when a new qTD is written into the queue head overlay area, the previous value of the *Ping State* bit is preserved.

77.4.3.12 Split Transactions

USB 2.0 defines extensions to the bus protocol for managing USB 1.x data streams through USB 2.0 Hubs.

This section describes how the host controller uses the interface data structures to manage data streams with full- and low-speed devices, connected below USB 2.0 hub, utilizing the split transaction protocol.

Refer to USB 2.0 Specification for the complete definition of the split transaction protocol. Full- and Low-speed devices are enumerated identically as high-speed devices, but the transactions to the Full- and Low-speed endpoints use the split-transaction protocol on the high-speed bus. The split transaction protocol is an encapsulation of (or wrapper around) the Full- or Low-speed transaction. The high-speed wrapper portion of the protocol is addressed to the USB 2.0 Hub and Transaction Translator below which the Full- or Low-speed device is attached.

The EHCI interface uses dedicated data structures for managing full-speed isochronous data streams (see Split Transaction Isochronous Transfer Descriptor (siTD)). Control, Bulk and Interrupt are managed using the queuing data structures (see Queue Head). The interface data structures need to be programmed with the device address and the Transaction Translator number of the USB 2.0 Hub operating as the Low-/Full-speed host controller for this link. The following sections describe the details of how the host controller must process and manage the split transaction protocol.

77.4.3.12.1 Split Transactions for Asynchronous Transfers

A queue head in the asynchronous schedule with an *EPS* field indicating a full-or low-speed device indicates to the host controller that it must use split transactions to stream data for this queue head.

All full-speed bulk and full-, low-speed control are managed through queue heads in the asynchronous schedule.

Software must initialize the queue head with the appropriate device address and port number for the transaction translator that is serving as the full/low-speed host controller for the links connecting the endpoint. Software must also initialize the split transaction state bit (*SplitXState*) to Do-Start-Split. Finally, if the endpoint is a control endpoint, then system software must set the *Control Transfer Type* (*C*) bit in the queue head to one. If this is not a control transfer type endpoint, the *C* bit must be initialized by software to be zero. This information is used by the host controller to properly set the Endpoint Type (ET) field in the split transaction bus token. When the *C* bit is zero, the split transaction token's ET field is set to indicate a bulk endpoint. When the *C* bit is one, the split transaction token's ET field is set to indicate a control endpoint. Refer to Chapter 8 of USB Specification Revision 2.0 for details.

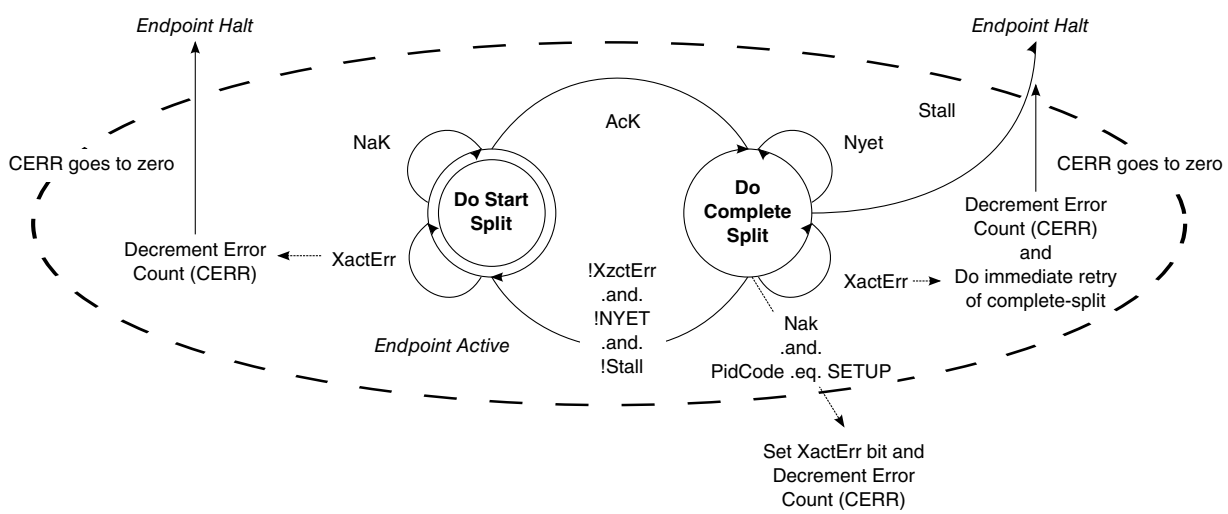


Figure 77-21. Host Controller Asynchronous Schedule Split-Transaction State Machine

77.4.3.12.1.1 Asynchronous - Do Start Split

This is the state which software must initialize a full- or low-speed asynchronous queue head. This state is entered from the Do Complete Split state only after a complete-split transaction receives a valid response from the transaction translator that is not a Nyet handshake.

For queue heads in this state, the host controller executes a start-split transaction to the appropriate transaction translator. If the bus transaction completes without an error and *PidCode* indicates an IN or OUT transaction, then the host controller reloads the error counter (*CErr*). If it is a successful bus transaction and the *PidCode* indicates a SETUP, the host controller does not reload the error counter. If the transaction translator responds with a Nak, the queue head is left in this state, and the host controller proceeds to the next queue head in the asynchronous schedule.

If the host controller times out the transaction (no response, or bad response) the host controller decrements *Cerr* and proceeds to the next queue head in the asynchronous schedule.

77.4.3.12.1.2 Asynchronous - Do Complete Split

This state is entered from the Do Start Split state only after a start-split transaction receives an Ack handshake from the transaction translator.

For queue heads in this state, the host controller executes a complete-split transaction to the appropriate transaction translator. If the transaction translator responds with a Nyet handshake, the queue head is left in this state, the error counter is reset and the host controller proceeds to the next queue head in the asynchronous schedule. When a Nyet handshake is received for a bus transaction where the queue head's *PidCode* indicates an IN or OUT, the host controller reloads the error counter (*CErr*). When a Nyet handshake is received for a complete-split bus transaction where the queue head's *PidCode* indicates a SETUP, the host controller must not adjust the value of *CErr*.

Independent of *PIDCode*, the following responses have the effects:

- Transaction Error (XactErr). Timeout or data CRC failure, and so on. The error counter (*Cerr*) is decremented by one and the complete split transaction is *immediately* retried (if possible). If there is not enough time in the micro-frame to execute the retry, the host controller MUST ensure that the next time the host controller begins executing from the Asynchronous schedule, it must begin executing from this queue head. If another start-split (for some other endpoint) is sent to the transaction translator before the complete-split is really completed, the transaction translator could dump the results (which were never delivered to the host). This is why the core specification states the retries must be immediate. A method to

accomplish this behavior is to not advance the asynchronous schedule. When the host controller returns to the asynchronous schedule in the next micro-frame, the first transaction from the schedule is the retry for this endpoint.

If *Cerr* went to zero, the host controller must halt the queue.

- NAK. The target endpoint Nak'd the full- or low-speed transaction. The state of the transfer is not advanced and the state is exited. If the *PidCode* is a SETUP, then the Nak response is a protocol error. The *XactErr* status bit is set to one and the *CErr* field is decremented.
- STALL. The target endpoint responded with a STALL handshake. The host controller sets the *halt* bit in the status byte, retires the qTD but does not attempt to advance the queue.

If the *PidCode* indicates an IN, then any of following responses are expected:

- DATA0/1. On reception of data, the host controller ensures the PID matches the expected data toggle and checks CRC. If the packet is *good*, the host controller advances the state of the transfer, for example move the data pointer by the number of bytes received, decrement *BytesToTransfer* field by the number of bytes received, and toggle the *dt* bit. The host controller then exit this state. The response and advancement of transfer may trigger other processing events, such as retirement of the qTD and advancement of the queue.

If the data sequence PID does not match the expected, the data is ignored, the transfer state is not advanced and this state is exited. If the *PidCode* indicates an OUT/SETUP, then any of following responses are expected:

- ACK. The target endpoint accepted the data, so the host controller must advance the state of the transfer. The *Current Offset* field is incremented by *Maximum Packet Length* or *Bytes to Transfer*, whichever is less. The field *Bytes To Transfer* is decremented by the same amount and the data toggle bit (*dt*) is toggled. The host controller then exit this state.
- Advancing the transfer state may cause other processing events such as retirement of the qTD and advancement of the queue (see Managing Control/Bulk/Interrupt Transfers through Queue Heads).

77.4.3.12.2 Split Transaction Interrupt

Split-transaction Interrupt-IN/OUT endpoints are managed through the same data structures used for high-speed interrupt endpoints. They both co-exist in the periodic schedule.

Queue heads/qTDs offer the set of features required for reliable data delivery, which is characteristic to interrupt transfer types. The split-transaction protocol is managed completely within this defined functional transfer framework. For example, for a high-speed endpoint, the host controller visits a queue head, execute a high-speed transaction (if criteria are met) and advance the transfer state (or not) depending on the results of the entire transaction. For low- and full-speed endpoints, the details of the *execution* phase are different (that is takes more than one bus transaction to complete), but the remainder of the operational framework is intact. This means that the transfer advancement, and so on, occurs as defined in Managing Control/Bulk/Interrupt Transfers through Queue Heads, but only occurs on the completion of a split transaction.

77.4.3.12.2.1 Split Transaction Scheduling Mechanisms for Interrupt

Full- and low-speed Interrupt queue heads have an *EPS* field indicating full- or low-speed and have a non-zero *S-mask* field.

The host controller can detect this combination of parameters and assume the endpoint is a periodic endpoint. Low- and full-speed interrupt queue heads require the use of the split transaction protocol. The host controller sets the Endpoint Type (ET) field in the split token to indicate the transaction is an interrupt. These transactions are managed through a transaction translator's periodic pipeline. Software should not set these fields to indicate the queue head is an interrupt unless the queue head is used in the periodic schedule.

System software manages the per/transaction translator periodic pipeline by budgeting and scheduling exactly during which micro-frames the start-splits and complete-splits for each endpoint occurs. The characteristics of the transaction translator are such that the high-speed transaction protocol must execute during explicit micro-frames, or the data or response information in the pipeline is lost.

The following figure illustrates the general scheduling boundary conditions that are supported by the EHCI periodic schedule and queue head data structure. The S and ^{C}X labels indicate micro-frames where software can schedule start-splits and complete splits (respectively).

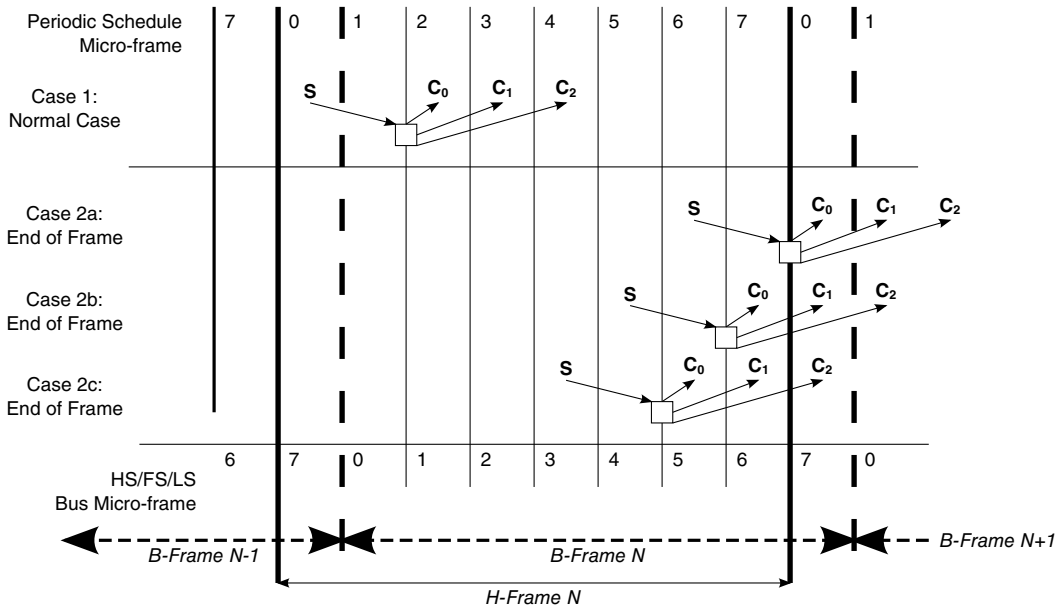


Figure 77-22. Split Transaction, Interrupt Scheduling Boundary Conditions

The scheduling cases are:

- Case 1: The normal scheduling case is where the entire split transaction is completely bounded by a frame (*H-Frame* in this case).
- Case 2a through Case 2c: The USB 2.0 Hub pipeline rules states clearly, when and how many complete-splits must be scheduled to account for earliest to latest execution on the full/low-speed link. The complete-splits may span the *H-Frame* boundary when the start-split is in micro-frame 4 or later. When this occurs, the *H-Frame* to *B-Frame* alignment requires that the queue head be reachable from consecutive periodic frame list locations. System software cannot build an efficient schedule that satisfies this requirement unless it uses FSTNs.

The figure below illustrates the general layout of the periodic schedule.

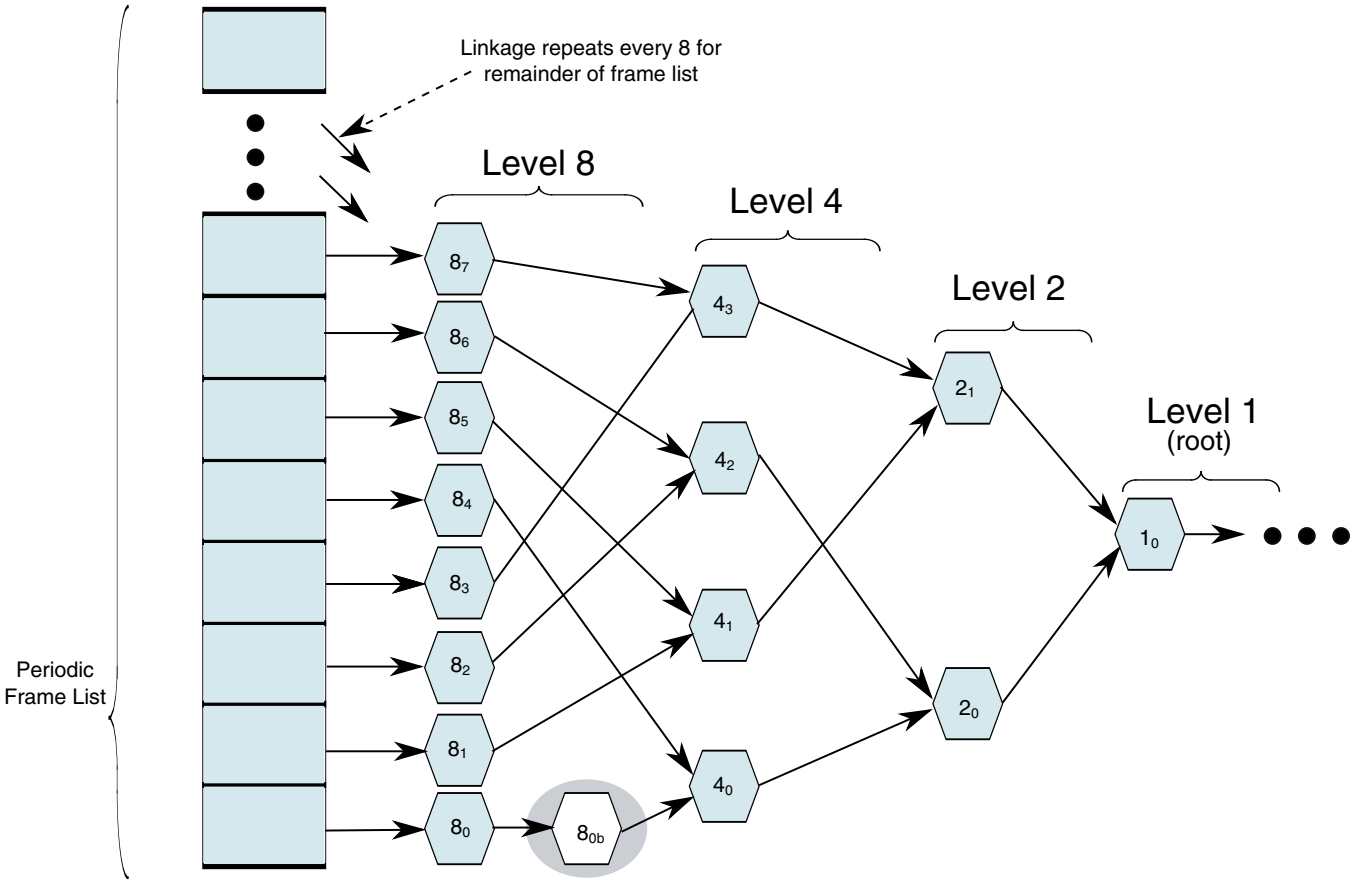


Figure 77-23. General Structure of EHCI Periodic Schedule Utilizing Interrupt Spreading

The periodic frame list is effectively the leaf level a binary tree, which is always traversed leaf to root. Each level in the tree corresponds to a 2^N poll rate. Software can efficiently manage periodic bandwidth on the USB by *spreading* interrupt queue heads that have the same poll rate requirement across all the available paths from the frame list. For example, system software can schedule eight poll rate 8 queue heads and account for them once in the high-speed bus bandwidth allocation.

When an endpoint is allocated an execution footprint that spans a frame boundary, the queue head for the endpoint must be reachable from consecutive locations in the frame list. An example would be if 8_{0b} where such an endpoint. Without additional support on the interface, to get 8_{0b} reachable at the correct time, software would have to link 8_1 to 8_{0b}. It would then have to move 4_1 and everything linked after into the same path as 4_0. This upsets the integrity of the binary tree and disallows the use of the spreading technique.

FSTN data structures are used to preserve the integrity of the binary-tree structure and enable the use of the spreading technique. Host Controller Operational Model for FSTNs defines the hardware and software operational model requirements for using FSTNs.

The following queue head fields are initialized by system software to instruct the host controller when to execute portions of the split-transaction protocol:

- *SplitXState*. This is single bit residing in the *Status* field of a queue head (see Table 77-23). This bit is used to track the current state of the split transaction.
- *Frame S-mask*. This is a bit-field where-in system software sets a bit corresponding to the micro-frame (within an *H-Frame*) that the host controller should execute a start-split transaction. This is always qualified by the value of the *SplitXState* bit in the *Status* field of the queue head. For example, referring to Figure 77-22, case one, the *S-mask* would have a value of 00000001b indicating that if the queue head is traversed by the host controller, and the *SplitXState* indicates Do_Start, and the current micro-frame as indicated by FRINDEX[2:0] is 0, then execute a start-split transaction.
- *Frame C-mask.* This is a bit-field where system software sets one or more bits corresponding to the micro-frames (within an *H-Frame*) that the host controller should execute complete-split transactions. The interpretation of this field is always qualified by the value of the *SplitXState* bit in the *Status* field of the queue head. For example, referring to Figure 77-22, case one, the *C-mask* would have a value of 00011100b indicating that if the queue head is traversed by the host controller, and the *SplitXState* indicates Do_Complete, and the current micro-frame as indicated by FRINDEX[2:0] is 2, 3, or 4, then execute a complete-split transaction. It is software's responsibility to ensure that the translation between *H-Frames* and *B-Frames* is correctly performed when setting bits in *S-mask* and *C-mask*

77.4.3.12.2.2 Host Controller Operational Model for FSTNs

The FSTN data structure is used to manage Low/Full-speed interrupt queue heads that need to be reached from consecutive frame list locations (that is boundary cases 2a through 2c).

An FSTN is essentially a *back pointer*, similar in intent to the back pointer field in the siTD data structure (see siTD Back Link Pointer).

This feature provides software a simple primitive to save a schedule position, redirect the host controller to traverse the necessary queue heads in the previous frame, then restore the original schedule position and complete normal traversal.

The four components to the use of FSTNs:

- FSTN data structure.
- A *Save Place* indicator. This is always an FSTN with its *Back Path Link Pointer.T-bit* set to zero.

- A *Restore* indicator. This is always an FSTN with its *Back Path Link Pointer.T-bit* set to one.
- Host controller FSTN traversal rules.

When the host controller encounters an FSTN during micro-frames 2 through 7 it simply follows the node's *Normal Path Link Pointer* to access the next schedule data structure.

NOTE

The FSTN's *Normal Path Link Pointer.T-bit* may set to one, which the host controller must interpret as the end of periodic list mark.

When the host controller encounters a *Save-Place* FSTN in micro-frames 0 or 1, it saves the value of the *Normal Path Link Pointer* and set an internal flag indicating that it is executing in *Recovery Path* mode. *Recovery Path* mode modifies the host controller's rules for how it traverses the schedule and limits which data structures is considered for execution of bus transactions. The host controller continues executing in *Recovery Path* mode until it encounters a *Restore* FSTN or it determines that it has reached the end of the micro-frame (see details in the list below).

The rules for schedule traversal and limited execution while in *Recovery Path* mode are:

- Always follow the *Normal Path Link Pointer* when it encounters an FSTN that is a *Save-Place* indicator. The host controller must not recursively follow *Save-Place* FSTNs. Therefore, while executing in *Recovery Path* mode, it must never follow an FSTN's *Back Path Link Pointer*.
- Do not process an siTD or, iTD data structure. Simply follow its *Next Link Pointer*.
- Do not process a QH (Queue Head) whose *EPS* field indicates a high-speed device. Simply follow its *Horizontal Link Pointer*.
- When a QH's *EPS* field indicates a Full/Low-speed device, the host controller considers only it for execution if its *SplitXState* is DoComplete (note: this applies whether the *PID Code* indicates an IN or an OUT). See Execute Transaction and Tracking Split Transaction Progress for Interrupt Transfers for a complete list of additional conditions that must be met in general for the host controller to issue a bus transaction.
  - The host controller must not execute a Start-split transaction while executing in *Recovery Path* mode. See Periodic Isochronous - Do Complete Split for special handling when in *Recovery Path* mode.
- Stop traversing the *recovery path* when it encounters an FSTN that is a *Restore* indicator. The host controller unconditionally uses the saved value of the *Save-Place* FSTN's *Normal Path Link Pointer* when returning to the normal path traversal. The

host controller must clear the context of executing a *Recovery Path* when it restores schedule traversal to the *Save-Place* FSTN's *Normal Path Link Pointer*.

- If the host controller determines that there is not enough time left in the micro-frame to complete processing of the periodic schedule, it abandons traversal of the recovery path, and clears the context of executing a recovery path. The result is that at the start of the next consecutive micro-frame, the host controller starts traversal at the frame list.

An example traversal of a periodic schedule that includes FSTNs is illustrated in the following figure.

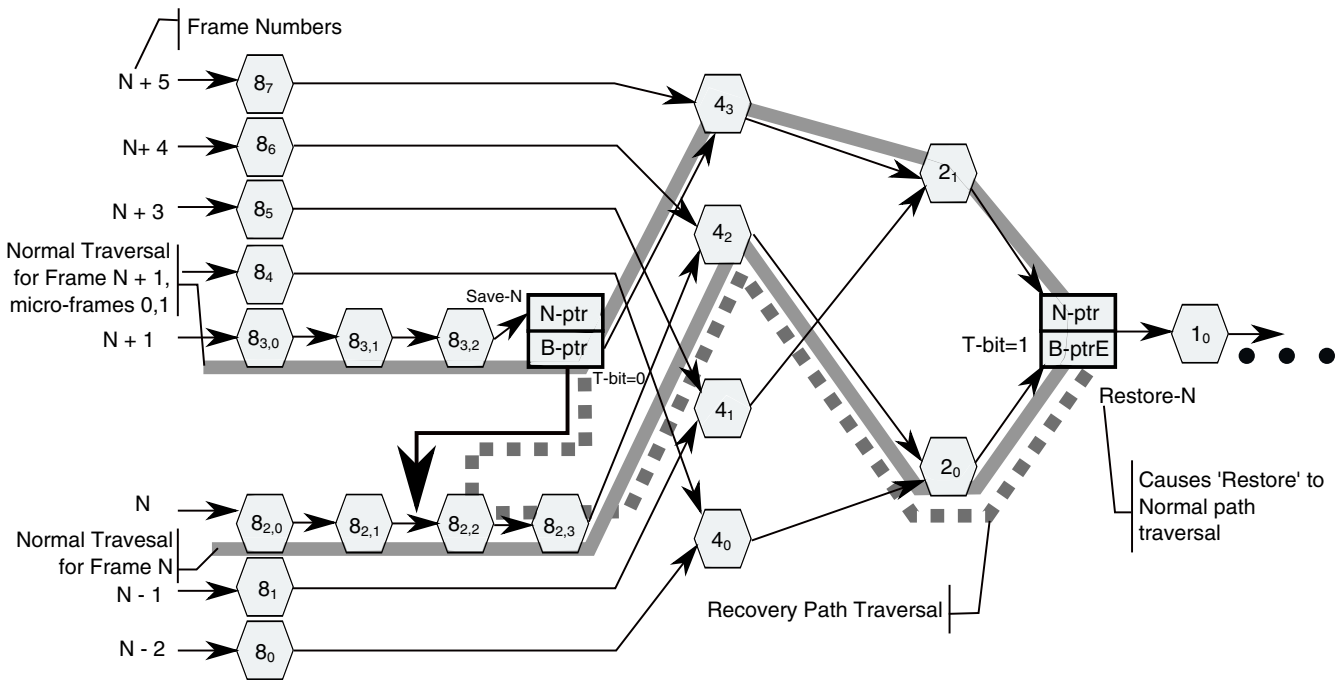


Figure 77-24. Example Host Controller Traversal of Recovery Path via FSTNs

In frame N+1 (micro-frames 0 and 1), when the host controller encounters Save-Path FSTN (Save-N), it observes that Save-N.Back Path Link Pointer.T-bit is zero (definition of a Save-Path indicator). The host controller saves the value of Save-N.Normal Path Link Pointer and follows Save-N.Back Path Link Pointer. At the same time, it sets an internal flag indicating that it is now in *Recovery Path* mode (the recovery path is annotated in the figure above with a large dashed line). The host controller continues traversing data structures on the recovery path and executing only those bus transactions as noted above, on the recovery path until it reaches Restore FSTN (Restore-N). Restore-N.Back Path Link Pointer.T-bit is set to one (definition of a Restore indicator), so the host controller exits *Recovery Path* mode by clearing the internal *Recovery Path* mode flag and commences (restores) schedule traversal using the saved value of the *Save-Place* FSTN's *Normal Path Link Pointer* (for example Save-N.Normal Path Link Pointer). The nodes traversed during these micro-frames include: {$8_{3,0}$, $8_{3,1}$, $8_{3,2}$, Save-A, $8_{2,2}$, $8_{2,3}$, 4_2,

2_0, Restore-N, 4_3, 2_1, Restore-N, 1_0 …}. The nodes on the recovery-path are in bold. In frame N (micro-frames 0-7), for this example, the host controller traverses all of the schedule data structures utilizing the *Normal Path Link Pointers* in any FSTNs it encounters. This is because the host controller has not yet encountered a *Save-Place* FSTN so it not executing in *Recovery Path* mode. When it encounters the *Restore* FSTN, (Restore-N), during micro-frames 0 and 1, it uses Restore-N.Normal Path Link Pointer to traverse to the next data structure (that is normal schedule traversal). This is because the host controller must use a Restore FSTN's *Normal Path Link Pointer* when not executing in a *Recovery-Path* mode. The nodes traversed during frame N include: {$8_{2.0}$, $8_{2.1}$, $8_{2.2}$, $8_{2.3}$, 4_2, 2_0, Restore-N, 1_0 …}.

In frame N+1 (micro-frames 2-7), when the host controller encounters Save-Path FSTN Save-N, it unconditionally follows Save-N.Normal Path Link Pointer. The nodes traversed during these micro-frames include: {$8_{3.0}$, $8_{3.1}$, $8_{3.2}$, Save-A, 4_3, 2_1, Restore-N, 1_0 …}.

77.4.3.12.2.3 Software Operational Model for FSTNs

Software must create a consistent, coherent schedule for the host controller to traverse.

When using FSTNs, system software must adhere to the following rules:

- Each *Save-Place* indicator requires a matching *Restore* indicator.
  - The *Save-Place* indicator is an FSTN with a valid *Back Path Link Pointer and T-bit* equal to zero.
    - *Back Path Link Pointer.Typ* field must be set to indicate the referenced data structure is a queue head. The *Restore* indicator is an FSTN with its *Back Path Link Pointer.T-bit* set to one.
  - A *Restore* FSTN may be matched to one or more *Save-Place* FSTNs. For example, if the schedule includes a poll-rate 1 level, then system software only needs to place a *Restore* FSTN at the beginning of this list in order to match all possible *Save-Place* FSTNs.
- If the schedule does not have elements linked at a poll-rate level of one, and one or more *Save-Place* FSTNs are used, then System Software must ensure the *Restore* FSTN's *Normal Path Link Pointer*'s *T-bit* is set to one, as this is used to mark the end of the periodic list.
- When the schedule does have elements linked at a poll rate level of one, a *Restore* FSTN must be the first data structure on the poll rate one list. All traversal paths from the frame list converge on the poll-rate one list. System software must ensure that *Recovery Path* mode is exited before the host controller is allowed to traverse the poll rate level one list.
- A *Save-Place* FSTN's *Back Path Link Pointer* must reference a queue head data structure. The referenced queue head must be reachable from the previous frame list

location. In other words, if the *Save-Place* FSTN is reachable from frame list offset N, then the FSTN's *Back Path Link Pointer* must reference a queue head that is reachable from frame list offset N-1.

Software should make the schedule as efficient as possible. What this means in this context is that software should have no more than one *Save-Place* FSTN reachable in any single frame. Note there is times when two (or more, depending on the implementation) could exist as full/low-speed footprints change with bandwidth adjustments. This could occur, for example when a bandwidth re-balance causes system software to move the *Save-Place* FSTN from one poll rate level to another. During the transition, software must preserve the integrity of the previous schedule until the new schedule is in place.

77.4.3.12.2.4 Tracking Split Transaction Progress for Interrupt Transfers

To correctly maintain the data stream, the host controller must be able to detect and report errors where data is lost.

For interrupt-IN transfers, data is lost when it makes it into the USB 2.0 hub, but the USB 2.0 host system is unable to get it from the USB 2.0 Hub and into the system before it expires from the transaction translator pipeline.

When a lost data condition is detected, the queue must be halted, thus signaling system software to recover from the error. A data-loss condition exists whenever a start-split is issued, accepted and successfully executed by the USB 2.0 Hub, but the complete-splits get unrecoverable errors on the high-speed link, or the complete-splits do not occur at the correct times. One reason complete-splits might not occur at the right time would be due to host-induced system hold-offs that cause the host controller to miss bus transactions because it cannot get timely access to the schedule in system memory.

The same condition can occur for an interrupt-OUT, but the result is not an endpoint halt condition, but rather effects only the progress of the transfer. The queue head has the following fields to track the progress of each split transaction. These fields are used to keep incremental state about which (and when) portions have been executed.

- *C-prog-mask*. This is an eight-bit bit-vector where the host controller keeps track of which complete-splits have been executed. Due to the nature of the Transaction Translator periodic pipeline, the complete-splits need to be executed in-order. The host controller needs to detect when the complete-splits have not been executed in order. This can only occur due to system hold-offs where the host controller cannot get to the memory-based schedule. *C-prog-mask* is a simple bit-vector that the host controller sets one of the *C-prog*-mask bits for each complete-split executed. The bit position is determined by the micro-frame number in which the complete-split was executed. The host controller always checks *C-prog-mask* before executing a

complete-split transaction. If the previous complete-splits have not been executed then it means one (or more) have been skipped and data has potentially been lost.

- *FrameTag*. This field is used by the host controller during the complete-split portion of the split transaction to tag the queue head with the frame number (*H-Frame* number) when the next complete split must be executed.
- *S-bytes*. This field can be used to store the number of data payload bytes sent during the start-split (if the transaction was an OUT). The *S-bytes* field must be used to accumulate the data payload bytes received during the complete-splits (for an IN).

77.4.3.12.2.5 Split Transaction Execution State Machine for Interrupt

In the following presentation, all references to micro-frame are in the context of a micro-frame within an *H-Frame.*

As with asynchronous Full- and Low-speed endpoints, a split-transaction state machine is used to manage the split transaction sequence. Aside from the fields defined in the queue head for scheduling and tracking the split transaction, the host controller calculates one internal mechanism that is also used to manage the split transaction. The internal calculated mechanism is:

- *cMicroFrameBit*. This is a single-bit encoding of the current micro-frame number. It is an eight-bit value calculated by the host controller at the beginning of every micro-frame. It is calculated from the three least significant bits of the *FRINDEX* register (that is, *cMicroFrameBit* = (1 shifted-left(*FRINDEX*[2:0])). The *cMicroFrameBit* has at most one bit asserted, which always corresponds to the current micro-frame number. For example, if the current micro-frame is 0, then *cMicroFrameBit* will equal 00000001b. The variable *cMicroFrameBit* is used to compare against the *S-mask* and *C-mask* fields to determine whether the queue head is marked for a start- or complete-splt transaction for the current micro-frame.

The following figure illustrates the state machine for managing a complete interrupt split transaction. There are two phases to each split transaction. The first is a single start-split transaction, which occurs when the *SplitXState* is at Do_Start and the single bit in *cMicroFrameBit* has a corresponding bit active in *QH.S-mask*. The transaction translator does not acknowledge the receipt of the periodic start-split, so the host controller unconditionally transitions the state to Do_Complete. Due to the available jitter in the transaction translator pipeline, there will be more than one complete-split transaction scheduled by software for the Do_Complete state. This translates simply to the fact that there are multiple bits set to a one in the *QH.C-mask* field.

The host controller keeps the queue head in the Do_Complete state until the split transaction is complete (see definition below), or an error condition triggers the *three-strikes-rule* (for example, after the host tries the same transaction three times, and each

encounters an error, the host controller will stop retrying the bus transaction and halt the endpoint, thus requiring system software to detect the condition and perform system-dependent recovery).

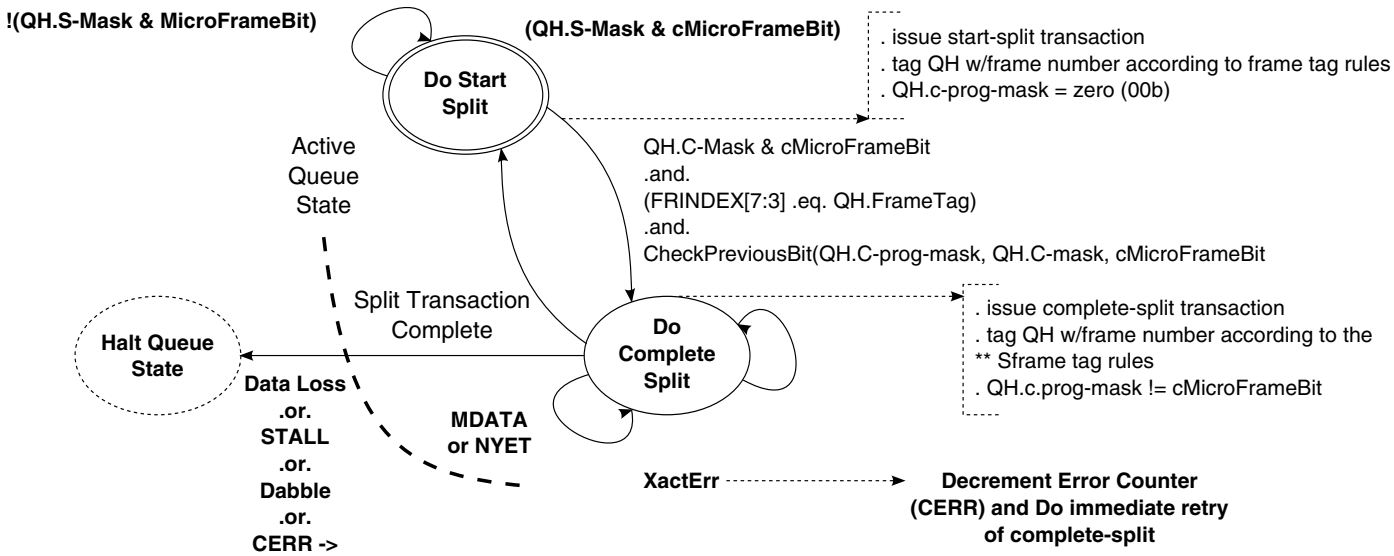


Figure 77-25. Split Transaction State Machine for Interrupt

See Previous Section for the frame tag management rules.

Periodic Interrupt - Do Start Split

This is the state software must initialize a full- or low-speed interrupt queue head *StartXState* bit. This state is entered from the Do_Complete Split state only after the split transaction is complete. This occurs when one of the following events occur: The transaction translator responds to a complete-split transaction with one of the following:

- NAK. A NAK response is a propagation of the full- or low-speed endpoint's NAK response.
- ACK. An ACK response is a propagation of the full- or low-speed endpoint's ACK response. Only occurs on an OUT endpoint.
- DATA 0/1. Only occurs for INs. Indicates that this is the last of the data from the endpoint for this split transaction.
- ERR. The transaction on the low-/full-speed link below the transaction translator had a failure (for example, timeout, bad CRC, etc.).
- NYET (and Last). The host controller issued the last complete-split and the transaction translator responded with a NYET handshake. This means that the start-split was not correctly received by the transaction translator, so it never executed a transaction to the full- or low-speed endpoint, see Section Periodic Isochronous - Do Complete Split for the definition of 'Last'.

Each time the host controller visits a queue head in this state (once within the Execute Transaction state), it performs the following test to determine whether to execute a start-split.

- *QH.S-mask* is bit-wise anded with *cMicroFrameBit*.

If the result is non-zero, then the host controller will issue a start-split transaction. If the *PIDCode* field indicates an IN transaction, the host controller must zero-out the *QH.S-bytes* field. After the split-transaction has been executed, the host controller sets up state in the queue head to track the progress of the complete-split phase of the split transaction. Specifically, it records the expected frame number into *QH.FrameTag* field (see Section), set *C-prog-mask* to zero (00h), and exits this state. Note that the host controller must not adjust the value of *CErr* as a result of completion of a start-split transaction.

Periodic Interrupt - Do Complete Split

This state is entered unconditionally from the Do Start Split state after a start-split transaction is executed on the bus. Each time the host controller visits a queue head in this state (once within the Execute Transaction state), it checks to determine whether a complete-split transaction should be executed now.

There are four tests to determine whether a complete-split transaction should be executed.

- Test A. *cMicroFrameBit* is bit-wise anded with *QH.C-mask* field. A non-zero result indicates that software scheduled a complete-split for this endpoint, during this micro-frame.
- Test B. *QH.FrameTag* is compared with the current contents of *FRINDEX*[7:3]. An equal indicates a match.
- Test C. The complete-split progress bit vector is checked to determine whether the previous bit is set, indicating that the previous complete-split was appropriately executed. An example algorithm for this test is provided below:

```
Algorithm Boolean CheckPreviousBit(QH.C-prog-mask, QH.C-mask, cMicroFrameBit)
Begin
-- Return values:
-- TRUE - no error
-- FALSE - error
--
Boolean rvalue = TRUE;
previousBit = cMicroframeBit logical-rotate-right(1)
-- Bit-wise anding previousBit with C-mask indicates
-- whether there was an intent
-- to send a complete split in the previous micro-frame. So,
-- if the
-- 'previous bit' is set in C-mask, check C-prog-mask to
-- make sure it
-- happened.
If (previousBit bitAND QH.C-mask)then
                                If not(previousBit bitAND QH.C-prog-mask) then
                    rvalue = FALSE;
          End if
End If
-- If the C-prog-mask already has a one in this bit position,
```

```
-- then an aliasing
-- error has occurred. It will probably get caught by the
-- FrameTag Test, but
-- at any rate it is an error condition that as detectable here
-- should not allow
-- a transaction to be executed.
If (cMicroFrameBit bitAND QH.C-prog-mask) then
 rvalue = FALSE;
End if
return (rvalue)
End Algorithm
```

- Test D. Check to see if a start-split should be executed in this micro-frame. Note this is the same test performed in the Do Start Split state (see Section Periodic Isochronous - Do Start Split). Whenever it evaluates to TRUE and the controller is NOT processing in the context of a *Recovery Path* mode, it means a start-split should occur in this micro-frame. Test D and Test A evaluating to TRUE at the same time is a system software error. Behavior is undefined.

If (A .and. B .and. C .and. not(D)) then the host controller will execute a complete-split transaction. When the host controller commits to executing the complete-split transaction, it updates *QH.C-prog-mask* by bit-ORing with *cMicroFrameBit*. On completion of the complete-split transaction, the host controller records the result of the transaction in the queue head and sets *QH.FrameTag* to the expected *H-Frame* number (see Section). The effect to the state of the queue head and thus the state of the transfer depends on the response by the transaction translator to the complete-split transaction. The following responses have the effects (note that any responses that result in decrementing of the *CErr* will result in the queue head being halted by the host controller if the result of the decrement is zero):

- NYET (and Last). On each NYET response, the host controller checks to determine whether this is the last complete-split for this split transaction. Last is defined in this context as the condition where all of the scheduled complete-splits have been executed. If it is the last complete-split (with a NYET response), then the transfer state of the queue head is not advanced (never received any data) and this state exited. The transaction translator must have responded to all the clompete-splits with NYETs, meaning that the start-split issued by the host controller was not received. The start-split should be retried at the next poll period.
- The test for whether this is the Last complete split can be performed by XOR *QH.C-mask* with *QH.C-prog-mask*. If the result is all zeros then all complete-splits have been executed. When this condition occurs, the *XactErr* status bit is set to a one and the *CErr* field is decremented.
- NYET (and not Last). See above description for testing for Last. The complete-split transaction received a NYET response from the transaction translator. Do not update any transfer state (except for *C-prog-mask* and *FrameTag*) and stay in this state. The host controller must not adjust *CErr* on this response.

- Transaction Error (XactErr). Timeout, data CRC failure, etc. The *CErr* field is decremented and the *XactErr* bit in the *Status* field is set to a one. The complete split transaction is *immediately* retried (if *Cerr* is non-zero).If there is not enough time in the micro-frame to complete the retry and the endpoint is an IN, or *CErr* is decremented to a zero from a one, the queue is halted. If there is not enough time in the micro-frame to complete the retry and the endpoint is an OUT and *CErr* is not zero, then this state is exited (that is, return to Do Start Split). This results in a retry of the entire OUT split transaction, at the next poll period. Refer to Chapter 11 Hubs (specifically the section full- and low-speed Interrupts) in the USB Specification Revision 2.0 for detailed requirements on why these errors must be immediately retried.
- ACK. This can only occur if the target endpoint is an OUT. The target endpoint ACK'd the data and this response is a propagation of the endpoint ACK up to the host controller. The host controller must advance the state of the transfer. The *Current Offset* field is incremented by *Maximum Packet Length* or *Bytes to Transfer*, whichever is less. The field *Bytes To Transfer* is decremented by the same amount. And the data toggle bit (*dt*) is toggled. The host controller will then exit this state for this queue head. The host controller must reload *CErr* with maximum value on this response. Advancing the transfer state may cause other process events such as retirement of the qTD and advancement of the queue (see Section Managing Control/Bulk/Interrupt Transfers through Queue Heads).
- MDATA. This response will only occur for an IN endpoint. The transaction translator responded with zero or more bytes of data and an MDATA PID. The incremental number of bytes received is accumulated in *QH.S-bytes*. The host controller must not adjust *CErr* on this response.
- DATA0/1. This response may only occur for an IN endpoint. The number of bytes received is added to the accumulated byte count in *QH.S-bytes*. The state of the transfer is advanced by the result and the host controller will exit this state for this queue head.
- Advancing the transfer state may cause other processing events such as retirement of the qTD and advancement of the queue (see Section Managing Control/Bulk/Interrupt Transfers through Queue Heads).
- If the data sequence PID does not match the expected, the entirety of the data received in this split transaction is ignored, the transfer state is not advanced and this state is exited.
- NAK. The target endpoint Nak'd the full- or low-speed transaction. The state of the transfer is not advanced, and this state is exited. The host controller must reload *CErr* with maximum value on this response.

- ERR. There was an error during the full- or low-speed transaction. The ERR status bit is set to a one, *Cerr* is decremented, the state of the transfer is not advanced, and this state is exited.
- STALL. The queue is halted (an exit condition of the Execute Transaction state). The status field bits: *Active* bit is set to zero and the *Halted* bit is set to a one and the qTD is retired. Responses which are not enumerated in the list or which are received out of sequence are illegal and may result in undefined host controller behavior. The other possible combinations of tests A, B, C, and D may indicate that data or response was lost. The table below lists the possible combinations and the appropriate action.

Table 77-47. Interrupt IN/OUT Do Complete Split State Execution Criteria

Condition	Action	Description
not(A) not(D)	Ignore QHD	Neither a start nor complete-split is scheduled for the current micro-frame.Host controller should continue walking the schedule.
A not(C)	If PIDCode = IN Halt QHD If PIDCode = OUT Retry start-split	Progress bit check failed. These means a complete-split has been missed. There is the possibility of lost data. If *PIDCode* is an IN, then the Queue head must be halted. If *PIDCode* is an OUT, then the transfer state is not advanced and the state exited (for example, start-split is retried). This is a host-induced error and does not effect *CER*R. In either case, set the *Missed Micro-frame* bit in the status field to a one.
A not(B) C	If PIDCode = IN Halt QHD If PIDCode = OUT Retry start-split	*QH.FrameTag* test failed. This means that exactly one or more *H-Frames* have been skipped. This means complete-splits and have missed. There is the possibility of lost data. If *PIDCode* is an IN, then the Queue head must be halted. If *PIDCode* is an OUT, then the transfer state is not advanced and the state exited (for example, start-split is retried). This is a host-induced error and does not effect *CER*R. In either case, set the *Missed Micro-frame* bit in the status field to a one.
A B C not(D)	Execute complete-split	This is the non-error case where the host controller executes a complete-split transaction.

Table continues on the next page...

Table 77-47. Interrupt IN/OUT Do Complete Split State Execution Criteria (continued)

D	If PIDCode = IN Halt QHD If PIDCode = OUT Retry start-split	This is a degenerate case where the start-split was issued, but all of the complete-splits were skipped and all possible intervening opportunities to detect the missed data failed to fire. If *PIDCode* is an IN, then the Queue head must be halted. If *PIDCode* is an OUT, then the transfer state is not advanced and the state exited (for example, start-split is retried). This is a host-induced error and does not effect *CERR*. In either case, set the *Missed Micro-frame* bit in the status field to a one. Note: When executing in the context of a *Recovery Path* mode, the host controller is allowed to process the queue head and take the actions indicated above, or it may wait until the queue head is visited in the normal processing mode. Regardless, the host controller must not execute a start-split in the context of a executing in a *Recovery Path* mode.

Managing QH.FrameTag Field

The *QH.FrameTag* field in a queue head is completely managed by the host controller. The rules for setting *QH.FrameTag* are simple:

- Rule 1: If transitioning from Do Start Split to Do Complete Split and the current value of *FRINDEX*[2:0] is 6 *QH.FrameTag* is set to *FRINDEX*[7:3] + 1. This accommodates split transactions whose start-split and complete-splits are in different *H-Frames* (case 2a, see Figure 77-22).
- Rule 2: If the current value of *FRINDEX*[2:0] is 7, *QH.FrameTag* is set to *FRINDEX*[7:3] + 1. This accommodates staying in Do Complete Split for cases 2a, 2b, and 2c (Figure 77-22).
- Rule 3: If transitioning from Do_Start Split to Do Complete Split and the current value of *FRINDEX*[2:0] is not 6, or currently in Do Complete Split and the current value of (*FRINDEX*[2:0]) is not 7, *FrameTag* is set to *FRINDEX*[7:3]. This accommodates all other cases (Figure 77-22).

77.4.3.12.2.6 Rebalancing the Periodic Schedule

System software must occasionally adjust a periodic queue head's S-mask and C-mask fields during operation. This need occurs when adjustments to the periodic schedule create a new bandwidth budget and one or more queue head's are assigned new execution footprints (that is, new S-mask and C-mask values).

It is imperative that System software must not update these masks to new values in the midst of a split transaction. In order to avoid any race conditions with the update, the EHCI host controller provides a simple assist to system software. System software sets the *Inactivate-on-next-Transaction* (*I*) bit to a one to signal the host controller that it intends to update the S-mask and C-mask on this queue head. System software will then

wait for the host controller to observe the *I-bit* is a one and transition the *Active* bit to a zero. The rules for how and when the host controller sets the *Active* bit to zero are enumerated below:

- If the *Active* bit is a zero, no action is taken. The host controller does not attempt to advance the queue when the *I-bit* is a one.
- If the *Active* bit is a one and the *SplitXState* is DoStart (regardless of the value of *S-mask*), the host controller will simply set *Active* bit to a zero. The host controller is not required to write the transfer state back to the *current* qTD. Note that if the *S-mask* indicates that a start-split is scheduled for the current micro-frame, the host controller must not issue the start-split bus transaction. It must set the *Active* bit to zero.

System software must save transfer state before setting the *I-bit* to a one. This is required so that it can correctly determine what transfer progress (if any) occurred after the *I-bit* was set to a one and the host controller executed its final bus-transaction and set *Active* to a zero.

After system software has updated the S-mask and C-mask, it must then reactivate the queue head. Because the *Active* bit and the *I-bit* cannot be updated with the same write, system software needs to use the following algorithm to coherently re-activate a queue head that has been stopped via the *I-bit*.

1. Set the *Halted* bit to a one, then
2. Set the *I-bit* to a zero, then
3. Set the *Active* bit to a one and the *Halted* bit to a zero in the same write.

Setting the *Halted* bit to a one inhibits the host controller from attempting to advance the queue between the time the *I-bit* goes to a zero and the *Active* bit goes to a one.

77.4.3.12.3 Split Transaction Isochronous

Full-speed isochronous transfers are managed using the split-transaction protocol through a USB 2.0 transaction translator in a USB2.0 Hub. The EHCI controller utilizes siTD data structure to support the special requirements of isochronous split-transactions.

This data structure uses the scheduling model of isochronous TDs (iTD, Section Isochronous (High-Speed) Transfer Descriptor (iTD)) (see Section Managing Isochronous Transfers Using iTDs for the operational model of iTDs) with the contiguous data feature provided by queue heads. This simple arrangement allows a single isochronous scheduling model and adds the additional feature that all data received from the endpoint (per split transaction) must land into a contiguous buffer.

77.4.3.12.3.1 Split Transaction Scheduling Mechanisms for Isochronous

Full-speed isochronous transactions are managed through a transaction translator's periodic pipeline. As with full- and low-speed interrupt, system software manages each transaction translator's periodic pipeline by budgeting and scheduling exactly during which micro-frames the start-splits and complete-splits for each full-speed isochronous endpoint occur.

The requirements described in Section Split Transaction Scheduling Mechanisms for Interrupt apply. The following figure illustrates the general scheduling boundary conditions that are supported by the EHCI periodic schedule. The ^{S}X and ^{C}X labels indicate micro-frames where software can schedule start- and complete-splits (respectively). The *H-Frame* boundaries are marked with a large, solid bold vertical line. The *B-Frame* boundaries are marked with a large, bold, dashed line. The bottom of the figure illustrates the relationship of an siTD to the *H-Frame*.

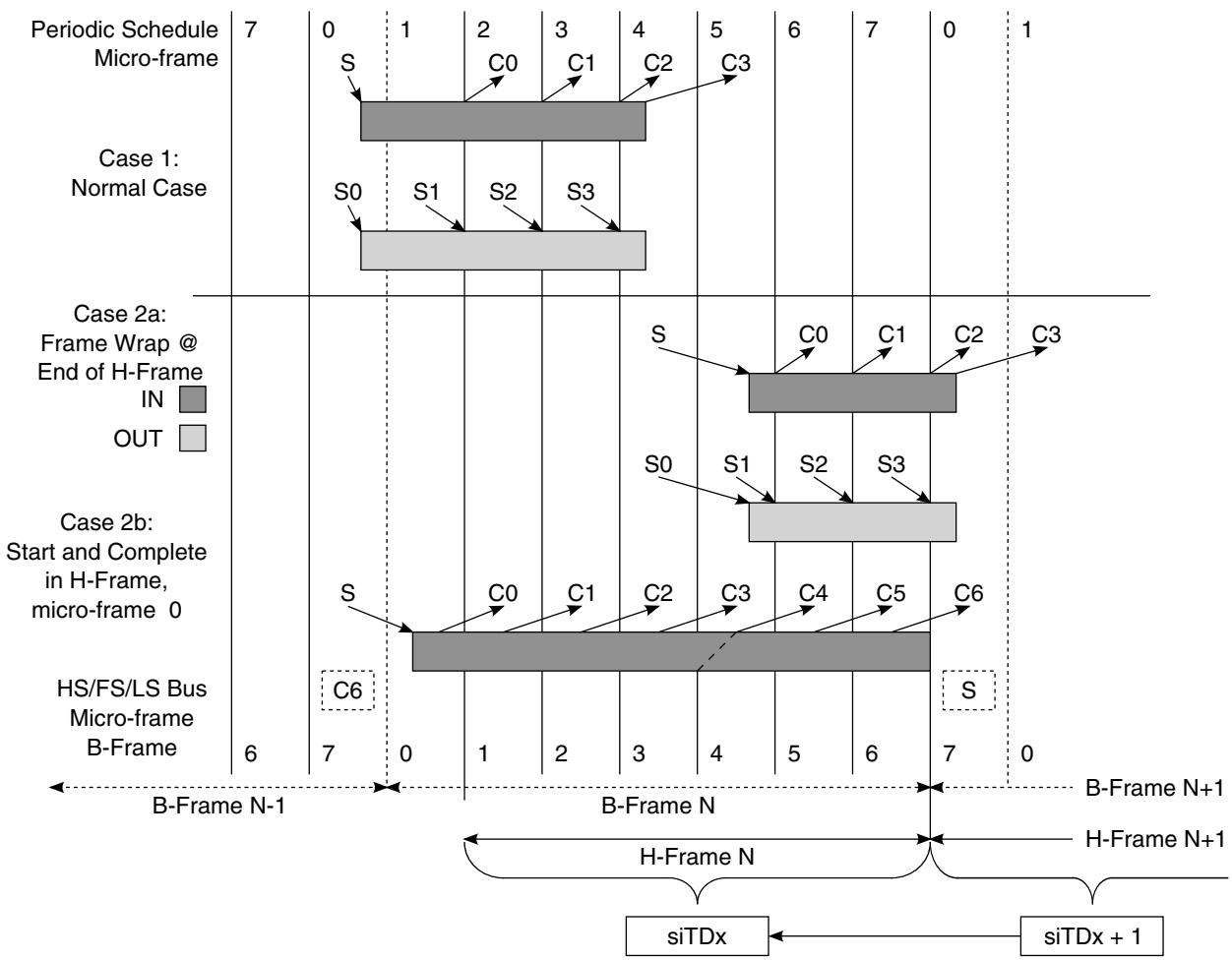


Figure 77-26. Split Transaction, Isochronous Scheduling Boundary Conditions

When the endpoint is an isochronous OUT, there are only start-splits, and no complete-splits. When the endpoint is an isochronous IN, there is at most one start-split and one to *N* complete-splits. The scheduling boundary cases are:

- *Case 1*: The entire split transaction is completely bounded by an *H-Frame*. For example: the start-splits and complete-splits are all scheduled to occur in the same *H-Frame*.
- *Case 2a*: This boundary case is where one or more (at most two) complete-splits of a split transaction IN are scheduled across an *H-Frame* boundary. This can only occur when the split transaction has the possibility of moving data in *B-Frame,* micro-frames 6 or 7 (*H-Frame* micro-frame 7 or 0). When an *H-Frame* boundary wrap condition occurs, the scheduling of the split transaction spans more than one location in the periodic list.(For example, it takes two siTDs in adjacent periodic frame list locations to fully describe the scheduling for the split transaction.)
- Although the scheduling of the split transaction may take two data structures, all of the complete-splits for each full-speed IN isochronous transaction must use only one data pointer. For this reason, siTDs contain a back pointer, the use of which is described below.
- Software must never schedule full-speed isochronous OUTs across an *H-Frame* boundary.
- *Case 2b*: This case can only occur for a very large isochronous IN. It is the only allowed scenario where a start-split and complete-split for the same endpoint can occur in the same micro-frame. Software must enforce this rule by scheduling the large transaction first. Large is defined to be anything larger than 579 byte maximum packet size.

A subset of the same mechanisms employed by full- and low-speed interrupt queue heads are employed in siTDs to schedule and track the portions of isochronous split transactions. The following fields are initialized by system software to instruct the host controller when to execute portions of the split transaction protocol.

- *SplitXState*. This is a single bit residing in the *Status* field of an siTD (see Figure 77-27). This bit is used to track the current state of the split transaction. The rules for managing this bit are described in Section Split Transaction Execution State Machine for Interrupt .
- *Frame S-mask*. This is a bit-field where-in system software sets a bit corresponding to the micro-frame (within an *H-Frame*) that the host controller should execute a start-split transaction. This is always qualified by the value of the *SplitXState* bit. For example, referring to the IN example in Figure 77-26, case one, the *S-mask* would have a value of 00000001b indicating that if the siTD is traversed by the host controller, and the *SplitXState* indicates Do Start Split, and the current micro-frame as indicated by USB.FRINDEX[2:0] is 0, then execute a start-split transaction.

- *Frame C-mask.* This is a bit-field where system software sets one or more bits corresponding to the micro-frames (within an *H-Frame*) that the host controller should execute complete-split transactions. The interpretation of this field is always qualified by the value of the *SplitXState* bit. For example, referring to the IN example in Figure 77-26, case one, the *C-mask* would have a value of 00111100b indicating that if the siTD is traversed by the host controller, and the *SplitXState* indicates Do Complete Split, and the current micro-frame as indicated by *USB.FRINDEX*[2:0] is 2, 3, 4, or 5, then execute a complete-split transaction.
- *Back Pointer*. This field in a siTD is used to complete an IN split-transaction using the previous *H-Frame*'s siTD. This is only used when the scheduling of the complete-splits span an *H-Frame* boundary.

There exists a one-to-one relationship between a high-speed isochronous split transaction (including all start- and complete-splits) and one full-speed isochronous transaction. An siTD contains (amongst other things) buffer state and split transaction scheduling information. An siTD's buffer state always maps to one full-speed isochronous data payload. This means that for any full-speed transaction payload, a single siTD's data buffer must be used. This rule applies to both IN an OUTs. An siTD's scheduling information usually also maps to one high-speed isochronous split transaction. The exception to this rule is the *H-Frame* boundary wrap cases mentioned above.

The siTD data structure describes at most, one frame's worth of high-speed transactions and that description is strictly bounded within a frame boundary. The figure below illustrates some examples. On the top are examples of the full-speed transaction footprints for the boundary scheduling cases described above. In the middle are time-frame references for both the *B-Frames* (HS/FS/LS Bus) and the *H-Frames*. On the bottom is illustrated the relationship between the scope of an siTD description and the time references. Each *H-Frame* corresponds to a single location in the periodic frame list. The implication is that each siTD is reachable from a single periodic frame list location at a time.

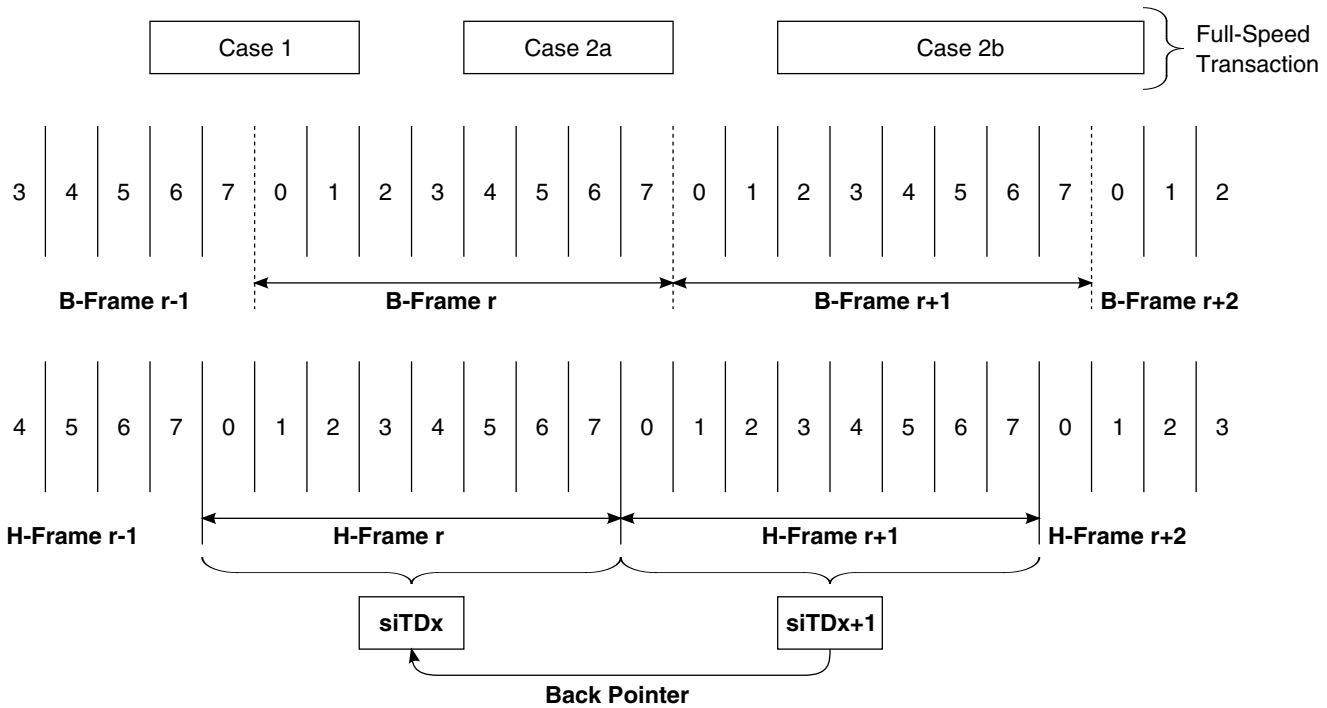


Figure 77-27. siTD Scheduling Boundary Examples

Each case is described below:

- *Case 1*: One siTD is sufficient to describe and complete the isochronous split transaction because the whole isochronous split transaction is tightly contained within a single *H-Frame*.
- *Case 2a, 2b*: Although both INs and OUTs can have these footprints, OUTs always take only one siTD to schedule. However, INs (for these boundary cases) require two siTDs to complete the scheduling of the isochronous split transaction $siTD_X$ is used to always issue the start-split and the first *N* complete-splits. The full-speed transaction (for these cases) can deliver data on the full-speed bus segment during micro-frame 7 of $H\text{-}Frame_{Y+1}$, or micro-frame 0 of $H\text{-}Frame_{Y+2}$. The complete splits are scheduled using $siTD_{X+2}$ (not shown). The complete-splits to extract this data must use the buffer pointer from $siTD_{X+1}$. The only way for the host controller to reach $siTD_{X+1}$ from $H\text{-}Frame_{Y+2}$is to use $siTD_{X+2}$'s back pointer. The host controller rules for when to use the back pointer are described is Section Periodic Isochronous - Do Complete Split .

Software must apply the following rules when calculating the schedule and linking the schedule data structures into the periodic schedule:

- Software must ensure that an isochronous split-transaction is started so that it will complete before the end of the *B-Frame*.
- Software must ensure that for a single full-speed isochronous endpoint, there is never a start-split and complete-split in *H-Frame, micro-frame 1*. This is mandated as a rule so that case 2a and case 2b can be discriminated. According to the core USB specification, the long isochronous transaction illustrated in Case 2b, could be scheduled so that the start-split was in micro-frame 1 of *H-Frame* N and the last complete-split would need to occur in micro-frame 1 of H-Frame N+1. However, it is impossible to discriminate between cases 2a and case 2b, which has significant impact on the complexity of the host controller.

77.4.3.12.3.2 Tracking Split Transaction Progress for Isochronous Transfers

To correctly maintain the data stream, the host controller must be able to detect and report errors where device to host data is lost. Isochronous endpoints do not employ the concept of a halt on error, however the host is required to identify and report per-packet errors observed in the data stream. This includes schedule traversal problems (skipped micro-frames), timeouts and corrupted data received.

In similar kind to interrupt split-transactions, the portions of the split transaction protocol must execute in the micro-frames they are scheduled. The queue head data structure used to manage full- and low-speed interrupt has several mechanisms for tracking when portions of a transaction have occurred. Isochronous transfers use siTDs, for their transfers, and the data structures are only reachable via the schedule in the exact micro-frame in which they are required (so all the mechanism employed for tracking in queue heads is not required for siTDs). Software has the option of reusing siTD several times in the complete periodic schedule. However, it must ensure that the results of split transaction N are consumed and the siTD reinitialized (activated) before the host controller gets back to the siTD (in a future micro-frame).

Split-transaction isochronous OUTs utilize a low-level protocol to indicate which portions of the split transaction data have arrived. Control over the low-level protocol is exposed in an siTD via the fields *Transaction Position* (*TP*) and *Transaction Count* (*T-count*). If the entire data payload for the OUT split transaction is larger than 188 bytes, there will be more than one start-split transaction, each of which require proper annotation. If host hold-offs occur, then the sequence of annotations received from the host will not be complete, which is detected and handled by the transaction translator. See Section Periodic Isochronous - Do Start Split for a description on how these fields are used during a sequence of start-split transactions.

The fields *siTD.T-Count* and *siTD.TP* are used by the host controller to drive and sequence the transaction position annotations. It is the responsibility of system software to properly initialize these fields in each siTD. Once the budget for a split-transaction

isochronous endpoint is established, *S-mask*, *T-Count*, and *TP* initialization values for all the siTD associated with the endpoint are constant. They remain constant until the budget for the endpoint is recalculated by software and the periodic schedule adjusted.

For IN-endpoints, the transaction translator simply annotates the response data packets with enough information to allow the host controller to identify the last data. As with split transaction Interrupt, it is the host controller's responsibility to detect when it has missed an opportunity to execute a complete-split. The following field in the siTD is used to track and detect errors in the execution of a split transaction for an IN isochronous endpoint.

- *C-prog-mask*. This is an eight-bit bit-vector where the host controller keeps track of which complete-splits have been executed. Due to the nature of the Transaction Translator periodic pipeline, the complete-splits need to be executed in-order. The host controller needs to detect when the complete-splits have not been executed in order. This can only occur due to system hold-offs where the host controller cannot get to the memory-based schedule. *C-prog-mask* is a simple bit-vector that the host controller sets a bit for each complete-split executed. The bit position is determined by the micro-frame (USB.FRINDEX[2:0]) number in which the complete-split was executed. The host controller always checks *C-prog-mask* before executing a complete-split transaction. If the previous complete-splits have not been executed, then it means one (or more) have been skipped and data has potentially been lost. System software is required to initialize this field to zero before setting an siTD's *Active* bit to a one.

If a transaction translator returns with the final data before all of the complete-splits have been executed, the state of the transfer is advanced so that the remaining complete-splits are not executed. Refer to Section Asynchronous - Do Complete Split for a description on how the state of the transfer is advanced. It is important to note that an IN siTD is retired based solely on the responses from the Transaction Translator to the complete-split transactions. This means, for example, that it is possible for a transaction translator to respond to a complete-split with an MDATA PID. The number of bytes in the MDATA's data payload could cause the siTD field *Total Bytes to Transfer* to decrement to zero. This response can occur, before all of the scheduled complete-splits have been executed. In other interface, data structures (for example, high-speed data streams through queue heads), the transition of *Total Bytes to Transfer* to zero signals the end of the transfer and results in setting of the *Active* bit to zero. However, in this case, the result has not been delivered by the Transaction Translator and the host must continue with the next complete-split transaction to extract the residual transaction state. This scenario occurs because of the pipeline rules for a Transaction Translator (see Chapter 11 of the Universal Serial Bus Revision 2.0). In summary the periodic pipeline rules require that on a micro-frame boundary, the Transaction Translator will hold the final two bytes received (if it has not seen an End Of Packet (EOP)) in the full-speed bus pipe stage and give the

remaining bytes to the high-speed pipeline stage. At the micro-frame boundary, the Transaction Translator could have received the entire packet (including both CRC bytes) but not received the packet EOP. In the next micro-frame, the Transaction Translator will respond with an MDATA and send all of the data bytes (with the two CRC bytes being held in the full-speed pipeline stage). This could cause the siTD to decrement its *Total Bytes to Transfer* field to zero, indicating it has received all expected data. The host must still execute one more (scheduled) complete-split transaction in order to extract the results of the full-speed transaction from the Transaction Translator (for example, the Transaction Translator may have detected a CRC failure, and this result must be forwarded to the host).

If the host experiences hold-offs that cause the host controller to skip one or more (but not all) scheduled split transactions for an isochronous OUT, then the protocol to the transaction translator will not be consistent and the transaction translator will detect and react to the problem. Likewise, for host hold-offs that cause the host controller to skip one or more (but not all) scheduled split transactions for an isochronous IN, the *C-prog-mask* is used by the host controller to detect errors. However, if the host experiences a hold-off that causes it to skip all of an siTD, or an siTD expires during a host hold off (for example, a hold-off occurs and the siTD is no longer reachable by the host controller in order for it to report the hold-off event), then system software must detect that the siTDs have not been processed by the host controller (that is, state not advanced) and report the appropriate error to the client driver.

77.4.3.12.3.3 Split Transaction Execution State Machine for Isochronous

In the following presentation, all references to micro-frame are in the context of a micro-frame within an *H-Frame*.

If the *Active* bit in the *Status* byte is a zero, the host controller will ignore the siTD and continue traversing the periodic schedule. Otherwise the host controller will process the siTD as specified below. A split transaction state machine is used to manage the split-transaction protocol sequence. The host controller uses the fields defined in Section Tracking Split Transaction Progress for Interrupt Transfers , plus the variable *cMicroFrameBit* defined in Section Split Transaction Execution State Machine for Interrupt to track the progress of an isochronous split transaction. The figure below illustrates the state machine for managing an siTD through an isochronous split transaction. Bold, dotted circles denote the state of the *Active* bit in the *Status* field of a siTD. The Bold, dotted arcs denote the transitions between these states. Solid circles denote the states of the split transaction state machine and the solid arcs denote the transitions between these states. Dotted arcs and boxes reference actions that take place either as a result of a transition or from being in a state.

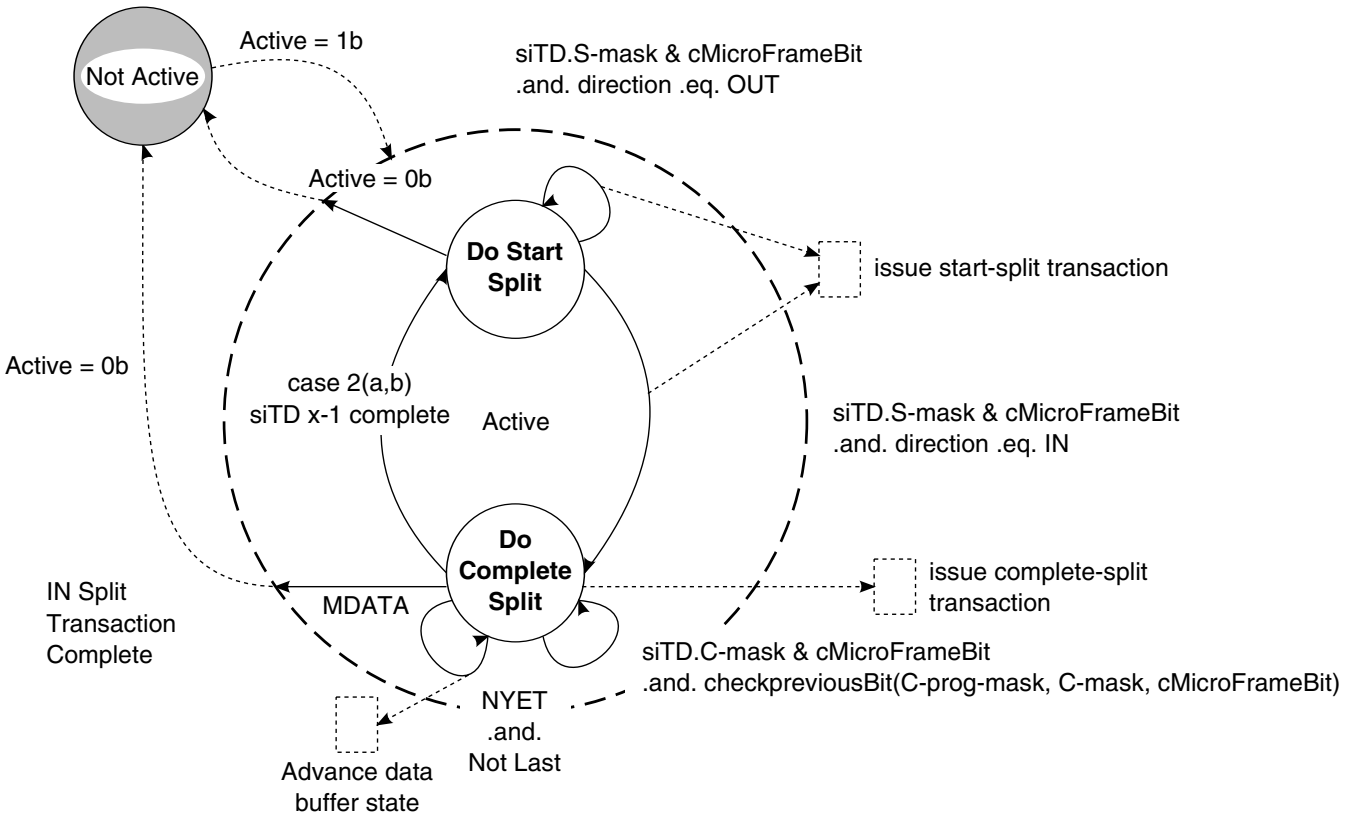


Figure 77-28. Split Transaction State Machine for Isochronous

77.4.3.12.3.4 Periodic Isochronous - Do Start Split

Isochronous split transaction OUTs use only this state. An siTD for a split-transaction isochronous IN is either initialized to this state, or the siTD transitions to this state from Do Complete Split when a case 2a (IN) or 2b scheduling boundary isochronous split-transaction completes.

Each time the host controller reaches an active siTD in this state, it checks the *siTD.S-mask* against *cMicroFrameBit*. If there is a one in the appropriate position, the siTD will execute a start-split transaction. By definition, the host controller cannot *reach* an siTD at the wrong time. If the *I/O* field indicates an IN, then the start-split transaction includes only the extended token plus the full-speed token. Software must initialize the *siTD.Total Bytes To Transfer* field to the number of bytes expected. This is usually the maximum packet size for the full-speed endpoint. The host controller exits this state when the start-split transaction is complete.

The remainder of this section is specific to an isochronous OUT endpoint (that is, the *I/O* field indicates an OUT). When the host controller executes a start-split transaction for an isochronous OUT it includes a data payload in the start-split transaction. The memory buffer address for the data payload is constructed by concatenating *siTD.Current Offset*

with the page pointer indicated by the page selector field (*siTD.P*). A zero in this field selects Page 0 and a 1 selects Page 1. During the start-split for an OUT, if the data transfer crosses a page boundary during the transaction, the host controller must detect the page cross, update the *siTD.P*-bit from a zero to a one, and begin using the *siTD.Page 1* with *siTD.Current Offset* as the memory address pointer. The field *siTD.TP* is used to annotate each start-split transaction with the indication of which part of the split-transaction data the current payload represents (ALL, BEGIN, MID, END). In all cases the host controller simply uses the value in *siTD.TP* to mark the start-split with the correct transaction position code.

T-Count is always initialized to the number of start-splits for the current frame. *TP* is always initialized to the first required transaction position identifier. The scheduling boundary case (see Figure 77-27) is used to determine the initial value of *TP*. The initial cases are summarized in the following table.

Table 77-48. Initial Conditions for OUT siTD's TP and T-count Fields

Case	T-count	TP	Description
1, 2a	=1	ALL	When the OUT data payload is less than (or equal to) 188 bytes, only one start-split is required to move the data. The one start-split must be marked with an ALL.
1, 2a	!=1	BEGIN	When the OUT data payload is greater than 188 bytes more than one start-split must be used to move the data. The initial start-split must be marked with a BEGIN.

After each start-split transaction is complete, the host controller updates *T-Count* and *TP* appropriately so that the next start-split is correctly annotated.

The table below illustrates all of the *TP* and *T-count* transitions, which must be accomplished by the host controller.

.

Table 77-49. Transaction Position (TP)/Transaction Count (T-Count) Transition Table

TP	T-count next	TP next	Description
ALL	0	N/A	Transition from ALL, to done.
BEGIN	1	END	Transition from BEGIN to END. Occurs when *T-count* starts at 2.
BEGIN	!=1	MID	Transition from BEGIN to MID. Occurs when *T-count* starts at greater than 2.
MID	!=1	MID	*TP* stays at MID while *T-count* is not equal to 1 (that is, greater than 1). This case can occur for any of the scheduling boundary cases where the *T-count* starts greater than 3.
MID	1	END	Transition from MID to END. This case can occur for any of the scheduling boundary cases where the *T-count* starts greater than 2.

The start-split transactions do not receive a handshake from the transaction translator, so the host controller always advances the transfer state in the siTD after the bus transaction is complete. To advance the transfer state the following operations take place:

- The *siTD.Total Bytes To Transfer* and the *siTD.Current Offset* fields are adjusted to reflect the number of bytes transferred.
- The *siTD.P* (page selector) bit is updated appropriately.
- The *siTD.TP* and *siTD.T-count* fields are updated appropriately as defined in Table 77-49.

These fields are then written back to the memory based siTD. The *S-mask* is fixed for the life of the current budget. As mentioned above, *TP* and *T-count* are set specifically in each siTD to reflect the data to be sent from this siTD. Therefore, regardless of the value of *S-mask*, the actual number of start-split transactions depends on *T-count* (or equivalently, *Total Bytes to Transfer*). The host controller must set the *Active* bit to a zero when it detects that all of the schedule data has been sent to the bus. The preferred method is to detect when *T-Count* decrements to zero as a result of a start-split bus transaction. Equivalently, the host controller can detect when *Total Bytes to Transfer* decrements to zero. Either implementation must ensure that if the initial condition is *Total Bytes to Transfer* equal to zero and *T-count* is equal to a one, then the host controller will issue a single start-split, with a zero-length data payload. Software must ensure that *TP*, *T-count* and *Total Bytes to Transfer* are set to deliver the appropriate number of bus transactions from each siTD. An inconsistent combination will yield undefined behavior.

If the host experiences hold-offs that cause the host controller to skip start-split transactions for an OUT transfer, the state of the transfer will not progress appropriately. The transaction translator will observe protocol violations in the arrival of the start-splits for the OUT endpoint (that is, the transaction position annotation will be incorrect as received by the transaction translator).

Example scenarios are described in Section Split Transaction for Isochronous - Processing Examples .

A host controller implementation can optionally track the progress of an OUT split transaction by setting appropriate bits in the *siTD.C-prog-mask* as it executes each scheduled start-split. The *checkPreviousBit*() algorithm defined in Periodic Isochronous - Do Complete Split can be used prior to executing each start-split to determine whether start-splits were skipped. The host controller can use this mechanism to detect missed micro-frames. It can then set the siTD's *Active* bit to zero and stop execution of this siTD. This saves on both memory and high-speed bus bandwidth.

77.4.3.12.3.5 Periodic Isochronous - Do Complete Split

This state is only used by a split-transaction isochronous IN endpoint. This state is entered unconditionally from the Do Start State after a start-split transaction is executed for an IN endpoint.

Each time the host controller visits an siTD in this state, it conducts a number of tests to determine whether it should execute a complete-split transaction. The individual tests are listed below. The sequence they are applied depends on which micro-frame the host controller is currently executing which means that the tests might not be applied until after the siTD referenced from the back pointer has been fetched.

- Test A. *cMicroFrameBit* is bit-wise anded with *siTD.C-mask* field. A non-zero result indicates that software scheduled a complete-split for this endpoint, during this micro-frame. This test is always applied to a newly fetched siTD that is in this state.
- Test B. The *siTD.C-prog-mask* bit vector is checked to determine whether the previous complete splits have been executed. An example algorithm is below (this is slightly different than the algorithm used in Section Periodic Isochronous - Do Complete Split). The sequence in which this test is applied depends on the current value of USB.FRINDEX[2:0]. If USB.FRINDEX[2:0] is 0 or 1, it is not applied until the back pointer has been used. Otherwise it is applied immediately.

```
Algorithm Boolean CheckPreviousBit(siTD.C-prog-mask, siTD.C-mask, cMicroFrameBit)
Begin
          Boolean rvalue = TRUE;
          previousBit = cMicroFrameBit rotate-right(1)
          -- Bit-wise anding previousBit with C-mask indicates whether there was an intent
          -- to send a complete split in the previous micro-frame. So, if the
          -- 'previous bit' is set in C-mask, check C-prog-mask to make sure it
          -- happened.
          if previousBit bitAND siTD.C-mask then
                    if not (previousBit bitAND siTD.C-prog-mask) then
                              rvalue = FALSE
                    End if
          End if
          Return rvalue
End Algorithm
```

If Test A is true and USB.FRINDEX[2:0] is zero or one, then this is a case 2a or 2b scheduling boundary (see Figure 77-26). See Section Periodic Isochronous - Do Complete Split for details in handling this condition.

If Test A and Test B evaluate to true, then the host controller will execute a complete-split transaction using the transfer state of the current siTD. When the host controller commits to executing the complete-split transaction, it updates *QH.C-prog-mask* by bit-ORing with *cMicroFrameBit*. The transfer state is advanced based on the completion status of the complete-split transaction. To advance the transfer state of an IN siTD, the host controller must:

- Decrement the number of bytes received from *siTD.Total Bytes To Transfer,*
- Adjust *siTD.Current Offset* by the number of bytes received,

- Adjust *siTD.P* (page selector) field if the transfer caused the host controller to use the next page pointer, and
- Set any appropriate bits in the *siTD.Status* field, depending on the results of the transaction.

Note that if the host controller encounters a condition where *siTD.Total Bytes To Transfer* is zero, and it receives more data, the host controller must not write the additional data to memory. The *siTD.Status.Active* bit must be set to zero and the *siTD.Status.Babble Detected* bit must be set to a one. The fields *siTD.Total Bytes To Transfer*, *siTD.Current Offset*, and *siTD.P* (page selector) are not required to be updated as a result of this transaction attempt.

The host controller must accept (assuming good data packet CRC and sufficient room in the buffer as indicated by the value of *siTD.Total Bytes To Transfer*) MDATA and DATA0/1 data payloads up to and including 192 bytes. A host controller implementation may optionally set *siTD.Status Active* to a zero and *siTD.Status.Babble Detected* to a one when it receives and MDATA or DATA0/1 with a data payload of more than 192 bytes. The following responses have the noted effects:

- ERR. The full-speed transaction completed with a time-out or bad CRC and this is a reflection of that error to the host. The host controller sets the *ERR* bit in the *siTD.Status* field and sets the *Active* bit to a zero.
- Transaction Error (XactErr). The complete-split transaction encounters a Timeout, CRC16 failure, etc. The *siTD.Status* field *XactErr* field is set to a one and the complete-split transaction must be retried immediately. The host controller must use an internal error counter to count the number of retries as a counter field is not provided in the siTD data structure. The host controller will not retry more than two times. If the host controller exhausts the retries or the end of the micro-frame occurs, the *Active* bit is set to zero.
- DATAx (0 or 1). This response signals that the final data for the split transaction has arrived. The transfer state of the siTD is advanced and the *Active* bit is set to a zero. If the *Bytes To Transfer* field has not decremented to zero (including the reception of the data payload in the DATAx response), then less data than was expected, or allowed for was actually received. This *short packet* event does not set the USBINT status bit in the USBSTS register to a one. The host controller will not detect this condition.
- NYET (and Last). On each NYET response, the host controller also checks to determine whether this is the last complete-split for this split transaction. Last was defined in Section Periodic Isochronous - Do Complete Split . If it is the last complete-split (with a NYET response), then the transfer state of the siTD is not advanced (never received any data) and the *Active* bit is set to a zero. No bits are set in the *Status* field because this is essentially a skipped transaction. The transaction translator must have responded to all the scheduled clompete-splits with NYETs

meaning that the start-split issued by the host controller was not received. This result should be interpreted by system software as if the transaction was completely skipped. The test for whether this is the last complete split can be performed by XORing C-mask with C-prog-mask. A zero result indicates that all complete-splits have been executed.

- MDATA (and Last). See above description for testing for Last. This can only occur when there is an error condition. Either there has been a babble condition on the full-speed link, which delayed the completion of the full-speed transaction, or software set up the *S-mask* and/or *C-mask*s incorrectly. The host controller must set *XactErr* bit to a one and the *Active* bit is set to a zero.
- NYET (and not Last). See above description for testing for Last. The complete-split transaction received a NYET response from the transaction translator. Do not update any transfer state (except for *C-prog-mask*) and stay in this state.
- MDATA (and not Last). The transaction translator responds with an MDATA when it has partial data for the split transaction. For example, the full-speed transaction data payload spans from micro-frame X to X+1 and during micro-frame X, the transaction translator will respond with an MDATA and the data accumulated up to the end of micro-frame X. The host controller advances the transfer state to reflect the number of bytes received.

If Test A succeeds, but Test B fails, it means that one or more of the complete-splits have been skipped. The host controller sets the *Missed Micro-Frame* status bit and sets the *Active* bit to a zero.

77.4.3.12.3.6 Complete-Split for Scheduling Boundary Cases 2a, 2b

Boundary cases 2a and 2b (INs only) (see Figure 77-26) require that the host controller use the transaction state context of the previous siTD to finish the split transaction. The table below enumerates the transaction state fields.

Table 77-50. Summary siTD Split Transaction State

Buffer State	Status	Execution Progress
Total Bytes To Transfer P (page select) Current Offset TP (transaction position) T-count (transaction count)	All bits in the status field	C-prog-mask

NOTE

TP and *T-count* are used only for Host to Device (OUT) endpoints.

If software has budgeted the schedule of this data stream with a frame wrap case, then it must initialize the *siTD.Back Pointer* field to reference a valid siTD and will have the *siTD.Back Pointer.T-bit* in the *siTD.Back Pointer*

field set to a zero. Otherwise, software must set the *siTD.Back Pointer.T-bit* in the *siTD.Back Pointer* field to a one. The host controller's rules for interpreting when to use the *siTD.Back Pointer* field are listed below. These rules apply only when the siTD's *Active* bit is a one and the *SplitXState* is Do Complete Split.

- When cMicroFrameBit is a 1h and the siTDX.Back Pointer.T-bit is a zero, or
- If cMicroFrameBit is a 2h and siTDX.S-mask[0] is a zero

When either of these conditions apply, then the host controller must use the transaction state from $siTD_{X-1}$.

In order to access $siTD_{X-1}$, the host controller reads on-chip the siTD referenced from $siTD_X$.*Back Pointer*.

The host controller must save the entire state from $siTD_X$ while processing $siTD_{X-1}$. This is to accommodate for case 2b processing. The host controller must not recursively walk the list of *siTD.Back Pointers*.

If $siTD_{X-1}$ is active (*Active* bit is a one and *SplitXStat* is Do Complete Split), then both Test A and Test B are applied as described above. If these criteria to execute a complete-split are met, the host controller executes the complete split and evaluates the results as described above. The transaction state (see Table 77-50) of $siTD_{X-1}$ is appropriately advanced based on the results and written back to memory. If the resultant state of $siTD_{X-1}$'s *Active* bit is a one, then the host controller returns to the context of $siTD_X$, and follows its next pointer to the next schedule item. No updates to $siTD_X$ are necessary.

If $siTD_{X-1}$ is active (*Active* bit is a one and *SplitXStat* is Do Start Split), then the host controller must set *Active* bit to a zero and *Missed Micro-Frame* status bit to a one and the resultant status written back to memory.

If $siTD_{X-1}$*'sActive* bit is a zero, (because it was zero when the host controller first visited $siTD_{X-1}$ via $siTD_X$*'s* back pointer, it transitioned to zero as a result of a detected error, or the results of $siTD_{X-1}$*'s* complete-split transaction transitioned it to zero), then the host controller returns to the context of $siTD_X$ and transitions its *SplitXState* to Do Start Split. The host controller then determines whether the case 2b start split boundary condition exists (that is, if *cMicroframeBit* is a 1b and $siTD_X$.*S-mask*[0] is a 1b). If this criterion is met the host controller immediately executes a start-split transaction and appropriately advances the transaction state of $siTD_X$, then follows $siTD_X$.*Next Pointer* to the next schedule item. If the criterion is not met, the host controller simply follows $siTD_X$.*Next Pointer* to the next schedule item. Note that in the case of a 2b boundary case, the split-transaction of $siTD_{X-1}$ will have its *Active* bit set to zero when the host controller returns

to the context of $siTD_X$. Also, note that software should not initialize an siTD with *C-mask* bits 0 and 1 set to a one and an *S-mask* with bit zero set to a one. This scheduling combination is not supported and the behavior of the host controller is undefined.

77.4.3.12.3.7 Split Transaction for Isochronous - Processing Examples

There is an important difference between how the hardware/software manages the isochronous split transaction state machine and how it manages the asynchronous and interrupt split transaction state machines. The asynchronous and interrupt split transaction state machines are encapsulated within a single queue head. The progress of the data stream depends on the progress of each split transaction. In some respects, the split-transaction state machine is sequenced via the Execute Transaction queue head traversal state machine (see Figure 77-19).

Isochronous is a pure time-oriented transaction/data stream. The interface data structures are optimized to efficiently describe transactions that need to occur at specific times. The isochronous split-transaction state machine must be managed across these time-oriented data structures. This means that system software must correctly describe the scheduling of split-transactions across more than one data structure.

Then the host controller must make the appropriate state transitions at the appropriate times, in the correct data structures.

For example, the table below illustrates a couple of frames worth of scheduling required to schedule a case 2a full-speed isochronous data stream.

Table 77-51. Example Case 2a - Software Scheduling siTDs for an IN Endpoint

siTDX		Micro-Frames								Initial
#	Masks	0	1	2	3	4	5	6	7	SplitXState
X	S-Mask	-	-	-	-	1	-	-	-	Do Start Split
	C-Mask	1	1	-	-	-	-	1	1	
X+1	S-Mask	-	-	-	-	1	-	-	-	Do Complete Split
	C-Mask	1	1					1	1	
X+2	S-Mask	-	-	-	-	1	-	-	-	Do Complete Split
	C-Mask	1	1					1	1	
X+3	S-Mask	Repeats previous pattern								Do Complete Split
	C-Mask									

This example shows the first three siTDs for the transaction stream. Because this is the case-2a frame-wrap case, *S-masks* of all siTDs for this endpoint have a value of 10h (a one bit in micro-frame 4) and *C-mask* value of C3h (one-bits in micro-frames 0,1, 6 and 7). Additionally, sofware ensures that the *Back Pointer* field of each siTD references the appropriate siTD data structure (and the *Back PointerT-bits* are set to zero).

The initial *SplitXState* of the first siTD is Do Start Split. The host controller will visit the first siTD eight times during frame X. The C-mask bits in micro-frames 0 and 1 are ignored because the state is Do Start Split. During micro-frame 4, the host controller determines that it can run a start-split (and does) and changes *SplitXState* to Do Complete Split. During micro-frames 6 and 7, the host controller executes complete-splits. Notice the siTD for frame X+1 has its *SplitXState* initialized to Do Complete Split. As the host controller continues to traverse the schedule during *H-Frame* X+1, it will visit the second siTD eight times. During micro-frames 0 and 1 it will detect that it must execute complete-splits.

During *H-Frame* X+1, micro-frame 0, the host controller detects that $siTD_{X+1}$'s *Back Pointer.T-bit* is a zero, saves the state of $siTD_{X+1}$ and fetches $siTD_X$. It executes the complete split transaction using the transaction state of $siTD_X$. If the $siTD_X$ split transaction is complete, siTD's *Active* bit is set to zero and results written back to $siTD_X$. The host controller retains the fact that $siTD_X$ is retired and transitions the *SplitXState* in the $siTD_{X+1}$ to Do Start Split. At this point, the host controller is prepared to execute the start-split for $siTD_{X+1}$ when it reaches micro-frame 4. If the split-transaction completes early (transaction-complete is defined in Section Periodic Isochronous - Do Complete Split), that is, before all the scheduled complete-splits have been executed, the host controller will transition $siTD_X$.*SplitXState* to Do Start Split early and naturally skip the remaining scheduled complete-split transactions. For this example, $siTD_{X+1}$ does not receive a DATA0 response until *H-Frame* X+2, micro-frame 1.

During *H-Frame X+2,* micro-frame 0, the host controller detects that $siTD_{X+2}$'s *Back Pointer.T-bit* is a zero, saves the state of $siTD_{X+2}$ and fetches $siTD_{X+1}$. As described above, it executes another split transaction, receives an MDATA response, updates the transfer state, but does not modify the *Active* bit. The host controller returns to the context of $siTD_{X+2}$, and traverses its next pointer without any state change updates to $siTD_{X+2}$. S

During *H-Frame X+2*, micro-frame 1, the host controller detects $siTD_{X+2}$'s *S-mask[0]* is a zero, saves the state of $siTD_{X+2}$ and fetches $siTD_{X+1}$. It executes another complete-split transaction, receives a DATA0 response, updates the transfer state and sets the *Active* bit to a zero. It returns to the state of $siTD_{X+2}$ and changes its *SplitXState* to Do Start Split. At this point, the host controller is prepared to execute start-splits for $siTD_{X+2}$ when it

reaches micro-frame 4. <TBD… describe how software detects that there was missing micro-frames (don't think we care about missing out micro-frames. There is enough residual state to identify than not all transactions were executed.).

77.4.3.13 Host Controller Pause

When the host controller's *HCHalted* bit in the USBSTS register is a zero, the host controller is sending SOF (Start OF Frame) packets down all enabled ports.

When the schedules are enabled, the EHCI host controller will access the schedules in main memory each micro-frame. This constant pinging of main memory is known to create ARM platform power management problems for mobile systems. Specifically, mobile systems aggressively manage the state of the ARM platform, based on recent history usage. In the more aggressive power saving modes, the ARM platform can disable its caches. Current PC architectures assume that bus-master accesses to main memory must be cache-coherent. So, when bus masters are busy touching memory, the ARM platform power management software can detect this activity over time and inhibit the transition of the ARM platform into its lowest power savings mode. USB controllers are bus-masters and the frequency at which they access their memory-based schedules keeps the ARM platform power management software from placing the ARM platform into its lowest power savings state.

USB Host controllers don't access main memory when they are suspended. However, there are a variety of reasons why placing the USB controllers into suspend won't work, but they are beyond the scope of this document. The base requirement is that the USB controller needs to be kept out of main memory, while at the same time, the USB bus is kept from going into suspend.

EHCI controllers provide a large-grained mechanism that can be manipulated by system software to change the memory access pattern of the host controller. System software can manipulate the schedule enable bits in the USBCMD register to turn on/off the scheduling traversal. A software heuristic can be applied to implement an on/off duty cycle that allows the USB to make reasonable progress and allow the ARM platform power management to get the ARM platform into its lowest power state. This method is not intended to be applied at all times to throttle USB, but should only be applied in very specific configurations and usage loads. For example, when only a keyboard or mouse is attached to the USB, the heuristic could detect times when the USB is attempting to move data only very infrequently and can adjust the duty cycle to allow the ARM platform to reach its low power state for longer periods of time. Similarly, it could detect increases in the USB load and adjust the duty cycle appropriately, even to the point where the schedules are never disabled. The assumption here is that the USB is moving data and the ARM platform will be required to process the data streams.

It is suggested that in order to provide a complete solution for the system, the companion host controllers should also provide a similar method to allow system software to inhibit the companion host controller from accessing its shared memory based data structures (schedule lists or otherwise).

77.4.3.14 Port Test Modes -Host Operational Model

EHCI host controllers must implement the port test modes Test J_State, Test K_State, Test_Packet, Test Force_Enable, and Test SE0_NAK as described in the USB Specification Revision 2.0. The system is only allowed to test ports that are owned by the EHCI controller (for example, *CF-bit* is a one and *PortOwner* bit is a zero).

System software is allowed to have at most one port in test mode at a time. Placing more than one port in test mode will yield undefined results. The required, per port test sequence is (assuming the *CF-bit* in the USB.CONFIGFLAG register is a one):

- Disable the periodic and asynchronous schedules by setting the *Asynchronous Schedule Enable* and *Periodic Schedule Enable* bits in the USBCMD register to a zero.
- Place all enabled root ports into the suspended state by setting the *Suspend* bit in each appropriate USB.PORTSC register to a one.
- Set the *Run/Stop* bit in the USBCMD register to a zero and wait for the *HCHalted* bit in the USBSTS register, to transition to a one. Note that an EHCI host controller implementation may optionally allow port testing with the *Run/Stop* bit set to a one. However, all host controllers must support port testing with *Run/Stop* set to a zero and *HCHalted* set to a one.
- Set the *Port Test Control* field in the port under test PORTSC register to the value corresponding to the desired test mode. If the selected test is Test_Force_Enable, then the *Run/Stop* bit in the USBCMD register must then be transitioned back to one, in order to enable transmission of SOFs out of the port under test.
- When the test is complete, system software must ensure the host controller is halted (*HCHalted* bit is a one) then it terminates and exits test mode by setting *HCReset* to a one.

77.4.3.15 Interrupts-Host Operational Model

The EHCI Host Controller hardware provides interrupt capability based on a number of sources.

There are several general groups of interrupt sources:

- Interrupts as a result of executing transactions from the schedule (success and error conditions),
- Host controller events (Port change events, etc.), and
- Host Controller error events

All transaction-based sources are maskable through the Host Controller's Interrupt Enable register (USBINTR, see Section Interrupt Enable Register (USB__USBINTR)). Additionally, individual transfer descriptors can be marked to generate an interrupt on completion. This section describes each interrupt source and the processing that occurs in response to the interrupt.

During normal operation, interrupts may be immediate or deferred until the next interrupt threshold occurs. The interrupt threshold is a tunable parameter via the *Interrupt Threshold Control* field in the USBCMD register. The value of this register controls when the host controller will generate an interrupt on behalf of normal transaction execution. When a transaction completes during an interrupt interval period, the interrupt signaling the completion of the transfer will not occur until the interrupt threshold occurs. For example, the default value is eight micro-frames. This means that the host controller will not generate interrupts any more frequently than once every eight micro-frames.

Section Host System Error details effects of a host system error.

If an interrupt has been scheduled to be generated for the current interrupt threshold interval, the interrupt is not signaled until after the status for the last complete transaction in the interval has been written back to host memory. This may sometimes result in the interrupt not being signaled until the next interrupt threshold.

Initial interrupt processing is the same, regardless of the reason for the interrupt. When an interrupt is signaled by the hardware, ARM platform control is transferred to host controller's USB interrupt handler. The precise mechanism to accomplish the transfer is OS specific. For this discussion it is just assumed that control is received. When the interrupt handler receives control, its first action is to reads the USBSTS (USB Status Register). It then acknowledges the interrupt by clearing all of the interrupt status bits by writing ones to these bit positions. The handler then determines whether the interrupt is due to schedule processing or some other event. After acknowledging the interrupt, the handler (via an OS-specific mechanism), schedules a deferred procedure call (DPC) which will execute later. The DPC routine processes the results of the schedule execution. The precise mechanisms used are beyond the scope of this document.

Note: the host controller is not required to de-assert a currently active interrupt condition when software sets the interrupt enables (in the USBINR register, see Section Interrupt Enable Register (USB__USBINTR)) to a zero. The only reliable method software should use for acknowledging an interrupt is by transitioning the appropriate status bits in the USBSTS register (Section USB Status Register (USB__USBSTS)) from a one to a zero.

77.4.3.15.1 Transfer/Transaction Based Interrupts

These interrupt sources are associated with transfer and transaction progress. They are all dependent on the next interrupt threshold.

77.4.3.15.1.1 Transaction Error

A transaction error is any error that caused the host controller to think that the transfer did not complete successfully.

The table below lists the events/responses that the host can observe as a result of a transaction. The effects of the error counter and interrupt status are summarized in the following paragraphs. Most of these errors set the *XactErr* status bit in the appropriate interface data structure.

There is a small set of protocol errors that relate only when executing a queue head and fit under the umbrella of a WRONG PID error that are significant to explicitly identify. When these errors occur, the *XactErr* status bit in the queue head is set and the *CErr* field is decremented. When the *PIDCode* indicates a SETUP, the following responses are protocol errors and result in *XactErr* bit being set to a one and the *CErr* field being decremented.

- *EPS* field indicates a high-speed device and it returns a Nak handshake to a SETUP.
- *EPS* field indicates a high-speed device and it returns a Nyet handshake to a SETUP.
- *EPS* field indicates a low- or full-speed device and the complete-split receives a Nak handshake.

Table 77-52. Summary of Transaction Errors

Event / Result	Queue Head/qTD/iTD/siTD Side-effects		USB Status Register (USBSTS)
	Cerr	Status Field	USBERRINT
CRC	-1	XactErr set to a one.	1[1]
Timeout	-1	XactErr set to a one.	1[1]
Bad PID[2]	-1	XactErr set to a one.	1[1]
Babble	N/A	Section Serial Bus Babble	1
Buffer Error	N/A	Section Data Buffer Error	

1. If occurs in a queue head, then *USBERRINT* is asserted only when *CErr* counts down from a one to a zero. In addition the queue is halted, see Halting a Queue Head .
2. The host controller received a response from the device, but it could not recognize the PID as a valid PID.

77.4.3.15.1.2 Serial Bus Babble

When a device transmits more data on the USB than the host controller is expecting for this transaction, it is defined to be babbling. In general, this is called a *Packet Babble*.

When a device sends more data than the *Maximum Length* number of bytes, the host controller sets the *Babble Detected* bit to a one and halts the endpoint if it is using a queue head (see Halting a Queue Head). *Maximum Length* is defined as the minimum of *Total Bytes to Transfer* and *Maximum Packet Size*. The *CErr* field is not decremented for a packet babble condition (only applies to queue heads). A babble condition also exists if IN transaction is in progress at High-speed EOF2 point. This is called a frame babble. A frame babble condition is recorded into the appropriate schedule data structure. In addition, the host controller must disable the port to which the frame babble is detected.

The *USBERRINT* bit in the USB.USBSTS register is set to a one and if the *USB Error Interrupt Enable* bit in the USB.USBINTR register is a one, then a hardware interrupt is signaled to the system at the next interrupt threshold. The host controller must never start an OUT transaction that will babble across a micro-frame EOF.

NOTE

When a host controller detects a data PID mismatch, it must either: disable the packet babble checking for the duration of the bus transaction or do packet babble checking based solely on *Maximum Packet Size*. The USB core specification defines the requirements on a data receiver when it receives a data PID mismatch (for example, expects a DATA0 and gets a DATA1 or visa-versa). In summary, it must ignore the received data and respond with an ACK handshake, in order to advance the transmitter's data sequence.

The EHCI interface allows System software to provide buffers for a Control, Bulk or Interrupt IN endpoint that are not an even multiple of the maximum packet size specified by the device. Whenever a device misses an ACK for an IN endpoint, the host and device are out of synchronization with respect to the progress of the data transfer. The host controller may have advanced the transfer to a buffer that is less than maximum packet size. The device will re-send its maximum packet size data packet, with the original data PID, in response to the next IN token. In order to properly manage the bus protocol, the host controller must disable the packet babble check when it observes the data PID mismatch.

77.4.3.15.1.3 Data Buffer Error

This event indicates that an overrun of incoming data or a underrun of outgoing data has occurred for this transaction. This would generally be caused by the host controller not being able to access required data buffers in memory within necessary latency requirements.

These conditions are not considered transaction errors, and do not effect the error count in the queue head. When these errors do occur, the host controller records the fact the error occurred by setting the *Data Buffer Error* bit in the queue head, iTD or siTD.

If the data buffer error occurs on a non-isochronous IN, the host controller will not issue a handshake to the endpoint. This will force the endpoint to resend the same data (and data toggle) in response to the next IN to the endpoint.

If the data buffer error occurs on an OUT, the host controller must corrupt the end of the packet so that it cannot be interpreted by the device as a good data packet. Simply truncating the packet is not considered acceptable. An acceptable implementation option is to 1's complement the CRC bytes and send them. There are other options suggested in the Transaction Translator section of the USB Specification Revision 2.0.

77.4.3.15.1.4 USB Interrupt (Interrupt on Completion (IOC))

Transfer Descriptors (iTDs, siTDs, and queue heads (qTDs)) contain a bit that can be set to cause an interrupt on their completion. The completion of the transfer associated with that schedule item causes the USB Interrupt (USBINT) bit in the USB.USBSTS register to be set to a one.

In addition, if a short packet is encountered on an IN transaction associated with a queue head, then this event also causes USBINT to be set to a one. If the USB Interrupt Enable bit in the USB.USBINTR register is set to a one, a hardware interrupt is signaled to the system at the next interrupt threshold. If the completion is because of errors, the *USBERRINT* bit in the USB.USBSTS register is also set to a one.

77.4.3.15.1.5 Short Packet

Reception of a data packet that is less than the endpoint's Max Packet size during Control, Bulk or Interrupt transfers signals the completion of the transfer. Whenever a short packet completion occurs during a queue head execution, the *USBINT* bit in the USB.USBSTS register is set to a one.

If the *USB Interrupt Enable* bit is set in the USB.USBINTR register, a hardware interrupt is signaled to the system at the next interrupt threshold.

77.4.3.15.2 Host Controller Event Interrupts

These interrupt sources are independent of the interrupt threshold (with the one exception being the Interrupt on Async Advance, see Section Interrupt on Async Advance).

77.4.3.15.2.1 Port Change Events

Port registers contain status and status change bits. When the status change bits are set to a one, the host controller sets the *Port Change Detect* bit in the USBSTS register to a one.

If the *Port Change Interrupt Enable* bit in the USB.USBINTR register is a one, then the host controller will issue a hardware interrupt. The port status change bits include:

- Connect Status Change
- Port Enable/Disable Change
- Over-current Change
- Force Port Resume

77.4.3.15.2.2 Frame List Rollover

This event indicates that the host controller has wrapped the frame list. The current programmed size of the frame list effects how often this interrupt occurs.

If the frame list size is 1024, then the interrupt will occur every 1024 milliseconds, if it is 512, then it will occur every 512 milliseconds, etc. When a frame list rollover is detected, the host controller sets the *Frame List Rollover* bit in the USB.USBSTS register to a one. If the *Frame List Rollover Enable* bit in the USB.USBINTR register is set to a one, the host controller issues a hardware interrupt. This interrupt is not delayed to the next interrupt threshold.

77.4.3.15.2.3 Interrupt on Async Advance

This event is used for deterministic removal of queue heads from the asynchronous schedule. Whenever the host controller advances the on-chip context of the asynchronous schedule, it evaluates the value of the *Interrupt on Async Advance Doorbell* bit in the USB.USBCMD register.

If it is a one, it sets the *Interrupt on Async Advance* bit in the USB.USBSTS register to a one. If the *Interrupton Async Advance Enable* bit in the USB.USBINTR register is a one, the host controller issues a hardware interrupt at the next interrupt threshold. A detailed explanation of this feature is described in Section Removing Queue Heads from Asynchronous Schedule .

77.4.3.15.2.4 Host System Error

The host controller is a bus master and any interaction between the host controller and the system may experience errors. The type of host error may be catastrophic to the host controller (such as a Master Abort) making it impossible for the host controller to continue in a coherent fashion.

In the presence of non-catastrophic host errors, such as parity errors, the host controller could potentially continue operation. The recommended behavior for these types of errors is to escalate it to a catastrophic error and halt the host controller. Host-based error must result in the following actions:

- The *Run/Stop* bit in the USB.USBCMD register is set to a zero.
- The following bits in the USB.USBSTS register are set:
  - *Host System Error* bit is to a one.
  - *HCHalted* bit is set to a one.
- If the *Host System Error Enable* bit in the USB.USBINTR register is a one, then the host controller will issue a hardware interrupt. This interrupt is not delayed to the next interrupt threshold. The following table summarizes the required actions taken on the various host errors.

Table 77-53. Summary Behavior of EHCI Host Controller on Host System Errors

Cycle Type	Master Abort	Target Abort	Data Phase Parity
Frame list pointer fetch (read)	Fatal	Fatal	Fatal [o]
siTD fetch (read)	Fatal	Fatal	Fatal [o]
siTD status write-back (write)	Fatal [o]	Fatal [o]	Fatal [o]
iTD fetch (read)	Fatal	Fatal	Fatal [o]
iTD status write-back (write)	Fatal [o]	Fatal [o]	Fatal [o]
qTD fetch (read)	Fatal	Fatal	Fatal [o]
qHD status write-back (write)	Fatal [o]	Fatal [o]	Fatal [o]
Data write	Fatal [o]	Fatal [o]	Fatal [o]
Data read	Fatal	Fatal	Fatal [o]

Potentially, a host controller implementation could continue operation without a halt. However, the recommended behavior is to halt the host controller.

NOTE

After a *Host System Error*, Software must reset the host controller through *HCReset* in the USB.USBCMD register before re-initializing and restarting the host controller.

77.4.4 EHCI Deviation

For the purposes a dual-role Host/Device controller with support for On-The-Go applications, it is necessary to deviate from the EHCI specificationEnhanced Host Controller Interface Specification for Universal Serial Bus, Revision 0.95, November 2000, Intel Corporation. http://www.intel.com. Device operation & On-The-Go operation is not specified in the EHCI and thus the implementation supported in this core is proprietary. The host mode operation of the core is near EHCI compatible with few minor differences documented in this section.

The particulars of the deviations occur in the areas summarized here:

- Embedded Transaction Translator - Allows direct attachment of FS and LS devices in host mode without the need for a companion controller.
- Device operation - In host mode the device operational registers are generally disabled and thus device mode is mostly transparent when in host mode. However, there are a couple exceptions documented in the following sections.
- Embedded design interface - This core does not a PCI Interface and therefore the PCI configuration registers described in the EHCI specification are not applicable.
- On-The-Go Operation - This design includes an On-The-Go controller for Port #1.

77.4.4.1 Embedded Transaction Translator Function

The USB-HS OTG High-Speed USB On-The-Go OTG controller supports directly connected full and low speed devices without requiring a companion controller by including the capabilities of a USB 2.0 high speed hub transaction translator.

Although there is no separate Transaction Translator block in the system, the transaction translator function normally associated with a high speed hub has been implemented within the DMA and Protocol engine blocks. The embedded transaction translator function is an extension to EHCI interface, but makes use of the standard data structures and operational models that exist in the EHCI specification to support full and low speed devices.

77.4.4.1.1 Capability Registers

The following additions have been added to the capability registers to support the embedded Transaction Translator Function:

- N_TT added to USB.HCSPARAMS - Host Control Structural Parameters
- N_PTT added to USB.HCSPARAMS - Host Control Structural Parameters

77.4.4.1.2 Operational Registers

The following additions have been added to the operational registers to support the embedded TT:

- is a new register.
- Addition of two-bit Port Speed (PSPD) to the Port Status & Control (USB__PORTSC1) register.

77.4.4.1.3 Discovery-EHCI Deviation

In a standard EHCI controller design, the EHCI host controller driver detects a Full speed (FS) or Low speed (LS) device by noting if the port enable bit is set after the port reset operation.

The port enable will only be set in a standard EHCI controller implementation after the port reset operation and when the host and device negotiate a High-Speed connection (that is, Chirp completes successfully).

Because this controller has an embedded Transaction Translator, the port enable will always be set after the port reset operation regardless of the result of the host device chirp result and the resulting port speed will be indicated by the PSPD field in USB.PORTSCx.

Therefore, the standard EHCI host controller driver requires an alteration to handle directly connected Full and Low speed devices or hubs.

The change is a fundamental one in that is summarized in the following table.

Table 77-54. Summary of EHCI

Standard EHCI	EHCI with embedded Transaction Translator
After port enable bit is set following a connection and reset sequence, the device/hub is assumed to be HS.	After port enable bit is set following a connection and reset sequence, the device/hub speed is noted from USB.PORTSCx.
FS and LS devices are assumed to be downstream from a HS hub thus, all port-level control is performed through the Hub Class to the nearest Hub.	FS and LS device can be either downstream from a HS hub or directly attached. When the FS/LS device is downstream from a HS hub, then port-level control is done using the Hub Class through the nearest Hub. When a FS/LS device is directly attached, then port-level control is accomplished using USB.PORTSCx.
FS and LS devices are assumed to be downstream from a HS hub with HubAddr=X. [where HubAddr > 0 and HubAddr is the address of the Hub where the bus transitions from HS to FS/LS (ie. Split target hub)]	FS and LS device can be either downstream from a HS hub with HubAddr = X [HubAddr > 0] or directly attached [where HubAddr = 0 and HubAddr is the address of the Root Hub where the bus transitions from HS to FS/LS (ie. Split target hub is the root hub)]

77.4.4.1.4 Data Structures

The same data structures used for FS/LS transactions though a HS hub are also used for transactions through the Root Hub with sm embedded Transaction Translator.

Here it is demonstrated how the Hub Address and Endpoint Speed fields should be set for directly attached FS/LS devices and hubs:

1. QH (for direct attach FS/LS) - Async. (Bulk/Control Endpoints) Periodic (Interrupt)
   - Hub Address = 0
   - Transactions to direct attached device/hub.
     - QH.EPS = Port Speed
   - Transactions to a device downstream from direct attached FS hub.
     - QH.EPS = Downstream Device Speed

NOTE

When QH.EPS = 01 (LS) and PORTSCx.PSPD = 00 (FS), a LS-pre-pid will be sent before the transmitting LS traffic.

Maximum Packet Size must be less than or equal 64 or undefined behaviour may result.

2. siTD (for direct attach FS) - Periodic (ISO Endpoint)
   - All FS ISO transactions:
     - Hub Address = 0
     - siTD.EPS = 00 (full speed)
       - Maximum Packet Size must less than or equal to 1023 or undefined behaviour may result.

77.4.4.1.5 Operational Model

The operational models are well defined for the behavior of the Transaction Translator (see USB 2.0 specificationUniversal Serial Bus Specification, Revision 2.0, April 2000, Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips. http://www.usb.org) and for the EHCI controller moving packets between system memory and a USB-HS hub. Because the embedded Transaction Translator exists within the host controller there is no physical bus between EHCI host controller driver and the USB FS/LS bus. These sections will briefly discuss the operational model for how the EHCI and Transaction Translator operational models are combined without the physical bus between. The following sections assume the reader is familiar with both the EHCI and USB 2.0 Transaction Translator operational models.

77.4.4.1.5.1 Micro- frame Pipeline

The EHCI operational model uses the concept of H-frames and B-frames to describe the pipeline between the Host (H) and the Bus (B). The embedded Transaction Translator shall use the same pipeline algorithms specified in the USB 2.0 specification for a Hub-based Transaction Translator.

All periodic transfers always begin at B-frame 0 (after SOF) and continue until the stored periodic transfers are complete. As an example of the micro-frame pipeline implemented in the embedded Transaction Translator, all periodic transfers that are tagged in EHCI to execute in H-frame 0 will be ready to execute on the bus in B-frame 0.

It is important to note that when programming the S-mask and C-masks in the EHCI data structures to schedule periodic transfers for the embedded Transaction Translator, the EHCI host controller driver must follow the same rules specified in EHCI for programming the S-mask and C-mask for downstream Hub-based Transaction Translators.

Once periodic transfers are exhausted, any stored asynchronous transfer will be moved. Asynchronous transfers are opportunistic in that they shall execute whenever possible and their operation is not tied to H-frame and B-frame boundaries with the exception that an asynchronous transfer can not babble through the SOF (start of B-frame 0.)

77.4.4.1.5.2 Split State Machines

The start and complete split operational model differs from EHCI slightly because there is no bus medium between the EHCI controller and the embedded Transaction Translator.

Where a start or complete-split operation would occur by requesting the split to the HS hub, the start/complete split operation is simple an internal operation to the embedded Transaction Translator. The following table summarizes the conditions where handshakes are emulated from internal state instead of actual handshakes to HS split bus traffic.

Table 77-55. Summary of the Conditons of Handshakes[1]

Condition	Emulate TT Response
Start-Split: All asynchronous buffers full.	NAK
Start-Split: All periodic buffers full.	ERR
Start-Split: Success for start of Async. Transaction.	ACK
Start-Split: Start Periodic Transaction.	No Handshake (Ok)
Complete-Split: Failed to find transaction in queue.	Bus Time Out
Complete-Split: Transaction in Queue is Busy.	NYET

Table continues on the next page...

Table 77-55. Summary of the Conditons of Handshakes[1] (continued)

Complete-Split: Transaction in Queue is Complete.	[Actual Handshake from LS/FS device]

1. The un-shaded cells represent Start-Splits and the shaded cells represent Complete-Splits

77.4.4.1.5.3 Asynchronous Transaction Scheduling and Buffer Management

The following USB 2.0 specification items are implemented in the embedded Transaction Translator:

77.4.4.1.5.4 USB 2.0 - 11.17.3

- Sequencing is provided & a packet length estimator ensures no full-speed/low-speed packet babbles into SOF time.

77.4.4.1.5.5 USB 2.0 - 11.17.4

- Transaction tracking for 2 data pipes.

77.4.4.1.5.6 Periodic Transaction Scheduling and Buffer Management

The following USB 2.0 specification items are implemented in the embedded Transaction Translator:

77.4.4.1.5.7 USB 2.0 - 11.18.6.[1-2]

- Abort of pending start-splits
  - EOF (and not started in micro-frames 6)
  - Idle for more than 4 micro-frames
- Abort of pending complete-splits
  - EOF
  - Idle for more than 4 micro-frames

77.4.4.1.5.8 USB 2.0 - 11.18.[7-8]

- Transaction tracking for up to 16 data pipes.
- Some applications may not require transaction tracking up to a maximum of 16 periodic data pipes. The option to limit the tracking to only 4 periodic data pipes exists in the by changing the configuration constant VUSB_HS_TT_PERIODIC_CONTEXTS to 4. The result is a significant gate count savings to the core given the limitations implied.

NOTE

Limiting the number of tracking pipes in the EMBedded -TT to four (4) will impose the restriction that no more than 4 pe

transactions (INTERRUPT/ISOCHRONOUS) can be scheduled through the embedded-tt per frame. the number 16 was chosen in the USB specification because it is sufficient to ensure that the high-speed to full-speed periodic pipeline can remain full. keeping the pipeline full puts no constraint on the number of periodic transactions that can be scheduled in a frame and the only limit becomes the flight time of the packets on the bus.

- Complete-split transaction searching.

NOTE

There is no data schedule mechanism for these transactions other than the micro-frame pipeline. The embedded TT assumes the number of packets scheduled in a frame does not exceed the frame duration (1 ms) or else undefined behavior may result.

77.4.4.1.5.9 Multiple Transaction Translators

The maximum number of embedded Transaction Translators that is currently supported is one as indicated by the N_TT field in the Host Controller Structural Parameters (USB__HCSPARAMS) register.

77.4.4.2 Device Operation

The co-existence of a device operational controller within the host controller has little effect on EHCI compatibility for host operation except as noted in this section.

77.4.4.3 USB.USBMODE Register

Given that the dual-role controller is initialized in neither host nor device mode, the USB Device Mode (USB__USBMODE) register must be programmed for host operation before the EHCI host controller driver can begin EHCI host operations.

77.4.4.3.1 Non-Zero Fields the Register File

Some of the reserved fields and reserved addresses in the capability registers and operational register have use in device mode, the following must be adhered to:

- Write operations to all EHCI reserved fields (some of which are device fields) with the operation registers should always be written to zero. This is an EHCI requirement of the device controller driver that must be adhered to.
- Read operations by the host controller must properly mask EHCI reserved fields (some of which are device fields) because fields that are used exclusive for device are undefined in host mode .

77.4.4.3.2 SOF Interrupt

This SOF Interrupt used for device mode is shared as a free running 125us interrupt for host mode.

EHCI does not specify this interrupt but it has been added for convenience and as a potential software time base. See USB Status Register (USB__USBSTS) and Interrupt Enable Register (USB__USBINTR) registers.

77.4.4.4 Embedded Design Interface

This is an Embedded USB Host Controller as defined by the EHCI specification and thus does not implement the PCI configuration registers.

77.4.4.4.1 Frame Adjust Register

Given that the optional PCI configuration registers are not included in this implementation, there is no corresponding bit level timing adjustments like is provided by the Frame Adjust register in the PCI configuration registers. Starts of micro-frames are timed precisely to 125 us using the transceiver clock as a reference clock. That is, a 60 Mhz transceiver clock for 8-bit physical interfaces & full-speed serial interfaces or 30 Mhz transceiver clock for 16-bit physical interfaces.

77.4.4.5 Miscellaneous variations from EHCI

77.4.4.5.1 Programmable Physical Interface Behaviour

This design supports multiple Physical interfaces which can operate in differing modes when the core is configured with software programmable Physical Interface Modes.

Software programmability allows the selection of the Physical interface part during the board design phase instead of during the chip design phase.

The control bits for selecting the Physical Interface operating mode have been added to the Port Status & Control (USB__PORTSC1) register providing a capability that is not defined by EHCI.

77.4.4.5.2 Discovery

77.4.4.5.2.1 Port Reset

The port connect methods specified by EHCI require setting the port reset bit in the Port Status & Control (USB__PORTSC1) register for a duration of 10ms. Due to the complexity required to support the attachment of devices that are not high speed there are counter already present in the design that can count the 10ms reset pulse to alleviate the requirement of the software to measure this duration. Therefore, the basic connection is then summarized as the following:

- [Port Change Interrupt] Port connect change occurs to notify the host controller driver that a device has attached.
- Software shall write a '1' to the reset the device.
- Software shall write a '0' to the reset the device after 10 ms.
  - This step, which is necessary in a standard EHCI design, may be omitted with this implementation. Should the EHCI host controller driver attempt to write a '0' to the reset bit while a reset is in progress the write will simple be ignored and the reset will continue until completion.
- [Port Change Interrupt] Port enable change occurs to notify the host controller that the device in now operational and at this point the port speed has been determined.

77.4.4.5.2.2 Port Speed Detection

After the port change interrupt indicates that a port is enabled, the EHCI stack should determine the port speed. Unlike the EHCI implementation, which will re-assign the port owner for any device that does not connect at High-Speed, this host controller supports direct attach of non High-Speed devices.

Therefore, the following differences are important regarding port speed detection:

- Port Owner is read-only and always reads 0.
- A 2-bit Port Speed indicator has been added to PORTSC to provide the current operating speed of the port to the host controller driver.
- A 1-bit High Speed indicator has been added to PORTSC to signify that the port is in High-Speed vs. Full/Low Speed - *This information is redundant with the 2-bit Port Speed indicator above.*

77.4.4.5.3 Port Test Mode

Port Test Control mode behaves fully as described in EHCI since the release of revision 3.2.1. In earlier product revisions, the test packet mode was not EHCI compatible. An alternate host controller driver procedure is no longer necessary or supported.

77.4.5 Device Data Structures

This section defines the interface data structures used to communicate control, status, and data between Device Controller Driver (DCD) Software and the Device Controller.

The data structure definitions in this chapter support a 32-bit memory buffer address space. The interface consists of device Queue Heads and Transfer Descriptors.

NOTE

Software must ensure that no interface data structure reachable by the Device Controller spans a 4K-page boundary.

The data structures defined in the chapter are (from the device controller's perspective) a mix of read-only and read/ writable fields. The device controller must preserve the read-only fields on all data structure writes.

The USB-HS OTG High-Speed USB On-The-Go core includes DCD Software called the USB 2.0 Device API. The Device API provides an easy to use Application Program Interface for developing device (peripheral) applications using the USB-HS OTG High-Speed USB On-The-Go core. The Device API incorporates and abstracts for the application developer all of the elements of the program interface.

The figure below shows the organization of the End Point Queue Head.

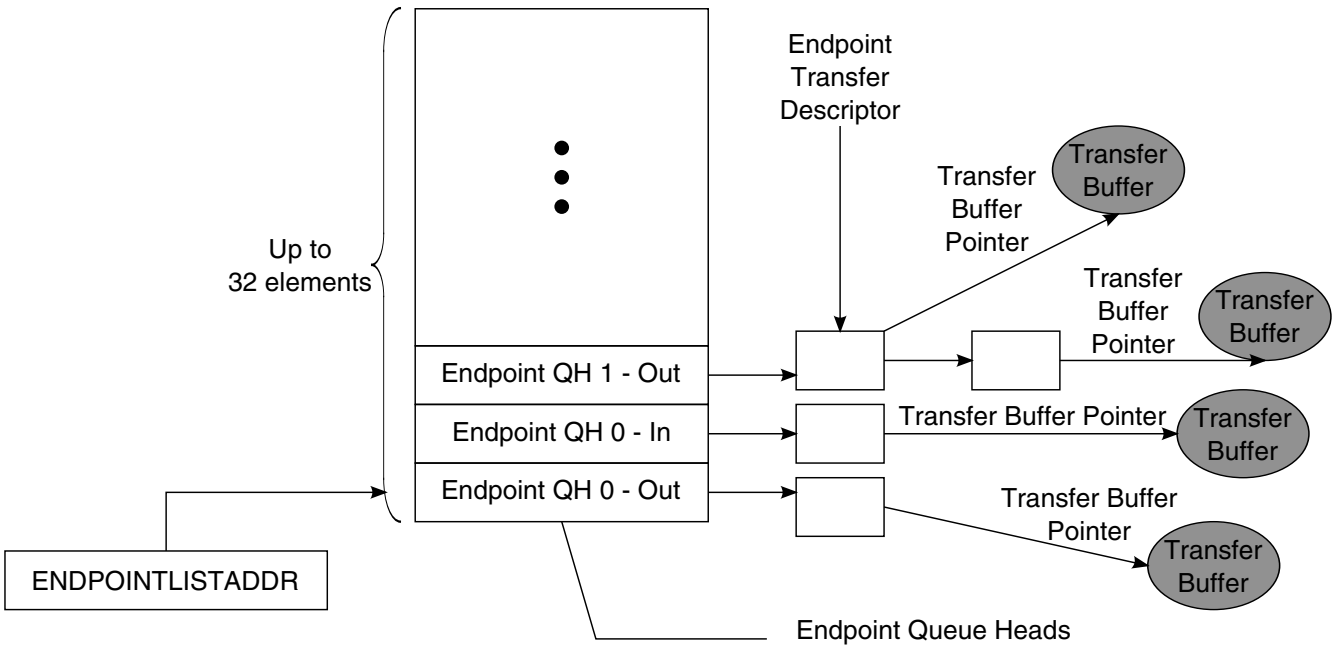


Figure 77-29. End Point Queue Head Organization

Device queue heads are arranged in an array in a continuous area of memory pointed to by the USB.ENDPOINTLISTADDR pointer. The even -numbered device queue heads in the list support receive endpoints (OUT/SETUP) and the odd-numbered queue heads in the list are used for transmit endpoints (IN/INTERRUPT). The device controller will index into this array based upon the endpoint number received from the USB bus. All information necessary to respond to transactions for all primed transfers is contained in this list so the Device Controller can readily respond to incoming requests without having to traverse a linked list.

NOTE

The Endpoint Queue Head List must be aligned to a 2k boundary.

77.4.5.1 Endpoint Queue Head (dQH)

The device Endpoint Queue Head (dQH) is where all transfers are managed. The dQH is a 48-byte data structure, but must be aligned on 64-byte boundaries.

During priming of an endpoint, the dTD (device transfer descriptor) is copied into the overlay area of the dQH, which starts at the nextTD pointer DWord and continues through the end of the buffer pointers DWords. After a transfer is complete, the dTD

status DWord is updated in the dTD pointed to by the currentTD pointer. While a packet is in progress, the overlay area of the dQH is used as a staging area for the dTD so that the Device Controller can access needed information with little minimal latency.

Table 77-56. Endpoint Queue Head (dQH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Mult			zlt	0	Maximum Packet Length											ios	0															
Current dTD Pointer																											0					
Next dTD Pointer																											0				T	
0	Total Bytes															ioc	0			Mult O		0			Status								
Buffer Pointer (Page 0)																				Current Offset												
Buffer Pointer (Page 1)																				-												
Buffer Pointer (Page 2)																				-												
Buffer Pointer (Page 3)																				-												
Buffer Pointer (Page 4)																				-												
-																																
Set-up Buffer Bytes 3…0																																
Set-up Buffer Bytes 7…4																																

	Device Controller Read/Write		Device Controller Read Only.

77.4.5.1.1 Endpoint Capabilities/Characteristics

This DWord specifies static information about the endpoint, in other words, this information does not change over the lifetime of the endpoint. Device Controller software should not attempt to modify this information while the corresponding endpoint is enabled.

Table 77-57 describes the endpoint capabilities.

Table 77-57. Endpoint Capabilities/Characteristics

Bit	Description

Table continues on the next page...

Table 77-57. Endpoint Capabilities/Characteristics (continued)

31-30	Mult. This field is used to indicate the number of packets executed per transaction description as given by the following: 00 - Execute N Transactions as demonstrated by the USB variable length packet protocol where N is computed using the Maximum Packet Length (dQH) and the Total Bytes field (dTD) 01 Execute 1 Transaction. 10 Execute 2 Transactions. 11 Execute 3 Transactions. **NOTE:** Non-ISO endpoints must set Mult="00". ISO endpoints must set Mult="01", "10", or "11" as needed.
29	Zero Length Termination Select. This bit is used to indicate when a zero length packet is used to terminate transfers where to total transfer length is a multiple . This bit is not relevant for Isochronous 0 - Enable zero length packet to terminate transfers equal to a multiple of the Maximum Packet Length. (default). 1 - Disable the zero length packet on transfers that are equal in length to a multiple Maximum Packet Length.
28-27	Reserved. These bit reserved for future use and should be set to zero.
26-16	Maximum Packet Length. This directly corresponds to the maximum packet size of the associated endpoint (wMaxPacketSize). The maximum value this field may contain is 0x400 (1024).
15	Interrupt On Setup (IOS). This bit is used on control type endpoints to indicate if USBINT is set in response to a setup being received.
14-0	Reserved. Bits reserved for future use and should be set to zero.

77.4.5.1.2 Transfer Overlay-Endpoint Queue Head

The seven DWords in the overlay area represent a transaction working space for the device controller.

The general operational model is that the device controller can detect whether the overlay area contains a description of an active transfer. If it does not contain an active transfer, then it will not read the associated endpoint.

After an endpoint is readied, the dTD will be copied into this queue head overlay area by the device controller. Until a transfer is expired, software must not write the queue head overlay area or the associated transfer descriptor. When the transfer is complete, the device controller will write the results back to the original transfer descriptor and advance the queue.

See dTD for a description of the overlay fields.

77.4.5.1.3 Current dTD Pointer

The current dTD pointer is used by the device controller to locate the transfer in progress. This word is for Device Controller (hardware) use only and should not be modified by DCD software.

The following table describes the dTD Pointer.

Table 77-58. Next dTD Pointer

Bit	Description
31-5	Current dTD. This field is a pointer to the dTD that is represented in the transfer overlay area. This field will be modified by the Device Controller to next dTD pointer during endpoint priming or queue advance.
4-0	Reserved. Bit reserved for future use and should be set to zero.

77.4.5.1.4 Set-up Buffer

The set-up buffer is dedicated storage for the 8-byte data that follows a set-up PID.

NOTE

Each endpoint has a TX and an RX dQH associated with it, and only the RX queue head is used for receiving setup data packets.

The following table describes the Multiple Mode Control.

Table 77-59. Multiple Mode Control (HCCPARAMS)

DWord	Bits	Description
1	31-0	Setup Buffer 0. This buffer contains bytes 3 to 0 of an incoming setup buffer packet and is written by the device controller to be read by software.
2	31-0	Setup Buffer 1. This buffer contains bytes 7 to 4 of an incoming setup buffer packet and is written by the device controller to be read by software.

77.4.5.2 Endpoint Transfer Descriptor (dTD)

The dTD describes to the device controller the location and quantity of data to be sent/ received for given transfer.

The DCD should not attempt to modify any field in an active dTD except the Next Like Pointer, which should only be modified as described in section Managing Transfers with Transfer Descriptors.

Table below shows the Endpoint Transfer Descriptor (dTD).

Table 77-60. Endpoint Transfer Descriptor (dTD)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Next Link Pointer																											0				T	
0	Total Bytes															ioc	0			MultO		0		Status								

Table continues on the next page...

Table 77-60. Endpoint Transfer Descriptor (dTD) (continued)

Buffer Pointer (Page 0)	Current Offset	
Buffer Pointer (Page 1)	0 Frame Number	
Buffer Pointer (Page 2)	-	
Buffer Pointer (Page 3)	-	
Buffer Pointer (Page 4)	Reserved	

	Device Controller Read/Write		Device Controller Read Only.

The following table describes the dTD Pointer.

Table 77-61. Next dTD Pointer

Bit	Description
31-5	Next Transfer Element Pointer. This field contains the physical memory address of the next dTD to be processed. The field corresponds to memory address signals [31:5], respectively.
4-1	Reserved. Bits reserved for future use and should be set to zero.
0	Terminate (T). 1=pointer is invalid. 0=Pointer is valid (points to a valid Transfer Element Descriptor). This bit indicates to the Device Controller that there are no more valid entries in the queue.

The following table describes the dTD Token.

Table 77-62. dTD Token

Bit	Description
31	Reserved. Bit reserved for future use and should be set to zero.
30-16	Total Bytes. This field specifies the total number of bytes to be moved with this transfer descriptor. This field is decremented by the number of bytes actually moved during the transaction and only on the successful completion of the transaction. The maximum value software may store in the field is 5*4K(5000H). This is the maximum number of bytes 5 page pointers can access. Although it is possible to create a transfer up to 20K this assumes the 1st offset into the first page is 0. When the offset cannot be predetermined, crossing past the 5th page can be guaranteed by limiting the total bytes to 16K**. Therefore, the maximum recommended transfer is 16K(4000H). If the value of the field is zero when the host controller fetches this transfer descriptor (and the active bit is set), the device controller executes a zero-length transaction and retires the transfer descriptor. It is not a requirement for IN transfers that Total Bytes To Transfer be an even multiple of *Maximum Packet Length*. If software builds such a transfer descriptor for an IN transfer, the last transaction will always be less that *Maximum Packet Length*.
15	Interrupt On Complete (IOC). This bit is used to indicate if USBINT is to be set in response to device controller being finished with this dTD.
14-12	Reserved. Bits reserved for future use and should be set to zero.

Table continues on the next page...

Table 77-62. dTD Token (continued)

11-10	Multiplier Override (MultO). This field can be used for transmit ISO's (ie. ISO-IN) to override the multiplier in the QH. This field must be zero for all packet types that are not transmit-ISO. Example: if QH.multiplier = 3; Maximum packet size = 8; Total Bytes = 15; MultiO = 0 [default] Three packets are sent: {Data2(8); Data1(7); Data0(0)} if QH.multiplier = 3; Maximum packet size = 8; Total Bytes = 15; MultiO = 2 Two packets are sent: {Data1(8); Data0(7)} For maximal efficiency, software should compute MultO = greatest integer of (Total Bytes / Max. Packet Size) except for the case when Total Bytes = 0; then MultO should be 1. Note: Non-ISO and Non-TX endpoints must set MultO="00".
9-8	Reserved. Bits reserved for future use and should be set to zero.
7-0	Status. This field is used by the Device Controller to communicate individual command execution states back to the Device Controller software. This field contains the status of the last transaction performed on this qTD. The bit encodings are: Bit Status Field Description 7 Active. 6 Halted. 5 Data Buffer Error. 3 Transaction Error. 4,2,0 Reserved.

The table below describes the dTD Buffer Page Pointer List.

Table 77-63. dTD Buffer Page Pointer List

Bit	Description
31-12	Buffer Pointer. Selects the page offset in memory for the packet buffer. Non virtual memory systems will typically set the buffer pointers to a series of incrementing integers.
0,11-0	Current Offset. Offset into the 4kb buffer where the packet is to begin.
1,10-0	Frame Number. Written by the device controller to indicate the frame number in which a packet finishes. This is typically be used to correlate relative completion times of packets on an ISO endpoint.

77.4.6 Device Operational Model

The function of the device operation is to transfer a request in the memory image to and from the Universal Serial Bus.

Using a set of linked list transfer descriptors, pointed to by a queue head, the device controller will perform the data transfers. The following sections explain the use of the device controller from the device controller driver (DCD) point-of-view and further describe how specific USB bus events relate to status changes in the device controller programmer's interface.

77.4.6.1 Device Controller Initialization

After hardware reset, the device is disabled until the Run/Stop bit is set to a '1'. In the disabled state, the pull-up on the USB D+ is not active which prevents an attach event from occurring. At a minimum, it is necessary to have the queue heads setup for endpoint zero before the device attach occurs.

Shortly after the device is enabled, a USB reset will occur followed by setup packet arriving at endpoint 0. A Queue head must be prepared so that the device controller can store the incoming setup packet.

In order to initialize a device, the software should perform the following steps:

1. Set Controller Mode in the USB.USBMODE register to device mode.
2. Allocate and Initialize device queue heads in system memory.

NOTE

Transitioning from host mode to device mode requires a device controller reset before modifying USB.USBMODE.

- Minimum: Initialize device queue heads 0 Tx & 0 Rx.
- For information on device queue heads, refer to section Device Data Structures.

NOTE

All device queue heads must be initialized for control endpoints must be initialized before the endpoint is enabled. Non-Control device queue heads before the endpoint can be used.

3. Configure USB.ENDPOINTLISTADDR Pointer.
   - For additional information on USB.ENDPOINTLISTADDR, refer to the register table.
4. Enable the microprocessor interrupt associated with the USB-HS OTG High-Speed USB On-The-Go core.
   - Recommended: enable all device interrupts including: USBINT, USBERRINT, Port Change Detect, USB Reset Received, DCSuspend.
   - For a list of available interrupts refer to the Interrupt Enable Register (USB__USBINTR) and the USB Status Register (USB__USBSTS) register tables.
5. Set Run/Stop bit to Run Mode.
   - After the Run bit is set, a device reset will occur. The DCD must monitor the reset event and adjust the software state as described in the Bus Reset section of the following Port State and Control section below.

NOTE

Endpoint 0 is designed as a control endpoint only and does not need to be configured using ENDPTCTRL0 register.

It is also not necessary to initially prime Endpoint 0 because the first packet received will always be a setup packet. The contents of the first setup packet will require a response in accordance with USB device framework (Chapter 9) command set.

77.4.6.2 Port State and Control

From a chip or system reset, the device controller enters the *powered* state. A transition from the *powered* state to the *attach* state occurs when the Run/Stop bit is set to a '1'.

After receiving a reset on the bus, the port will enter the *defaultFS or defaultHS* state in accordance with the reset protocol described in Appendix C.2 of the USB Specification Rev. 2.0.

The following state diagram depicts the state of a USB 2.0 device.

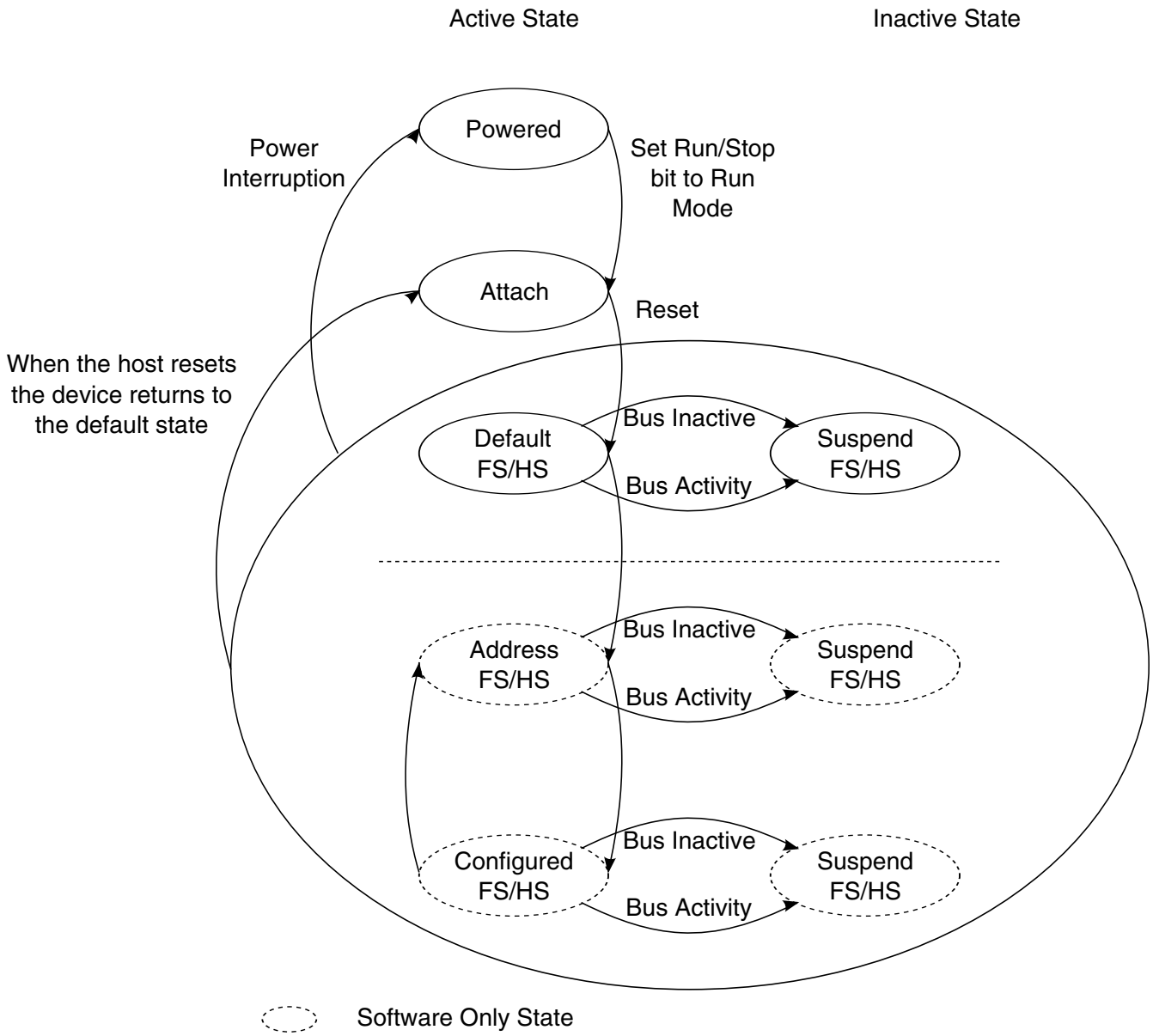


Figure 77-30. Device State Diagram

States *powered, attach, defaultFS/HS, suspendFS/HS* are implemented in the device controller and are communicated to the DCD using the following status bits:

The following table describes the Device Controller State Information Bits.

Table 77-64. Device Controller State Information Bits

Bit	Register
DCSuspend	USB Status Register (USB__USBSTS)
USB Reset Received	USBSTS

Table continues on the next page...

Table 77-64. Device Controller State Information Bits (continued)

Port Change Detect	USBSTS
High-Speed Port	Port Status & Control (USB__PORTSC1)

It is the responsibility of the DCD to maintain a state variable to differentiate between the *DefaultFS/HS* state and the *Address/Configured* states. Change of state from *Default* to *Address* and the *Configured* states is part of the enumeration process described in the device framework section of the USB 2.0 Specification.

As a result of entering the *Address* state, the device address register (DEVICEADDR) must be programmed by the DCD.

Entry into the *Configured* indicates that all endpoints to be used in the operation of the device have been properly initialized by programming the ENDPTCTRLx registers and initializing the associated queue heads.

77.4.6.2.1 Bus Reset

A bus reset is used by the host to initialize downstream devices.

When a bus reset is detected, the device controller will renegotiate its attachment speed, reset the device address to 0, and notify the DCD by interrupt (assuming the USB Reset Interrupt Enable is set). After a reset is received, all endpoints (except endpoint 0) are disabled and any primed transactions will be cancelled by the device controller. The concept of priming will be clarified below, but the DCD must perform the following tasks when a reset is received:

Clear all setup token semaphores by reading the Endpoint Status (USB_UOG_ENDPTSTAT) register and writing the same value back to the Endpoint Status (USB_UOG_ENDPTSTAT) register.

Clear all the endpoint complete status bits by reading the Endpoint Complete (USB_UOG_ENDPTCOMPLETE) register and writing the same value back to the Endpoint Complete (USB_UOG_ENDPTCOMPLETE) register.

Cancel all primed status by waiting until all bits in the Endpoint Initialization (USB_UOG_ENDPTPRIME) are 0 and then writing 0xFFFFFFFF to Endpoint De-Initialize (USB_UOG_ENDPTFLUSH).

Read the reset bit in the Port Status & Control (USB__PORTSC1) register and make sure that it is still active. A USB reset will occur for a minimum of 3 ms and the DCD must reach this point in the reset cleanup before end of the reset occurs, otherwise a hardware reset of the device controller is recommended (rare.)

- A hardware reset can be performed by writing a one to the device controller reset bit in the USBCMD reset. Note: a hardware reset will cause the device to detach from the bus by clearing the Run/Stop bit. Thus, the DCD must completely re-initialize the device controller after a hardware reset.

Free all allocated dTDs because they will no longer be executed by the device controller. If this is the first time the DCD is processing a USB reset event, then it is likely that no dTDs have been allocated.

At this time, the DCD may release control back to the OS because no further changes to the device controller are permitted until a Port Change Detect is indicated.

After a Port Change Detect, the device has reached the default state and the DCD can read the Port Status & Control (USB__PORTSC1) to determine if the device is operating in FS or HS mode. At this time, the device controller has reached normal operating mode and DCD can begin enumeration according to the USB Chapter 9 - Device Framework.

NOTE

The device DCD may use the FS/HS mode information to determine the bandwidth mode of the device

In some applications, it may not be possible to enable one or more pipes while in FS mode. *Beyond the data rate issue, there is no difference in DCD operation between FS and HS modes.*

77.4.6.2.2 Suspend/Resume

77.4.6.2.2.1 Suspend

Suspend Description

In order to conserve power, USB devices automatically enter the suspended state when the device has observed no bus traffic for a specified period. When suspended, the USB device maintains any internal status, including its address and configuration. Attached devices must be prepared to suspend at any time they are powered, regardless of if they have been assigned a non-default address, are configured, or neither. Bus activity may cease due to the host entering a suspend mode of its own. In addition, a USB device shall also enter the suspended state when the hub port it is attached to is disabled.

A USB device exits suspend mode when there is bus activity. A USB device may also request the host to exit suspend mode or selective suspend by using electrical signaling to indicate remote wakeup. The ability of a device to signal remote wakeup is optional. If the USB device is capable of remote wakeup signaling, the device must support the ability of the host to enable and disable this capability. When the device is reset, remote wakeup signaling must be disabled.

Suspend Operational Model

The device controller moves into the suspend state when suspend signaling is detected or activity is missing on the upstream port for more than a specific period. After the device controller enters the suspend state, the DCD is notified by an interrupt (assuming *DC Suspend Interrupt* is enabled). When the *DCSuspend* bit in the Port Status & Control (USB__PORTSC1) is set to a '1', the device controller is suspended.

DCD response when the device controller is suspended is application specific and may involve switching to low power operation.

Information on the bus power limits in suspend state can be found in USB 2.0 specification.

NOTE

Review system level clocking issues defined in section (Ref: Signals-Clocking) for the clocking requirements of a suspended device controller.

77.4.6.2.2.2 Resume

If the device controller is suspended, its operation is resumed when any non-idle signaling is received on its upstream facing port. In addition, the device can signal the system to resume operation by forcing resume signaling to the upstream port.

Resume signaling is sent upstream by writing a '1' to the Resume bit in the in the Port Status & Control (USB__PORTSC1) while the device is in suspend state. Sending resume signal to an upstream port should cause the host to issue resume signaling and bring the suspended bus segment (one more devices) back to the active condition.

NOTE

Before resume signaling can be used, the host must enable it by using the Set Feature command defined in device framework (chapter 9) of the USB 2.0 Specification.

77.4.6.2.2.3 Port Test Modes

Contact ARC International for port test mode capabilities.

77.4.6.2.3 Managing Endpoints

The USB 2.0 specification defines an endpoint, also called a device endpoint or an address endpoint as a uniquely addressable portion of a USB device that can source or sink data in a communications channel between the host and the device.

The endpoint address is specified by the combination of the endpoint number and the endpoint direction.

The channel between the host and an endpoint at a specific device represents a data pipe. Endpoint 0 for a device is always a *control* type data channel used for device discovery and enumeration. Other types of endpoints support by USB include *bulk, interrupt*, and *isochronous*. Each endpoint type has specific behavior related to packet response and error handling. More detail on endpoint operation can be found in the USB 2.0 specification.

The USB-HS OTG High-Speed USB On-The-Go device controller hardware supports up to the USB 2.0 maximum of 32 endpoint specified numbers. Each additional endpoint beyond the required endpoint position adds additional hardware logic. The maximum number of endpoint numbers available to the DCD is configured at hardware synthesis timer. After synthesis, the DCD can enable, disable and configure endpoint type up to the maximum selected during synthesis.

Each endpoint direction is essentially independent and can be configured with differing behavior in each direction. For example, the DCD can configure endpoint 1-IN to be a bulk endpoint and endpoint 1-OUT to be an isochronous endpoint. This helps to conserve the total number of endpoints required for device operation. The only exception is that control endpoints must use both directions on a single endpoint number to function as a control endpoint. Endpoint 0 is, for example, is always a control endpoint and uses the pair of directions.

Each endpoint direction requires a *queue head* allocated in memory. If the maximum of 16 endpoint numbers, one for each endpoint direction are being used by the device controller, then 32 *queue heads* are required. The operation of an endpoint and use of *queue heads* are described later in this document.

77.4.6.2.4 Endpoint Initialization

After hardware reset, all endpoints except endpoint zero are uninitialized and disabled. The DCD must configure and enable each endpoint by writing to configuration bit in the ENDPTCTRLx register.

Each 32-bit ENDPTCTRLx is split into an upper and lower half. The lower half of ENDPTCTRLx is used to configure the receive or OUT endpoint and the upper half is likewise used to configure the corresponding transmit or IN endpoint. Control endpoints must be configured the same in both the upper and lower half of the ENDPTCTRLx

register otherwise the behavior is undefined. The following table shows how to construct a configuration word for endpoint initialization. The following table shows the fields and values for the Device Controller Endpoint initialization.

Table 77-65. Device Controller Endpoint Initialization

Field	Value
Data Toggle Reset	1
Data Toggle Inhibit	0
Endpoint Type	00 Control 01 Isochronous 10 Bulk 11 Interrupt
Endpoint Stall	0

77.4.6.2.5 Stalling

There are two occasions where the device controller may need to return to the host a STALL.

The first occasion is the functional stall, which is a condition set by the DCD as described in the USB 2.0 device framework. A functional stall is only used on non-control endpoints and can be enabled in the device controller by setting the endpoint stall bit in the ENDPTCTRLx register associated with the given endpoint and the given direction. In a functional stall condition, the device controller will continue to return STALL responses to all transactions occurring on the respective endpoint and direction until the endpoint stall bit is cleared by the DCD.

A protocol stall, unlike a function stall, is used on control endpoints is automatically cleared by the device controller at the start of a new control transaction (setup phase). When enabling a protocol stall, the DCD should enable the stall bits (both directions) as a pair. A single write to the ENDPTCTRLx register can ensure that both stall bits are set at the same instant.

NOTE

Any write to the ENDPTCTRLx register during operational mode must preserve the endpoint type field (that is, perform a read-modify-write).

The following table shows the response matrix for the Device Controller Stall.

Table 77-66. Device Controller Stall Response Matrix

USB Packet	Endpoint Stall Bit.	Effect on STALL bit.	USB Response

Table continues on the next page...

Table 77-66. Device Controller Stall Response Matrix (continued)

SETUP packet received by a non-control endpoint.	N/A	None.	STALL
IN/OUT/PING packet received by a non-control endpoint.	'1'	None.	STALL
IN/OUT/PING packet received by a non-control endpoint.	'0'	None.	ACK/ NAK/ NYET
SETUP packet received by a control endpoint.	N/A	Cleared	ACK
IN/OUT/PING packet received by a control endpoint	'1'	None	STALL
IN/OUT/PING packet received by a control endpoint.	'0'	None.	ACK/ NAK/ NYET

77.4.6.2.6 Data Toggle

Data toggle is a mechanism to maintain data coherency between host and device for any given data pipe.

For more information on data toggle, refer to the USB 2.0 specification.

77.4.6.2.6.1 Data Toggle Reset

The DCD may reset the data toggle state bit and cause the data toggle sequence to reset in the device controller by writing a '1' to the data toggle reset bit in the ENDPTCTRLx register.

This should only be necessary when configuring/initializing an endpoint or returning from a STALL condition.

77.4.6.2.6.2 Data Toggle Inhibit

NOTE

This feature is for test purposes only and should never be used during normal device controller operation.

Setting the *data toggle Inhibit bit* active ('1') causes the device controller to ignore the data toggle pattern that is normally sent and accept all incoming data packets regardless of the data toggle state.

In normal operation, the device controller checks the DATA0/DATA1 bit against the data toggle to determine if the packet is valid. If Data PID does not match the data toggle state bit maintained by the device controller for that endpoint, the Data toggle is considered not valid. If the data toggle is not valid, the device controller assumes the packet was already received and discards the packet (not reporting it to the DCD). To prevent the host controller from re-sending the same packet, the device controller will respond to the error packet by acknowledging it with either an ACK or NYET response.

77.4.6.2.6.3 Priming Transmit Endpoints

Priming a transmit endpoint will cause the device controller to fetch the device transfer descriptor (dTD) for the transaction pointed to by the device queue head (dQH).

After the dTD is fetched, it will be stored in the dQH until the device controller completes the transfer described by the dTD. Storing the dTD in the dQH allows the device controller to fetch the operating context needed to handle a request from the host without the need to follow the linked list, starting at the dQH when the host request is received.

After the device has loaded the dTD, the leading data in the packet is stored in a FIFO in the device controller. This FIFO is split into virtual channels so that the leading data can be stored for any endpoint up to the maximum number of endpoints configured at device synthesis time.

After a priming request is complete, an endpoint state of primed is indicated in the USB.ENDPTSTATUS register. For a primed transmit endpoint, the device controller can respond to an IN request from the host and meet the stringent bus turnaround time of High Speed USB.

Because only the leading data is stored in the device controller FIFO, it is necessary for the device controller to begin filling in behind leading data after the transaction starts. The FIFO must be sized to account for the maximum latency that can be incurred by the system memory bus. More information about FIFO sizing is presented in section .

77.4.6.2.6.4 Priming Receive Endpoints

Priming receive endpoints is identical to priming of transmit endpoints from the point of view of the DCD. At the device controller the major difference in the operational model is that there is no data movement of the leading packet data simply because the data is to be received from the host.

Note as part of the architecture, the FIFO for the receive endpoints is not partitioned into multiple channels like the transmit FIFO. Thus, the size of the RX FIFO does not scale with the number of endpoints.

77.4.6.3 Operational Model For Packet Transfers

All transactions on the USB bus are initiated by the host and in turn, the device must respond to any request from the host within the turnaround time stated in the USB 2.0 Specification.

At USB 1.1 Full or Low Speed rates, this turnaround time was significant and the USB 1.1 device controllers were architected so that the device controller could access main memory or interrupt a host protocol processor in order to respond to the USB 1.1 transaction. The architecture of the USB 2.0 device controller must be different because same methods will not meet USB 2.0 High-speed turnaround time requirements by simply increasing clock rate.

A USB host will send requests to the device controller in an order that can not be precisely predicted as a single pipeline, so it is not possible to prepare a single packet for the device controller to execute. However, the order of packet requests is predictable when the endpoint number and direction is considered. For example, if endpoint 3 (transmit direction) is configured as a bulk pipe, then we can expect the host will send IN requests to that endpoint. This device controller is architected in such a way that it can prepare packets for each endpoint/direction in anticipation of the host request. The process of preparing the device controller to send or receive data in response to host initiated transaction on the bus is referred to as "priming" the endpoint. This term will be used throughout the following documentation to describe the device controller operation so the DCD can be architected properly use priming. Further, note that the term "flushing" is used to describe the action of clearing a packet that was queued for execution.

77.4.6.3.1 Interrupt/Bulk Endpoint Operational Model

The behaviors of the device controller for interrupt and bulk endpoints are identical. All valid IN and OUT transactions to bulk pipes will handshake with a NAK unless the endpoint had been primed. Once the endpoint has been primed, data delivery will commence.

A dTD will be retired by the device controller when the packets described in the transfer descriptor have been completed. Each dTD describes N packets to be transferred according to the USB Variable Length transfer protocol. The formula and table on the following page describe how the device controller computes the number and length of the packets to be sent/received by the USB vary according to the total number of bytes and maximum packet length.

With Zero Length Termination (ZLT) = 0

N = INT(Number Of Bytes/Max. Packet Length) + 1

With Zero Length Termination (ZLT) = 1

N = MAXINT(Number Of Bytes/Max. Packet Length)

Table 77-67. Variable Length Transfer Protocol Example (ZLT = 0)

Bytes (dTD)	Max. Packet Length (dQH)	N	P1	P2	P3
511	256	2	256	255	
512	256	3	256	256	0
512	512	2	512	0	

Table 77-68. Variable Length Transfer Protocol Example (ZLT = 1)

Bytes (dTD)	Max. Packet Length (dQH)	N	P1	P2	P3
511	256	2	256	255	
512	256	2	256	256	
512	512	1	512		

NOTE

The MULT field in the dQH must be set to "00" for bulk, interrupt, and control endpoints.

TX-dTD is complete when:

- All packets described dTD were successfully transmitted. *** Total bytes in dTD will equal zero when this occurs.

RX-dTD is complete when:

- All packets described in dTD were successfully received. *** Total bytes in dTD will equal zero when this occurs.
- A short packet (number of bytes < maximum packet length) was received. *** This is a successful transfer completion; DCD must check Total Bytes in dTD to determine the number of bytes that are remaining. From the total bytes remaining in the dTD, the DCD can compute the actual bytes received.
- A long packet was received (number of bytes > maximum packet size) OR (total bytes received > total bytes specified). *** This is an error condition. The device controller will discard the remaining packet, and set the Buffer Error bit in the dTD. In addition, the endpoint will be flushed and the USBERR interrupt will become active.

On the successful completion of the packet(s) described by the dTD, the active bit in the dTD will be cleared and the next pointer will be followed when the Terminate bit is clear. When the Terminate bit is set, the device controller will flush the endpoint/direction and cease operations for that endpoint/direction.

On the unsuccessful completion of a packet (see long packet above), the dQH will be left pointing to the dTD that was in error. In order to recover from this error condition, the DCD must properly reinitialize the dQH by clearing the active bit and update the nextTD pointer before attempting to re-prime the endpoint.

NOTE

All packet level errors such as a missing handshake or CRC error will be retried automatically by the device controller.

There is no required interaction with the DCD for handling such errors.

77.4.6.3.1.1 Interrupt/Bulk Endpoint Bus Response Matrix

The table below shows the response matrix for Interrput/Bulk Endpoint Bus.

Table 77-69. Interrupt/Bulk Endpoint Bus Response Matrix

	Stall	Not Primed	Primed	Underflow	Overflow
Setup	Ignore	Ignore	Ignore	N/A	N/A
In	STALL	NAK	Transmit	BS Error	N/A
Out	STALL	NAK	Receive + NYET/ ACK	N/A	NAK
Ping	STALL	NAK	ACK	N/A	N/A
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore

NOTE

BS Error = Force Bit Stuff Error

NYET/ACK - NYET unless the Transfer Descriptor has packets remaining according to the USB variable length protocol then ACK.

SYSERR - System error should never occur when the latency FIFOs are correctly sized and the DCD is responsive.

77.4.6.3.2 Control Endpoint Operation Model

77.4.6.3.2.1 Setup Phase

All requests to a control endpoint begin with a setup phase followed by an optional data phase and a required status phase. The device controller will always accept the setup phase unless the setup lockout is engaged.

The setup lockout will engage so that future setup packets are ignored. Lockout of setup packets ensures that while software is reading the setup packet stored in the queue head, that data is not written as it is being read potentially causing an invalid setup packet.

In hardware versions 2.3 and later, the setup lockout mechanism can be disabled and a new tripwire type semaphore will ensure that the setup packet payload is extracted from the queue head without being corrupted be an incoming setup packet. This is the preferred behavior because ignoring repeated setup packets due to long software interrupt latency would be a compliance issue.

Setup Packet Handling (Pre-2.3 hardware)

- After receiving an interrupt and inspecting USB Device Mode (USB__USBMODE) to determine that a setup packet was received on a particular pipe:

1. Duplicate contents of dQH.SsetupBuffer into local software byte array.
2. Write '1' to clear corresponding Endpoint Setup Status (USB_UOG_ENDPTSETUPSTAT) bit and thereby disabling Setup Lockout. (That is, the Setup Lockout activates as soon as a setup arrives. By writing to the Endpoint Setup Status (USB_UOG_ENDPTSETUPSTAT), the device controller will accept new setup packets.)
3. Process setup packet using local software byte array copy and execute status/ handshake phases.

NOTE

After receiving a new setup packet the status and/or handshake phases may still be pending from a previous control sequence. These should be flushed & deallocated before linking a new status and/or handshake dTD for the most recent setup packet.

NOTE

To limit the exposure of setup packets to the setup lockout mechanism (if used), the DCD should designate the priority of responding to setup packets above responding to other packet completions.

Setup Packet Handling (2.3 hardware and later)

- Disable Setup Lockout by writing 1 to Setup Lockout Mode (SLOM) in USB Device Mode (USB__USBMODE). (once at initialization). Setup lockout is not necessary when using the tripwire as described below.

NOTE

Leaving the Setup Lockout Mode As 0 will result in pre-2.3 hardware behavior.

- After receiving an interrupt and inspecting Endpoint Setup Status (USB_UOG_ENDPTSETUPSTAT) to determine that a setup packet was received on a particular pipe:
  a. Write 1 to clear corresponding bit Endpoint Setup Status (USB_UOG_ENDPTSETUPSTAT).
  b. Write 1 to Setup Tripwire (SUTW) in USB Command Register (USB__USBCMD) register.
  c. Duplicate contents of dQH.SetupBuffer into local software byte array.
  d. Read Setup TripWire (SUTW) in USB Command Register (USB__USBCMD) register. (if set - continue; if cleared - goto 2)
  e. Write 0 to clear Setup Tripwire (SUTW) in USB Command Register (USB__USBCMD) register.
  f. Process setup packet using local software byte array copy and execute status/ handshake phases.

NOTE

After receiving a new setup packet the status and/or handshake phases may still be pending from a previous control sequence. These should be flushed & deallocated before linking a new status and/or handshake dTD for the most recent setup packet.

77.4.6.3.2.2 Data Phase

Following the setup phase, the DCD must create a device transfer descriptor for the data phase and prime the transfer.

After priming the packet, the DCD must verify a new setup packet has not been received by reading the USB.ENDPTSETUPSTAT register immediately verifying that the prime had completed. A prime will complete when the associated bit in the Endpoint Initialization (USB_UOG_ENDPTPRIME) register is zero and the associated bit in the Endpoint Status (USB_UOG_ENDPTSTAT) register is a one. If a prime fails, ie. The Endpoint Initialization (USB_UOG_ENDPTPRIME) bit goes to zero and the Endpoint Status (USB_UOG_ENDPTSTAT) bit is not set, then the prime has failed. This can only

be due to improper setup of the dQH, dTD or a setup arriving during the prime operation. If a new setup packet is indicated after the ENDPTPRIME bit is cleared, then the transfer descriptor can be freed and the DCD must reinterpret the setup packet.

Should a setup arrive after the data stage is primed, the device controller will automatically clear the prime status (Endpoint Status (USB_UOG_ENDPTSTAT)) to enforce data coherency with the setup packet.

NOTE

- The MULT field in the dQH must be set to "00" for bulk, interrupt, and control endpoints.
- Error handling of data phase packets is the same as bulk packets described previously.

77.4.6.3.2.3 Status Phase

Similar to the data phase, the DCD must create a transfer descriptor (with byte length equal zero) and prime the endpoint for the status phase.

The DCD must also perform the same checks of the USB.ENDPTSETUPSTAT as described above in the data phase.

NOTE

- The MULT field in the dQH must be set to 00 for bulk, interrupt, and control endpoints.
- Error handling of data phase packets is the same as bulk packets described previously.

77.4.6.3.2.4 Control Endpoint Bus Response Matrix

Shown in the following table is the device controller response to packets on a control endpoint according to the device controller state.

The table below shows the response matrix for the Control Endpoint Bus.

Table 77-70. Control Endpoint Bus Response Matrix

Token Type	Endpoint State					Setup Lockout
	Stall	Not Primed	Primed	Underflow	Overflow	
Setup	ACK	ACK	ACK	N/A	SYSERR	
In	STALL	NAK	Transmit	BS Error	N/A	N/A
Out	STALL	NAK	Receive + NYET/ ACK	N/A	NAK	N/A
Ping	STALL	NAK	ACK	N/A	N/A	N/A
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore	Ignore

BS Error = Force Bit Stuff Error

NYET/ACK - NYET unless the Transfer Descriptor has packets remaining according to the USB variable length protocol then ACK.

SYSERR - System error should never occur when the latency FIFOs are correctly sized and the DCD is responsive.

77.4.6.3.3 Isochronous Endpoint Operational Model

Isochronous endpoints are used for real-time scheduled delivery of data and their operational model is significantly different than the host throttled Bulk, Interrupt, and Control data pipes.

Real time delivery by the device controller will is accomplished by the following:

- Exactly MULT Packets per (micro)Frame are transmitted/received. Note: MULT is a two-bit field in the device Queue Head. The variable length packet protocol is not used on isochronous endpoints.
- NAK responses are not used. Instead, zero length packets and sent in response to an IN request to an unprimed endpoints. For unprimed RX endpoints, the response to an OUT transaction is to ignore the packet within the device controller.
- Prime requests always schedule the transfer described in the dTD for the next (micro)frame. If the ISO-dTD is still active after that frame, then the ISO-dTD will be held ready until executed or canceled by the DCD.

An EHCI compatible host controller uses the periodic frame list to schedule data exchanges to Isochronous endpoints. The operational model for device mode does not use such a data structure. Instead, the same dTD used for Control/Bulk/Interrupt endpoints is also used for isochronous endpoints. The difference is in the handling of the dTD.

The first difference between bulk and ISO-endpoints is that priming an ISO-endpoint is a delayed operation such that an endpoint will become primed only after a SOF is received. After the DCD writes the prime bit, the prime bit will be cleared as usual to indicate to software that the device controller completed a priming the dTD for transfer. Internal to the design, the device controller hardware masks that prime start until the next frame boundary. This behavior is hidden from the DCD but occurs so that the device controller can match the dTD to a specific (micro)frame.

Another difference with isochronous endpoints is that the transaction must wholly complete in a (micro)frame. Once an ISO transaction is started in a (micro)frame it will retire the corresponding dTD when MULT transactions occur or the device controller finds a fulfillment condition.

The transaction error bit set in the status field indicates a fulfillment error condition. When a fulfillment error occurs, the frame after the transfer failed to complete wholly, the device controller will force retire the ISO-dTD and move to the next ISO-dTD.

It is important to note that fulfillment errors are only caused due to partially completed packets. If no activity occurs to a primed ISO-dTD, the transaction will stay primed indefinitely. This means it is up to software discard transmit ISO-dTDs that pile up from a failure of the host to move the data.

Finally, the last difference with ISO packets is in the data level error handling. When a CRC error occurs on a received packet, the packet is not retried similar to bulk and control endpoints. Instead, the CRC is noted by setting the *Transaction Error* bit and the data is stored as usual for the application software to sort out.

- TX Packet Retired
  - MULT counter reaches zero.
  - Fulfillment Error [*Transaction Error* bit is set]
    - # Packets Occurred > 0 AND # Packets Occurred < MULT

NOTE

For TX-ISO, MULT Counter can be loaded with a lesser value in the dTD Multiplier Override field in hardware versions 2.3 and later. If the Multiplier Override is zero, the MULT Counter is initialized to the Multiplier in the QH.

- RX Packet Retired:
  - MULT counter reaches zero.
  - Non-MDATA Data PID is received**
    - ** Exit criteria only valid in hardware version 2.3 or later. Previous to hardware version 2.3, any PID sequence that did not match the MULT field exactly would be flagged as a transaction error due to PID mismatch or fulfillment error.
  - Overflow Error:
    - Packet received is > maximum packet length. [*Buffer Error* bit is set]
    - Packet received exceeds total bytes allocated in dTD. [*Buffer Error* bit is set]
  - Fulfillment Error [*Transaction Error* bit is set]
    - # Packets Occurred > 0 AND # Packets Occurred < MULT
  - CRC Error [*Transaction Error* bit is set

NOTE

For ISO, when a dTD is retired, the next dTD is primed for the next frame. For continuous (micro)frame to (micro)frame

operation the DCD should ensure that the dTD linked-list is out ahead of the device controller by at least two (micro)frames.

77.4.6.3.3.1 Isochronous Pipe Synchronization

When it is necessary to synchronize an isochroous data pipe to the host, the (micro)frame number (USB.FRINDEX register) can be used as a marker.

To cause a packet transfer to occur at a specific (micro)frame number [N], the DCD should interrupt on SOF during frame N-1. When the USB.FRINDEX=N-1, the DCD must write the prime bit. The device controller will prime the isochronous endpoint in (micro)frame N-1 so that the device controller will execute delivery during (micro)frame N.

NOTE

Priming an endpoint towards the end of (micro)frame N-1 will not guarantee delivery in (micro)frame N. The delivery may actually occur in (micro)frame N+1 if device controller does not have enough time to complete the prime before the SOF for packet N is received.

77.4.6.3.3.2 Isochronous Endpoint Bus Response Matrix

The following table shows the response matrix for the Isochronous Endpoint Bus.

Table 77-71. Isochronous Endpoint Bus Response Matrix

	Stall	Not Primed	Primed	Underflow	Overflow
Setup	STALL	STALL	STALL	N/A	N/A
In	NULL Packet	NULL Packet	Transmit	BS Error	N/A
Out	Ignore	Ignore	Receive	N/A	Drop Packet
Ping	Ignore	Ignore	Ignore	Ignore	Ignore
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore

1. BS Error = Force Bit Stuff Error

   NULL Packet = Zero Length Packet

77.4.6.4 Managing Queue Heads

The following figure shows the End Point Queue Head.

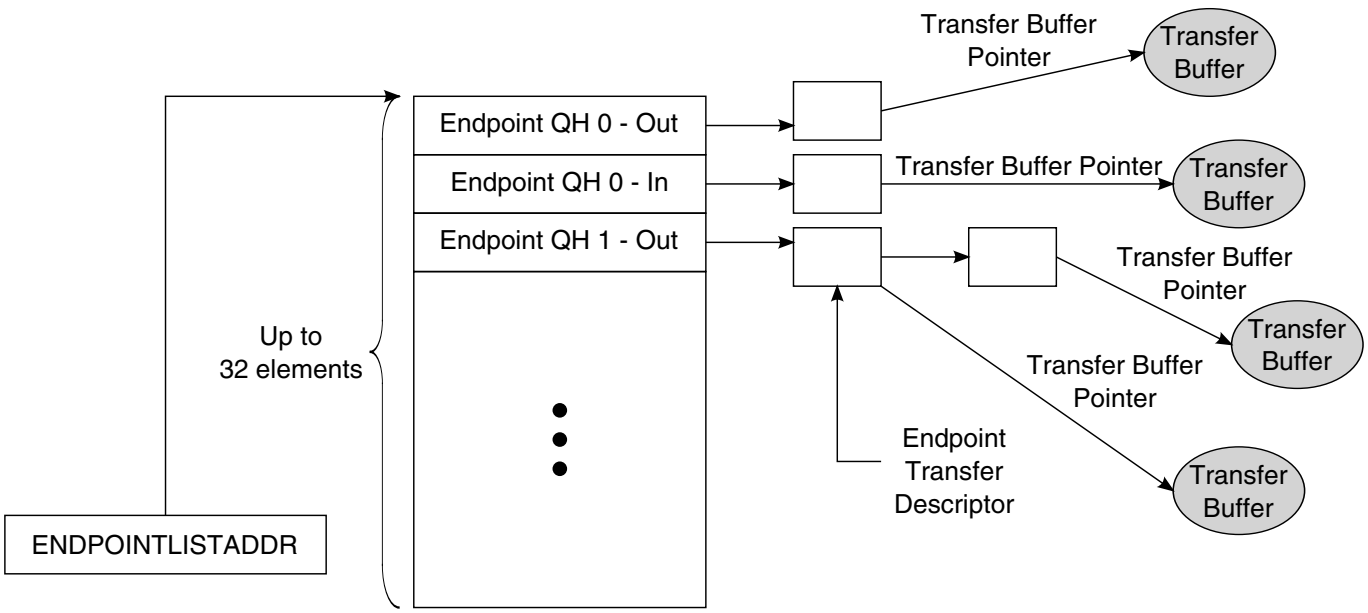


Figure 77-31. End Point Queue Head Diagram

The device queue head (dQH) points to the linked list of transfer tasks, each depicted by the device Transfer Descriptor (dTD). An area of memory pointed to by USB.ENDPOINTLISTADDR contains a group of all dQH's in a sequential list as shown in Figure 77-31. The even elements in the list of dQH's are used for receive endpoints (OUT/SETUP) and the odd elements are used for transmit endpoints (IN/INTERRUPT). Device transfer descriptors are linked head to tail starting at the queue head and ending at a terminate bit. Once the dTD has been retired, it will no longer be part of the linked list from the queue head. Therefore, software is required to track all transfer descriptors because pointers will no longer exist within the queue head once the dTD is retired (see section Software Link Pointers).

In addition to the current and next pointers and the dTD overlay examined in section Operational Model For Packet Transfers, the dQH also contains the following parameters for the associated endpoint: Multipler, Maximum Packet Length, Interrupt On Setup. The complete initialization of the dQH including these fields is demonstrated in the next section.

77.4.6.4.1 Queue Head Initialization

One pair of device queue heads must be initialized for each active endpoint.

To initialize a device queue head:

- Write the wMaxPacketSize field as required by the USB Chapter 9 or application specific protocol.
- Write the multiplier field to 0 for control, bulk, and interrupt endpoints. For ISO endpoints, set the multiplier to 1,2, or 3 as required bandwidth an in conjunction with the USB Chapter 9 protocol.

NOTE

In FS mode, the multiplier field can only be 1 for ISO endpoints.

- Write the next dTD Terminate bit field to 1.
- Write the Active bit in the status field to 0.
- Write the Halt bit in the status field to 0.

NOTE

The DCD must only modify dQH if the associated endpoint is not primed and there are no outstanding dTD's.

77.4.6.4.2 Operational Model For Setup Transfers

As discussed in section Control Endpoint Operation Model, setup transfer requires special treatment by the DCD. A setup transfer does not use a dTD but instead stores the incoming data from a setup packet in an 8-byte buffer within the dQH.

Upon receiving notification of the setup packet, the DCD should handle the setup transfer as demonstrated here:

1. Copy setup buffer contents from dQH - RX to software buffer.
2. Acknowledge setup backup by writing a "1" to the corresponding bit in ENDPTSETUPSTAT.

NOTE

- The acknowledge must occur before continuing to process the setup packet.
- After the acknowledge has occurred, the DCD must not attempt to access the setup buffer in the dQH - RX. Only the local software copy should be examined.

3. Check for pending data or status dTD's from previous control transfers and flush if any exist as discussed in section Flushing/De-priming an Endpoint.
4. Decode setup packet and prepare data phase [optional] and status phase transfer as required by the USB Chapter 9 or application specific protocol.

NOTE

It is possible for the device controller to receive setup packets before previous control transfers complete. Existing control packets in progress must be flushed and the new control packet completed.

77.4.6.5 Managing Transfers with Transfer Descriptors

77.4.6.5.1 Software Link Pointers

It is necessary for the DCD software to maintain head and tail pointers to the for the linked list of dTDs for each respective queue head.

This is necessary because the dQH only maintains pointers to the current working dTD and the next dTD to be executed. The operations described in next section for managing dTD will assume the DCD can use reference the head and tail of the dTD linked list. The following figure shows the Software Link Pointers.

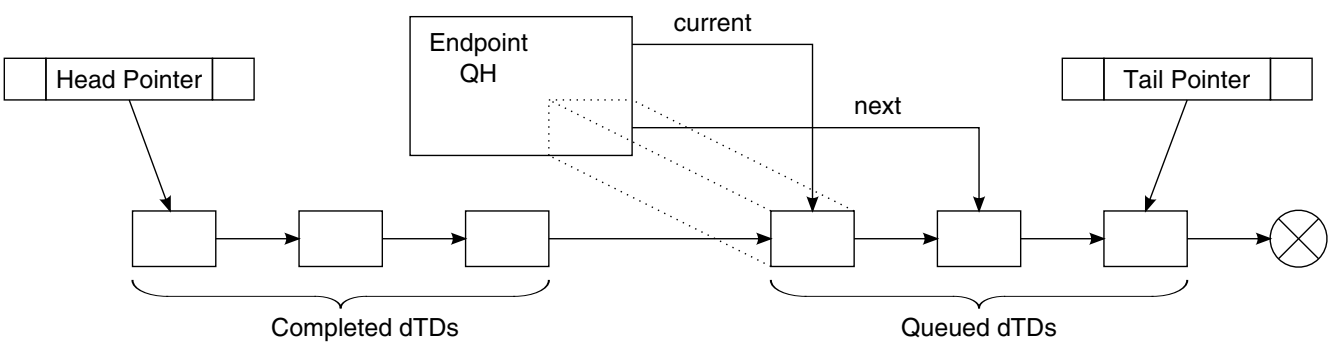


Figure 77-32. Software Link Pointers

NOTE

To conserve memory, the reserved fields at the end of the dQH can be used to store the Head & Tail pointers, but it still remains the responsibility of the DCD to maintain the pointers.

77.4.6.5.2 Building a Transfer Descriptor

Before a transfer can be executed from the linked list, a dTD must be built to describe the transfer.

Use the following procedure for building dTDs.

Allocate 8-DWord dTD block of memory aligned to 8-DWord boundaries. Example: bit address 4:0 would be equal to "00000"

Write the following fields:

1. Initialize first 7 DWords to 0.
2. Set the terminate bit to 1.
3. Fill in total bytes with transfer size.
4. Set the interrupt on complete if desired.
5. Initialize the status field with the active bit set to 1 and all remaining status bits set to 0.
6. Fill in buffer pointer page 0 and the current offset to point to the start of the data buffer.
7. Initialize buffer pointer page 1 through page 4 to be one greater than each of the previous buffer pointer.

77.4.6.5.3 Executing A Transfer Descriptor

To safely add a dTD, the DCD must be follow this procedure which will handle the event where the device controller reaches the end of the dTD list at the same time a new dTD is being added to the end of the list.

Determine whether the link list is empty:

- Check DCD driver to see if pipe is empty (internal representation of linked-list should indicate if any packets are outstanding)
- Case 1: Link list is empty
  - 1. Write dQH next pointer AND dQH terminate bit to 0 as a single DWord operation.
  - 2. Clear active & halt bit in dQH (in case set from a previous error).
  - 3. Prime endpoint by writing 1 to correct bit position in Endpoint Initialization (USB_UOG_ENDPTPRIME).
- Case 2: Link list is not empty
  - 1. Add dTD to end of linked list.
  - 2. Read correct prime bit in Endpoint Initialization (USB_UOG_ENDPTPRIME)- if 1 DONE.
  - 3. Set ATDTW bit in USBCMD register to 1.
  - 4. Read correct status bit in Endpoint Status (USB_UOG_ENDPTSTAT). (store in tmp. variable for later)
  - 5. Read ATDTW bit in USBCMD register.
  - If 0 goto 3.
  - If 1 continue to 6.

- 6. Write ATDTW bit in USBCMD register to 0.
- 7. If status bit read in (3) is 1 DONE.
- 8. If status bit read in (3) is 0 then Goto Case 1: Step 1.

77.4.6.5.4 Transfer Completion

After a dTD has been initialized and the associated endpoint primed the device controller will execute the transfer upon the host-initiated request. The DCD will be notified with a USB interrupt if the Interrupt On Complete bit was set or alternately, the DCD can poll the endpoint complete register to find when the dTD had been executed. After a dTD has been executed, DCD can check the status bits to determine success or failure.

NOTE

Multiple dTD can be completed in a single endpoint complete notification. After clearing the notification, DCD must search the dTD linked list and retire all dTDs that have finished (Active bit cleared).

By reading the status fields of the completed dTDs, the DCD can determine if the transfers completed successfully. Success is determined with the following combination of status bits:

- Active = 0
- Halted = 0
- Transaction Error = 0
- Data Buffer Error = 0

Should any combination other than the one shown above exist, the DCD must take proper action. Transfer failure mechanisms are indicated in the Device Error Matrix.

In addition to checking the status bit the DCD must read the Transfer Bytes field to determine the actual bytes transferred. When a transfer is complete, the Total Bytes transferred is by decremented by the actual bytes transferred. For Transmit packets, a packet is only complete after the actual bytes reaches zero, but for receive packets, the host may send fewer bytes in the transfer according the USB variable length packet protocol.

77.4.6.5.5 Flushing/De-priming an Endpoint

It is necessary for the DCD to flush to de-prime one more endpoints on a USB device reset or during a broken control transfer. There may also be application specific requirements to stop transfers in progress.

The following procedure can be used by the DCD to stop a transfer in progress:

1. Write a '1' to the corresponding bit(s) in Endpoint De-Initialize (USB_UOG_ENDPTFLUSH).
2. Wait until all bits in Endpoint De-Initialize (USB_UOG_ENDPTFLUSH) are '0'.
   - Software note: this operation may take a large amount of time depending on the USB bus activity. It is not desirable to have this wait loop within an interrupt service routine.
3. Read Endpoint Status (USB_UOG_ENDPTSTAT) to ensure that for all endpoints commanded to be flushed, that the corresponding bits are now '0'. If the corresponding bits are '1' after step #2 has finished, then the flush failed as described in the following:
   - Explanation: In very rare cases, a packet is in progress to the particular endpoint when commanded flush using Endpoint De-Initialize (USB_UOG_ENDPTFLUSH). A safeguard is in place to refuse the flush to ensure that the packet in progress completes successfully. The DCD may need to repeatedly flush any endpoints that fail to flush be repeating steps 1-3 until each endpoint is successfully flushed.

77.4.6.5.6 Device Error Matrix

The following table summarizes packet errors that are not automatically handled by the Device Controller.

Table 77-72. Device Error Matrix

Error	Direction	Packet Type	Data Buffer Error Bit	Transaction Error Bit
Overflow **	RX	Any	1	0
ISO Packet Error	RX	ISO	0	1
ISO Fulfillment Error	Both	ISO	0	1

Notice that the device controller handles all errors on Bulk/Control/Interrupt Endpoints except for a data buffer overflow. However, for ISO endpoints, errors packets are not retried and errors are tagged as indicated. The table below describes the errors.

Table 77-73. Error Descriptions

Error	Description
Overflow	Number of bytes received exceeded max. packet size or total buffer length.
	** This error will also set the Halt bit in the dQH and if there are dTDs remaining in the linked list for the endpoint, then those will not be executed.
ISO Packet Error	CRC Error on received ISO packet. Contents not guaranteed to be correct.

Table continues on the next page...

Table 77-73. Error Descriptions (continued)

Error	Description
ISO Fulfillment Error	Hst failed to complete the number of packets defined in the dQH mult field within the given (micro)frame. For scheduled data delivery the DCD may need to readjust the data queue because a fulfillment error will cause Device Controller to cease data transfers on the pipe for one (micro)frame. During the "dead" (micro)frame, the Device Controller reports error on the pipe and primes for the following frame.

77.4.6.6 Servicing Interrupts

The interrupt service routine must consider that there are high-frequency, low-frequency operations, and error operations and order accordingly.

77.4.6.6.1 High-Frequency Interrupts

High frequency interrupts in particular should be handed in the order below. The most important of these is listed first because the DCD must acknowledge a setup buffer in the timeliest manner possible.

The table below describes the High frequency interrupt events.

Table 77-74. High Frequency Interrupt Events

Execution Order	Interrupt	Action
1a	USB Interrupt - USB.ENDPTSETUPSTATUS	Copy contents of setup buffer and acknowledge setup packet (as indicated in Figure 77-31 shows the End Point Queue Head). Process setup packet according to USB 2.0 Chapter 9 or application specific protocol.
1b	USB Interrupt[1] - USB.ENDPTCOMPLETE	Handle completion of dTD as indicated in Figure 77-31 shows the End Point Queue Head.
2	SOF Interrupt	Action as deemed necessary by application. This interrupt may not have a use in all applications.

1. It is likely that multiple interrupts to stack up on any call to the Interrupt Service Routine AND during the Interrupt Service Routine.

77.4.6.6.2 Low-Frequency Interrupts

The low frequency events include the following interrupts. These interrupt can be handled in any order because they do not occur often in comparison to the high-frequency interrupts.

The table below shows the Low frequency interrupt events.

Table 77-75. Low Frequency Interrupt Events

Interrupt	Action
Port Change	Change software state information.
Sleep Enable (Suspend)	Change software state information. Low power handling as necessary.
Reset Received	Change software state information. Abort pending transfers.

77.4.6.6.3 Error Interrupts

Error interrupts will be least frequent and should be placed last in the interrupt service routine.

The following table shows the error interrput events

.

Table 77-76. Error Interrupt Events

Interrupt	Action
USB Error Interrupt	This error is redundant because it combines USB Interrupt and an error status in the dTD. The DCD will more aptly handle packet-level errors by checking dTD status field upon receipt of USB Interrupt (w/ USB.ENDPTCOMPLETE).
System Error	Unrecoverable error. Immediate Reset of core; free transfers buffers in progress and restart the DCD.

77.5 USB Non-Core Memory Map/Register Definition

There are two kinds of registers in the USB module: USB core registers and USB non-core registers.

USB core registers are used to control USB core functions, and more independent of USB features. Each USB controller core has its own core registers.

USB non-core registera are additional to USB core registers, and more dependent on USB features. Therefore, these registers could be very different in different i.MX products.

This section describes only the USB non-core registers. For detailed descriptions of USB core registers, please refer to Register Interface .

NOTE

For reserved bits, please preserve the value when writing (read its reset value, then write this value back).

"USB_UOG_", "USB_UH1_", "USB_UH2_", "USB_UH3_" prefix in register name indicates it is a core register for OTG/ Host1/Host2/Host3 controller core respectively.

"USB_USB_" prefix in register name indicates it is a USB non-core register.

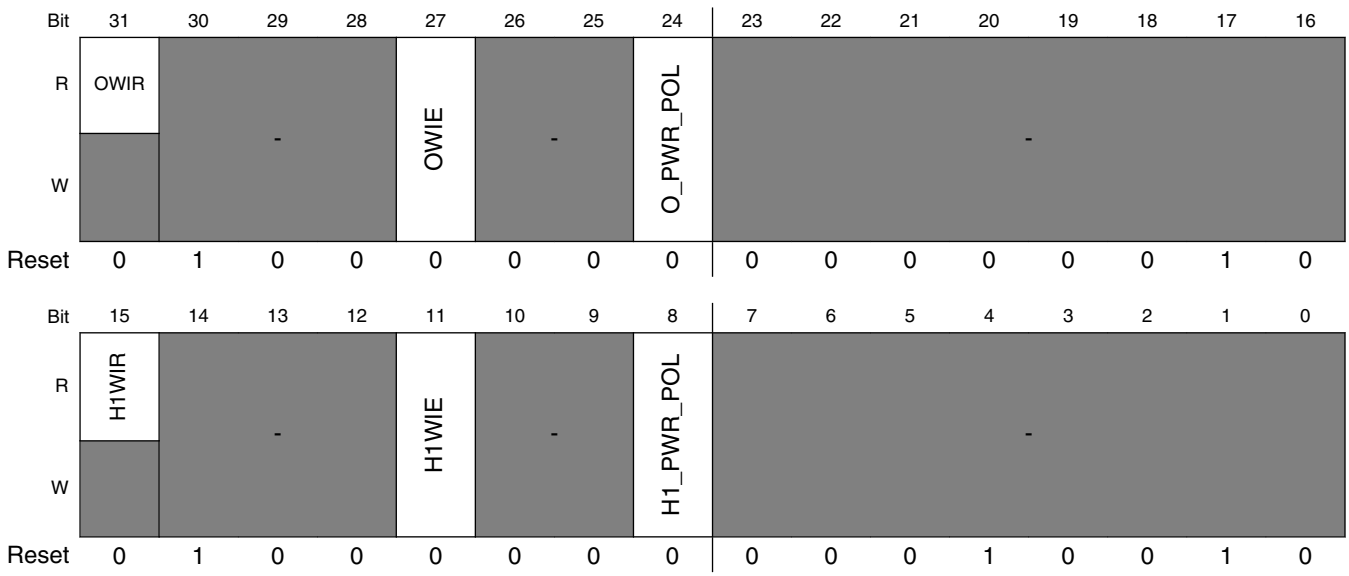
USB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53F8_0800	USB Control Register 0 (USB_USB_CTRL_0)	32	R/W	4002_4012h	77.5.1/ 4881
53F8_0808	USB OTG UTMI PHY Control Register 0 (USB_USB_OTG_PHY_CTRL_0)	32	R/W	8000_1400h	77.5.2/ 4883
53F8_080C	USB OTG UTMI PHY Control Register 1 (USB_USB_OTG_PHY_CTRL_1)	32	R/W	0054_1402h	77.5.3/ 4885
53F8_0810	USB Control Register 1 (USB_USB_CTRL_1)	32	R/W	0000_0000h	77.5.4/ 4888
53F8_0814	USB Host2 Control Register (USB_USB_UH2_CTRL)	32	R/W	0000_1402h	77.5.5/ 4890
53F8_0818	USB Host3 Control Register (USB_USB_UH3_CTRL)	32	R/W	0000_1402h	77.5.6/ 4892
53F8_081C	USB Host1 UTMI PHY Control Register 0 (USB_USB_UH1_PHY_CTRL_0)	32	R/W	8000_1408h	77.5.7/ 4895
53F8_0820	USB Host1 UTMI PHY Control Register 1 (USB_USB_UH1_PHY_CTRL_1)	32	R/W	0054_1401h	77.5.8/ 4897
53F8_0824	USB Clock on/off control Register (USB_USB_CLKONOFF_CTRL)	32	R/W	8980_1042h	77.5.9/ 4899

77.5.1 USB Control Register 0 (USB_USB_CTRL_0)

The USB control register 0 controls the integration specific features of the USB. These features are not directly related to the USB functionality, but control special features, interfacing on the USB ports, as well as power control and wake-up functionality.

Address: USB_USB_CTRL_0 is 53F8_0000h base + 800h offset = 53F8_0800h



USB_USB_CTRL_0 field descriptions

Field	Description
31 OWIR	OTG Wake-up Interrupt Request This bit indicates that a wake-up interrupt request is issued by the OTG port. This bit is cleared when disabling the wake-up interrupt (clearing bit "OWIE"). 1 Wake-up interrupt request detected 0 No wake-up interrupt request detected
30–28 -	Reserved
27 OWIE	OTG Wake-up Interrupt Enable This bit enables or disables the OTG wake-up interrupt. Disabling the interrupt also clears the Interrupt request bit "OWIR". Wake-up interrupt enable should be turned off after receiving a wake-up interrupt and turned on again prior to entering suspend mode 1 Interrupt Enabled 0 Interrupt Disabled (Default)
26–25 -	Reserved
24 O_PWR_POL	OTG power Pin polarity The polarity of OTG Power pin (to enable external PMIC to drive VBUS)

Table continues on the next page...

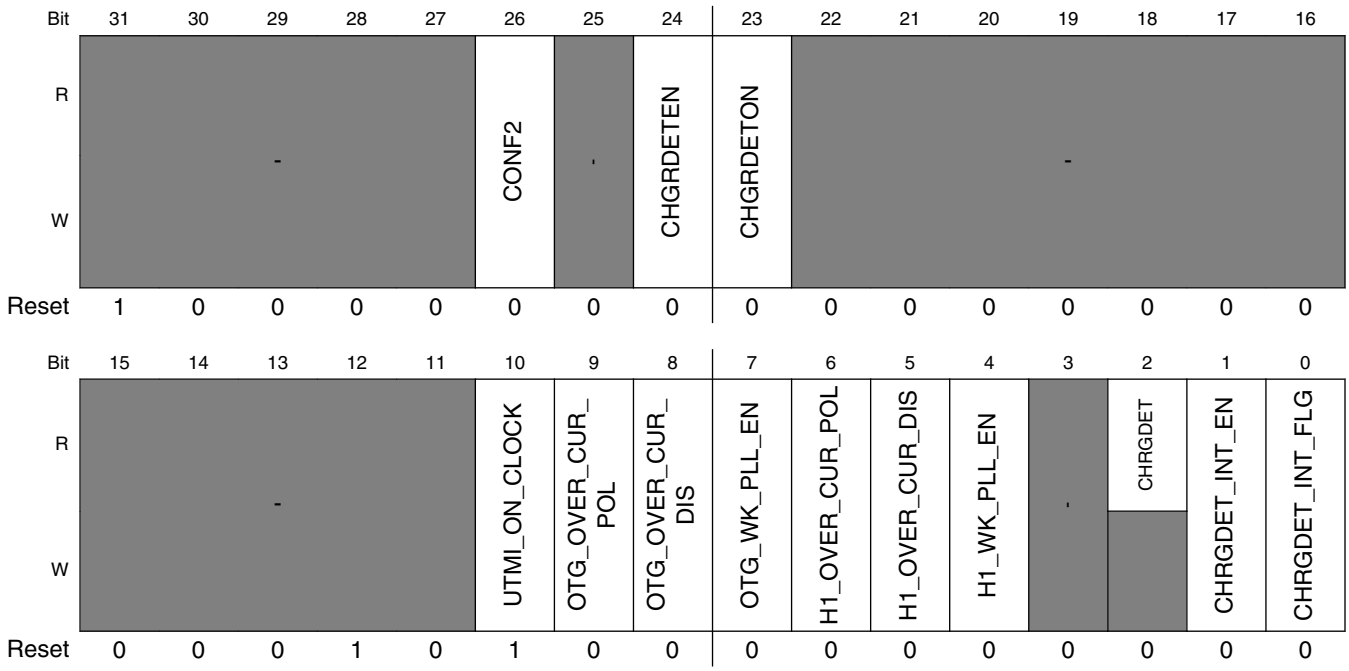
USB_USB_CTRL_0 field descriptions (continued)

Field	Description
	1 High active 0 Low active
23–16 -	Reserved
15 H1WIR	Host1 Wake-up Interrupt Request This bit indicates that a wake-up interrupt request is issued by the Host1 port. This bit is cleared when disabling the wake-up interrupt (clearing bit "H1WIE"). 1 Wake-up interrupt request detected 0 No wake-up interrupt request detected (Default)
14–12 -	Reserved
11 H1WIE	Host1 Wake-up Interrupt Enable This bit enables or disables the Host1 wake-up interrupt. Disabling the interrupt also clears the Interrupt request bit "H1WIR". Wake-up interrupt enable should be turned off after receiving a wake-up interrupt and turned on again prior to entering suspend mode 1 Interrupt Enabled 0 Interrupt Disabled
10–9 -	Reserved
8 H1_PWR_POL	Host1 power Pin polarity The polarity of Host1 Power pin (to enable external PMIC to drive VBUS) 1 High active 0 Low active
7–0 -	Reserved

77.5.2 USB OTG UTMI PHY Control Register 0 (USB_USB_OTG_PHY_CTRL_0)

USB OTG UTMI PHY control register 0 controls the on-chip UTMI PHY.

Address: USB_USB_OTG_PHY_CTRL_0 is 53F8_0000h base + 808h offset = 53F8_0808h



USB_USB_OTG_PHY_CTRL_0 field descriptions

Field	Description
31–27 -	Reserved
26 CONF2	UTMI PHY CONF2 When this bit is active, OTG comparators are turn on during suspend. 1 Turns on OTG comparators during suspend 0 Inactive
25 -	Reserved
24 CHGRDETEN	UTMI PHY Charger Detector Enable This bit should be set for 300 ms (or more) after CHGRDETON bit is set to '1b'. 1 Active 0 Inactive
23 CHGRDETON	UTMI PHY Charger Detector Power On This bit controls the internal current mirrors used for charger detection. When this bit is set '1b', internal current mirrors are powered on.

Table continues on the next page...

USB_USB_OTG_PHY_CTRL_0 field descriptions (continued)

Field	Description
	This bit should be set to '1b' at least 20 ms before CHGRDETEN bit is set to '1b'.
22–11 -	Reserved
10 UTMI_ON_CLOCK	UTMI PHY Clock On When this bit is set to '1b', The DPLL in OTG UTMI PHY will be force on even if the PHY is in suspend mode. When this bit is set to '0b', DPLL clock will stop when the PHY entering suspend mode. This feature is used when Host2 controller core and/or Host3 controller core use the PHY DPLL as clock source. 1 Enable 0 Disable
9 OTG_OVER_CUR_POL	OTG Polarity of Overcurrent This bit sets the active level of OTG port overcurrent detector output. 1 Low active 0 High active
8 OTG_OVER_CUR_DIS	OTG Disable Overcurrent Detector This bit enables/disables OTG overcurrent detector. 1 Disables the overcurrent detector 0 Enables the overcurrent detector
7 OTG_WK_PLL_EN	OTG PHY 480MHz DPLL automatic power-up by OTG wake-up event enable. If set '1b', the OTG UTMI DPLL will automatically be enabled when a wake-up event is detected on OTG port. 1 Enable automatic power-up of OTG PHY DPLL 0 Disable automatic power-up of OTG PHY DPLL
6 H1_OVER_CUR_POL	Host1 Polarity of Overcurrent This bit sets the active level of Host1 port overcurrent detector output. 1 Low active 0 High active
5 H1_OVER_CUR_DIS	Host1 Disable Overcurrent Detector This bit enables/disables Host1 overcurrent detector. 1 Disables the overcurrent detector 0 Enables the overcurrent detector
4 H1_WK_PLL_EN	Host1 PHY 480MHz DPLL automatic power-up by Host1 wake-up event enable. If set '1b', the Host1 UTMI DPLL will automatically be enabled when a wake-up event is detected on Host1 port.

Table continues on the next page...

USB_USB_OTG_PHY_CTRL_0 field descriptions (continued)

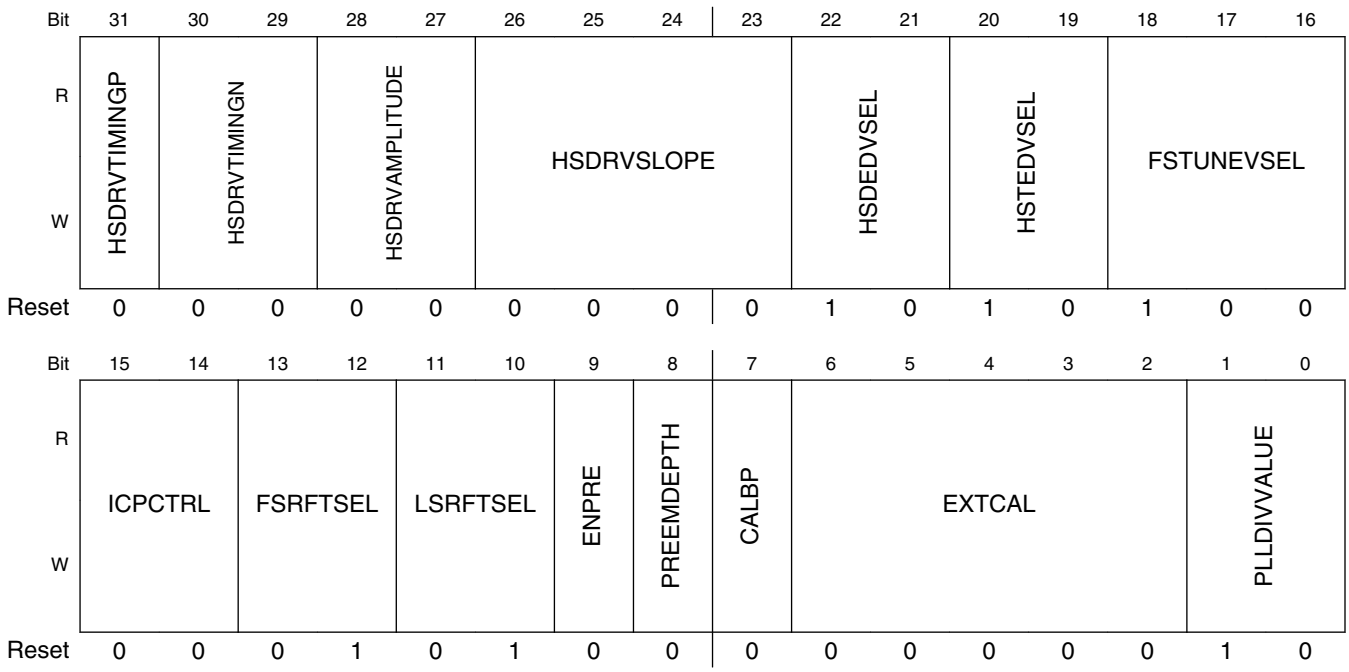
Field	Description
	1 Enable automatic power-up of Host1 PHY DPLL 0 Disable automatic power-up of Host1 PHY DPLL
3 -	Reserved
2 CHRGDET	OTG UTMI PHY Charger Detector Output. **NOTE:** Maximum response time = 1ms 1 Detected 0 not detected
1 CHRGDET_INT_EN	OTG UTMI PHY Charger detected interrupt enable 1 Enable 0 Disable
0 CHRGDET_INT_FLG	OTG UTMI PHY Charger detected interrupt flag This bit is cleared when CHRGDET_INT_EN is set '0b'. 1 Interrupt generated 0 No interrupt

77.5.3 USB OTG UTMI PHY Control Register 1 (USB_USB_OTG_PHY_CTRL_1)

USB OTG UTMI PHY control register 1 also controls the on-chip UTMI PHY.

.

Address: USB_USB_OTG_PHY_CTRL_1 is 53F8_0000h base + 80Ch offset = 53F8_080Ch



USB_USB_OTG_PHY_CTRL_1 field descriptions

Field	Description
31 HSDRVTIMINGP	HS driver timing control for PMOS 1 8x 0 2x
30–29 HSDRVTIMINGN	HS driver timing control for NMOS 00 2x 01 4x 10 6x 11 8x
28–27 HSDRVAMPLITUDE	HS driver amplitude control 00 I (I = 17.78 mA) 01 I + 2.5% 10 I + 5% 11 I + 7.5%
26–23 HSDRVSLOPE	HS driver slope control The HS Driver may have its rise/fall times controlled using the hsdrvslope[3:0] control pins. Additional charge is injected, so the HS driver Rise/Fall times change (RC constant changes).
22–21 HSDEDVSEL	Reference voltage for high speed disconnect envelope detector 00 (46 + 2)/100 x vbg (556.8 mV) 01 (46 + 3)/100 x vbg (568.4 mV) 10 (46 + 4)/100 x vbg (580 mV) 11 (46 + 5)/100 x vbg (591.6 mV)

Table continues on the next page...

USB_USB_OTG_PHY_CTRL_1 field descriptions (continued)

Field	Description
20–19 HSTEDVSEL	Reference voltage for high speed transmission envelope detector 00 vbg x (9)/100 (104.4 mV) 01 vbg x (9+1)/100 (116 mV) 10 vbg x (9+2)/100 (127.6 mV) 11 vbg x (9+3)/100 (139.2 mV)
18–16 FSTUNEVSEL	Reference voltage control for Calibration circuit 000 (46)/100 x vbg (533.6 mV) 001 (46 + 1)/100 x vbg (545.2 mV) 010 (46 + 2)/100 x vbg (556.8 mV) 011 (46 + 3)/100 x vbg (568.4 mV) 100 (46 + 4)/100 x vbg (580 mV) 101 (46 + 5)/100 x vbg (591.6 mV) 110 (46 + 6)/100 x vbg (603.2 mV) 111 (46 + 7)/100 x vbg (614.8 mV)
15–14 ICPCTRL	PLL charge pump current control 00 Icp (Icp = 40 mA) 01 Icp x 0.5 10 Icp x 1.5 11 Icp x 2
13–12 FSRFTSEL	FS driver rise/fall time control 00 Nominal FS rise time - 30% 01 Nominal FS rise time 10 Nominal FS rise time 11 Nominal FS rise time + 30%
11–10 LSRFTSEL	LS driver rise/fall time control 00 Nominal LS rise time - 30% 01 Nominal LS rise time 10 Nominal LS rise time 11 Nominal LS rise time + 30%
9 ENPRE	HS driver pre-emphasis Enable 1 Enable 0 Disable
8 PREEMDEPTH	HS driver pre-emphasis depth 00 I (I = 17.78 mA) 01 I + 5% 10 I + 10% 11 I + 20%
7 CALBP	Calibration Bypass Enable for both dp and dn lines

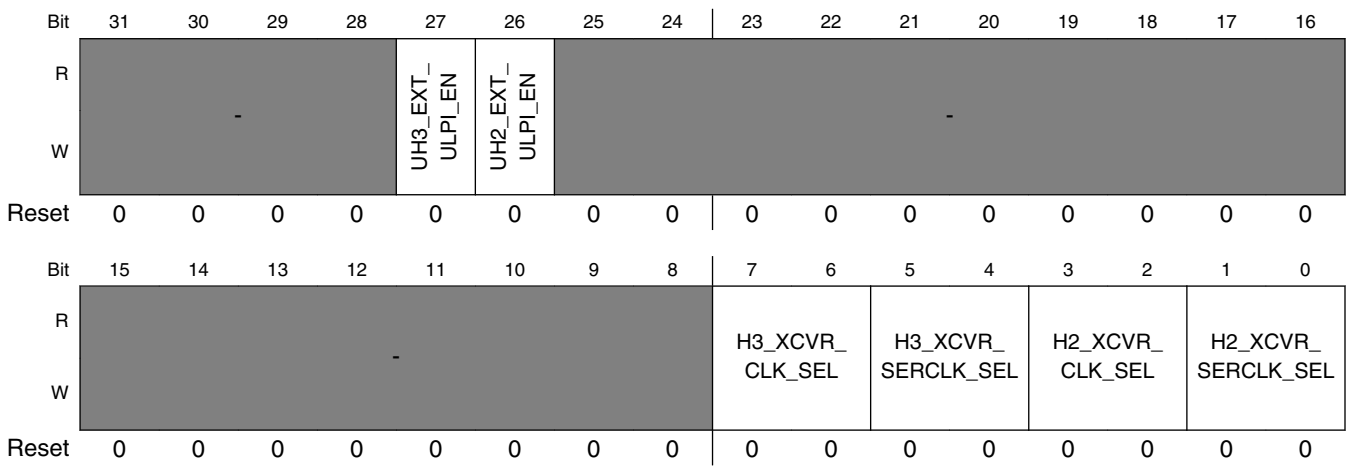
Table continues on the next page...

USB_USB_OTG_PHY_CTRL_1 field descriptions (continued)

Field	Description
6–2 EXTCAL	Controls calibration value externally. This bit is valid only if calbp bit is set to '1b'.
1–0 PLLDIVVALUE	Selects reference clock frequence 00 sysclock uses 19.2 MHz 01 sysclock uses 24 MHz 10 sysclock uses 26 MHz 11 sysclock uses 27 MHz

77.5.4 USB Control Register 1 (USB_USB_CTRL_1)

Address: USB_USB_CTRL_1 is 53F8_0000h base + 810h offset = 53F8_0810h



USB_USB_CTRL_1 field descriptions

Field	Description
31–28 -	Reserved
27 UH3_EXT_ULPI_EN	Host3 selects ULPI data interface or Serial interface This bit must be set to '1b' before Host3 core is set to ULPI mode. It should be clear to '0b' before Host3 core is set to FS/LS serial mode. 1 Select ULPI data interface 0 Select Serial data interface
26 UH2_EXT_ULPI_EN	Host2 selects ULPI data interface or Serial interface This bit must be set to '1b' before Host2 core is set to ULPI mode. It should be clear to '0b' before Host2 core is set to FS/LS serial mode. 1 Select ULPI data interface 0 Select Serial data interface

Table continues on the next page...

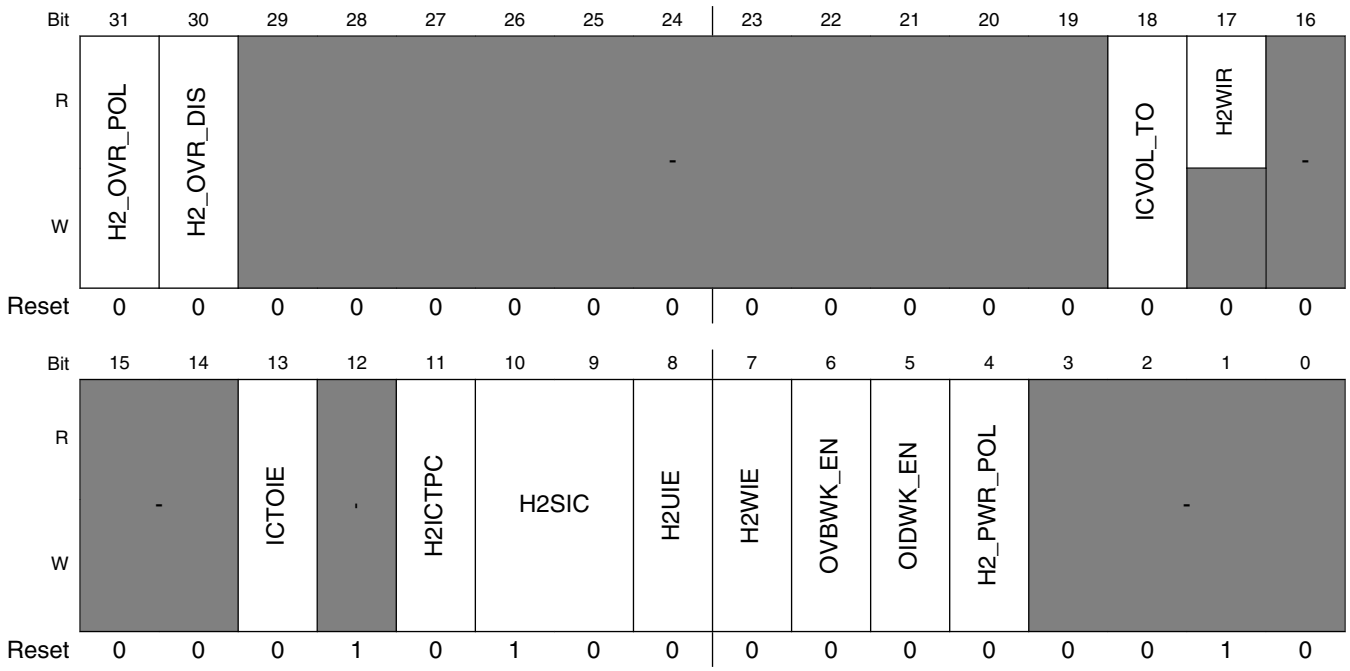
USB_USB_CTRL_1 field descriptions (continued)

Field	Description
25–8 -	Reserved
7–6 H3_XCVR_CLK_SEL	Select the clock source for Host3 Transceiver clock. Host3 can support Serial and ULPI transceivers. When used with ULPI PHY, the ULPI PHY clock must be used as source. For Serial PHY either settings 00 or 11 can be used. 00 Clock sourced from OTG UTMI PHY DPLL 01 Clock from ULPI PHY (Host3 ULPI clock PAD) 10 Reserved 11 Clock from on-chip system DPLL (less accurate)
5–4 H3_XCVR_SERCLK_SEL	Select the clock source for Host3 with serial transceiver. In general, the OTG UTMI PHY DPLL sould be used as it is more accurate.However, the system DPLL can be used if needed in special cases. 00 Clock sourced from OTG UTMI PHY DPLL 01 Reserved 10 Reserved 11 Clock from on-chip system DPLL (less accurate)
3–2 H2_XCVR_CLK_SEL	Select the clock source for Host2 Transceiver clock. Host2 can support Serial and ULPI transceivers. When used with ULPI PHY, the ULPI PHY clock must be used as source. For Serial PHY either settings 00 or 11 can be used. 00 Clock sourced from OTG UTMI PHY DPLL 01 Clock from ULPI PHY (Host2 ULPI clock PAD) 10 Reserved 11 Clock from on-chip system DPLL (less accurate)
1–0 H2_XCVR_SERCLK_SEL	Select the clock source for Host2 with serial transceiver. In general, the OTG UTMI PHY DPLL sould be used as it is more accurate.However, the system DPLL can be used if needed in special cases. 00 Clock sourced from OTG UTMI PHY DPLL 01 Reserved 10 Reserved 11 Clock from on-chip system DPLL (less accurate)

77.5.5 USB Host2 Control Register (USB_USB_UH2_CTRL)

The USB Host2 control register controls the integration specific features of the USB host2 block. These features are not directly related to the USB functionality, but control special features, interfacing on the USB ports, as well as power control and wake-up functionality.

Address: USB_USB_UH2_CTRL is 53F8_0000h base + 814h offset = 53F8_0814h



USB_USB_UH2_CTRL field descriptions

Field	Description
31 H2_OVR_POL	Host2 Polarity of Overcurrent This bit sets the active level of Host2 port overcurrent detector output. 1 Low active 0 High active
30 H2_OVR_DIS	Host2 Disable Overcurrent Detector This bit enables/disables Host2 overcurrent detector. 1 Disables the overcurrent detector 0 Enables the overcurrent detector
29–19 -	Reserved
18 ICVOL_TO	Host2 IC-USB voltage detection time out interrupt 1 Interrupt generated 0 No interrupt

Table continues on the next page...

USB_USB_UH2_CTRL field descriptions (continued)

Field	Description
17 H2WIR	Host2 Wake-up Interrupt Request This bit indicates that a wake-up interrupt request is issued by the OTG port. This bit is cleared when disabling the wake-up interrupt (clearing bit "H2WIE"). 1 Wake-up interrupt request detected 0 No wake-up interrupt request detected
16–14 -	Reserved
13 ICTOIE	Host2 IC-USB voltage detection time out interrupt enable 1 Enable 0 Disable
12 -	Reserved
11 H2ICTPC	Host2 Clear IC-USB TP interrupt Flag 1 Clear 0 Not clear (Default)
10–9 H2SIC	Host2 Serial Interface Configuration These bits control the interface type of the Host2 port when used with a serial transceiver. These bit fields allow for configuring the serial interface for single-ended or differential operation combined with bidirectional or unidirectional operation. The available settings are described below. **NOTE:** When using bidirectional mode, the IOMUX must be configured to place the DP/DM pad into loopback mode. 00 Differential/unidirectional (6-wire) 01 Differential/bidirectional (4-wire) 10 Single ended/unidirectional (6-wire) (Default) 11 Single ended/unidirectional (6-wire)
8 H2UIE	Host2 Waku-up by ULPI transceiver Interrupt Enable This bit controls whether Host2 controller core will be wake-up by interrupts from the ULPI transceiver. This bit is only meaningful when a ULPI transceiver is selected. 1 Enable 0 Disable (Default)
7 H2WIE	Host 2 Wake-up Interrupt Enable This bit enables or disables the Host 2 wake-up interrupt. Disabling the interrupt also clears the Interrupt request bit. Wake-up interrupt enable should be turned off after receiving a wake-up interrupt and turned on again prior to going in suspend mode 1 Interrupt Enabled 0 Interrupt Disabled
6 OVBWK_EN	OTG VBUS Wake-up Enable This bit enables or disables the interrupt which is caused OTG VBUS Change.

Table continues on the next page...

USB_USB_UH2_CTRL field descriptions (continued)

Field	Description
	1 Interrupt Enabled 0 Interrupt Disabled
5 OIDWK_EN	OTG ID change Wake-up Enable This bit enables or disables the interrupt which is caused OTG ID Change. 1 Interrupt Enabled 0 Interrupt Disabled
4 H2_PWR_POL	Host2 power Pin polarity The polarity of Host2 Power pin(to enable external PMIC to drive VBUS) 1 High active 0 Low active
3–0 -	Reserved

77.5.6 USB Host3 Control Register (USB_USB_UH3_CTRL)

The USB Host3 control register controls the integration specific features of the USB host3 block. These features are not directly related to the USB functionality, but control special features, interfacing on the USB ports, as well as power control and wake-up functionality.

Address: USB_USB_UH3_CTRL is 53F8_0000h base + 818h offset = 53F8_0818h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	H3_OVR_POL	H3_OVR_DIS	-									H2_SER_DRV_EN	H3_SER_DRV_EN	-	H3WIR	-
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-					H3SIC	H3UIE	H3WIE	-			H3_PWR_POL	-			
W																
Reset	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0

USB_USB_UH3_CTRL field descriptions

Field	Description
31 H3_OVR_POL	Host3 Polarity of Overcurrent This bit sets the active level of Host3 port overcurrent detector output. 1 Low active 0 High active
30 H3_OVR_DIS	Host3 Disable Overcurrent Detector This bit enables/disables Host3 overcurrent detector. 1 Disables the overcurrent detector 0 Enables the overcurrent detector
29–21 -	Reserved
20 H2_SER_DRV_EN	Host2 Serial Interface Drive Enable Host2 core serial interface output enable. By default all serial interface output pins are not driven by the controller. This bit should be enabled after Host2 core switches to serial mode and should be disabled before Host2 core switches to ULPI mode. 1 Serial interface drive enable 0 Serial interface drive disable (Default)
19 H3_SER_DRV_EN	Host3 Serial Interface Drive Enable Host3 core serial interface output enable. By default all serial interface output pins are not driven by the controller. This bit should be enabled after Host3 core switches to serial mode and should be disabled before Host3 core switches to ULPI mode. 1 Serial interface drive enable 0 Serial interface drive disable (Default)
18 -	Reserved
17 H3WIR	Host3 Wake-up Interrupt Request This bit indicates that a wake-up interrupt request is issued by the Host3 port. This bit is cleared when disabling the wake-up interrupt (clearing bit "H3WIE"). 1 Wake-up interrupt request detected 0 No wake-up interrupt request detected
16–11 -	Reserved
10–9 H3SIC	Host3 Serial Interface Configuration These bits Control the interface type of the Host3 port when used with a serial transceiver. These bit fields allow for configuring the serial interface for single-ended or differential operation combined with bidirectional or unidirectional operation. The available settings are described below. **NOTE:** In case of Bidirectional mode, IOMUX must set the DP/DM pad into loopback mode. 00 Differential / Unidirectional (6-wire) 01 Differential / Bidirectional (4-wire) 10 Single Ended / Unidirectional (6-wire) (Default) 11 Single Ended / Bidirectional (3-wire)

Table continues on the next page...

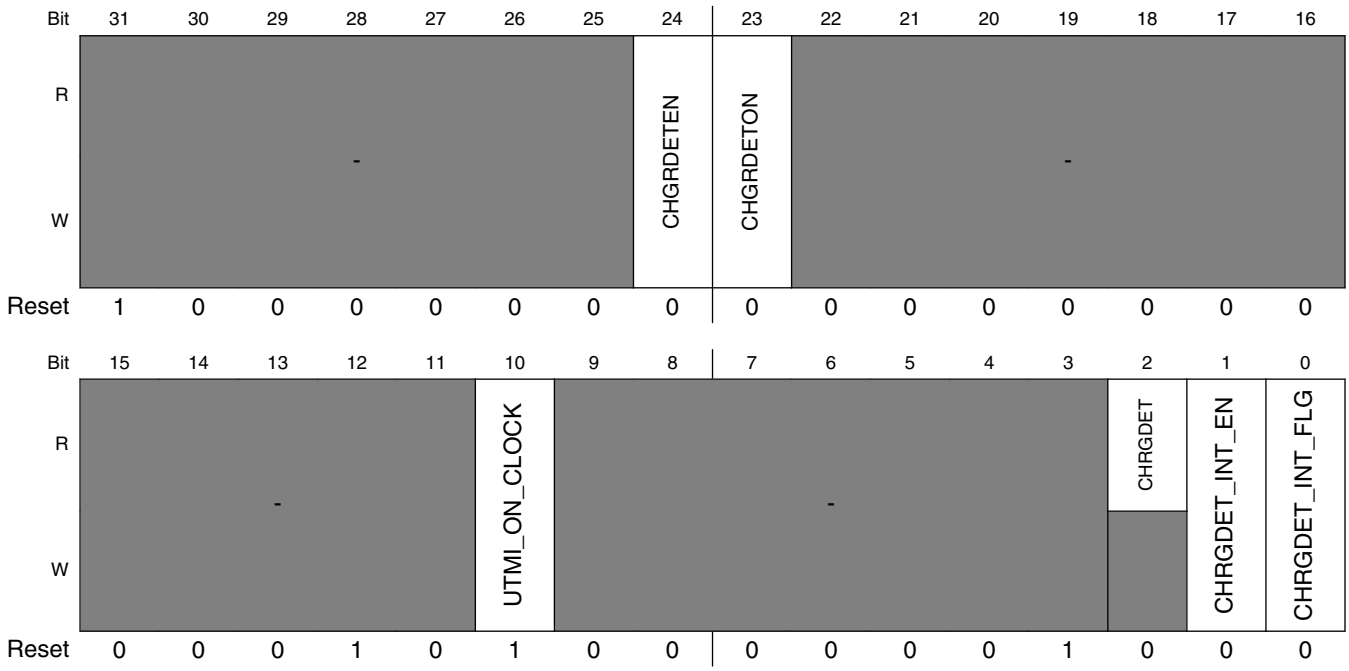
USB_USB_UH3_CTRL field descriptions (continued)

Field	Description
8 H3UIE	Host3 Waku-up by ULPI transceiver Interrupt Enable This bit controls whether Host3 controller core will be wake-up by interrupts from the ULPI transceiver. This bit is only meaningful when a ULPI transceiver is selected. 1 Enable 0 Disable (Default)
7 H3WIE	Host3 Wake-up Interrupt Enable This bit enables or disables the Host3 wake-up interrupt. Disabling the interrupt also clears the Interrupt request bit "H3WIR". Wake-up interrupt enable should be turned off after receiving a wake-up interrupt and turned on again prior to entering suspend mode 1 Interrupt Enabled 0 Interrupt Disabled (Default)
6–5 -	Reserved
4 H3_PWR_POL	Host3 power Pin polarity The polarity of Host3 Power pin (to enable external PMIC to drive VBUS) 1 High active 0 Low active
3–0 -	Reserved

77.5.7 USB Host1 UTMI PHY Control Register 0 (USB_USB_UH1_PHY_CTRL_0)

USB Host1 UTMI PHY Control Register 0 are used to control the on-chip UTMI PHY.

Address: USB_USB_UH1_PHY_CTRL_0 is 53F8_0000h base + 81Ch offset = 53F8_081Ch



USB_USB_UH1_PHY_CTRL_0 field descriptions

Field	Description
31–25 -	Reserved
24 CHGRDETEN	UTMI PHY Charger Detector Enable This bit should be set for 300 ms (or more) after CHGRDETON bit is set to '1b'. 1 Active 0 Inactive
23 CHGRDETON	UTMI PHY Charger Detector Power On This bit controls the internal current mirrors used for charger detection. When this bit is set '1b', internal current mirrors are powered on. This bit should be set to '1b' at least 20 ms before CHGRDETEN bit is set to '1b'.
22–11 -	Reserved
10 UTMI_ON_ CLOCK	UTMI PHY Clock On When this bit is set to '1b', The PLL in OTG UTMI PHY will be force on even if the PHY is in suspend mode. When this bit is set to '0b', PLL clock will stop when the PHY entering suspend mode.

Table continues on the next page...

USB_USB_UH1_PHY_CTRL_0 field descriptions (continued)

Field	Description
	This feature is used when Host2 controller core and/or Host3 controller core use the PHY PLL as clock source. 1 Enable 0 Disable
9–3 -	Reserved
2 CHRGDET	Host1 UTMI PHY Charger Detector Output. **NOTE:** Maximum response time = 1ms 1 Detected 0 not detected
1 CHRGDET_INT_EN	Host1 UTMI PHY Charger detected interrupt enable 1 Enable 0 Disable
0 CHRGDET_INT_FLG	Host1 UTMI PHY Charger detected interrupt flag This bit is cleared when CHRGDET_INT_EN is set '0b'. 1 Interrupt generated 0 No interrupt

77.5.8 USB Host1 UTMI PHY Control Register 1 (USB_USB_UH1_PHY_CTRL_1)

USB Host1 UTMI PHY Control Register 1 are also used to control the on-chip UTMI PHY.

Address: USB_USB_UH1_PHY_CTRL_1 is 53F8_0000h base + 820h offset = 53F8_0820h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HSDRVTIMINGP	HSDRVTIMINGN		HSDRVAMPLITUDE		HSDRVSLOPE			HSDEDVSEL		HSTEDVSEL			FSTUNEVSEL		
Reset	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ICPCTRL		FSRFTSEL		LSRFTSEL		ENPRE	PREEMDEPTH	CALBP	EXTCAL					PLLDIVVALUE	
Reset	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1

USB_USB_UH1_PHY_CTRL_1 field descriptions

Field	Description
31 HSDRVTIMINGP	HS driver timing control for PMOS 1 8x 0 2x
30–29 HSDRVTIMINGN	HS driver timing control for NMOS 00 2x 01 4x 10 6x 11 8x
28–27 HSDRVAMPLITUDE	HS driver amplitude control 00 (I = 17.78 mA) 01 I + 2.5% 10 I + 5% 11 I + 7.5%

Table continues on the next page...

USB_USB_UH1_PHY_CTRL_1 field descriptions (continued)

Field	Description
26–23 HSDRVSLOPE	HS driver slope control The HS Driver may have its rise/fall times controlled using the hsdrvslope[3:0] control pins. Additional charge is injected, so the HS driver Rise/Fall times change (RC constant changes). Rise/Fall times: Depends on the Package, PCB... Correct value result from Silicon tests
22–21 HSDEDVSEL	Reference voltage for high speed disconnect envelope detector 00 (46+2)/100 x vbg (556.8 mV) 01 (46+3)/100 x vbg (568.4 mV) 10 (46+4)/100 x vbg (580 mV) 11 (46+5)/100 x vbg (591.6 mV)
20–19 HSTEDVSEL	Reference voltage for high speed transmission envelope detector 00 vbg x (9)/100 (104.4 mV) 01 vbg x (9+1)/100 (116 mV) 10 vbg x (9+2)/100 (127.6 mV) 11 vbg x (9+3)/100 (139.2 mV)
18–16 FSTUNEVSEL	Reference voltage control for Calibration circuit 000 (46)/100 x vbg (533.6 mV) 001 (46+1)/100 x vbg (545.2 mV) 010 (46+2)/100 x vbg (556.8 mV) 011 (46+3)/100 x vbg (568.4 mV) 100 (46+4)/100 x vbg (580 mV) 101 (46+5)/100 x vbg (591.6 mV) 110 (46+6)/100 x vbg (603.2 mV) 111 (46+7)/100 x vbg (614.8 mV)
15–14 ICPCTRL	DPLL charge pump current control 00 Icp (Icp = 40 uA) 01 Icp x 0.5 10 Icp x 1.5 11 Icp x 2
13–12 FSRFTSEL	FS driver rise/fall time control 00 Nominal FS rise time - 30% 01 Nominal FS rise time 10 Nominal FS rise time 11 Nominal FS rise time + 30%
11–10 LSRFTSEL	LS driver rise/fall time control 00 Nominal LS rise time - 30% 01 Nominal LS rise time 10 Nominal LS rise time 11 Nominal LS rise time + 30%
9 ENPRE	HS driver pre-emphasis enable

Table continues on the next page...

USB_USB_UH1_PHY_CTRL_1 field descriptions (continued)

Field	Description
8 PREEMDEPTH	HS driver pre-emphasis depth 00 I (I = 17.78 mA) 01 I + 5% 10 I + 10% 11 I + 20%
7 CALBP	Calibration Bypass Enable for both dp and dn lines
6–2 EXTCAL	Controls calibration value externally. This bit is valid only if calbp bit is set to '1b'.
1–0 PLLDIVVALUE	Selects reference clock frequence 00 sysclock uses 19.2 MHz 01 sysclock uses 24 MHz 10 sysclock uses 26 MHz 11 sysclock uses 27 MHz

77.5.9 USB Clock on/off control Register (USB_USB_CLKONOFF_CTRL)

Address: USB_USB_CLKONOFF_CTRL is 53F8_0000h base + 824h offset = 53F8_0824h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	-	H3_PLL_CK_OFF	H2_PLL_CK_OFF	-						H3_INT60_CK_OFF	H2_INT60_CK_OFF	H3_AHBCLK_OFF	H2_AHBCLK_OFF	H1_AHBCLK_OFF	OTG_AHBCLK_OFF	-
Reset	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-															
Reset	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0

USB_USB_CLKONOFF_CTRL field descriptions

Field	Description
31 -	Reserved
30 H3_PLL_CK_ OFF	Enable/disable the clock which comes from the on-chip DPLL for Host3 1 Turn off the on-chip DPLL clock for the host3 0 Turn on the on-chip DPLL clock for the host3
29 H2_PLL_CK_ OFF	Enable/disable the clock which comes from the on-chip DPLL for Host2 1 Turn off the on-chip DPLL clock for the host2 0 Turn on the on-chip DPLL clock for the host2
28–23 -	Reserved
22 H3_INT60_CK_ OFF	Enable/disable the clock which comes from the internal 60Mhz clock generator for Host3 1 Turn off the internal 60 MHz clock for the host3 0 Turn on the internal 60 MHz clock for the host3
21 H2_INT60_CK_ OFF	Enable/disable the clock which comes from the internal 60Mhz clock generator for Host2 1turn off the internal 60 MHz clock for the host2 0turn on the internal 60 MHz clock for the host2
20 H3_AHBCLK_ OFF	Enable/disable the AHB clock for Host3 This bit is used to enable/disable the AHB clock for Host3. 1 Turn off the AHB clock for the host3 0 Turn on the AHB clock for the host3
19 H2_AHBCLK_ OFF	Enable/disable the AHB clock for Host2 This bit is used to enable/disable the AHB clock for Host2. 1 Turn off the AHB clock for the host2 0 Turn on the AHB clock for the host2
18 H1_AHBCLK_ OFF	Enable/disable the AHB clock for Host1 This bit is used to enable/disable the AHB clock for Host1. 1 Turn off the AHB clock for the host1 0 Turn on the AHB clock for the host1
17 OTG_AHBCLK_ OFF	Enable/disable the AHB clock for OTG This bit is used to enable/disable the AHB clock for OTG. 1 Turn off the AHB clock for the OTG 0 Turn on the AHB clock for the OTG
16–0 -	Reserved

77.6 USB Core Memory Map/Register Definition

USB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53F8_0000	Identification register (USB_UOG_ID)	32	R	E401_FA05h	77.6.1/ 4907
53F8_0004	Hardware General (USB_UOG_HWGENERAL)	32	R	See section	77.6.2/ 4908
53F8_0008	Host Hardware Parameters (USB_UOG_HWHOST)	32	R	1002_0001h	77.6.3/ 4909
53F8_000C	Device Hardware Parameters (USB_UOG_HWDEVICE)	32	R	0000_0011h	77.6.4/ 4910
53F8_0010	TX Buffer Hardware Parameters (USB_UOG_HWTXBUF)	32	R	8008_0B08h	77.6.5/ 4911
53F8_0014	RX Buffer Hardware Parameters (USB_UOG_HWRXBUF)	32	R	0000_0808h	77.6.6/ 4911
53F8_0080	General Purpose Timer #0 Load (USB_UOG_GPTIMER0LD)	32	R/W	0000_0000h	77.6.7/ 4912
53F8_0084	General Purpose Timer #0 Controller (USB_UOG_GPTIMER0CTRL)	32	R/W	0000_0000h	77.6.8/ 4913
53F8_0088	General Purpose Timer #1 Load (USB_UOG_GPTIMER1LD)	32	R/W	0000_0000h	77.6.9/ 4914
53F8_008C	General Purpose Timer #1 Controller (USB_UOG_GPTIMER1CTRL)	32	R/W	0000_0000h	77.6.10/ 4915
53F8_0090	System Bus Config (USB_UOG_SBUSCFG)	32	R/W	0000_0002h	77.6.11/ 4916
53F8_0100	Capability Register Length (USB_UOG_CAPLENGTH)	8	R	40h	77.6.12/ 4917
53F8_0102	Host Controller Interface Version (USB_UOG_HCIVERSION)	16	R	0100h	77.6.13/ 4917
53F8_0104	Host Controller Structural Parameters (USB_UOG_HCSPARAMS)	32	R	0001_0011h	77.6.14/ 4918
53F8_0108	Host Controller Capability Parameters (USB_UOG_HCCPARAMS)	32	R	0000_0006h	77.6.15/ 4920
53F8_0120	Device Controller Interface Version (USB_UOG_DCIVERSION)	16	R	0001h	77.6.16/ 4921
53F8_0124	Device Controller Capability Parameters (USB_UOG_DCCPARAMS)	32	R	0000_0188h	77.6.17/ 4922

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53F8_0140	USB Command Register (USB_UOG_USBCMD)	32	R/W	0008_0000h	77.6.18/ 4922
53F8_0144	USB Status Register (USB_UOG_USBSTS)	32	R/W	0000_0000h	77.6.19/ 4926
53F8_0148	Interrupt Enable Register (USB_UOG_USBINTR)	32	R/W	0000_0000h	77.6.20/ 4929
53F8_014C	USB Frame Index (USB_UOG_FRINDEX)	32	R/W	0000_0000h	77.6.21/ 4931
53F8_0154	Frame List Base Address (USB_UOG_PERIODICLISTBASE)	32	R/W	0000_0000h	77.6.22/ 4932
53F8_0154	Device Address (USB_UOG_DEVICEADDR)	32	R/W	0000_0000h	77.6.23/ 4933
53F8_0158	Next Asynch. Address (USB_UOG_ASYNCLISTADDR)	32	R/W	0000_0000h	77.6.24/ 4934
53F8_0158	Endpoint List Address (USB_UOG_ENDPTLISTADDR)	32	R/W	0000_0000h	77.6.25/ 4935
53F8_0160	Programmable Burst Size (USB_UOG_BURSTSIZE)	32	R/W	0000_0000h	77.6.26/ 4935
53F8_0164	TX FIFO Fill Tuning (USB_UOG_TXFILLTUNING)	32	R/W	See section	77.6.27/ 4936
53F8_016C	IC_USB enable and voltage negotiation (USB_UOG_IC_USB)	32	R/W	0000_0000h	77.6.28/ 4938
53F8_0178	Endpoint NAK (USB_UOG_ENDPTNAK)	32	R/W	0000_0000h	77.6.29/ 4939
53F8_017C	Endpoint Nake Enable (USB_UOG_ENDPTNAKEN)	32	R/W	0000_0000h	77.6.30/ 4939
53F8_0184	Port Status & Control (USB_UOG_PORTSC1)	32	R/W	1000_0000h	77.6.31/ 4940
53F8_01A4	On-The-Go Status & control (USB_UOG_OTGSC)	32	R/W	0000_0120h	77.6.32/ 4947
53F8_01A8	USB Device Mode (USB_UOG_USBMODE)	32	R/W	0000_0000h	77.6.33/ 4950
53F8_01AC	Endpoint Setup Status (USB_UOG_ENDPTSETUPSTAT)	32	R/W	0000_0000h	77.6.34/ 4951
53F8_01B0	Endpoint Initialization (USB_UOG_ENDPTPRIME)	32	R/W	0000_0000h	77.6.35/ 4952
53F8_01B4	Endpoint De-Initialize (USB_UOG_ENDPTFLUSH)	32	R/W	0000_0000h	77.6.36/ 4953
53F8_01B8	Endpoint Status (USB_UOG_ENDPTSTAT)	32	R	0000_0000h	77.6.37/ 4953

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53F8_01BC	Endpoint Complete (USB_UOG_ENDPTCOMPLETE)	32	R/W	0000_0000h	77.6.38/ 4954
53F8_01C0	Endpoint Control0 (USB_UOG_ENDPTCTRL0)	32	R/W	0080_0080h	77.6.39/ 4955
53F8_01C4	Endpoint Controln (USB_UOG_ENDPTCTRL1)	32	R/W	0000_0000h	77.6.40/ 4956
53F8_01C8	Endpoint Controln (USB_UOG_ENDPTCTRL2)	32	R/W	0000_0000h	77.6.40/ 4956
53F8_01CC	Endpoint Controln (USB_UOG_ENDPTCTRL3)	32	R/W	0000_0000h	77.6.40/ 4956
53F8_01D0	Endpoint Controln (USB_UOG_ENDPTCTRL4)	32	R/W	0000_0000h	77.6.40/ 4956
53F8_01D4	Endpoint Controln (USB_UOG_ENDPTCTRL5)	32	R/W	0000_0000h	77.6.40/ 4956
53F8_01D8	Endpoint Controln (USB_UOG_ENDPTCTRL6)	32	R/W	0000_0000h	77.6.40/ 4956
53F8_01DC	Endpoint Controln (USB_UOG_ENDPTCTRL7)	32	R/W	0000_0000h	77.6.40/ 4956
53F8_0200	Identification register (USB_UH1_ID)	32	R	E401_FA05h	77.6.1/ 4907
53F8_0204	Hardware General (USB_UH1_HWGENERAL)	32	R	See section	77.6.2/ 4908
53F8_0208	Host Hardware Parameters (USB_UH1_HWHOST)	32	R	1002_0001h	77.6.3/ 4909
53F8_0210	TX Buffer Hardware Parameters (USB_UH1_HWTXBUF)	32	R	8008_0B08h	77.6.5/ 4911
53F8_0214	RX Buffer Hardware Parameters (USB_UH1_HWRXBUF)	32	R	0000_0808h	77.6.6/ 4911
53F8_0280	General Purpose Timer #0 Load (USB_UH1_GPTIMER0LD)	32	R/W	0000_0000h	77.6.7/ 4912
53F8_0284	General Purpose Timer #0 Controller (USB_UH1_GPTIMER0CTRL)	32	R/W	0000_0000h	77.6.8/ 4913
53F8_0288	General Purpose Timer #1 Load (USB_UH1_GPTIMER1LD)	32	R/W	0000_0000h	77.6.9/ 4914
53F8_028C	General Purpose Timer #1 Controller (USB_UH1_GPTIMER1CTRL)	32	R/W	0000_0000h	77.6.10/ 4915
53F8_0290	System Bus Config (USB_UH1_SBUSCFG)	32	R/W	0000_0002h	77.6.11/ 4916
53F8_0300	Capability Register Length (USB_UH1_CAPLENGTH)	8	R	40h	77.6.12/ 4917

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53F8_0302	Host Controller Interface Version (USB_UH1_HCIVERSION)	16	R	0100h	77.6.13/ 4917
53F8_0304	Host Controller Structural Parameters (USB_UH1_HCSPARAMS)	32	R	0001_0011h	77.6.14/ 4918
53F8_0308	Host Controller Capability Parameters (USB_UH1_HCCPARAMS)	32	R	0000_0006h	77.6.15/ 4920
53F8_0340	USB Command Register (USB_UH1_USBCMD)	32	R/W	0008_0000h	77.6.18/ 4922
53F8_0344	USB Status Register (USB_UH1_USBSTS)	32	R/W	0000_0000h	77.6.19/ 4926
53F8_0348	Interrupt Enable Register (USB_UH1_USBINTR)	32	R/W	0000_0000h	77.6.20/ 4929
53F8_034C	USB Frame Index (USB_UH1_FRINDEX)	32	R/W	0000_0000h	77.6.21/ 4931
53F8_0354	Frame List Base Address (USB_UH1_PERIODICLISTBASE)	32	R/W	0000_0000h	77.6.22/ 4932
53F8_0358	Next Asynch. Address (USB_UH1_ASYNCLISTADDR)	32	R/W	0000_0000h	77.6.24/ 4934
53F8_0360	Programmable Burst Size (USB_UH1_BURSTSIZE)	32	R/W	0000_0000h	77.6.26/ 4935
53F8_0364	TX FIFO Fill Tuning (USB_UH1_TXFILLTUNING)	32	R/W	See section	77.6.27/ 4936
53F8_036C	IC_USB enable and voltage negotiation (USB_UH1_IC_USB)	32	R/W	0000_0000h	77.6.28/ 4938
53F8_0384	Port Status & Control (USB_UH1_PORTSC1)	32	R/W	1000_0000h	77.6.31/ 4940
53F8_03A8	USB Device Mode (USB_UH1_USBMODE)	32	R/W	0000_0000h	77.6.33/ 4950
53F8_0400	Identification register (USB_UH2_ID)	32	R	E401_FA05h	77.6.1/ 4907
53F8_0404	Hardware General (USB_UH2_HWGENERAL)	32	R	See section	77.6.2/ 4908
53F8_0408	Host Hardware Parameters (USB_UH2_HWHOST)	32	R	1002_0001h	77.6.3/ 4909
53F8_0410	TX Buffer Hardware Parameters (USB_UH2_HWTXBUF)	32	R	8008_0B08h	77.6.5/ 4911
53F8_0414	RX Buffer Hardware Parameters (USB_UH2_HWRXBUF)	32	R	0000_0808h	77.6.6/ 4911
53F8_0480	General Purpose Timer #0 Load (USB_UH2_GPTIMER0LD)	32	R/W	0000_0000h	77.6.7/ 4912

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53F8_0484	General Purpose Timer #0 Controller (USB_UH2_GPTIMER0CTRL)	32	R/W	0000_0000h	77.6.8/ 4913
53F8_0488	General Purpose Timer #1 Load (USB_UH2_GPTIMER1LD)	32	R/W	0000_0000h	77.6.9/ 4914
53F8_048C	General Purpose Timer #1 Controller (USB_UH2_GPTIMER1CTRL)	32	R/W	0000_0000h	77.6.10/ 4915
53F8_0490	System Bus Config (USB_UH2_SBUSCFG)	32	R/W	0000_0002h	77.6.11/ 4916
53F8_0500	Capability Register Length (USB_UH2_CAPLENGTH)	8	R	40h	77.6.12/ 4917
53F8_0502	Host Controller Interface Version (USB_UH2_HCIVERSION)	16	R	0100h	77.6.13/ 4917
53F8_0504	Host Controller Structural Parameters (USB_UH2_HCSPARAMS)	32	R	0001_0011h	77.6.14/ 4918
53F8_0508	Host Controller Capability Parameters (USB_UH2_HCCPARAMS)	32	R	0000_0006h	77.6.15/ 4920
53F8_0540	USB Command Register (USB_UH2_USBCMD)	32	R/W	0008_0000h	77.6.18/ 4922
53F8_0544	USB Status Register (USB_UH2_USBSTS)	32	R/W	0000_0000h	77.6.19/ 4926
53F8_0548	Interrupt Enable Register (USB_UH2_USBINTR)	32	R/W	0000_0000h	77.6.20/ 4929
53F8_054C	USB Frame Index (USB_UH2_FRINDEX)	32	R/W	0000_0000h	77.6.21/ 4931
53F8_0554	Frame List Base Address (USB_UH2_PERIODICLISTBASE)	32	R/W	0000_0000h	77.6.22/ 4932
53F8_0558	Next Asynch. Address (USB_UH2_ASYNCLISTADDR)	32	R/W	0000_0000h	77.6.24/ 4934
53F8_0560	Programmable Burst Size (USB_UH2_BURSTSIZE)	32	R/W	0000_0000h	77.6.26/ 4935
53F8_0564	TX FIFO Fill Tuning (USB_UH2_TXFILLTUNING)	32	R/W	See section	77.6.27/ 4936
53F8_056C	IC_USB enable and voltage negotiation (USB_UH2_IC_USB)	32	R/W	0000_0000h	77.6.28/ 4938
53F8_0584	Port Status & Control (USB_UH2_PORTSC1)	32	R/W	1000_0000h	77.6.31/ 4940
53F8_05A8	USB Device Mode (USB_UH2_USBMODE)	32	R/W	0000_0000h	77.6.33/ 4950
53F8_0600	Identification register (USB_UH3_ID)	32	R	E401_FA05h	77.6.1/ 4907

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53F8_0604	Hardware General (USB_UH3_HWGENERAL)	32	R	See section	77.6.2/ 4908
53F8_0608	Host Hardware Parameters (USB_UH3_HWHOST)	32	R	1002_0001h	77.6.3/ 4909
53F8_0610	TX Buffer Hardware Parameters (USB_UH3_HWTXBUF)	32	R	8008_0B08h	77.6.5/ 4911
53F8_0614	RX Buffer Hardware Parameters (USB_UH3_HWRXBUF)	32	R	0000_0808h	77.6.6/ 4911
53F8_0680	General Purpose Timer #0 Load (USB_UH3_GPTIMER0LD)	32	R/W	0000_0000h	77.6.7/ 4912
53F8_0684	General Purpose Timer #0 Controller (USB_UH3_GPTIMER0CTRL)	32	R/W	0000_0000h	77.6.8/ 4913
53F8_0688	General Purpose Timer #1 Load (USB_UH3_GPTIMER1LD)	32	R/W	0000_0000h	77.6.9/ 4914
53F8_068C	General Purpose Timer #1 Controller (USB_UH3_GPTIMER1CTRL)	32	R/W	0000_0000h	77.6.10/ 4915
53F8_0690	System Bus Config (USB_UH3_SBUSCFG)	32	R/W	0000_0002h	77.6.11/ 4916
53F8_0700	Capability Register Length (USB_UH3_CAPLENGTH)	8	R	40h	77.6.12/ 4917
53F8_0702	Host Controller Interface Version (USB_UH3_HCIVERSION)	16	R	0100h	77.6.13/ 4917
53F8_0704	Host Controller Structural Parameters (USB_UH3_HCSPARAMS)	32	R	0001_0011h	77.6.14/ 4918
53F8_0708	Host Controller Capability Parameters (USB_UH3_HCCPARAMS)	32	R	0000_0006h	77.6.15/ 4920
53F8_0740	USB Command Register (USB_UH3_USBCMD)	32	R/W	0008_0000h	77.6.18/ 4922
53F8_0744	USB Status Register (USB_UH3_USBSTS)	32	R/W	0000_0000h	77.6.19/ 4926
53F8_0748	Interrupt Enable Register (USB_UH3_USBINTR)	32	R/W	0000_0000h	77.6.20/ 4929
53F8_074C	USB Frame Index (USB_UH3_FRINDEX)	32	R/W	0000_0000h	77.6.21/ 4931
53F8_0754	Frame List Base Address (USB_UH3_PERIODICLISTBASE)	32	R/W	0000_0000h	77.6.22/ 4932
53F8_0758	Next Asynch. Address (USB_UH3_ASYNCLISTADDR)	32	R/W	0000_0000h	77.6.24/ 4934
53F8_0760	Programmable Burst Size (USB_UH3_BURSTSIZE)	32	R/W	0000_0000h	77.6.26/ 4935

Table continues on the next page...

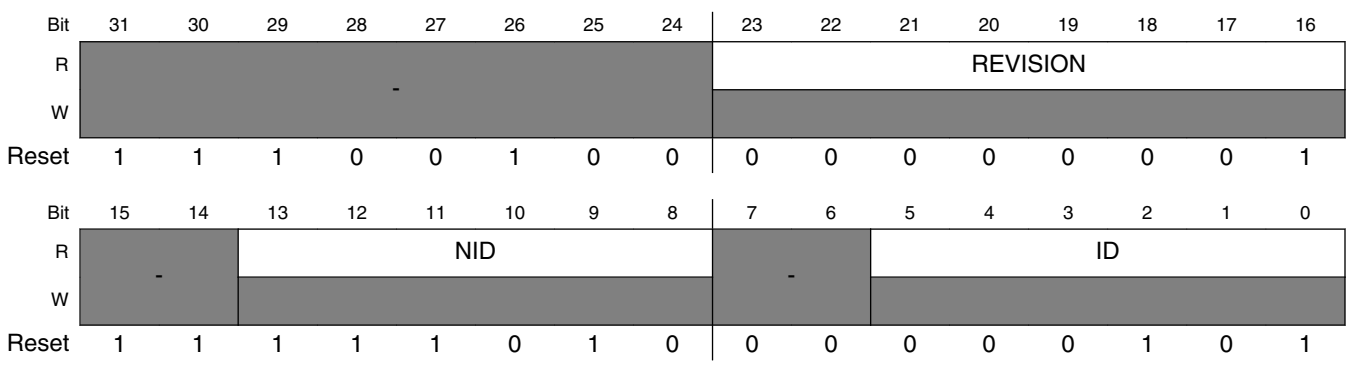
USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53F8_0764	TX FIFO Fill Tuning (USB_UH3_TXFILLTUNING)	32	R/W	See section	77.6.27/ 4936
53F8_076C	IC_USB enable and voltage negotiation (USB_UH3_IC_USB)	32	R/W	0000_0000h	77.6.28/ 4938
53F8_0770	ULPI Viewport (USB_UOG_ULPIVIEW)	32	R/W	0000_0000h	77.6.41/ 4959
53F8_0784	Port Status & Control (USB_UH3_PORTSC1)	32	R/W	1000_0000h	77.6.31/ 4940
53F8_07A8	USB Device Mode (USB_UH3_USBMODE)	32	R/W	0000_0000h	77.6.33/ 4950
53F8_0970	ULPI Viewport (USB_UH1_ULPIVIEW)	32	R/W	0000_0000h	77.6.41/ 4959
53F8_0B70	ULPI Viewport (USB_UH2_ULPIVIEW)	32	R/W	0000_0000h	77.6.41/ 4959
53F8_0D70	ULPI Viewport (USB_UH3_ULPIVIEW)	32	R/W	0000_0000h	77.6.41/ 4959

77.6.1 Identification register (USB__ID)

The ID register identifies the USB 2.0 OTG High-Speed core and its revision.

Addresses: USB_UOG_ID is 53F8_0000h base + 0h offset = 53F8_0000h
USB_UH1_ID is 53F8_0000h base + 200h offset = 53F8_0200h
USB_UH2_ID is 53F8_0000h base + 400h offset = 53F8_0400h
USB_UH3_ID is 53F8_0000h base + 600h offset = 53F8_0600h



USB_*n*_ID field descriptions

Field	Description
31–24 -	Reserved

Table continues on the next page...

USB_*n*_ID field descriptions (continued)

Field	Description
23–16 REVISION	Revision number of the controller core.
15–14 -	Reserved
13–8 NID	Complement version of ID
7–6 -	Reserved
5–0 ID	Configuration number. This number is set to 0x05 and indicates that the peripheral is USB 2.0 OTG High-Speed core.

77.6.2 Hardware General (USB__HWGENERAL)

General hardware parameters as defined in System Level Issues and Core Configuration.

Addresses: USB_UOG_HWGENERAL is 53F8_0000h base + 4h offset = 53F8_0004h
USB_UH1_HWGENERAL is 53F8_0000h base + 204h offset = 53F8_0204h
USB_UH2_HWGENERAL is 53F8_0000h base + 404h offset = 53F8_0404h
USB_UH3_HWGENERAL is 53F8_0000h base + 604h offset = 53F8_0604h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	-															
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-					SM		PHYM			PHYW		-			
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	1*	0*	1*	0*	1*

* Notes:
- The reset value dependson the implementation.

USB_*n*_HWGENERAL field descriptions

Field	Description
31–11 -	Reserved
10–9 SM	Serial interface mode capability For OTG/Host1/Host2/Host3 controller core, reset value is '10b'. 00 No Serial Engine, always use parallel signalling.

Table continues on the next page...

USB_*n*_HWGENERAL field descriptions (continued)

Field	Description
	01 Serial Engine present, always use serial signalling for FS/LS. 10 Software programmable - Reset to use parallel signalling for FS/LS 11 Software programmable - Reset to use serial signalling for FS/LS
8–6 PHYM	Transciever type For OTG controller core, reset value is '0h'; For Host1/Host2/Host3 controller core, reset value is '7h'. 000 UTMI/UMTI+ 001 ULPI DDR 010 ULPI 011 Serial Only 100 Software programmable - reset to UTMI/UTMI+ 101 Software programmable - reset to ULPI DDR 110 Software programmable - reset to ULPI 111 Software programmable - reset to Serial 1000 IC-USB 1001 Software programmable - reset to IC-USB 1010 HSIC 1011 Software programmable - reset to HSIC
5–4 PHYW	Data width of the transciever connected to the controller core. For OTG/Host1/Host2/Host3 controller core, reset value is '11b'. 00 8 bit wide data bus [60MHz clock from the transciever] 01 16 bit wide data bus [30MHZ clock from the transciever] 10 software programmable reset to 8-bit width 11 software programmable reset to 16-bit width
3–0 -	Reserved

77.6.3 Host Hardware Parameters (USB__HWHOST)

Host hardware parameters as defined in System Level Issues and Core Configuration.

Addresses: USB_UOG_HWHOST is 53F8_0000h base + 8h offset = 53F8_0008h
USB_UH1_HWHOST is 53F8_0000h base + 208h offset = 53F8_0208h
USB_UH2_HWHOST is 53F8_0000h base + 408h offset = 53F8_0408h
USB_UH3_HWHOST is 53F8_0000h base + 608h offset = 53F8_0608h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																												NPORT			HC
W																																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

USB_*n*_HWHOST field descriptions

Field	Description
31–4 -	Reserved
3–1 NPORT	The Nmber of downstream ports supported by the host controller is NPORT+1. As all 4 controller cores are single port, these bits are set to '000b'.
0 HC	Host Capable. All 4 controller cores support host operation mode. 1 support host operation mode 0 not support

77.6.4 Device Hardware Parameters (USB_UOG_HWDEVICE)

Device hardware parameters are as defined in System Level Issues and Core Configuration.

This register is only available in OTG core.

Address: USB_UOG_HWDEVICE is 53F8_0000h base + Ch offset = 53F8_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																										DEVEP					DC
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

USB_UOG_HWDEVICE field descriptions

Field	Description
31–6 -	Reserved
5–1 DEVEP	Device Endpoint Number OTG controller core has 8 Endpoints.
0 DC	Device Capable. 1 support device operation mode 0 not support

77.6.5 TX Buffer Hardware Parameters (USB__HWTXBUF)

TX buffer hardware parameters are as defined in System Level Issues and Core Configuration.

Addresses: USB_UOG_HWTXBUF is 53F8_0000h base + 10h offset = 53F8_0010h
USB_UH1_HWTXBUF is 53F8_0000h base + 210h offset = 53F8_0210h
USB_UH2_HWTXBUF is 53F8_0000h base + 410h offset = 53F8_0410h
USB_UH3_HWTXBUF is 53F8_0000h base + 610h offset = 53F8_0610h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-								TXCHANADD								-								TXBURST							
W																																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1	0	0	0

USB_*n*_HWTXBUF field descriptions

Field	Description
31–24 -	Reserved
23–16 TXCHANADD	TX Buffer size is: (2^TXCHANADD) * 4 Bytes. These bits are set to '08h', so buffer size is 256*4 Bytes. For OTG controller core, there is one TX Buffer for each endpoint, so totally 8 TX Buffers. For Host1/Host2/Host3 controller core, there is only one TX Buffer.
15–8 -	Reserved
7–0 TXBURST	Default burst size for memory to TX buffer transfer. This is reset value of TXPBURST bits in USB core regsiter UOG_BURSTSIZE. Please see Programmable Burst Size (USB__BURSTSIZE) .

77.6.6 RX Buffer Hardware Parameters (USB__HWRXBUF)

RX buffer hardware parameters are as defined in System Level Issues and Core Configuration.

Addresses: USB_UOG_HWRXBUF is 53F8_0000h base + 14h offset = 53F8_0014h
USB_UH1_HWRXBUF is 53F8_0000h base + 214h offset = 53F8_0214h
USB_UH2_HWRXBUF is 53F8_0000h base + 414h offset = 53F8_0414h
USB_UH3_HWRXBUF is 53F8_0000h base + 614h offset = 53F8_0614h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																RXADD								RXBURST							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0							

USB_*n*_HWRXBUF field descriptions

Field	Description
31–16 -	Reserved
15–8 RXADD	Buffer total size for all receive endpoints is (2^RXADD). RX Buffer size is: (2^RXADD) * 4 Bytes. These bits are set to '08h', so buffer size is 256*4 Bytes. For OTG controller core, there is one RX Buffer, shared by 8 Endpoints. For Host1/Host2/Host3 controller core, there is only one RX Buffer.
7–0 RXBURST	Default burst size for memory to RX buffer transfer. This is reset value of RXPBURST bits in USB core regsiter UOG_BURSTSIZE. Please see Programmable Burst Size (USB__BURSTSIZE) .

77.6.7 General Purpose Timer #0 Load (USB__GPTIMER0LD)

This register controls load value of the count timer in register n_GPTIMER0CTRL. Please see _GPTIMER0CTRLGeneral Purpose Timer #0 Controller .

Addresses: USB_UOG_GPTIMER0LD is 53F8_0000h base + 80h offset = 53F8_0080h
USB_UH1_GPTIMER0LD is 53F8_0000h base + 280h offset = 53F8_0280h
USB_UH2_GPTIMER0LD is 53F8_0000h base + 480h offset = 53F8_0480h
USB_UH3_GPTIMER0LD is 53F8_0000h base + 680h offset = 53F8_0680h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-								GPTLD																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*_GPTIMER0LD field descriptions

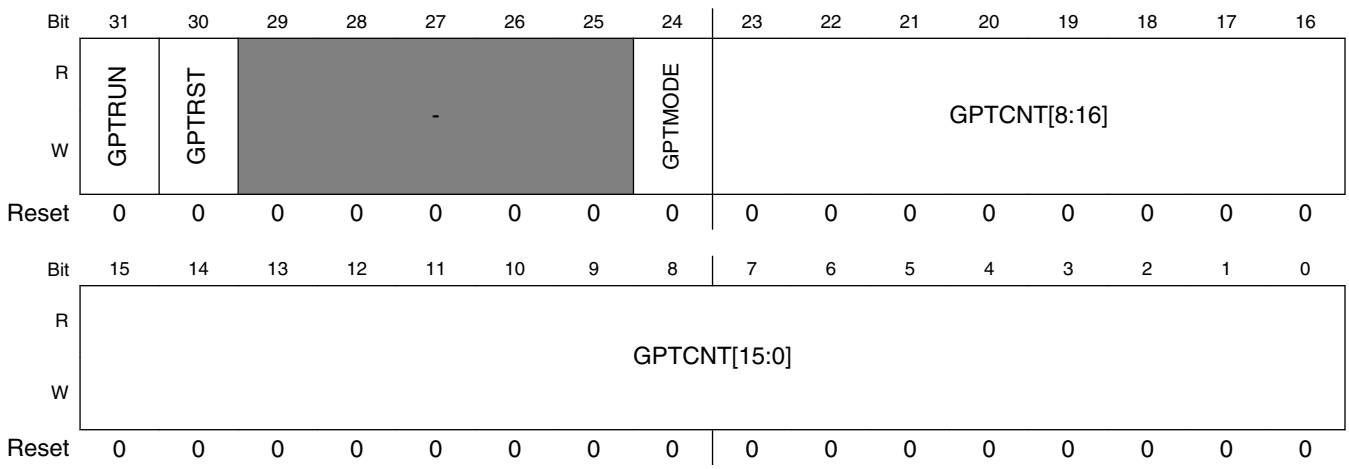
Field	Description
31–24 -	Reserved
23–0 GPTLD	General Purpose Timer Load Value These bit fields are loaded to GPTCNT bits when GPTRST bit is set '1b'. This value represents the time in microseconds minus 1 for the timer duration. Example: for a one millisecond timer, load 1000-1=999 or 0x0003E7. **NOTE:** Max value is 0xFFFFFF or 16.777215 seconds.

77.6.8 General Purpose Timer #0 Controller (USB__GPTIMER0CTRL)

This register contains the control for this countdown timer and a data field can be queried to determine the running count value. This timer has granularity on 1 us and can be programmed to a little over 16 seconds. There are two counter modes which are described in the register table below. When the timer counter value transitions to zero, an interrupt could be generated if enable.

Interrupt status bit is TI0 bit in n_USBSTS register (See USB Status Register (USB__USBSTS)), interrupt enable bit is TIE0 bit in n_USBINTR register. (See Interrupt Enable Register (USB__USBINTR) .)

Addresses: USB_UOG_GPTIMER0CTRL is 53F8_0000h base + 84h offset = 53F8_0084h
USB_UH1_GPTIMER0CTRL is 53F8_0000h base + 284h offset = 53F8_0284h
USB_UH2_GPTIMER0CTRL is 53F8_0000h base + 484h offset = 53F8_0484h
USB_UH3_GPTIMER0CTRL is 53F8_0000h base + 684h offset = 53F8_0684h



USB_*n*_GPTIMER0CTRL field descriptions

Field	Description
31 GPTRUN	General Purpose Timer Run GPTCNT bits are not effected when setting or clearing this bit. 0 Stop counting 1 Run
30 GPTRST	General Purpose Timer Reset 0 No action 1 Load counter value from GPTLD bits in n_GPTIMER0LD
29–25 -	Reserved

Table continues on the next page...

USB_*n*_GPTIMER0CTRL field descriptions (continued)

Field	Description
24 GPTMODE	General Purpose Timer Mode In one shot mode, the timer will count down to zero, generate an interrupt, and stop until the counter is reset by software; In repeat mode, the timer will count down to zero, generate an interrupt and automatically reload the counter value from GPTLD bits to start again. 0 One Shot Mode 1 Repeat Mode
23–0 GPTCNT	General Purpose Timer Counter. This field is the count value of the countdown timer.

77.6.9 General Purpose Timer #1 Load (USB__GPTIMER1LD)

This register controls load value of the count timer in register n_GPTIMER1CTRL. Please see _GPTIMER1CTRLGeneral Purpose Timer #1 Controller .

Addresses: USB_UOG_GPTIMER1LD is 53F8_0000h base + 88h offset = 53F8_0088h
USB_UH1_GPTIMER1LD is 53F8_0000h base + 288h offset = 53F8_0288h
USB_UH2_GPTIMER1LD is 53F8_0000h base + 488h offset = 53F8_0488h
USB_UH3_GPTIMER1LD is 53F8_0000h base + 688h offset = 53F8_0688h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-								GPTLD																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*_GPTIMER1LD field descriptions

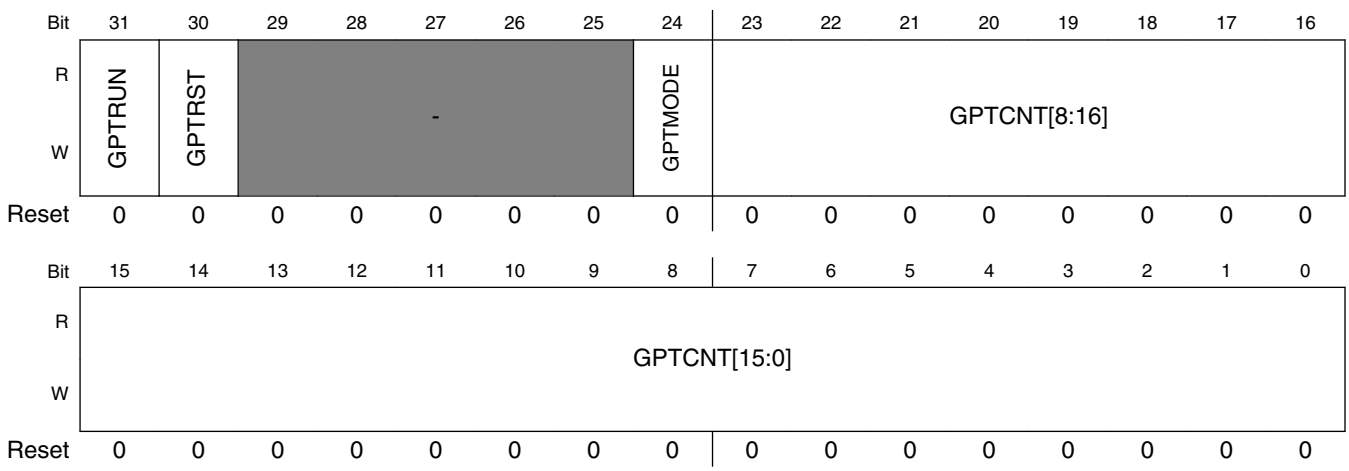
Field	Description
31–24 -	Reserved
23–0 GPTLD	General Purpose Timer Load Value These bit fields are loaded to GPTCNT bits when GPTRST bit is set '1b'. This value represents the time in microseconds minus 1 for the timer duration. Example: for a one millisecond timer, load 1000-1=999 or 0x0003E7. **NOTE:** Max value is 0xFFFFFF or 16.777215 seconds.

77.6.10 General Purpose Timer #1 Controller (USB__GPTIMER1CTRL)

This register contains the control for this countdown timer and a data field can be queried to determine the running count value. This timer has granularity on 1 us and can be programmed to a little over 16 seconds. There are two counter modes which are described in the register table below. When the timer counter value transitions to zero, an interrupt could be generated if enable.

Interrupt status bit is TI1 bit in UOG_USBSTS register (See USB Status Register (USB__USBSTS)), interrupt enable bit is TIE1 bit in n_USBINTR register (See Interrupt Enable Register (USB__USBINTR)).

Addresses: USB_UOG_GPTIMER1CTRL is 53F8_0000h base + 8Ch offset = 53F8_008Ch
USB_UH1_GPTIMER1CTRL is 53F8_0000h base + 28Ch offset = 53F8_028Ch
USB_UH2_GPTIMER1CTRL is 53F8_0000h base + 48Ch offset = 53F8_048Ch
USB_UH3_GPTIMER1CTRL is 53F8_0000h base + 68Ch offset = 53F8_068Ch



USB_*n*_GPTIMER1CTRL field descriptions

Field	Description
31 GPTRUN	General Purpose Timer Run GPTCNT bits are not effected when setting or clearing this bit. 0 Stop counting 1 Run
30 GPTRST	General Purpose Timer Reset 0 No action 1 Load counter value from GPTLD bits in UOG_GPTIMER0LD
29–25 -	Reserved

Table continues on the next page...

USB_*n*_GPTIMER1CTRL field descriptions (continued)

Field	Description
24 GPTMODE	General Purpose Timer Mode In one shot mode, the timer will count down to zero, generate an interrupt, and stop until the counter is reset by software. In repeat mode, the timer will count down to zero, generate an interrupt and automatically reload the counter value from GPTLD bits to start again. 0 One Shot Mode 1 Repeat Mode
23–0 GPTCNT	General Purpose Timer Counter. This field is the count value of the countdown timer.

77.6.11 System Bus Config (USB__SBUSCFG)

Addresses: USB_UOG_SBUSCFG is 53F8_0000h base + 90h offset = 53F8_0090h
USB_UH1_SBUSCFG is 53F8_0000h base + 290h offset = 53F8_0290h
USB_UH2_SBUSCFG is 53F8_0000h base + 490h offset = 53F8_0490h
USB_UH3_SBUSCFG is 53F8_0000h base + 690h offset = 53F8_0690h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-																													AHBBRST		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

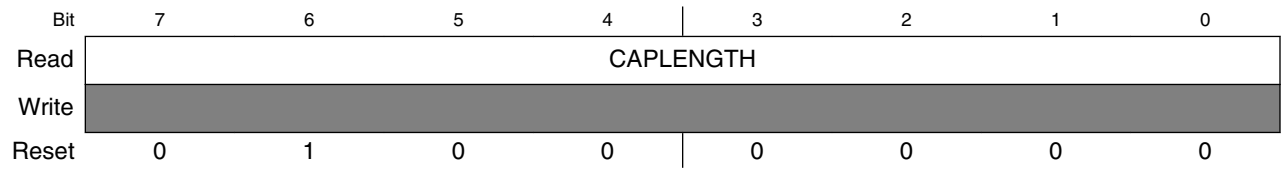
USB_*n*_SBUSCFG field descriptions

Field	Description
31–3 -	Reserved
2–0 AHBBRST	AHB master interface Burst configuration These bits controls AHB master transfer type sequence (or priority). **NOTE:** This register overrides n_BURSTSIZE register when its value is not zero. 000 Incremental burst of unspecified length only 001 INCR4 burst, then single transfer 010 INCR8 burst, INCR4 burst, then single transfer 011 INCR16 burst, INCR8 burst, INCR4 burst, then single transfer 100 Reserved, don't use 101 INCR4 burst, then incremental burst of unspecified length 110 INCR8 burst, INCR4 burst, then incremental burst of unspecified length 111 INCR16 burst, INCR8 burst, INCR4 burst, then incremental burst of unspecified length

77.6.12 Capability Register Length (USB__CAPLENGTH)

The following figure Capability Register Length (n_CAPLENGTH) indicates the offset that should be added to the register base address at the beginning of the Operational Register.

Addresses: USB_UOG_CAPLENGTH is 53F8_0000h base + 100h offset = 53F8_0100h
USB_UH1_CAPLENGTH is 53F8_0000h base + 300h offset = 53F8_0300h
USB_UH2_CAPLENGTH is 53F8_0000h base + 500h offset = 53F8_0500h
USB_UH3_CAPLENGTH is 53F8_0000h base + 700h offset = 53F8_0700h



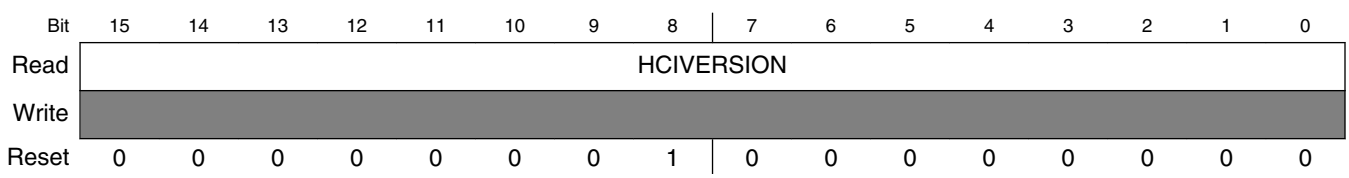
USB_n_CAPLENGTH field descriptions

Field	Description
7–0 CAPLENGTH	These bits are used as an offset to add to register base to find the beginning of the Operational Register. Default value is '40h'.

77.6.13 Host Controller Interface Version (USB__HCIVERSION)

The following figure shows the Host Interface version number (n_HCIVERSION), which is a 2-byte register containing a BCD encoding of the EHCI revision number supported by this host controller. The most significant byte of this register represents a major revision and the least significant byte is the minor revision.

Addresses: USB_UOG_HCIVERSION is 53F8_0000h base + 102h offset = 53F8_0102h
USB_UH1_HCIVERSION is 53F8_0000h base + 302h offset = 53F8_0302h
USB_UH2_HCIVERSION is 53F8_0000h base + 502h offset = 53F8_0502h
USB_UH3_HCIVERSION is 53F8_0000h base + 702h offset = 53F8_0702h



USB_n_HCIVERSION field descriptions

Field	Description
15–0 HCIVERSION	Host Controller Interface Version Number Default value is '10h', which means EHCI rev1.0.

77.6.14 Host Controller Structural Parameters (USB__HCSPARAMS)

The following figure shows the port steering logic capabilities of Host Control Structural Parameters (n_HCSPARAMS).

Addresses: USB_UOG_HCSPARAMS is 53F8_0000h base + 104h offset = 53F8_0104h
USB_UH1_HCSPARAMS is 53F8_0000h base + 304h offset = 53F8_0304h
USB_UH2_HCSPARAMS is 53F8_0000h base + 504h offset = 53F8_0504h
USB_UH3_HCSPARAMS is 53F8_0000h base + 704h offset = 53F8_0704h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-				N_TT				N_PTT				-			PI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	N_CC				N_PCC				-			PPC	N_PORTS			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

USB_*n*_HCSPARAMS field descriptions

Field	Description
31–28 -	Reserved
27–24 N_TT	Number of Transaction Translators (N_TT). Default value '0000b' This field indicates the number of embedded transaction translators associated with the USB2.0 host controller. These bits would be set to '0001b' for Multi-Port Host, and '0000b' for Single-Port Host.
23–20 N_PTT	Number of Ports per Transaction Translator (N_PTT). Default value '0000b' This field indicates the number of ports assigned to each transaction translator within the USB2.0 host controller. These bits would be set to equal N_PORTS for Multi-Port Host, and '0000b' for Single-Port Host.
19–17 -	Reserved
16 PI	Port Indicators (P INDICATOR) This bit indicates whether the ports support port indicator control. When set to one, the port status and control registers include a read/writeable field for controlling the state of the port indicator This bit is "1b" in all 4 controller core.
15–12 N_CC	Number of Companion Controller (N_CC). This field indicates the number of companion controllers associated with this USB2.0 host controller. These bits are '0000b' in all 4 controller core.

Table continues on the next page...

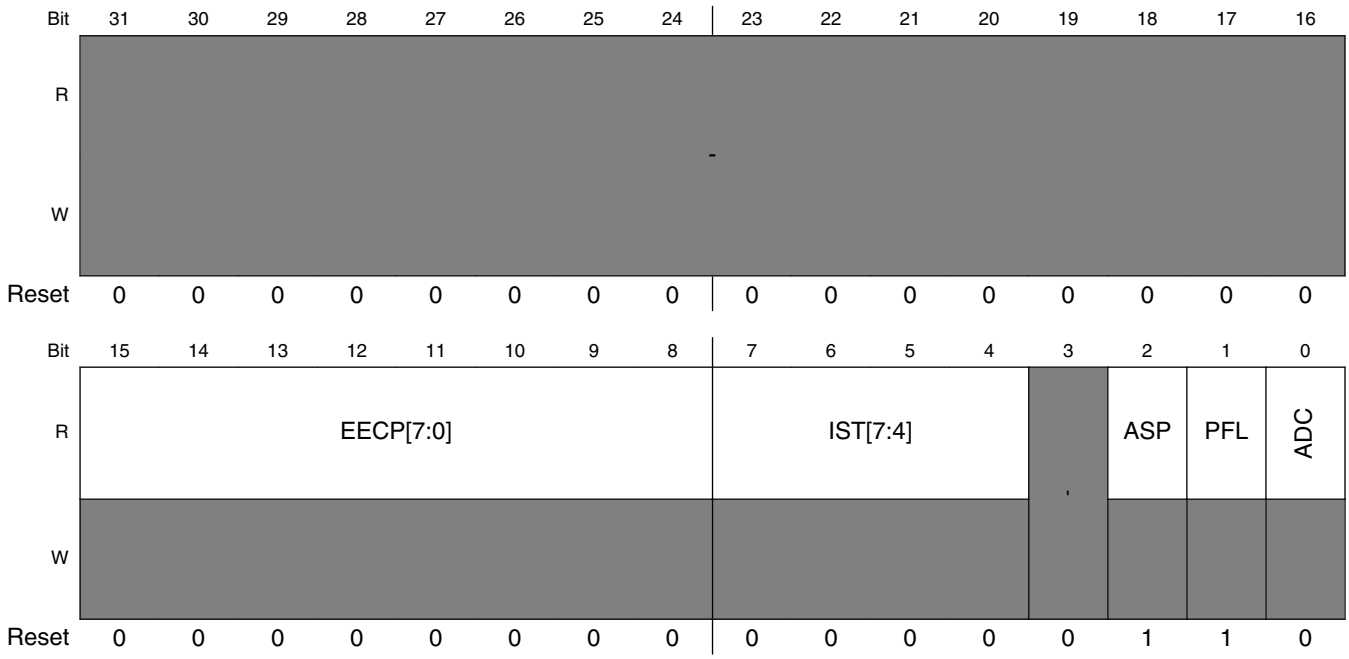
USB_*n*_HCSPARAMS field descriptions (continued)

Field	Description
	0 There is no internal Companion Controller and port-ownership hand-off is not supported. 1 There are internal companion controller(s) and port-ownership hand-offs is supported.
11–8 N_PCC	Number of Ports per Companion Controller This field indicates the number of ports supported per internal Companion Controller. It is used to indicate the port routing configuration to the system software. For example, if N_PORTS has a value of 6 and N_CC has a value of 2 then N_PCC could have a value of 3. The convention is that the first N_PCC ports are assumed to be routed to companion controller 1, the next N_PCC ports to companion controller 2, etc. In the previous example, the N_PCC could have been 4, where the first 4 are routed to companion controller 1 and the last two are routed to companion controller 2. The number in this field must be consistent with N_PORTS and N_CC. These bits are '0000b' in all 4 controller core.
7–5 -	Reserved
4 PPC	Port Power Control This field indicates whether the host controller implementation includes port power control. A one indicates the ports have port power switches. A zero indicates the ports do not have port power switches. The value of this field affects the functionality of the Port Power field in each port status and control register
3–0 N_PORTS	Number of downstream ports. This field specifies the number of physical downstream ports implemented on this host controller. The value of this field determines how many port registers are addressable in the Operational Register. Valid values are in the range of 1h to Fh. A zero in this field is undefined. These bits are always set to '0001b' because all 4 controller cores are Single-Port Host.

77.6.15 Host Controller Capability Parameters (USB__HCCPARAMS)

This register identifies multiple mode control (time-base bit functionality), addressing capability.

Addresses: USB_UOG_HCCPARAMS is 53F8_0000h base + 108h offset = 53F8_0108h
USB_UH1_HCCPARAMS is 53F8_0000h base + 308h offset = 53F8_0308h
USB_UH2_HCCPARAMS is 53F8_0000h base + 508h offset = 53F8_0508h
USB_UH3_HCCPARAMS is 53F8_0000h base + 708h offset = 53F8_0708h



USB_*n*_HCCPARAMS field descriptions

Field	Description
31–16 -	Reserved
15–8 EECP[7:0]	EHCI Extended Capabilities Pointer. This field indicates the existence of a capabilities list. A value of 00h indicates no extended capabilities are implemented. A non-zero value in this register indicates the offset in PCI configuration space of the first EHCI extended capability. The pointer value must be 40h or greater if implemented to maintain the consistency of the PCI header defined for this class of device. These bits are set '00h' in all 4 controller core.
7–4 IST[7:4]	Isochronous Scheduling Threshold. This field indicates, relative to the current position of the executing host controller, where software can reliably update the isochronous schedule. When bit [7] is zero, the value of the least significant 3 bits indicates the number of micro-frames a host controller can hold a set of isochronous data structures (one or more) before flushing the state. When bit [7] is a one, then host software assumes the host controller may cache an isochronous data structure for an entire frame. These bits are set '00h' in all 4 controller core.

Table continues on the next page...

USB_*n*_HCCPARAMS field descriptions (continued)

Field	Description
3 -	Reserved
2 ASP	Asynchronous Schedule Park Capability If this bit is set to a one, then the host controller supports the park feature for high-speed queue heads in the Asynchronous Schedule. The feature can be disabled or enabled and set to a specific level by using the *Asynchronous Schedule Park Mode Enable* and *Asynchronous Schedule Park Mode Count* fields in the USBCMD register. This bit is set '1b' in all 4 controller core.
1 PFL	Programmable Frame List Flag If this bit is set to zero, then the system software must use a frame list length of 1024 elements with this host controller. The USBCMD register Frame List Size field is a read-only register and must be set to zero. If set to a one, then the system software can specify and use a smaller frame list and configure the host controller via the USBCMD register Frame List Size field. The frame list must always be aligned on a 4K-page boundary. This requirement ensures that the frame list is always physically contiguous. This bit is set '1b' in all 4 controller core.
0 ADC	64-bit Addressing Capability This bit is set '0b' in all 4 controller core, no 64-bit addressing capability is supported.

77.6.16 Device Controller Interface Version (USB_UOG_DCIVERSION)

This register indicates the two-byte BCD encoding of the device controller interface version number.

Address: USB_UOG_DCIVERSION is 53F8_0000h base + 120h offset = 53F8_0120h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	DCIVERSION															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

USB_UOG_DCIVERSION field descriptions

Field	Description
15–0 DCIVERSION	Device Controller Interface Version Number Default value is '01h', which means rev0.1.

77.6.17 Device Controller Capability Parameters (USB_UOG_DCCPARAMS)

These fields describe the overall device capability of the controller.

NOTE

This register is only available in OTG controller core.

Address: USB_UOG_DCCPARAMS is 53F8_0000h base + 124h offset = 53F8_0124h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-							HC	DC	-		DEN[4:0]				
W																
Reset	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0

USB_UOG_DCCPARAMS field descriptions

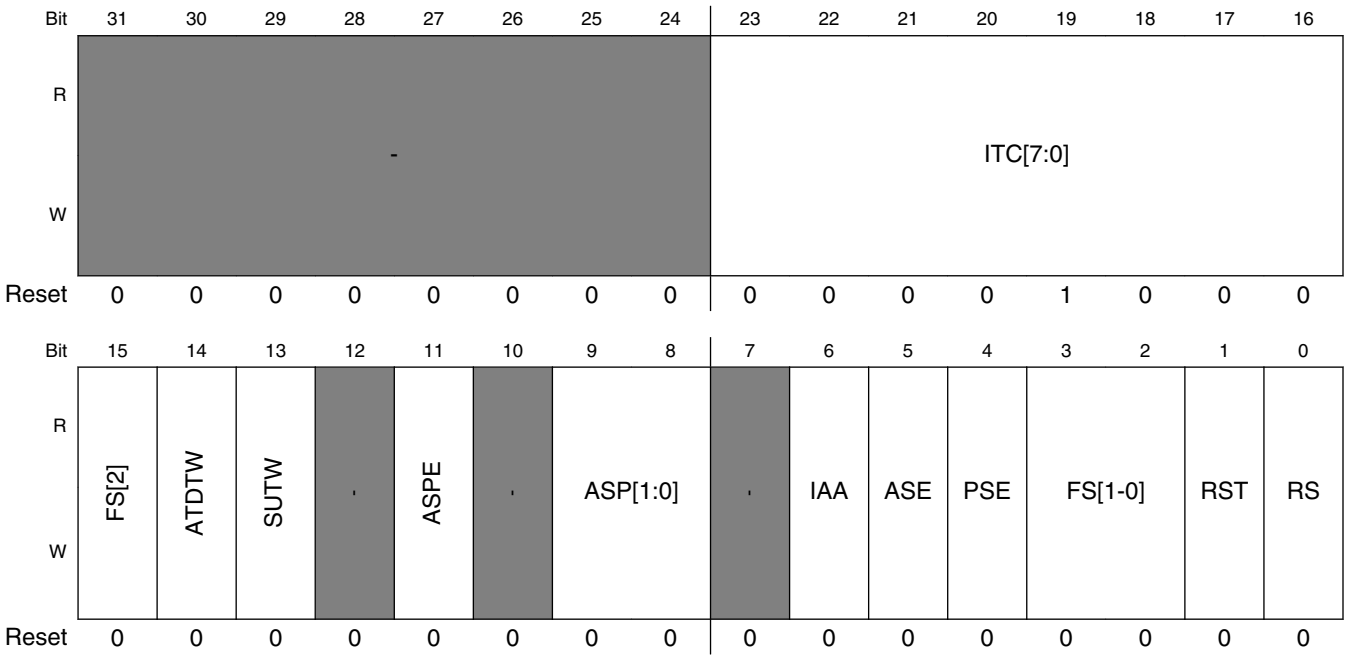
Field	Description
31–9 -	Reserved
8 HC	Host Capable When this bit is 1, this controller is capable of operating as an EHCI compatible USB 2.0 host controller.
7 DC	Device Capable When this bit is 1, this controller is capable of operating as a USB 2.0 device.
6–5 -	Reserved
4–0 DEN[4:0]	Device Endpoint Number This field indicates the number of endpoints built into the device controller. If this controller is not device capable, then this field will be zero. Valid values are 0 - 15.

77.6.18 USB Command Register (USB__USBCMD)

The Command Register indicates the command to be executed by the serial bus host/device controller. Writing to the register causes a command to be executed.

*: ASPE,ASP[1],ASP[0] reset value: '0b' for OTG core; '1b' for Host1/Host2/Host3 core.

Addresses: USB_UOG_USBCMD is 53F8_0000h base + 140h offset = 53F8_0140h
USB_UH1_USBCMD is 53F8_0000h base + 340h offset = 53F8_0340h
USB_UH2_USBCMD is 53F8_0000h base + 540h offset = 53F8_0540h
USB_UH3_USBCMD is 53F8_0000h base + 740h offset = 53F8_0740h



USB_*n*_USBCMD field descriptions

Field	Description
31–24 -	Reserved
23–16 ITC[7:0]	Interrupt Threshold Control -Read/Write. The system software uses this field to set the maximum rate at which the host/device controller will issue interrupts. ITC contains the maximum interrupt interval measured in micro-frames. Valid values are shown below. Value Maximum Interrupt Interval 0x00 Immediate (no threshold) 0x01 1 micro-frame 0x02 2 micro-frames 0x04 4 micro-frames 0x08 8 micro-frames 0x10 16 micro-frames 0x20 32 micro-frames 0x40 64 micro-frames
15 FS[2]	See also bits 3-2 Frame List Size - (Read/Write or Read Only). [host mode only] This field is Read/Write only if Programmable Frame List Flag in the HCCPARAMS registers is set to one. This field specifies the size of the frame list that controls which bits in the Frame Index Register should be used for the Frame List Current index.

Table continues on the next page...

USB_*n*_USBCMD field descriptions (continued)

Field	Description
	NOTE: This field is made up from USBCMD bits 15, 3 and 2. Value Meaning 000 1024 elements (4096 bytes) Default value 001 512 elements (2048 bytes) 010 256 elements (1024 bytes) 011 128 elements (512 bytes) 100 64 elements (256 bytes) 101 32 elements (128 bytes) 110 16 elements (64 bytes) 111 8 elements (32 bytes)
14 ATDTW	Add dTD TripWire - Read/Write. [device mode only] This bit is used as a semaphore to ensure proper addition of a new dTD to an active (primed) endpoint's linked list. This bit is set and cleared by software. This bit would also be cleared by hardware when state machine is hazard region for which adding a dTD to a primed endpoint may go unrecognized.
13 SUTW	Setup TripWire - Read/Write. [device mode only] This bit is used as a semaphore to ensure that the setup data payload of 8 bytes is extracted from a QH by the DCD without being corrupted. If the setup lockout mode is off (SLOM bit in USB core register n_USBMODE, see USB Device Mode (USB__USBMODE)) then there is a hazard when new setup data arrives while the DCD is copying the setup data payload from the QH for a previous setup packet. This bit is set and cleared by software. This bit would also be cleared by hardware when a hazard detected.
12 -	Reserved
11 ASPE	Asynchronous Schedule Park Mode Enable - Read/Write. If the *Asynchronous Park Capability* bit in the HCCPARAMS register is a one, then this bit defaults to a 1h and is R/W. Otherwise the bit must be a zero and is RO. Software uses this bit to enable or disable Park mode. When this bit is one, Park mode is enabled. When this bit is a zero, Park mode is disabled. This field is set to '1b' in this implementation.
10 -	Reserved
9–8 ASP[1:0]	Asynchronous Schedule Park Mode Count - Read/Write. If the *Asynchronous Park Capability* bit in the HCCPARAMS register is a one, then this field defaults to 3h and is R/W. Otherwise it defaults to zero and is Read-Only. It contains a count of the number of successive transactions the host controller is allowed to execute from a high-speed queue head on the Asynchronous schedule before continuing traversal of the Asynchronous schedule. Valid values are 1h to 3h. Software must not write a zero to this bit when *Park Mode Enable* is a one as this will result in undefined behavior. This field is set to 3h in all 4 controller core.
7 -	Reserved
6 IAA	Interrupt on Async Advance Doorbell - Read/Write.

Table continues on the next page...

USB_*n*_USBCMD field descriptions (continued)

Field	Description
	This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule. Software must write a 1 to this bit to ring the doorbell. When the host controller has evicted all appropriate cached schedule states, it sets the Interrupt on Async Advance status bit in the USBSTS register. If the Interrupt on Sync Advance Enable bit in the USBINTR register is one, then the host controller will assert an interrupt at the next interrupt threshold. The host controller sets this bit to zero after it has set the Interrupt on Sync Advance status bit in the USBSTS register to one. Software should not write a one to this bit when the asynchronous schedule is inactive. Doing so will yield undefined results. This bit is only used in host mode. Writing a one to this bit when device mode is selected will have undefined results.
5 ASE	Asynchronous Schedule Enable - Read/Write. Default 0b. This bit controls whether the host controller skips processing the Asynchronous Schedule. Only the host controller uses this bit. Values Meaning 0 Do not process the Asynchronous Schedule. 1 Use the ASYNCLISTADDR register to access the Asynchronous Schedule.
4 PSE	Periodic Schedule Enable- Read/Write. Default 0b. This bit controls whether the host controller skips processing the Periodic Schedule. Only the host controller uses this bit. Values Meaning 0 Do not process the Periodic Schedule 1 Use the PERIODICLISTBASE register to access the Periodic Schedule.
3–2 FS[1-0]	See description at bit 15
1 RST	Controller Reset (RESET) - Read/Write. Software uses this bit to reset the controller. This bit is set to zero by the Host/Device Controller when the reset process is complete. Software cannot terminate the reset process early by writing a zero to this register. Host operation mode: When software writes a one to this bit, the Controller resets its internal pipelines, timers, counters, state machines etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports. Software should not set this bit to a one when the HCHalted bit in the USBSTS register is a zero. Attempting to reset an actively running host controller will result in undefined behavior. Device operation mode: When software writes a one to this bit, the Controller resets its internal pipelines, timers, counters, state machines etc. to their initial value. Writing a one to this bit when the device is in the attached state is not recommended, because the effect on an attached host is undefined. In order to ensure that the device is not in an attached state before initiating a device controller reset, all primed endpoints should be flushed and the USBCMD Run/Stop bit should be set to 0.
0 RS	Run/Stop (RS) - Read/Write. Default 0b. 1=Run. 0=Stop. Host operation mode:

Table continues on the next page...

USB_*n*_USBCMD field descriptions (continued)

Field	Description
	When set to '1b', the Controller proceeds with the execution of the schedule. The Controller continues execution as long as this bit is set to a one. When this bit is set to 0, the Host Controller completes the current transaction on the USB and then halts. The HC Halted bit in the status register indicates when the Controller has finished the transaction and has entered the stopped state. Software should not write a one to this field unless the controller is in the Halted state (that is, HCHalted in the USBSTS register is a one). Device operation mode: Writing a one to this bit will cause the controller to enable a pull-up on D+ and initiate an attach event. This control bit is not directly connected to the pull-up enable, as the pull-up will become disabled upon transitioning into high-speed mode. Software should use this bit to prevent an attach event before the controller has been properly initialized. Writing a 0 to this will cause a detach event.

77.6.19 USB Status Register (USB__USBSTS)

This register indicates various states of the Host/Device Controller and any pending interrupts. This register does not indicate status resulting from a transaction on the serial bus.

*HCH bit reset value: '0b' for OTG core, '1b' for Host1/Host2/Host3 core.

Addresses: USB_UOG_USBSTS is 53F8_0000h base + 144h offset = 53F8_0144h
USB_UH1_USBSTS is 53F8_0000h base + 344h offset = 53F8_0344h
USB_UH2_USBSTS is 53F8_0000h base + 544h offset = 53F8_0544h
USB_UH3_USBSTS is 53F8_0000h base + 744h offset = 53F8_0744h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-						TI1	TI0	-							NAKI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AS	PS	RCL	HCH	-	ULPII	-	SLI	SRI	URI	AAI	SEI	FRI	PCI	UEI	UI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*_USBSTS field descriptions

Field	Description
31–26 -	Reserved
25 TI1	General Purpose Timer Interrupt 1(GPTINT1)--R/WC. This bit is set when the counter in the GPTIMER1CTRL register transitions to zero, writing a one to this bit will clear it.
24 TI0	General Purpose Timer Interrupt 0(GPTINT0)--R/WC. This bit is set when the counter in the GPTIMER0CTRL register transitions to zero, writing a one to this bit clears it.
23–17 -	Reserved
16 NAKI	NAK Interrupt Bit--RO. This bit is set by hardware when for a particular endpoint both the TX/RX Endpoint NAK bit and corresponding TX/RX Endpoint NAK Enable bit are set. This bit is automatically cleared by hardware when all Enabled TX/RX Endpoint NAK bits are cleared.
15 AS	Asynchronous Schedule Status - Read Only. This bit reports the current real status of the Asynchronous Schedule. When set to zero the asynchronous schedule status is disabled and if set to one the status is enabled. The Host Controller is not required to immediately disable or enable the Asynchronous Schedule when software transitions the Asynchronous Schedule Enable bit in the USBCMD register. When this bit and the Asynchronous Schedule Enable bit are the same value, the Asynchronous Schedule is either enabled (1) or disabled (0). Only used in the host operation mode.
14 PS	Periodic Schedule Status - Read Only. This bit reports the current real status of the Periodic Schedule. When set to zero the periodic schedule is disabled, and if set to one the status is enabled. The Host Controller is not required to immediately disable or enable the Periodic Schedule when software transitions the Periodic Schedule Enable bit in the USBCMD register. When this bit and the Periodic Schedule Enable bit are the same value, the Periodic Schedule is either enabled (1) or disabled (0). Only used in the host operation mode.
13 RCL	Reclamation - Read Only. This is a read-only status bit used to detect an empty asynchronous schedule. Only used in the host operation mode.
12 HCH	HCHalted - Read Only. This bit is a zero whenever the Run/Stop bit is a one. The Controller sets this bit to one after it has stopped executing because of the Run/Stop bit being set to 0, either by software or by the Controller hardware (for example, an internal error). Only used in the host operation mode. Default value is '0b' for OTG core, and '1b' for Host1/Host2/Host3 core. This is because OTG core is not operating as host in default. Please see CM bit in UOG_USBMODE register.
11 -	Reserved
10 ULPII	ULPI Interrupt - R/WC.

Table continues on the next page...

USB_*n*_USBSTS field descriptions (continued)

Field	Description
	This bit will be set '1b' by hardware when there is an event completion in ULPI viewport. This bit is usable only if the controller support UPLI interface mode.
9 -	Reserved
8 SLI	DCSuspend - R/WC. When a controller enters a suspend state from an active state, this bit will be set to a one. The device controller clears the bit upon exiting from a suspend state. Only used in device operation mode.
7 SRI	SOF Received - R/WC. When the device controller detects a Start Of (micro) Frame, this bit will be set to a one. When a SOF is extremely late, the device controller will automatically set this bit to indicate that an SOF was expected. Therefore, this bit will be set roughly every 1ms in device FS mode and every 125ms in HS mode and will be synchronized to the actual SOF that is received. Because the device controller is initialized to FS before connect, this bit will be set at an interval of 1ms during the prelude to connect and chirp. In host mode, this bit will be set every 125us and can be used by host controller driver as a time base. Software writes a 1 to this bit to clear it.
6 URI	USB Reset Received - R/WC. When the device controller detects a USB Reset and enters the default state, this bit will be set to a one. Software can write a 1 to this bit to clear the USB Reset Received status bit. Only used in device operation mode.
5 AAI	Interrupt on Async Advance - R/WC. System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing a one to the Interrupt on Async Advance Doorbell bit in the n_USBCMD register. This status bit indicates the assertion of that interrupt source. Only used in host operation mode.
4 SEI	System Error- R/WC. This bit is will be set to '1b' when an Error response is seen to a read on the system interface.
3 FRI	Frame List Rollover - R/WC. The Host Controller sets this bit to a one when the Frame List Index rolls over from its maximum value to zero. The exact value at which the rollover occurs depends on the frame list size. For example. If the frame list size (as programmed in the Frame List Size field of the UOG_USBCMD register) is 1024, the Frame Index Register rolls over every time FRINDEX [13] toggles. Similarly, if the size is 512, the Host Controller sets this bit to a one every time FHINDEX [12] toggles. Only used in host operation mode.
2 PCI	Port Change Detect - R/WC. The Host Controller sets this bit to a one when on any port a Connect Status occurs, a Port Enable/ Disable Change occurs, or the Force Port Resume bit is set as the result of a J-K transition on the suspended port. The Device Controller sets this bit to a one when the port controller enters the full or high-speed operational state. When the port controller exits the full or high-speed operation states due to Reset or Suspend events, the notification mechanisms are the USB Reset Received bit and the DCSuspend bits respectively.

Table continues on the next page...

USB_*n*_USBSTS field descriptions (continued)

Field	Description
1 UEI	USB Error Interrupt (USBERRINT) - R/WC. When completion of a USB transaction results in an error condition, this bit is set by the Host/Device Controller. This bit is set along with the USBINT bit, if the TD on which the error interrupt occurred also had its interrupt on complete (IOC) bit set The device controller detects resume signaling only.
0 UI	USB Interrupt (USBINT) - R/WC. This bit is set by the Host/Device Controller when the cause of an interrupt is a completion of a USB transaction where the Transfer Descriptor (TD) has an interrupt on complete (IOC) bit set. This bit is also set by the Host/Device Controller when a short packet is detected. A short packet is when the actual number of bytes received was less than the expected number of bytes.

77.6.20 Interrupt Enable Register (USB__USBINTR)

The interrupts to software are enabled with this register. An interrupt is generated when a bit is set and the corresponding interrupt source is active. The USB Status register (n_USBSTS) still shows interrupt sources even if they are disabled by the n_USBINTR register, allowing polling of interrupt events by the software.

Addresses: USB_UOG_USBINTR is 53F8_0000h base + 148h offset = 53F8_0148h
USB_UH1_USBINTR is 53F8_0000h base + 348h offset = 53F8_0348h
USB_UH2_USBINTR is 53F8_0000h base + 548h offset = 53F8_0548h
USB_UH3_USBINTR is 53F8_0000h base + 748h offset = 53F8_0748h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	-						TIE1	TIE0	-				UPIE	UAIE	-	NAKE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-					ULPIE	-	SLE	SRE	URE	AAE	SEE	FRE	PCE	UEE	UE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*_USBINTR field descriptions

Field	Description
31–26 -	Reserved
25 TIE1	General Purpose Timer #1 Interrupt Enable When this bit is one and the TI1 bit in n_USBSTS register is a one the controller will issue an interrupt.
24 TIE0	General Purpose Timer #0 Interrupt Enable When this bit is one and the TI0 bit in n_USBSTS register is a one the controller will issue an interrupt.
23–20 -	Reserved
19 UPIE	USB Host Periodic Interrupt Enable When this bit is one, and the UPI bit in the n_USBSTS register is one, host controller will issue an interrupt at the next interrupt threshold.
18 UAIE	USB Host Asynchronous Interrupt Enable When this bit is one, and the UAI bit in the n_USBSTS register is one, host controller will issue an interrupt at the next interrupt threshold.
17 -	Reserved
16 NAKE	NAK Interrupt Enable When this bit is one and the NAKI bit in n_USBSTS register is a one the controller will issue an interrupt.
15–11 -	These bits are reserved and should be set to zero.
10 ULPIE	ULPI Interrupt Enable When this bit is one and the UPLII bit in n_USBSTS register is a one the controller will issue an interrupt. This bit is usable only if the controller support UPLI interface mode.
9 -	Reserved
8 SLE	Sleep Interrupt Enable When this bit is one and the SLI bit in n_n_USBSTS register is a one the controller will issue an interrupt. Only used in device operation mode.
7 SRE	SOF Received Interrupt Enable When this bit is one and the SRI bit in n_USBSTS register is a one the controller will issue an interrupt.
6 URE	USB Reset Interrupt Enable When this bit is one and the URI bit in n_UOG_USBSTS register is a one the controller will issue an interrupt. Only used in device operation mode.
5 AAE	Async Advance Interrupt Enable When this bit is one and the AAI bit in n_USBSTS register is a one the controller will issue an interrupt. Only used in host operation mode.
4 SEE	System Error Interrupt Enable When this bit is one and the SEI bit in n_USBSTS register is a one the controller will issue an interrupt.

Table continues on the next page...

USB_*n*_USBINTR field descriptions (continued)

Field	Description
	Only used in host operation mode.
3 FRE	Frame List Rollover Interrupt Enable When this bit is one and the FRI bit in n_USBSTS register is a one the controller will issue an interrupt. Only used in host operation mode.
2 PCE	Port Change Detect Interrupt Enable When this bit is one and the PCI bit in n_USBSTS register is a one the controller will issue an interrupt.
1 UEE	USB Error Interrupt Enable When this bit is one and the UEI bit in n_USBSTS register is a one the controller will issue an interrupt.
0 UE	USB Interrupt Enalbe When this bit is one and the UI bit in n_USBSTS register is a one the controller will issue an interrupt.

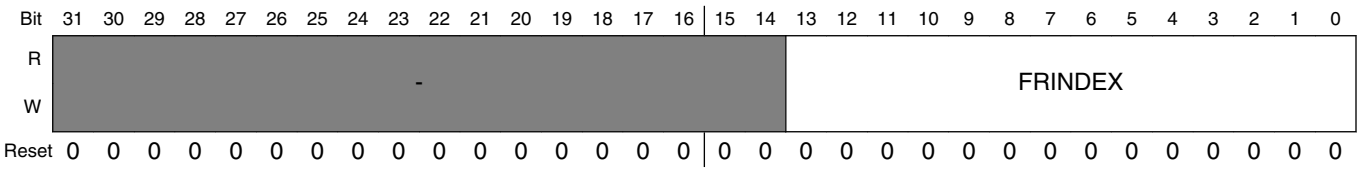
77.6.21 USB Frame Index (USB__FRINDEX)

This register is used by the host controller to index the periodic frame list. The register updates every 125 microseconds (once each micro-frame). Bits [N: 3] are used to select a particular entry in the Periodic Frame List during periodic schedule execution. The number of bits used for the index depends on the size of the frame list as set by system software in the Frame List Size field in the n_USBCMD register.

This register must be written as a DWord. Byte writes produce-undefined results. This register cannot be written unless the Host Controller is in the 'Halted' state as indicated by the HCHalted bit. A write to this register while the Run/Stop hit is set to a one produces undefined results. Writes to this register also affect the SOF value.

In device mode this register is read only and, the device controller updates the FRINDEX [13:3] register from the frame number indicated by the SOF marker. Whenever a SOF is received by the USB bus, FRINDEX [13:3] will be checked against the SOF marker. If FRINDEX [13:3] is different from the SOF marker, FRINDEX [13:3] will be set to the SOF value and FRINDEX [2:0] will be set to zero (that is, SOF for 1 ms frame). If FRINDEX [13:3] is equal to the SOF value, FRINDEX [2:0] will be increment (that is, SOF for 125 us micro-frame.).

Addresses: USB_UOG_FRINDEX is 53F8_0000h base + 14Ch offset = 53F8_014Ch
USB_UH1_FRINDEX is 53F8_0000h base + 34Ch offset = 53F8_034Ch
USB_UH2_FRINDEX is 53F8_0000h base + 54Ch offset = 53F8_054Ch
USB_UH3_FRINDEX is 53F8_0000h base + 74Ch offset = 53F8_074Ch



USB_*n*_FRINDEX field descriptions

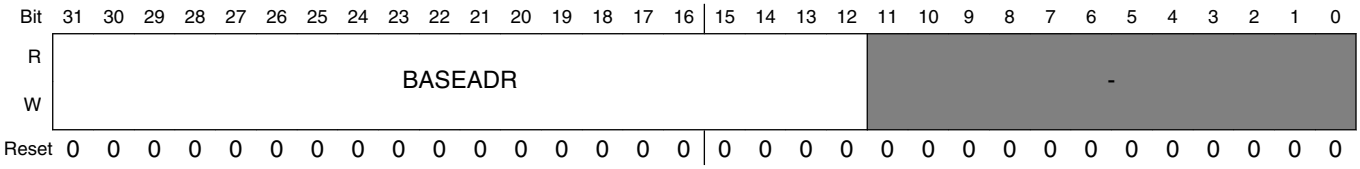
Field	Description
31–14 -	Reserved
13–0 FRINDEX	Frame Index. The value, in this register, increments at the end of each time frame (micro-frame). Bits [N: 3] are used for the Frame List current index. This means that each location of the frame list is accessed 8 times (frames or micro-frames) before moving to the next index. The following illustrates values of N based on the value of the Frame List Size field in the USBCMD register, when used in host mode. USBCMD [Frame List Size] Number Elements N In device mode the value is the current frame number of the last frame transmitted. It is not used as an index. In either mode bits 2:0 indicate the current microframe. 000 (1024) 12 001 (512) 11 010 (256) 10 011 (128) 9 100 (64) 8 101 (32) 7 110 (16) 6 111 (8) 5

77.6.22 Frame List Base Address (USB__PERIODICLISTBASE)

Host Controller only

This 32-bit register contains the beginning address of the Periodic Frame List in the system memory. HCD loads this register prior to starting the schedule execution by the Host Controller. The memory structure referenced by this physical memory pointer is assumed to be 4-Kbyte aligned. The contents of this register are combined with the Frame Index Register (UOG_FRINDEX) to enable the Host Controller to step through the Periodic Frame List in sequence.

Addresses: USB_UOG_PERIODICLISTBASE is 53F8_0000h base + 154h offset = 53F8_0154h
USB_UH1_PERIODICLISTBASE is 53F8_0000h base + 354h offset = 53F8_0354h
USB_UH2_PERIODICLISTBASE is 53F8_0000h base + 554h offset = 53F8_0554h
USB_UH3_PERIODICLISTBASE is 53F8_0000h base + 754h offset = 53F8_0754h



USB_*n*_PERIODICLISTBASE field descriptions

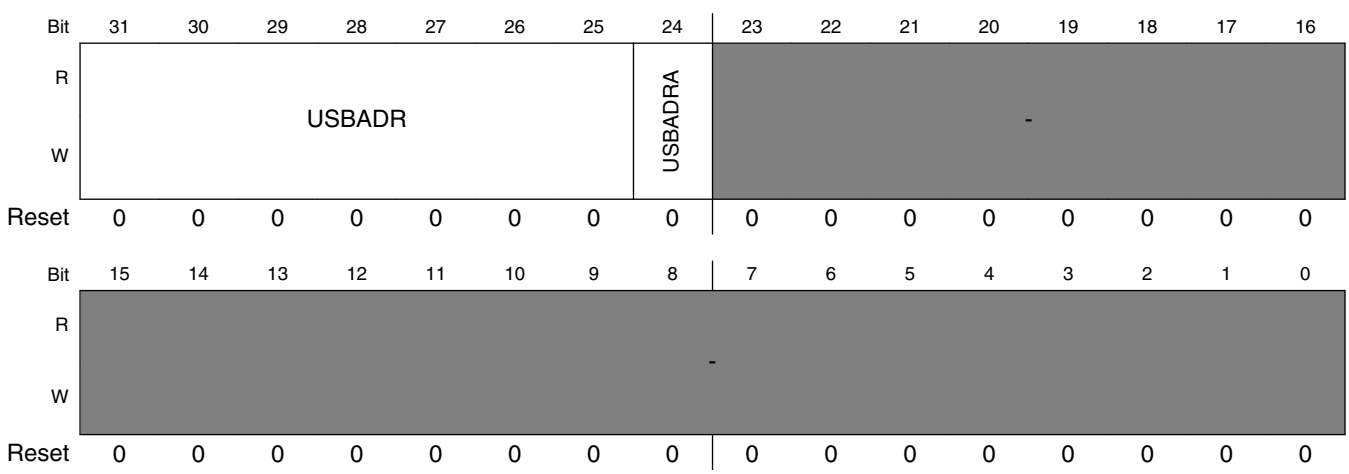
Field	Description
31–12 BASEADR	Base Address (Low). These bits correspond to memory address signals [31:12], respectively. Only used by the host controller.
11–0 -	Reserved

77.6.23 Device Address (USB_UOG_DEVICEADDR)

Device Controller only

The upper seven bits of this register represent the device address. After any controller reset or a USB reset, the device address is set to the default address (0). The default address will match all incoming addresses. Software shall reprogram the address after receiving a SET_ADDRESS descriptor.

Address: USB_UOG_DEVICEADDR is 53F8_0000h base + 154h offset = 53F8_0154h



USB_UOG_DEVICEADDR field descriptions

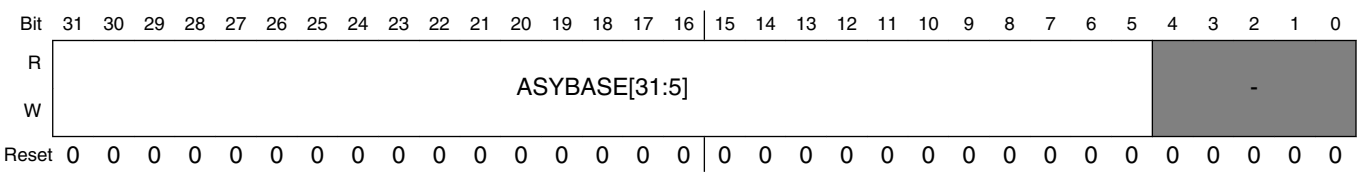
Field	Description
31–25 USBADR	Device Address. These bits correspond to the USB device address
24 USBADRA	Device Address Advance. Default=0. When this bit is '0', any writes to USBADR are instantaneous. When this bit is written to a '1' at the same time or before USBADR is written, the write to the USBADR field is staged and held in a hidden register. After an IN occurs on endpoint 0 and is ACKed, USBADR will be loaded from the holding register. Hardware will automatically clear this bit on the following conditions: 1) IN is ACKed to endpoint 0. (USBADR is updated from staging register). 2) OUT/SETUP occur to endpoint 0. (USBADR is not updated). 3) Device Reset occurs (USBADR is reset to 0). **NOTE:** After the status phase of the SET_ADDRESS descriptor, the DCD has 2 ms to program the USBADR field. This mechanism will ensure this specification is met when the DCD can not write of the device address within 2ms from the SET_ADDRESS status phase. If the DCD writes the USBADR with USBADRA=1 after the SET_ADDRESS data phase (before the prime of the status phase), the USBADR will be programmed instantly at the correct time and meet the 2ms USB requirement.
23–0 -	Reserved

77.6.24 Next Asynch. Address (USB__ASYNCLISTADDR)

Host Controller only

This 32-bit register contains the address of the next asynchronous queue head to be executed by the host. Bits [4:0] of this register cannot be modified by the system software and will always return a zero when read.

Addresses: USB_UOG_ASYNCLISTADDR is 53F8_0000h base + 158h offset = 53F8_0158h
USB_UH1_ASYNCLISTADDR is 53F8_0000h base + 358h offset = 53F8_0358h
USB_UH2_ASYNCLISTADDR is 53F8_0000h base + 558h offset = 53F8_0558h
USB_UH3_ASYNCLISTADDR is 53F8_0000h base + 758h offset = 53F8_0758h



USB_*n*_ASYNCLISTADDR field descriptions

Field	Description
31–5 ASYBASE[31:5]	Link Pointer Low (LPL).

Table continues on the next page...

USB_*n*_ASYNCLISTADDR field descriptions (continued)

Field	Description
	These bits correspond to memory address signals [31:5], respectively. This field may only reference a Queue Head (QH). Only used by the host controller.
4–0 -	Reserved

77.6.25 Endpoint List Address (USB_UOG_ENDPTLISTADDR)

Device Controller only

In device mode, this register contains the address of the top of the endpoint list in system memory. Bits [10:0] of this register cannot be modified by the system software and will always return a zero when read.

The memory structure referenced by this physical memory pointer is assumed 64-byte.

Address: USB_UOG_ENDPTLISTADDR is 53F8_0000h base + 158h offset = 53F8_0158h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EPBASE[31:11]																-															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_UOG_ENDPTLISTADDR field descriptions

Field	Description
31–11 EPBASE[31:11]	Endpoint List Pointer(Low). These bits correspond to memory address signals [31:11], respectively. This field will reference a list of up to 32 Queue Head (OH) (that is, one queue head per endpoint & direction).
10–0 -	Reserved

77.6.26 Programmable Burst Size (USB__BURSTSIZE)

This register is used to control the burst size used during data movement on the AHB master interface. This register is ignored if AHBBRST bits in SBUSCFG register is non-zero value.

Addresses: USB_UOG_BURSTSIZE is 53F8_0000h base + 160h offset = 53F8_0160h

USB_UH1_BURSTSIZE is 53F8_0000h base + 360h offset = 53F8_0360h

USB_UH2_BURSTSIZE is 53F8_0000h base + 560h offset = 53F8_0560h

USB_UH3_BURSTSIZE is 53F8_0000h base + 760h offset = 53F8_0760h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																TXPBURST								RXPBURST								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*_BURSTSIZE field descriptions

Field	Description
31–17 -	Reserved
16–8 TXPBURST	Programmable TX Burst Size. Default value is determined by TXBURST bits in n_HWTXBUF. This register represents the maximum length of a the burst in 32-bit words while moving data from system memory to the USB bus.
7–0 RXPBURST	Programmable RX Burst Size. Default value is determined by TXBURST bits in n_HWRXBUF. This register represents the maximum length of a the burst in 32-bit words while moving data from the USB bus to system memory.

77.6.27 TX FIFO Fill Tuning (USB__TXFILLTUNING)

The fields in this register control performance tuning associated with how the host controller posts data to the TX latency FIFO before moving the data onto the USB bus. The specific areas of performance include the how much data to post into the FIFO and an estimate for how long that operation should take in the target system.

Definitions:

T_0 = Standard packet overhead

T_1 = Time to send data payload

T_{ff} = Time to fetch packet into TX FIFO up to specified level.

T_s = Total Packet Flight Time (send-only) packet

$T_s = T_0 + T_1$

T_p = Total Packet Time (fetch and send) packet

$T_p = T_{ff} + T_0 + T_1$

Upon discovery of a transmit (OUT/SETUP) packet in the data structures, host controller checks to ensure T_p remains before the end of the [micro]frame. If so it proceeds to pre-fill the TX FIFO. If at anytime during the pre-fill operation the time remaining the [micro]frame is < T_s then the packet attempt ceases and the packet is tried at a later time. Although this is not an error condition and the host controller will eventually recover, a mark will be made the scheduler health counter to note the occurrence of a "back-off" event. When a back-off event is detected, the partial packet fetched may need to be discarded from the latency buffer to make room for periodic traffic that will begin after the next SOF. Too many back-off events can waste bandwidth and power on the system bus and thus should be minimized (not necessarily eliminated). Back-offs can be minimized with use of the n_TSCHHEALTH (T_{ff}) described below.

Addresses: USB_UOG_TXFILLTUNING is 53F8_0000h base + 164h offset = 53F8_0164h
USB_UH1_TXFILLTUNING is 53F8_0000h base + 364h offset = 53F8_0364h
USB_UH2_TXFILLTUNING is 53F8_0000h base + 564h offset = 53F8_0564h
USB_UH3_TXFILLTUNING is 53F8_0000h base + 764h offset = 53F8_0764h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-										TXFIFOTHRES						-			TXSCHHEALTH					TXSCHOH							
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	1*	0*	0*	0*	0*	0*	0*	0*	1*	0*	0*	0*

* Notes:
- Reset value is implementation dependent.

USB_*n*_TXFILLTUNING field descriptions

Field	Description
31–22 -	Reserved
21–16 TXFIFOTHRES	FIFO Burst Threshold. (Read/Write) This register controls the number of data bursts that are posted to the TX latency FIFO in host mode before the packet begins on to the bus. The minimum value is 2 and this value should be a low as possible to maximize USB performance. A higher value can be used in systems with unpredictable latency and/or insufficient bandwidth where the FIFO may underrun because the data transferred from the latency FIFO to USB occurs before it can be replenished from system memory. This value is ignored if the Stream Disable bit in UOG_USBMODE register is set. Default value is '00h' for OTG controller core, and '02h' for Host1/Host2/Host3 controller core.
15–13 -	Reserved
12–8 TXSCHHEALTH	Scheduler Health Counter. (Read/Write To Clear) This register increments when the host controller fails to fill the TX latency FIFO to the level programmed by TXFIFOTHRES before running out of time to send the packet before the next Start-Of-Frame. This health counter measures the number of times this occurs to provide feedback to selecting a proper TXSCHOH. Writing to this register will clear the counter and this counter will max. at 31. Default value is '08h' for OTG controller core, and '00h' for Host1/Host2/Host3 controller core.

Table continues on the next page...

USB_*n*_TXFILLTUNING field descriptions (continued)

Field	Description
7–0 TXSCHOH	Scheduler Overhead. (Read/Write) [Default = 0] This register adds an additional fixed offset to the schedule time estimator described above as Tff. As an approximation, the value chosen for this register should limit the number of back-off events captured in the TXSCHHEALTH to less than 10 per second in a highly utilized bus. Choosing a value that is too high for this register is not desired as it can needlessly reduce USB utilization. The time unit represented in this register is 1.267us when a device is connected in High-Speed Mode. The time unit represented in this register is 6.333us when a device is connected in Low/Full Speed Mode. Default value is '08h' for OTG controller core, and '00h' for Host1/Host2/Host3 controller core.

77.6.28 IC_USB enable and voltage negotiation (USB__IC_USB)

This register enable and controls the IC-USB FS/LS transceiver.

Addresses: USB_UOG_IC_USB is 53F8_0000h base + 16Ch offset = 53F8_016Ch
USB_UH1_IC_USB is 53F8_0000h base + 36Ch offset = 53F8_036Ch
USB_UH2_IC_USB is 53F8_0000h base + 56Ch offset = 53F8_056Ch
USB_UH3_IC_USB is 53F8_0000h base + 76Ch offset = 53F8_076Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-																												IC1	IC_VDD1		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*_IC_USB field descriptions

Field	Description
31–4 -	Reserved
3 IC1	Inter-Chip transceiver enable. This bits enable the Inter-Chip transceiver. To enable IC-USB interface, PTS bits in PORTSC1 registers also need to be set correspondingly. Please see Port Status & Control (USB__PORTSC1) . **NOTE:** If IC1 is enabled, IOMUX must set the DP/DM pad into loopback mode.
2–0 IC_VDD1	These bits indicates which voltage is being supplied to the peripheral This field is read-only and set to 000 in case of a device controller. 000 No voltage 001 1.0 V 010 1.2 V 011 1.5 V 100 1.8 V 101 3.0 V 110 Reserved 111 Reserved

77.6.29 Endpoint NAK (USB_UOG_ENDPTNAK)

Address: USB_UOG_ENDPTNAK is 53F8_0000h base + 178h offset = 53F8_0178h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-								EPTN[7:0]								-								EPRN[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_UOG_ENDPTNAK field descriptions

Field	Description
31–24 -	Reserved
23–16 EPTN[7:0]	TX Endpoint NAK - R/WC. Each TX endpoint has 1 bit in this field. The bit is set when the device sends a NAK handshake on a received IN token for the corresponding endpoint. Bit [N] - Endpoint #[N], N is 0-7
15–8 -	Reserved
7–0 EPRN[7:0]	RX Endpoint NAK - R/WC. Each RX endpoint has 1 bit in this field. The bit is set when the device sends a NAK handshake on a received OUT or PING token for the corresponding endpoint. Bit [N] - Endpoint #[N], N is 0-7

77.6.30 Endpoint Nake Enable (USB_UOG_ENDPTNAKEN)

Address: USB_UOG_ENDPTNAKEN is 53F8_0000h base + 17Ch offset = 53F8_017Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-								EPTNE[7:0]								-								EPRNE[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_UOG_ENDPTNAKEN field descriptions

Field	Description
31–24 -	Reserved
23–16 EPTNE[7:0]	TX Endpoint NAK Enable - R/W. Each bit is an enable bit for the corresponding TX Endpoint NAK bit. If this bit is set and the corresponding TX Endpoint NAK bit is set, the NAK Interrupt bit is set.

Table continues on the next page...

USB_UOG_ENDPTNAKEN field descriptions (continued)

Field	Description
	Bit [N] - Endpoint #[N], N is 0-7
15–8 -	Reserved
7–0 EPRNE[7:0]	RX Endpoint NAK Enable - R/W. Each bit is an enable bit for the corresponding RX Endpoint NAK bit. If this bit is set and the corresponding RX Endpoint NAK bit is set, the NAK Interrupt bit is set. Bit [N] - Endpoint #[N], N is 0-7

77.6.31 Port Status & Control (USB__PORTSC1)

Host Controller

A host controller could implement one to eight port status and control registers. The number is determined by N_PORTs bits in HWSPARAMs register (please see Host Controller Structural Parameters (USB__HCSPARAMS)). Software could read this parameter register to determine how many ports need service.

All 4 controller cores are Single-Port Host, so there is only one port status and control register for each controller core.

This register is only reset by power on reset or controller core reset. The initial conditions of a port are:

- No device connected
- Port disabled

If the port supports power control, this state remains until port power is supplied (by software).

Device Controller

A device controller has only port register one (PORTSC1) and it does not support power control. Port control in device mode is only used for status port reset, suspend, and current connect status. It is also used to initiate test mode or force signaling and allows software to put the PHY into low power suspend mode and disable the PHY clock.

Addresses: USB_UOG_PORTSC1 is 53F8_0000h base + 184h offset = 53F8_0184h
USB_UH1_PORTSC1 is 53F8_0000h base + 384h offset = 53F8_0384h
USB_UH2_PORTSC1 is 53F8_0000h base + 584h offset = 53F8_0584h
USB_UH3_PORTSC1 is 53F8_0000h base + 784h offset = 53F8_0784h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PTS[1:0]		STS	PTW	PSPD		PTS	PFSC	PHCD	WKOC	WKDC	WKCN	PTC[3:0]			
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PIC[1:0]		PO	PP	LS[1:0]		HSP	PR	SUSP	FPR	OCC	OCA	PEC	PE	CSC	CCS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_*n*_PORTSC1 field descriptions

Field	Description
31–30 PTS[1:0]	Parallel Transceiver Select (bit25, bit31, bi30). These register bits are implementation dependent. For OTG core, it's Read-Only. Reset value is 000b. For Host1/Host2/Host3 core, it's Read/Write. Reset value is 011b. Bit field {bit25, bit31, bit30}: "000b" UTMI/UTMI+ "001b" Reserved "010b" ULPI "011b" Serial/USB 1.1 PHY/IC-USB (FS Only) "100b" HSIC **NOTE:** All USB port interface modes are listed in this field description, but not all are supported. For detail feature of each controller core, please see Features . The behaviour is unknown when unsupported interface mode is selected.
29 STS	Serial Transceiver Select - Read/Write. Reset value is 0b. 1 Serial Interface Engine is selected 0 Parallel Interface signals is selected Serial Interface Engine can be used in combination with UTMI+/ULPI physical interface to provide FS/LS signaling instead of the parallel interface signals. When this bit is set '1b', serial interface engine will be used instead of parallel interface signals. This bit has no effect unless PTS bits is set to select UTMI+/ULPI interface. The Serial/USB1.1 PHY/IC-USB will use the serial interface engine for FS/LS signaling regardless of this bit value.
28 PTW	Parallel Transceiver Width

Table continues on the next page...

USB_*n*_PORTSC1 field descriptions (continued)

Field	Description
	This bit has no effect if serial interface engine is used. These register bits are implementation dependent. For OTG/Host1/Host2/Host3 core, it is Read/Write. Reset value is '1b'. 1 Select the 8-bit UTMI interface [60MHz] 2 Select the 16-bit UTMI interface [30MHz]
27–26 PSPD	Port Speed - Read Only. This register field indicates the speed at which the port is operating. 00 Full Speed 01 Low Speed 10 High Speed 11 Undefined
25 PTS	See description at bits 31-30
24 PFSC	Port Force Full Speed Connect - Read/Write. Default = 0b. When this bit is set to '1b', the port will be forced to only connect at Full Speed, It disables the chirp sequence that allows the port to identify itself as High Speed. 1 Forced to full speed 0 Normal operation
23 PHCD	PHY Low Power Suspend - Clock Disable (PLPSCD) - Read/Write. Default = 0b. When this bit is set to '1b', the PHY clock is disabled. Reading this bit will indicate the status of the PHY clock. **NOTE:** The PHY clock cannot be disabled if it is being used as the system clock. In device mode, The PHY can be put into Low Power Suspend when the device is not running (USBCMD Run/Stop=0b) or the host has signaled suspend (PORTSC1 SUSPEND=1b). PHY Low power suspend will be cleared automatically when the host initials resume. Before forcing a resume from the device, the device controller driver must clear this bit. In host mode, the PHY can be put into Low Power Suspend when the downstream device has been put into suspend mode or when no downstream device is connected. Low power suspend is completely under the control of software. 1 Disable PHY clock 0 Enable PHY clock
22 WKOC	Wake on Over-current Enable (WKOC_E) - Read/Write. Default = 0b. Writing this bit to a one enables the port to be sensitive to over-current conditions as wake-up events. This field is zero if *Port Power(*Port Status & Control (USB__PORTSC1)*)* is zero.
21 WKDC	Wake on Disconnect Enable (WKDSCNNT_E) - Read/Write. Default=0b. Writing this bit to a one enables the port to be sensitive to device disconnects as wake-up events. This field is zero if *Port Power(*Port Status & Control (USB__PORTSC1)*)* is zero or in device mode.
20 WKCN	Wake on Connect Enable (WKCNNT_E) - Read/Write. Default=0b. Writing this bit to a one enables the port to be sensitive to device connects as wake-up events. This field is zero if *Port Power(*Port Status & Control (USB__PORTSC1)*)* is zero or in device mode.

Table continues on the next page...

USB_*n*_PORTSC1 field descriptions (continued)

Field	Description
19–16 PTC[3:0]	Port Test Control - Read/Write. Default = 0000b. Refer to Section 4.14 for the operational model for using these test modes and the USB Specification Revision 2.0, Chapter 7 for details on each test mode. The FORCE_ENABLE_FS and FORCE ENABLE_LS are extensions to the test mode support specified in the EHCI specification. Writing the PTC field to any of the FORCE_ENABLE_{HS/FS/LS} values will force the port into the connected and enabled state at the selected speed. Writing the PTC field back to TEST_MODE_DISABLE will allow the port state machines to progress normally from that point. **NOTE:** *Low speed operations are not supported as a peripheral device.* Any other value than zero indicates that the port is operating in test mode. Value Specific Test 0000 TEST_MODE_DISABLE 0001 J_STATE 0010 K_STATE 0011 SE0 (host) / NAK (device) 0100 Packet 0101 FORCE_ENABLE_HS 0110 FORCE_ENABLE_FS 0111 FORCE_ENABLE_LS 1000-1111 Reserved
15–14 PIC[1:0]	Port Indicator Control - Read/Write. Default = Ob. Writing to this field has no effect if the P_INDICATOR bit in the HCSPARAMS register is a zero. Refer to the USB Specification Revision 2.0 for a description on how these bits are to be used. This field is zero if *Port Power* is zero. Bit Value Meaning 00 Port indicators are off 01 Amber 10 Green 11 Undefined
13 PO	Port Owner-Read/Write. Default = 0. This bit unconditionally goes to a 0 when the configured bit in the CONFIGFLAG register makes a 0 to 1 transition. This bit unconditionally goes to 1 whenever the Configured bit is zero System software uses this field to release ownership of the port to a selected host controller (in the event that the attached device is not a high-speed device). Software writes a one to this bit when the attached device is not a high-speed device. A one in this bit means that an internal companion controller owns and controls the port. Port owner handoff is not supported in all 4 controller cores, therefore this bit will always be 0.
12 PP	Port Power (PP)-Read/Write or Read Only. The function of this bit depends on the value of the Port Power Switching (PPC) field in the HCSPARAMS register. The behavior is as follows: PPC PP Operation 0 1b *Read Only - Host controller does not have port power control switches. Each port is hard-wired to power.*

Table continues on the next page...

USB_*n*_PORTSC1 field descriptions (continued)

Field	Description
	1 1b/0b - *Read/Write. Host/OTG controller requires port power control switches. This bit represents the current setting of the switch (0=off, 1=on). When power is not available on a port (that is, PP equals a 0), the port is non-functional and will not report attaches, detaches, etc.* When an over-current condition is detected on a powered port and PPC is a one, the PP bit in each affected port may be transitional by the host controller driver from a one to a zero (removing power from the port). This feature is implemented in all 4 controller cores (PPC = 1).
11–10 LS[1:0]	Line Status-Read Only. These bits reflect the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. In host mode, the use of linestate by the host controller driver is not necessary (unlike EHCI), because the port controller state machine and the port routing manage the connection of LS and FS. In device mode, the use of linestate by the device controller driver is not necessary. The encoding of the bits are: Bits [11:10] Meaning 00 SE0 10 J-state 01 K-state 11 Undefined
9 HSP	High-Speed Port - Read Only. Default = 0b. When the bit is one, the host/device connected to the port is in high-speed mode and if set to zero, the host/device connected to the port is not in a high-speed mode. **NOTE:** HSP is redundant with PSPD(bit 27, 26) but remained for compatibility.
8 PR	Port Reset - Read/Write or Read Only. Default = 0b. In Host Mode: Read/Write. 1=Port is in Reset. 0=Port is not in Reset. Default 0. When software writes a one to this bit the bus-reset sequence as defined in the USB Specification Revision 2.0 is started. *This bit will automatically change to zero after the reset sequence is complete. This behavior is different from EHCI where the host controller driver is required to set this bit to a zero after the reset duration is timed in the driver.* In Device Mode: This bit is a read only status bit. Device reset from the USB bus is also indicated in the USBSTS register. This field is zero if *Port Power(*Port Status & Control (USB__PORTSC1)*)* is zero.
7 SUSP	Suspend - Read/Write or Read Only. Default = 0b. 1=Port in suspend state. 0=Port not in suspend state. In Host Mode: Read/Write. Port Enabled Bit and Suspend bit of this register define the port states as follows: Bits [Port Enabled, Suspend] Port State 0x Disable 10 Enable 11 Suspend

Table continues on the next page...

USB_*n*_PORTSC1 field descriptions (continued)

Field	Description
	When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the The host controller will unconditionally set this bit to zero when software sets the *Force Port Resume* bit to zero. The host controller ignores a write of zero to this bit. If host software sets this bit to a one when the port is not enabled (that is, *Port enabled* bit is a zero) the results are undefined. This field is zero if *Port Power(*Port Status & Control (USB__PORTSC1)*)* is zero in host mode. In Device Mode: Read Only. In device mode this bit is a read only status bit.
6 FPR	Force Port Resume -Read/Write. 1= Resume detected/driven on port. 0=No resume (K-state) detected/ driven on port. Default = 0. In Host Mode: Software sets this bit to one to drive resume signaling. The Host Controller sets this bit to one if a J-to-K transition is detected while the port is in the Suspend state. When this bit transitions to a one because a J-to-K transition is detected, the *Port Change Detect* bit in the USBSTS register is also set to one. *This bit will automatically change to zero after the resume sequence is complete. This behavior is different from EHCI where the host controller driver is required to set this bit to a zero after the resume duration is timed in the driver.* Note that when the Host controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this bit remains a one. This bit will remain a one until the port has switched to the high-speed idle. Writing a zero has no affect because the port controller will time the resume operation clear the bit the port control state switches to HS or FS idle. This field is zero if *Port Power(*Port Status & Control (USB__PORTSC1)*)* is zero in host mode. This bit is not-EHCI compatible. In Device mode: After the device has been in Suspend State for 5ms or more, software must set this bit to one to drive resume signaling before clearing. The Device Controller will set this bit to one if a J-to-K transition is detected while the port is in the Suspend state. The bit will be cleared when the device returns to normal operation. Also, when this bit transitions to a one because a J-to-K transition detected, the *Port Change Detect* bit in the USBSTS register is also set to one.
5 OCC	Over-current Change-R/WC. Default=0. This bit is set '1b' by hardware when there is a change to Over-current Active. Software can clear this bit by writing a one to this bit position.
4 OCA	Over-current Active-Read Only. Default 0. This bit will automatically transition from one to zero when the over current condition is removed. 1 This port currently has an over-current condition 0 This port does not have an over-current condition.
3 PEC	Port Enable/Disable Change-R/WC. 1=Port enabled/disabled status has changed. 0=No change. Default = 0. In Host Mode:

Table continues on the next page...

USB_*n*_PORTSC1 field descriptions (continued)

Field	Description
	For the root hub, this bit is set to a one only when a port is disabled due to disconnect on the port or due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the USB Specification). Software clears this by writing a one to it. This field is zero if *Port Power(*Port Status & Control (USB__PORTSC1)*)* is zero. In Device mode: The device port is always enabled, so this bit is always '0b'.
2 PE	Port Enabled/Disabled-Read/Write. 1=Enable. 0=Disable. Default 0. In Host Mode: Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a one to this field. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by the host software. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events. When the port is disabled, (0b) downstream propagation of data is blocked except for reset. This field is zero if *Port Power(*Port Status & Control (USB__PORTSC1)*)* is zero in host mode. In Device Mode: The device port is always enabled, so this bit is always '1b'.
1 CSC	Connect Status Change-R/WC. 1 =Change in Current Connect Status. 0=No change. Default 0. In Host Mode: Indicates a change has occurred in the port's Current Connect Status. The host/device controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be 'setting' an already-set bit (that is, the bit will remain set). Software clears this bit by writing a one to it. This field is zero if *Port Power(*Port Status & Control (USB__PORTSC1)*)* is zero in host mode. In Device Mode: This bit is undefined in device controller mode.
0 CCS	Current Connect Status-Read Only. In Host Mode: 1=Device is present on port. 0=No device is present. Default = 0. This value reflects the current state of the port, and may not correspond directly to the event that caused the *Connect Status Change* bit (Bit 1) to be set. This field is zero if *Port Power(*Port Status & Control (USB__PORTSC1)*)* is zero in host mode. In Device Mode: 1=Attached. 0=Not Attached. Default=0. A one indicates that the device successfully attached and is operating in either high speed or full speed as indicated by the High Speed Port bit in this register. A zero indicates that the device did not attach successfully or was forcibly disconnected by the software writing a zero to the Run bit in the USBCMD register. It does not state the device being disconnected or suspended.

77.6.32 On-The-Go Status & control (USB_UOG_OTGSC)

This register is availabe only in OTG controller core. It has four sections:

- OTG Interrupt enables (Read/Write)
- OTG Interrupt status (Read/Write to Clear)
- OTG Status inputs (Read Only)
- OTG Controls (Read/Write)

The status inputs are debounced using a 1 ms time constant. Values on the status inputs that do not persist for more than 1 ms does not cause an update of the status input register, or cause an OTG interrupt.

See also USB Device Mode (USB__USBMODE) register.

Address: USB_UOG_OTGSC is 53F8_0000h base + 1A4h offset = 53F8_01A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	-	DPIE	1msE	BSEIE	BSVIE	ASVIE	AVVIE	IDIE	-	DPIS	1msS	BSEIS	BSVIS	ASVIS	AVVIS	IDIS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-	DPS	1msT	BSE	BSV	ASV	AVV	ID	-		IDPU	DP	OT	-	VC	VD
Reset	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0

USB_UOG_OTGSC field descriptions

Field	Description
31 -	Reserved
30 DPIE	Data Pulse Interrupt Enable
29 1msE	1 millisecond timer Interrupt Enable - Read/Write

Table continues on the next page...

USB_UOG_OTGSC field descriptions (continued)

Field	Description
28 BSEIE	B Session End Interrupt Enable - Read/Write. Setting this bit enables the B session end interrupt.
27 BSVIE	B Session Valid Interrupt Enable - Read/Write. Setting this bit enables the B session valid interrupt.
26 ASVIE	A Session Valid Interrupt Enable - Read/Write. Setting this bit enables the A session valid interrupt.
25 AVVIE	A VBus Valid Interrupt Enable - Read/Write. Setting this bit enables the A VBus valid interrupt.
24 IDIE	USB ID Interrupt Enable - Read/Write. Setting this bit enables the USB ID interrupt.
23 -	Reserved
22 DPIS	Data Pulse Interrupt Status - Read/Write to Clear. This bit is set when data bus pulsing occurs on DP or DM. Data bus pulsing is only detected when USBMODE.CM = Host (11) and PORTSC1(0)[PP] = 0. Software must write a one to clear this bit.
21 1msS	1 millisecond timer Interrupt Status - Read/Write to Clear. This bit is set once every millisecond. Software must write a one to clear this bit.
20 BSEIS	B Session End Interrupt Status - Read/Write to Clear. This bit is set when VBus has fallen below the B session end threshold. Software must write a one to clear this bit
19 BSVIS	B Session Valid Interrupt Status - Read/Write to Clear. This bit is set when VBus has either risen above or fallen below the B session valid threshold (0.8 VDC). Software must write a one to clear this bit.
18 ASVIS	A Session Valid Interrupt Status - Read/Write to Clear. This bit is set when VBus has either risen above or fallen below the A session valid threshold (0.8 VDC). Software must write a one to clear this bit.
17 AVVIS	A VBus Valid Interrupt Status - Read/Write to Clear. This bit is set when VBus has either risen above or fallen below the VBus valid threshold (4.4 VDC) on an A device. Software must write a one to clear this bit.
16 IDIS	USB ID Interrupt Status - Read/Write. This bit is set when a change on the ID input has been detected. Software must write a one to clear this bit.
15 -	Reserved

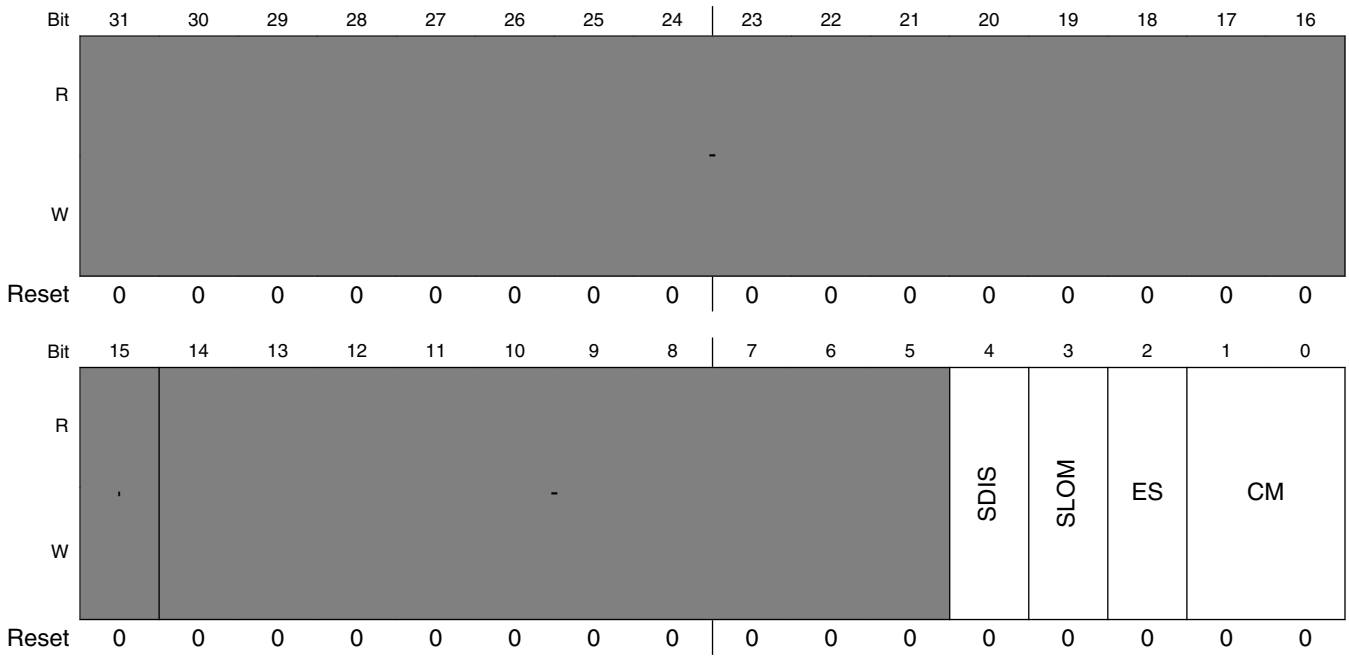
Table continues on the next page...

USB_UOG_OTGSC field descriptions (continued)

Field	Description
14 DPS	Data Bus Pulsing Status - Read Only. A '1' indicates data bus pulsing is being detected on the port.
13 1msT	1 millisecond timer toggle - Read Only. This bit toggles once per millisecond.
12 BSE	B Session End - Read Only. Indicates VBus is below the B session end threshold.
11 BSV	B Session Valid - Read Only. Indicates VBus is above the B session valid threshold.
10 ASV	A Session Valid - Read Only. Indicates VBus is above the A session valid threshold.
9 AVV	A VBus Valid - Read Only. Indicates VBus is above the A VBus valid threshold.
8 ID	USB ID - Read Only. 0 = A device, 1 = B device
7–6 -	Reserved
5 IDPU	ID Pullup - Read/Write This bit provide control over the ID pull-up resister; 0 = off, 1 = on [default]. When this bit is 0, the ID input will not be sampled.
4 DP	Data Pulsing - Read/Write. Setting this bit causes the pullup on DP to be asserted for data pulsing during SRP.
3 OT	OTG Termination - Read/Write. This bit must be set when the OTG device is in device mode, this controls the pulldown on DM.
2 -	Reserved
1 VC	VBUS Charge - Read/Write. Setting this bit causes the VBus line to be charged. This is used for VBus pulsing during SRP.
0 VD	VBUS_Discharge - Read/Write. Setting this bit causes VBus to discharge through a resistor.

77.6.33 USB Device Mode (USB__USBMODE)

Addresses: USB_UOG_USBMODE is 53F8_0000h base + 1A8h offset = 53F8_01A8h
USB_UH1_USBMODE is 53F8_0000h base + 3A8h offset = 53F8_03A8h
USB_UH2_USBMODE is 53F8_0000h base + 5A8h offset = 53F8_05A8h
USB_UH3_USBMODE is 53F8_0000h base + 7A8h offset = 53F8_07A8h



USB_*n*_USBMODE field descriptions

Field	Description
31–16 -	Reserved
15 -	Reserved
14–5 -	Reserved
4 SDIS	Stream Disable Mode. (0 - Inactive [default]; 1 - Active) Device Mode: Setting to a '1' disables double priming on both RX and TX for low bandwidth systems. This mode ensures that when the RX and TX buffers are sufficient to contain an entire packet that the standard double buffering scheme is disabled to prevent overruns/underruns in bandwidth limited systems. Note: In High Speed Mode, all packets received are responded to with a NYET handshake when stream disable is active. Host Mode: Setting to a '1' ensures that overruns/underruns of the latency FIFO are eliminated for low bandwidth systems where the RX and TX buffers are sufficient to contain the entire packet. Enabling stream disable also has the effect of ensuring the TX latency is filled to capacity before the packet is launched onto the USB.

Table continues on the next page...

USB_*n*_USBMODE field descriptions (continued)

Field	Description
	NOTE: Time duration to pre-fill the FIFO becomes significant when stream disable is active. See TX FIFO Fill Tuning (USB__TXFILLTUNING) and TXTTFILLTUNING [MPH Only] to characterize the adjustments needed for the scheduler when using this feature. **NOTE:** The use of this feature substantially limits of the overall USB performance that can be achieved.
3 SLOM	Setup Lockout Mode. In device mode, this bit controls behavior of the setup lock mechanism. See Control Endpoint Operation Model . 0 Setup Lockouts On (default); 1 Setup Lockouts Off (DCD requires use of Setup Data Buffer Tripwire in USB Command Register (USB__USBCMD) .
2 ES	Endian Select - Read/Write. This bit can change the byte alignment of the transfer buffers to match the host microprocessor. The bit fields in the microprocessor interface and the data structures are unaffected by the value of this bit because they are based upon the 32-bit word. Bit Meaning 0 Little Endian [Default] 1 Big Endian
1–0 CM	Controller Mode - R/WO. Controller mode is defaulted to the proper mode for host only and device only implementations. For those designs that contain both host & device capability, the controller defaults to an idle state and needs to be initialized to the desired operating mode after reset. For combination host/device controllers, this register can only be written once after reset. If it is necessary to switch modes, software must reset the controller by writing to the *RESET* bit in the USBCMD register before reprogramming this register. For OTG controller core, reset value is '00b'. For Host1/Host2/Host3 controller core, reset value is '11b'. Bit Meaning 00 Idle [Default for combination host/device] 01 Reserved 10 Device Controller [Default for device only controller] 11 Host Controller [Default for host only controller]

77.6.34 Endpoint Setup Status (USB_UOG_ENDPTSETUPSTAT)

Address: USB_UOG_ENDPTSETUPSTAT is 53F8_0000h base + 1ACh offset = 53F8_01ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-																ENDPTSETUPSTAT																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_UOG_ENDPTSETUPSTAT field descriptions

Field	Description
31–16 -	Reserved
15–0 ENDPTSETUPSTAT	Setup Endpoint Status. For every setup transaction that is received, a corresponding bit in this register is set to one. Software must clear or acknowledge the setup transfer by writing a one to a respective bit after it has read the setup data from Queue head. The response to a setup packet as in the order of operations and total response time is crucial to limit bus time outs while the setup lock our mechanism is engaged. See Managing Endpoints in the Device Operational Model. This register is only used in device mode.

77.6.35 Endpoint Initialization (USB_UOG_ENDPTPRIME)

This register is only used in device mode.

Address: USB_UOG_ENDPTPRIME is 53F8_0000h base + 1B0h offset = 53F8_01B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-								PETB[7:0]								-								PERB[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

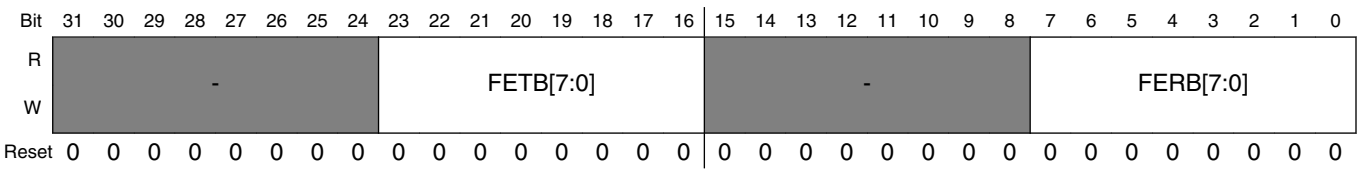
USB_UOG_ENDPTPRIME field descriptions

Field	Description
31–24 -	Reserved
23–16 PETB[7:0]	Prime Endpoint Transmit Buffer - R/WS. For each endpoint a corresponding bit is used to request that a buffer prepared for a transmit operation in order to respond to a USB IN/INTERRUPT transaction. Software should write a one to the corresponding bit when posting a new transfer descriptor to an endpoint. Hardware automatically uses this bit to begin parsing for a new transfer descriptor from the queue head and prepare a transmit buffer. Hardware clears this bit when the associated endpoint(s) is (are) successfully primed. **NOTE:** These bits are momentarily set by hardware during hardware re-priming operations when a dTD is retired, and the dQH is updated. PETB[N] - Endpoint #N, N is in 0..7
15–8 -	Reserved
7–0 PERB[7:0]	Prime Endpoint Receive Buffer - R/WS. For each endpoint, a corresponding bit is used to request a buffer prepare for a receive operation for when a USB host initiates a USB OUT transaction. Software should write a one to the corresponding bit whenever posting a new transfer descriptor to an endpoint. Hardware automatically uses this bit to begin parsing for a new transfer descriptor from the queue head and prepare a receive buffer. Hardware clears this bit when the associated endpoint(s) is (are) successfully primed. **NOTE:** These bits are momentarily set by hardware during hardware re-priming operations when a dTD is retired, and the dQH is updated. PERB[N] - Endpoint #N, N is in 0..7

77.6.36 Endpoint De-Initialize (USB_UOG_ENDPTFLUSH)

This register is only used in device mode.

Address: USB_UOG_ENDPTFLUSH is 53F8_0000h base + 1B4h offset = 53F8_01B4h



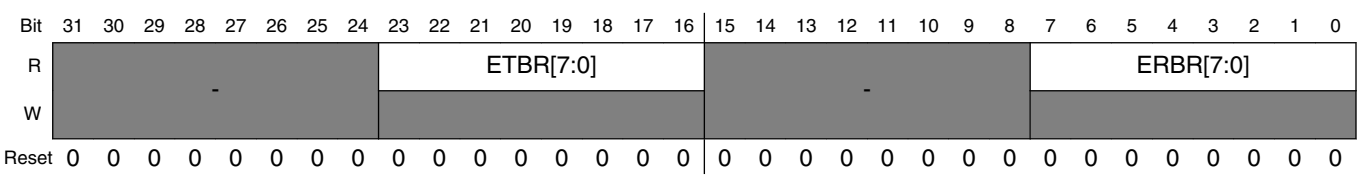
USB_UOG_ENDPTFLUSH field descriptions

Field	Description
31–24 -	Reserved
23–16 FETB[7:0]	Flush Endpoint Transmit Buffer - R/WS. Writing one to a bit(s) in this register causes the associated endpoint(s) to clear any primed buffers. If a packet is in progress for one of the associated endpoints, then that transfer continues until completion. Hardware clears this register after the endpoint flush operation is successful. FETB[N] - Endpoint #N, N is in 0..7
15–8 -	Reserved
7–0 FERB[7:0]	Flush Endpoint Receive Buffer - R/WS. Writing one to a bit(s) causes the assocUOGiated endpoint(s) to clear any primed buffers. If a packet is in progress for one of the associated endpoints, then that transfer continues until completion. Hardware clears this register after the endpoint flush operation is successful. FERB[N] - Endpoint #N, N is in 0..7\

77.6.37 Endpoint Status (USB_UOG_ENDPTSTAT)

This register is only used in device mode.

Address: USB_UOG_ENDPTSTAT is 53F8_0000h base + 1B8h offset = 53F8_01B8h



USB_UOG_ENDPTSTAT field descriptions

Field	Description
31–24 -	Reserved

Table continues on the next page...

USB_UOG_ENDPTSTAT field descriptions (continued)

Field	Description
23–16 ETBR[7:0]	Endpoint Transmit Buffer Ready -- Read Only. One bit for each endpoint indicates status of the respective endpoint buffer. This bit is set to one by the hardware as a response to receiving a command from a corresponding bit in the ENDPTPRIME register. There is always a delay between setting a bit in the ENDPTPRIME register and endpoint indicating ready. This delay time varies based upon the current USB traffic and the number of bits set in the ENDPRIME register. Buffer ready is cleared by USB reset, by the USB DMA system, or through the ENDPTFLUSH register. **NOTE:** These bits are momentarily cleared by hardware during hardware endpoint re-priming operations when a dTD is retired, and the dQH is updated. ETBR[N] - Endpoint #N, N is in 0..7
15–8 -	Reserved
7–0 ERBR[7:0]	Endpoint Receive Buffer Ready -- Read Only. One bit for each endpoint indicates status of the respective endpoint buffer. This bit is set to a one by the hardware as a response to receiving a command from a corresponding bit in the ENDPRIME register. There is always a delay between setting a bit in the ENDPRIME register and endpoint indicating ready. This delay time varies based upon the current USB traffic and the number of bits set in the ENDPRIME register. Buffer ready is cleared by USB reset, by the USB DMA system, or through the ENDPTFLUSH register. **NOTE:** These bits are momentarily cleared by hardware during hardware endpoint re-priming operations when a dTD is retired, and the dQH is updated. ERBR[N] - Endpoint #N, N is in 0..7

77.6.38 Endpoint Complete (USB_UOG_ENDPTCOMPLETE)

This register is only used in device mode.

Address: USB_UOG_ENDPTCOMPLETE is 53F8_0000h base + 1BCh offset = 53F8_01BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-								ETCE								-								ERCE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_UOG_ENDPTCOMPLETE field descriptions

Field	Description
31–24 -	Reserved
23–16 ETCE	Endpoint Transmit Complete Event - R/WC. Each bit indicates a transmit event (IN/INTERRUPT) occurred and software should read the corresponding endpoint queue to determine the endpoint status. If the corresponding IOC bit is set in the Transfer Descriptor, then this bit is set simultaneously with the *USBINT* . Writing one clears the corresponding bit in this register. ETCE[N] - Endpoint #N, N is in 0..7
15–8 -	Reserved

Table continues on the next page...

USB_UOG_ENDPTCOMPLETE field descriptions (continued)

Field	Description
7–0 ERCE	Endpoint Receive Complete Event - RW/C. Each bit indicates a received event (OUT/SETUP) occurred and software should read the corresponding endpoint queue to determine the transfer status. If the corresponding IOC bit is set in the Transfer Descriptor, then this bit is set simultaneously with the *USBINT* . Writing one clears the corresponding bit in this register. ERCE[N] - Endpoint #N, N is in 0..7

77.6.39 Endpoint Control0 (USB_UOG_ENDPTCTRL0)

Every Device implements Endpoint 0 as a control endpoint.

Address: USB_UOG_ENDPTCTRL0 is 53F8_0000h base + 1C0h offset = 53F8_01C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	-								TXE	-			TXT		-	TXS
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	-								RXE	-			RXT		-	RXS
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

USB_UOG_ENDPTCTRL0 field descriptions

Field	Description
31–24 -	Reserved
23 TXE	TX Endpoint Enable 1 Enabled Endpoint0 is always enabled.
22–20 -	Reserved
19–18 TXT	TX Endpoint Type - Read/Write 00 - Control

Table continues on the next page...

USB_UOG_ENDPTCTRL0 field descriptions (continued)

Field	Description
	Endpoint0 is fixed as a Control End Point.
17 -	Reserved
16 TXS	TX Endpoint Stall - Read/Write 0 End Point OK [Default] 1 End Point Stalled Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. It continues returning STALL until the bit is cleared by software or it is automatically cleared upon receipt of a new SETUP request.
15–8 -	Reserved
7 RXE	RX Endpoint Enable 1 Enabled Endpoint0 is always enabled.
6–4 -	Reserved
3–2 RXT	RX Endpoint Type - Read/Write 00 Control Endpoint0 is fixed as a Control End Point.
1 -	Reserved
0 RXS	RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. It continues returning STALL until the bit is cleared by software or it is automatically cleared upon receipt of a new SETUP request.

77.6.40 Endpoint ControlIn (USB_UOG_ENDPTCTRL*n*)

There is a UOG_ENDPTCTRLx register for each endpoint in a device.

NOTE

If one endpoint direction is enabled and the paired endpoint of opposite direction is disabled then the unused direction type must be changed from the default control-type to any other type (that is Bulk-type). leaving an unconfigured endpoint control causes undefined behavior for the data pid tracking on the active endpoint/direction.

Addresses: USB_UOG_ENDPTCTRL1 is 53F8_0000h base + 1C4h offset = 53F8_01C4h
USB_UOG_ENDPTCTRL2 is 53F8_0000h base + 1C8h offset = 53F8_01C8h
USB_UOG_ENDPTCTRL3 is 53F8_0000h base + 1CCh offset = 53F8_01CCh
USB_UOG_ENDPTCTRL4 is 53F8_0000h base + 1D0h offset = 53F8_01D0h
USB_UOG_ENDPTCTRL5 is 53F8_0000h base + 1D4h offset = 53F8_01D4h
USB_UOG_ENDPTCTRL6 is 53F8_0000h base + 1D8h offset = 53F8_01D8h
USB_UOG_ENDPTCTRL7 is 53F8_0000h base + 1DCh offset = 53F8_01DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-								TXE	TXR	TXI	-	TXT		TXD	TXS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-								RXE	RXR	RXI	-	RXT		RXD	RXS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_UOG_ENDPTCTRL*n* field descriptions

Field	Description
31–24 -	Reserved
23 TXE	TX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
22 TXR	TX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
21 TXI	TX Data Toggle Inhibit 0 PID Sequencing Enabled. [Default] 1 PID Sequencing Disabled. This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.

Table continues on the next page...

USB_UOG_ENDPTCTRL*n* field descriptions (continued)

Field	Description
20 -	Reserved
19–18 TXT	TX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
17 TXD	TX Endpoint Data Source - Read/Write 0 Dual Port Memory Buffer/DMA Engine [DEFAULT] Should always be written as 0.
16 TXS	TX Endpoint Stall - Read/Write 0 End Point OK 1 End Point Stalled This bit is set automatically upon receipt of a SETUP request if this Endpoint is not configured as a Control Endpoint. It is cleared automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. It continues to returning STALL until this bit is either cleared by software or automatically cleared as above. **NOTE:** For CONTROL type endpoint, there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. Take care that the STALL bit is not set immediately after writing a '1' to it. Please follow this procedure: continually write this STALL bit until it is set or until a new setup has ben received by checking the associated ENDPTSETUPSTAT bit.
15–8 -	Reserved
7 RXE	RX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
6 RXR	RX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the host and device.
5 RXI	RX Data Toggle Inhibit 0 Disabled [Default] 1 Enabled This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always accept data packet regardless of their data PID.
4 -	Reserved.

Table continues on the next page...

USB_UOG_ENDPTCTRL*n* field descriptions (continued)

Field	Description
3–2 RXT	RX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Reserved
1 RXD	RX Endpoint Data Sink - Read/Write - TBD 0 Dual Port Memory Buffer/DMA Engine [Default] Should always be written as zero.
0 RXS	RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled This bit is set automatically upon receipt of a SETUP request if this Endpoint is not configured as a Control Endpoint. It is cleared automatically upon receipt a SETUP request if this Endpoint is configured as a Control Endpoint, Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. It continues to returning STALL until this bit is either cleared by software or automatically cleared as above.

77.6.41 ULPI Viewport (USB__ULPIVIEW)

The register provides indirect access to the ULPI PHY register set. Although the core performs access to the ULPI PHY register set, there may be extraordinary circumstances where software may need direct access.

NOTE

Writes to the ULPI through the ULPI Viewport can substantially harm standard USB operations. Most of the ULPI register settings are under direct control of the USB core and should not be changed by software through viewport writes as this may harm USB operations.

NOTE

Executing read operations though the ULPI Viewport should have no harmful side effects to standard USB operations.

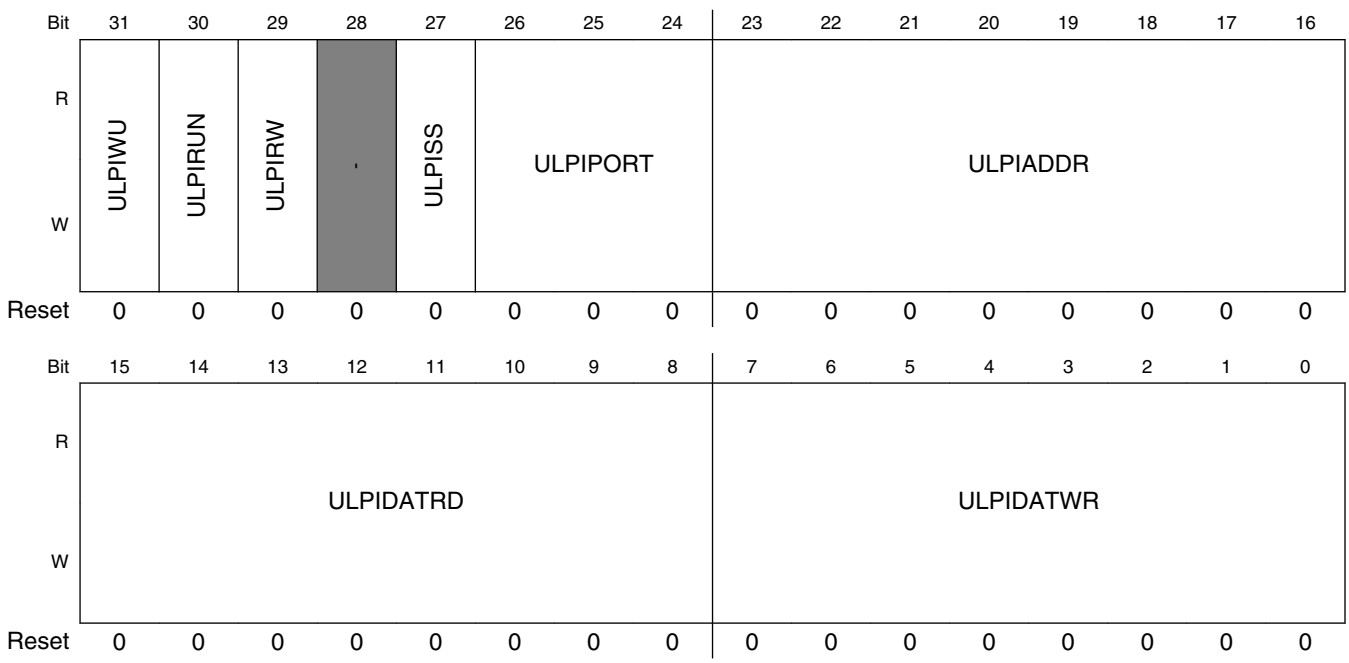
There are two operations that can be performed with the ULPI Viewport, wake-up and read /write operations. The wake-up operation is used to put the ULPI interface into normal operation mode and re-enable the clock if necessary. A wake-up operation is required before accessing the registers when the ULPI interface is operating in low power mode, serial mode, or carkit mode. The ULPI state can be determined by reading the sync. state bit (ULPISS). If this bit is a one, then ULPI interface is run

operation mode and can accept read/write operations. If the ULPISS indicates a '0' then read/write operations will not be able execute. Undefined behavior will result if ULPISS = 0 and a read or write operation is performed. To execute a wake-up operation, write all 32-bits of the ULPI Viewport where ULPIPORT is constructed appropriately and the ULPIWU bit is a '1' and ULPIRUN bit is a '0'. Poll the ULPI Viewport until ULPIWU is zero for the operation to complete.

To execute a read or write operation, write all 32-bits of the ULPI Viewport where ULPIDATWR, ULPIADDR, ULPIPORT, ULPIRW are constructed appropriately and the ULPIRUN bit is a '1'. Poll the ULPI Viewport until ULPIRUN is zero for the operation to complete. Once ULPIRUN is zero, the ULPIDATRD will be valid if the operation was a read.

The polling method above could also be replaced and interrupt driven using the ULPI interrupt defined in the USB Status Register (USB__USBSTS) and Interrupt Enable Register (USB__USBINTR) registers. When a wake-up or read/write operation complete, the ULPI interrupt will be set.

Addresses: USB_UOG_ULPIVIEW is 53F8_0000h base + 770h offset = 53F8_0770h
USB_UH1_ULPIVIEW is 53F8_0000h base + 970h offset = 53F8_0970h
USB_UH2_ULPIVIEW is 53F8_0000h base + B70h offset = 53F8_0B70h
USB_UH3_ULPIVIEW is 53F8_0000h base + D70h offset = 53F8_0D70h



USB_*n*_ULPIVIEW field descriptions

Field	Description
31 ULPIWU	ULPI wake-up - Read/Write.

Table continues on the next page...

USB_*n*_ULPIVIEW field descriptions (continued)

Field	Description
	Writing the 1 to this bit will begin the wake-up operation. The bit will automatically transition to 0 after the wake-up is complete. Once this bit is set, the driver can not set it back to 0. Note: The driver must never execute a wake-up and a read/write operation at the same time.
30 ULPIRUN	ULPI Read/Write Run - Read/Write. Writing the 1 to this bit will begin the read/write operation. The bit will automatically transition to 0 after the read/write is complete. Once this bit is set, the driver can not set it back to 0. **NOTE:** The driver must never execute a wake-up and a read/write operation at the same time.
29 ULPIRW	ULPI Read/Write Control - Read/Write. (0) - Read; (1) - Write. This bit selects between running a read or write operation.
28 -	Reserved
27 ULPISS	ULPI Sync State - Read Only. (1) - Normal Sync. State. (0) In another state (that is, carkit, serial, low power) This bit represents the state of the ULPI interface. Before reading this bit, the ULPIPORT field should be set accordingly if used with the multi-port host. Otherwise, this field should always remain 0.
26–24 ULPIPORT	ULPI Port Number - Read/Write. For the wake-up or read/write operation to be executed, this value selects the port number the ULPI PHY is attached to in the multi-port host. The range is 0 to 7. This field should always be written as a 0 for the non-multi port products.
23–16 ULPIADDR	ULPI Data Address - Read/Write. When a read or write operation is commanded, the address of the operation is written to this field.
15–8 ULPIDATRD	ULPI Data Read - Read Only. After a read operation completes, the result is placed in this field.
7–0 ULPIDATWR	ULPI Data Write - Read/Write. When a write operation is commanded, the data to be sent is written to this field.

Chapter 78
Video Processing Unit (VPU)

78.1 Introduction

The Video Processing Unit (VPU) is the multi-media video processing block in i.MX 6Dual/6Quad.

The following figure shows VPU top-level diagram.

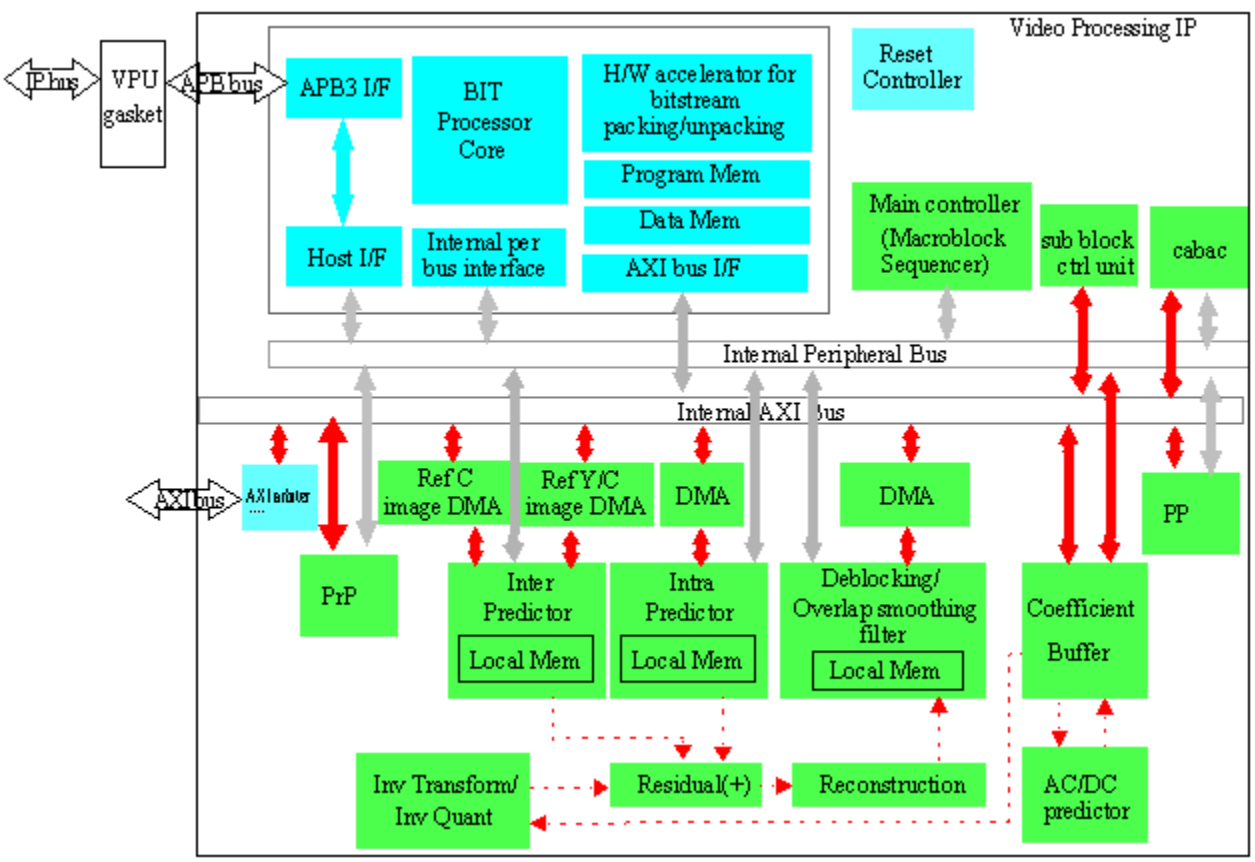


Figure 78-1. VPU Block Diagram

78.1.1 Overview

Video Processing Unit of i.MX53 is a high performance multi-standard video codec which can decode H.264 BP/MP/HP, VC-1 SP/MP/AP, MPEG-4 SP/ASP, H.263 P0/P3, MPEG-1/2 MP, Divx (Xvid) HP/PP/HTP/HDP, RV8/9/10, Sorenson Spark, VP8 (1280x720), AVS, H.264-MVC (1280x720), MJPEG BP (max. 8192x8192) up to full-HD 1920x1088 @30fps.

It can also perform H.264 BP/CBP, MPEG-4 SP, H.263 P0/P3, MJPEG BP (max. 8192x8192) encoding up to 1280x720 @30fps. VPU can support encode or decode multiple video clips with multiple standards simultaneously.

VPU has two bus interface: IP bus for register access controlling and AXI bus for data throughput.

VPU makes use of external memory space for bitstream buffer, parameter buffer, bit code buffer, working buffer and frame buffer. VPU has 51Kbyte internal memories to achieve high performance decoding.

VPU has an embedded BIT processor controlling internal video processing sub-blocks and communicating with host processor through host interface. Very low resources of the ARM platform is required to support VPU. Host processor only need to access VPU registers for initializing VPU or setting parameters during frame gap. In application, it is done through VPU API called by host processor. Large part of video codec (except BIT processor) is optimally shared which enables to achieve low power and low gate count.

VPU combined with processing capabilities of Image Processing Unit (IPU) can support high quality video processing applications for i.MX 6Dual/6Quad.

78.1.2 Features

VPU supports H.264 BP/MP/HP, VC-1 SP/MP/AP, MPEG-4 SP/ASP, H.263 P0/P3, MPEG-1/2 MP/HP, Divx (Xvid) HP/PP/HTP/HDP, RV8/9/10, Sorenson Spark, MJPEG BP decoding and H.264 BP/CBP, MPEG-4 SP, H.263 P0/P3, MJPEG BP encoding. The supported resolution is up to full-HD 1920x1088@30fps for decoding and 1280x720 @30fps for encoding except MJPEG codec (up to 8192x8192). VPU can support various error resilience tools and also support multiple decoding and full duplex multi-party-call simultaneously.

And the VPU provides programmability, flexibility and ease of upgrade in decoding/encoding or host interface because all the controls in decoding/encoding process and host interface are implemented as firmware in a programmable BIT processor

NOTE

The capabilities of VPU listed here is measured with a nearly ideal bus and memory system. But in real SOC, the capabilities of VPU are also depends on the bus and memory system of SOC.

The detailed feature of VPU is as follows:

Table 78-1. Supported Decoding/Encoding Standards

	Standard	Profile	Level	Resolution
Encoder	H.264	BP	3/3.1	1280x720
	MPEG-4	SP	5/6	1280x720
	H.263	P0/P3	70	1280x720
	MJPEG	BP		8192x8192
Decoder	H.264	BP/MP/HP	3/3.1/4.1	1920x1088
	MPEG-4	SP/ASP	5/6	1920x1088
	H.263	P0/3	70	1920x1088
	VC-1	SP/MP/AP	ML/L1,2,3	1920x1088
	MPEG-2	MP/HP	ML/HL	1920x1088
	DivX/Xvid	HP/PP/HTP/HDP		1920x1088
	RV	8/9/10		1920x1088
	H.264-MVC	BP/MP/HP		1280x720
	MJPEG	BP		8192x8192

- Multi-standard video codec
  - H.264/AVC decoder for baseline profile, main profile and high profile
  - VC-1 decoder for simple profile, main profile and advanced profile
  - MPEG-4 decoder for simple profile, advanced simple profile except GMC
  - H.263 decoder for baseline profile
  - Divx decoder for home theater and high definition profile (version 3.x, 4.x, 5.x, 6.x) and Xvid
  - MPEG-2 decoder for main profile @ main and high level
  - RV decoder for profile 8/9/10
  - MJPEG decoder for Baseline profile
  - H.264/AVC encoder for baseline profile
  - MPEG-4 encoder for simple profile
  - H.263 encoder for baseline profile
  - MJPEG encoder for baseline profile
  - Multiple codec: supports up to 4 decoding/encoding processes simultaneously, each process can have a different format.
- Other features

- Pre/Post rotator and mirror
- Built-in de-ringing filter.
- Built-in de-blocking filter for MPEG-2/MPEG-4/Divx.
- MPEG-2 encoder partial acceleration.
- Simultaneous multi-stream and multi-standard processing capability.
- Robust error detection/concealment.

- Programmability
  - Embeds 16-bit BIT processor that is dedicated to processing bitstream and controlling the hardware.
  - General purpose registers and interrupt generation for communication between host processor and VPU.

78.1.3 Modes of Operation

The VPU has two kinds of operating modes: normal operating mode and low power mode.

78.1.3.1 Normal Operating Mode

Normal operating mode is the video codec processing mode, which can be MPEG-4 encoding/decoding, Divx decoding, H.264 encoding/decoding, VC-1 decoding, or multiple bitstream encoding/decoding in multiple standards simultaneously.

VPU supports up to 4 decoding/encoding processes simultaneously. Each process can have a different format. The host processor creates and executes the specified process by sending parameters and commands to the BIT processor through the VPU API.

78.1.3.2 Low Power Mode

When VPU is in idle state, VPU interface signal vpu_idle is asserted and system can gate off its clock source according to this signal.

If the clock source is gated off and system wants to wake it up, it just re-enable the clock and send command to VPU. If power is off when the system wants to wake it up, VPU has to be re-initialized.

78.2 Functional Description

VPU is a high-performance multi-standard video processing block that supports up to full HD 1920 x 1088 at 30 fps decoding and 1280 x 720 at 30 fps encoding.

78.2.1 VPU Architecture

VPU is a high performance multi-standard video codec that can decode

- H.264 BP/MP/HP
- VC-1 SP/MP/AP
- MPEG-4 SP/ASP
- H.263 P0/P3
- MPEG-1/2 MP/HP
- Divx (Xvid) HP/PP/HTP/HDP
- RV8/9/10
- Sorenson Spark
- MJPEG BP (max. 8192x8192) up to full-HD 1920 x 1088 at 30 fps

It can also perform H.264 BP, MPEG-4 SP, H.263 P0/P3, and MJPEG BP (max. 8192x8192) encoding up to 1280 x 720 at 30 fps. VPU can support encode or decode multiple video clips with multiple standards simultaneously.

It connects with the system via the 32-bit IP bus for system control and 64-bit AMBA3 AXI for data throughput. It uses on-chip memories to achieve high performance.

VPU has a 16-bit DSP called a BIT processor. The BIT processor controls the internal video codec sub blocks and communicates with a host processor through the host interface. The required resource for the host ARM platform to control the VPU is very low, under 1 MIPS, because all functions such as rate control, FMO, ASO, video codec control and error resilience are implemented in the BIT processor. The majority of sub-blocks in VPU are optimally shared, which enables the ultra low power and low gate count.

VPU mainly consists of an embedded 16-bit BIT processor, video codec hardware, and bus arbiter/interface. Refer to for the block diagram of VPU.

78.2.1.1 Embedded BIT processor

The embedded BIT processor is used for parsing or forming bitstreams. It includes some hardware accelerators to speed up bitstream processing.

In addition to handling bitstreams, the BIT processor controls the video decoding hardware and communicates with host processor through IP bus and AXI bus interface.

78.2.1.2 Video CODEC Hardware

All video decoding processing, except handling coefficients for VLD, are implemented in hardware.

The video codec hardware is designed to reduce logic gate count by sharing parts of sub-blocks for multi-standard video decoding.

It mainly includes the following hardware components.

78.2.1.2.1 Inter-Predictor

The Inter-Predictor sub-block uses a reconstructed motion vector that represents the displacement between the block currently being decoded and the corresponding location in the reference frame, to calculate interpolated pixel data for motion compensation.

78.2.1.2.2 AC/DC and Intra-Predictor

There are two intra-prediction sub-blocks in the video decoding hardware. One is for the MPEG-4/H.263 P3/VC-1 AC/DC prediction and another for the H.264 intra-prediction.

In case of VC-1 decoding, the hardware prediction mode decision is used. It brings high performance and low power consumption. The coefficient data of decoding is re-ordered automatically based on the detected prediction mode. For H.264 intra-prediction mode in decoding, hardware mode decision or software based made decision is used.

78.2.1.2.3 Inverse transform/Inverse quantization

VPU has only one transform/quantization sub-block. It operates in several types of inverse transform/quantization for MPEG-2, VC-1 and H.264.

Each inverse transform/quantization for decoding uses different coefficients and different data processes, but it uses same control and data flow in many cases.

78.2.1.2.4 De-blocking/Overlap-smoothing filter

>De-blocking filter removes blocking artifacts resulted from quantization, different motion vectors. The filter processing is applied in both decoder and encoder.

The VPU de-blocking filter supports H.264/H.263/MPEG4/VC-1. Overlap-smoothing filter supports VC-1 overlap-smoothing feature. For H.264, H.263 and VC-1, the de-blocking filter operates within coding loop. Filtered frames are used as reference frames for motion compensation of subsequent coded frames. But for MPEG4, the de-blocking filter operates outside coding loop for only display.

78.2.1.2.5 Coefficient buffer interface

The coefficient buffer interface provides a channel for BIT processor to read quantized coefficients resulted from encoding process or to send variable-length-decoded coefficients to the video codec for decoding process.

78.2.1.2.6 Macroblock controller

VPU has a complex and large number of pipeline for high-performance. To manage it wholly by BIT processor is not suitable, so VPU embeds the macroblock controller to control all sub-blocks of the video codec based on the configuration of pipelining by BIT processor. This scheme reduces the load on BIT processor and guarantees the programmability of the block. Before the video codec encodes or decodes a macroblock, BIT processor configures how the pipeline of the codec is structured. If all processes are completed for encoding/decoding a macroblock, the macro-block controller indicates its completion. In summary,

BIT processor configures which sub-blocks are enabled for current macroblock processing, and the macroblock controller manages corresponding sub-blocks based on the configuration.

The figure below shows the connectivity of macroblock controller.

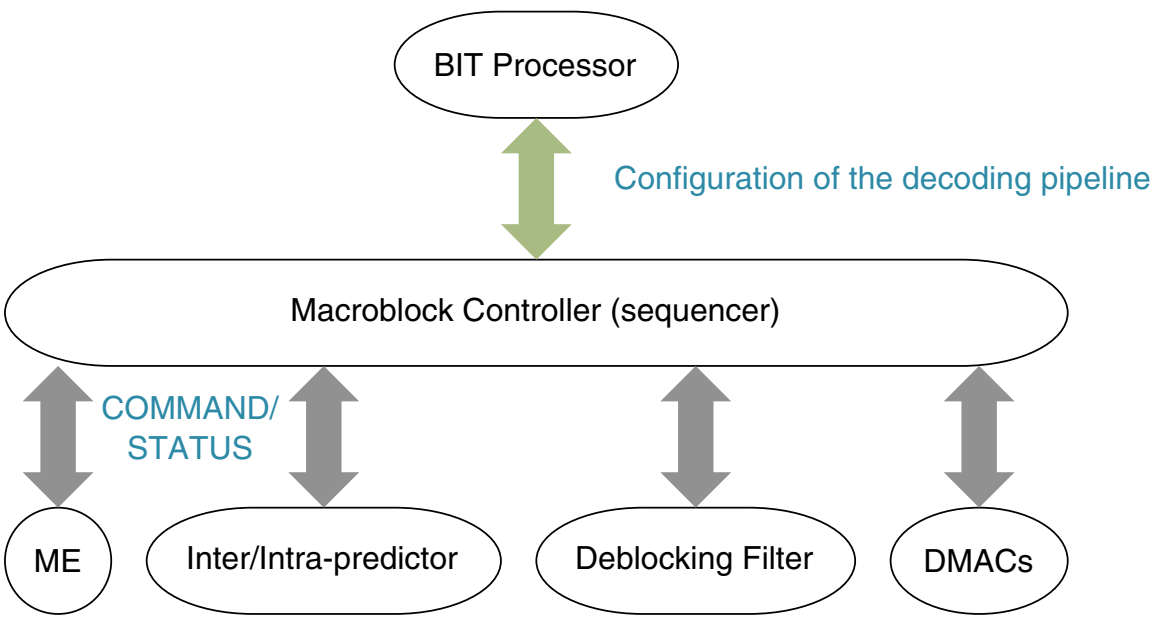


Figure 78-2. The connectivity of macroblock controller

78.2.1.2.7 Rotation/Mirroring

VPU supports rotation together with mirroring function for decoding output image to display. It is done by rotator/mirror sub-block.

The rotation/mirroring process in decoding process requires additional bandwidth because VPU has to re-use the un-rotated image for decoding the next image. So the rotated image is written to other memory space. In this scheme, the display I/F has not to change memory space for displaying the decoded image because subsequent rotated image is written to the same space.

The rotator sub-blocks support 8-types mode of 90 x n degree(n=0, 1, 2, 3) rotating and mirroring. The figure below gives architecture diagram of post rotation/mirroring sub-block.

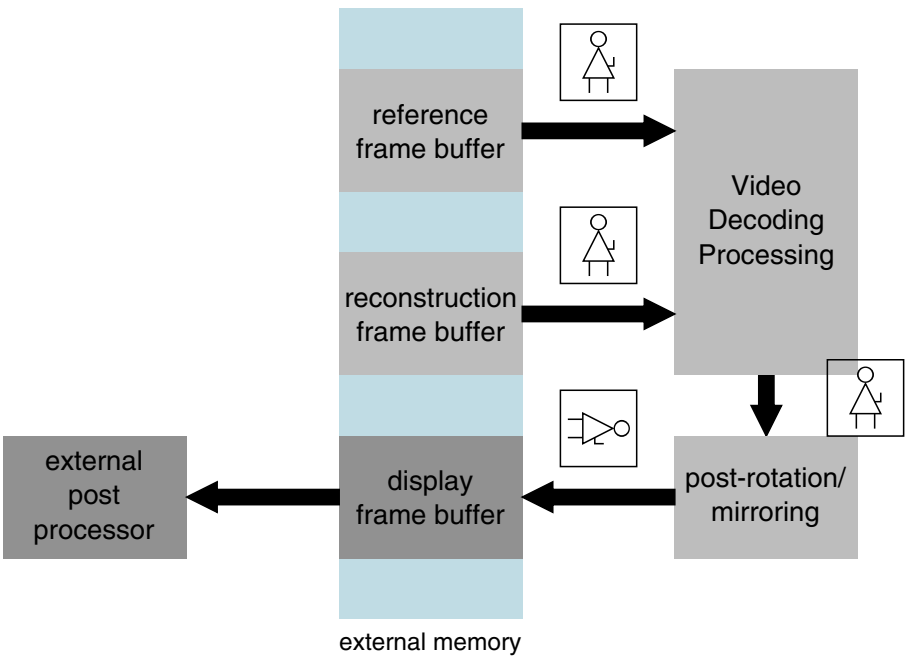


Figure 78-3. Post rotation/mirroring

78.2.2 Clocks

There are four clock domains in VPU.

- AXI bus clock domain (aclk): Controls all AXI bus related functions. Maximum frequency is 200MHz. The aclk derives from the same clock as other blocks performing AXI bus access in i.MX 6Dual/6Quad, the frequency is determined by the whole system performance requirement.
- Decoding core clock domain (cclk): This is VPU functional clock. It controls all VPU decoding functionality, with a maximum frequency of 200MHz. The actual value of the core clock is dependent upon the maximum use case. Clock frequency scaling will be done through the VPU API.

- IP bus clock domain (ipg_clk_s): Controls VPU registers read/write function, with a maximum frequency of 66.5 MHz. ipg_clk_s is a gated clock of IP green-line clock (ipg_clk) with ips_module_en, it is turned off when there is no registers read/write, for power saving.
- Reference clock domain (rclk): Used as reference clock for vpu_idle related logic.

VPU has all four clock domains because it involves in AXI signal generation, functional calculation, IP bus configuration, vpu_idle control logic. Only positive edge clocks is used in VPU design. Each clock is fully asynchronous with other clocks and separated from other clocks.

All clocks can be gated off when VPU is in the idle state to reduce power consumption. There is no restriction about the value of aclk and cclk. The actual frequency of aclk and cclk affects bus bandwidth and video decoding processing capability.

78.2.3 Reset

There are four reset signals going into VPU, corresponding to four clock domains, active low.

- areset_n: used in AXI bus interface. Corresponding clock is aclk.
- creset_n: used in BIT processor and video codec hardware. Corresponding clock is cclk.
- ipg_hard_pos_async_reset_b: used in IP bus interface. Corresponding clock is ipg_clk_s.
- rreset_n: used in reference counter for vpu_idle or vpu_underrun. Corresponding clock is rclk.

The number of cycles for each reset signal must be at least 16 cycles.

VPU embeds an internal reset controller for feature of the software reset from the BIT processor. If any of the VPU blocks except the BIT processor is needed to reset (software reset), the host processor can enable this software reset by setting the software reset register through VPU API. But, the BIT processor cannot be reset by this software reset scheme, because the reset signal of the BIT processor is connected directly from external reset signal.

If reset fires when VPU is processing a transaction through the AXI bus, there is no guarantee that the AXI will complete the transaction normally, because VPU will be reset. If there are any corrupted data in memory, it can be discarded by software. Basically, if the host processor needs to issue a reset, it must check that there is no

transaction on AXI bus between the VPU and external memory interface. In general, the AXI bus is free of VPU transactions when one frame decoding is completed. The start of next frame processing need software configuration

78.2.4 Interrupts

There isone interrupt signal output from VPU. Basically, this interrupt is used to indicate completion of decoding or encoding one frame.

It is generated when VPU interrupt is enabled and as well as the interrupt condition is met. The interrupt signal is active high and retains till the host processor clears it by writing "1" to the interrupt clear register. This signal is synchronized to the positive edge of cclk.

When getting frame completion interrupt, the software can set the parameters for next frame processing and start the BIT processor again. The parameters are mainly the source/destination frame buffer base address. It can be different from the previous frame buffer because the previous completed frame of data may be needed for other image block like display block or IPU in the system.

Basically, the following operation responding to interrupt is dependent upon the application. For example, the software can send the decoded frame to the Image Processing Unit for post-processing and display, at the same time the software should prepare the next bitstream to be processed to the external memory before starting a new processing. This can be done through VPU driver.

78.2.5 Endianness

VPU supports both little and big endian memory system.

User should specify the endianness of the bitstream buffer and frame buffer corresponding to the application scenario. The endianness configuration can be done through VPU API.

78.3 Initialization Information

VPU embedded BIT processor is highly optimized to handle bitstream data.

In addition to processing bitstreams, the BIT processor also controls the communication between VPU and host processor.

The BIT processor firmware to drive video decoding is divided into 2 parts.

One is boot code, which is downloaded by host processor through the IP bus to BIT processor internal memory, it is loaded only once at the VPU initialization stage. The size of the boot code is 1Kbyte. This boot code has also to be written to a region of external memory, and the base address of the firmware region is written through VPU API.

The other is a package of firmware for driving decoding processing. Before starting decoding, firmware has to be written to the continuous region of external memory following the boot code. At run-time, the BIT processor will self-download firmware into internal memory corresponding to the activated decoding standard. It is loaded through the AXI bus.

The VPU initialization process is shown in the figure below.

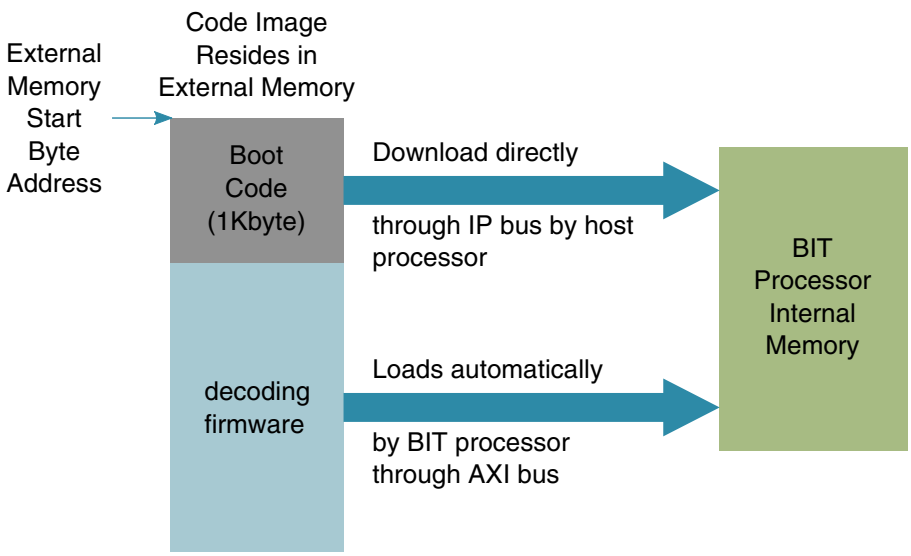


Figure 78-4. VPU initialization process

78.4 Application Information

The figure below shows roles of the BIT processor and VPU video processing core sub-block and how to interface with application software. Basically, at the frame level, host processor communicates with VPU through the provided API's. To give the video decoding more flexibility and debugging capability, all processes related to the bitstream are assigned to the BIT processor.

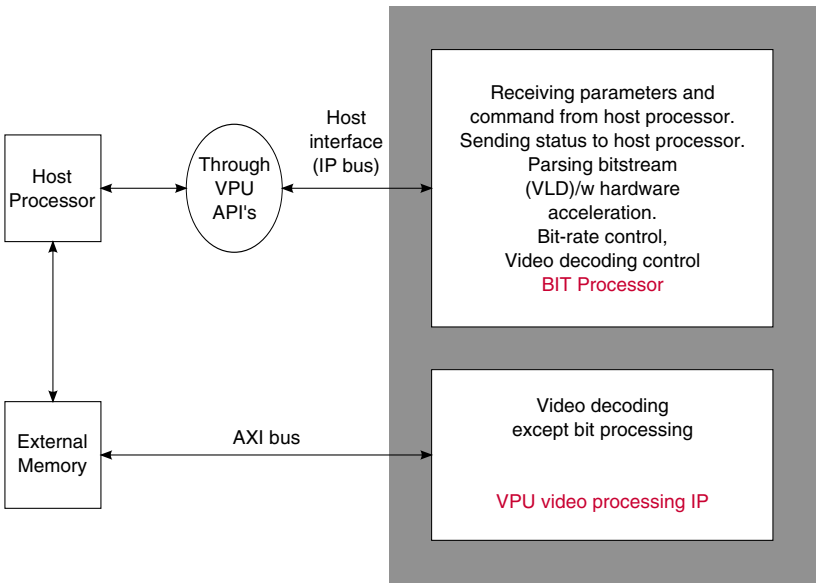


Figure 78-5. VPU interface with application software diagram

78.4.1 Video Decoding Processing Control

This section describes how BIT processor controls the video decoding and communicates with the host.

78.4.1.1 Video Decoding Process Flow

VPU can handle a maximum of 4 processes simultaneously. Each process can have a different format - MPEG-4, MPEG-2, H.264, VC-1 and etc. The figure below shows a simplified state diagram for running decoding process.

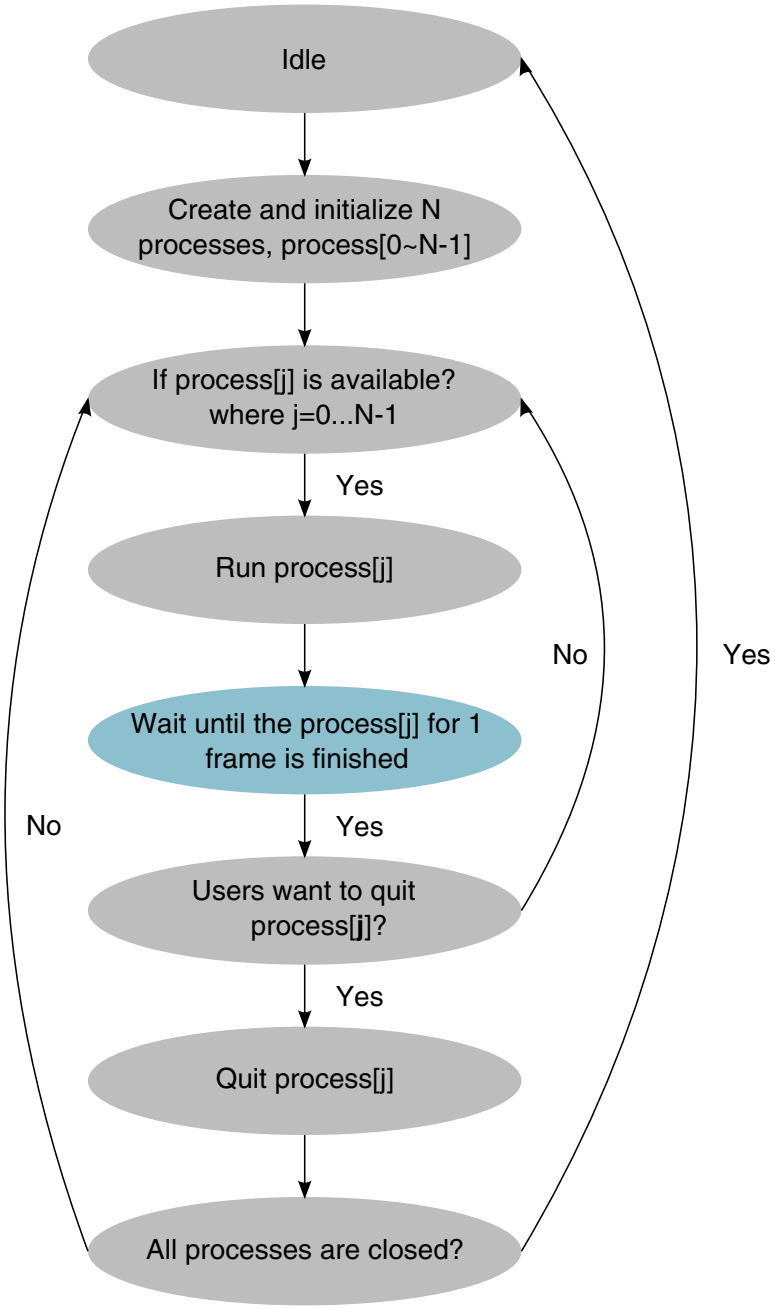


Figure 78-6. Decoding process state diagram

Each decoding process consists of three categories:

- Create processes- Software creates and configures processes.
- Running processes- At a proper time instance, software will begin a specific process. The proper time instance means when the decoding is in idle state and bitstream to be decoded is ready in the external memory.
- Quit Processes- Software can quit a specific process

If more than one process are ready to run, each process must be assigned to different process ID - RunIndex, which is range from 0 to 3. Basically, the ID is assigned based on the order of creation. For example, when 1 MPEG-4 Decoding + 1 H.264 Decoding + 1 MPEG-2 Decoding + 1 VC1 Decoding are running simultaneously, MPEG-4 Decoding is assigned to process index "0", H.264 Decoding is assigned to process index "1", MPEG-2 Decoding is assigned to index "2", and VC1 decoding is assigned to process index "3".

There is no priority rules for executing processes, after creating all processes at the initialization step, the host enables the BIT processor to execute process specified with the RunIndex. All processes are executed in time-division like mechanism, after one process finishes decoding or encoding a frame, another process then can be executed.

In conjunction with the process ID, the RunCodStd needs to be set, to define which coding standard is used with the created process and whether the created process will decode a bitstream. The table below shows the dedicated RunCodStd value for each decoding standard. All this can be done through VPU API.

Table 78-2. RuncCodStd Register Value for Coding Standard

Coding standard	RunCodStd
MPEG-4/H.263 decoding	0
MPEG-4/H.263 encoding	1
H.264 decoding	2
H.264 encoding	3
VC-1 decoding	4

78.4.1.2 Video Decoding Process Command

There are 7 execution commands to initialize, run, quit, set frame and set parameter processes. A command is sent by writing the command value to the RunCommand register through VPU API.

- DEC_SEQ_INIT- This command is to initiate a decoding process. At DEC_SEQ_INIT command, BIT processor finds sequence header and parses the header to extract bitstream information such as picture size then the information is reported to DEC_SEQ_INIT return registers. API should set following configuration parameters before sending this command to VPU.
  - Bitstream buffer base address and size
  - Frame buffers base addresses and stride lines
- DEC_SEQ_END- This command is to terminate a decoding process.
  - After this command, no more PICTURE_RUN commands will be accepted for this process.

- DEC_PIC_RUN- This command is to decode one picture.
  - In decoding case, frame destination address should be set before executing.
- SET_FRAME_BUF- This command informs frame buffer addresses to be used as a decoding/reconstructing image to the BIT processor. Total 63 frame buffers may be used for decoding/reconstructing.
  - The decoding image must be reserved for motion compensation reference until it is not used for reference. So the decoded frame buffer is re-used carefully. The BIT processor receives the whole frame buffer address by this command before picture decoding is started. Then the BIT processor manages the frame buffer allocation for next storage area for decoding.
  - The frame buffer addresses are stored to external memory address. The luminance and one/two chrominance buffer addresses for each frame index must be stored.
- DEC_PARA_SET- This command adds a sequence parameter set or a picture parameter set in H.264 decoder case.
  - The sequence parameter set or the picture parameter set may be conveyed via "out-of-band". In that case, the host must transfer the sequence parameter set or picture parameter set to decoder by this command.
  - The sequence parameter set or picture parameter set must be written to the Parameter buffer of the BIT processor in RBSP format prior to executing this command. The BIT processor decodes the transferred sequence/the picture parameter and stores decoded contents. The decoded sequence/picture parameter set will be activated at decoding slice header with the matched sequence/picture parameter set id.
  - Multiple sequence/picture parameter sets may be delivered to decoder. They are distinguished by different sequence/picture parameter set id. BIT processor can process 32 sequence parameter sets and 256 picture parameter sets.
  - The type (sequence or picture) and size (byte count) of conveyed sequence/ picture parameter set must be delivered to BIT processor by command argument register.
- DEC BUF FLUSH- Flush data in bitstream buffer.
  - After this command finished, bitstream buffer read pointer will be 0. So host must set bitstream buffer write pointer as 0.
- GET F/W VERSION- Get firmware version.

There is a busy status register in the VPU to show whether the BIT processor is executing a command. The busy status keeps "1" until the command is finished, then the BIT processor can accept a new command.

78.4.1.3 Video Decoding Process Finish Detection

VPU raises the interrupt signal or busy state is asserted when a process finishes decoding a frame. So there are three ways of detecting whether the process is finished:

- Polling VPU interrupt status register. When interrupt is generated, the interrupt status register can indicate that.
- Polling VPU busy status register. When a DEC_PIC_RUN command is issued to BIT processor and the process is under operation, the busy status keeps "1", as soon as the busy status becomes "0", the decoding process is finished.
- Capture the interrupt signal from system level, respond to interrupt request within system interrupt service routine.

There is a single busy status register for all processes, user knows which process is finished by the specified running process ID - RunIndex. Interrupt status can be cleared by writing 1 to interrupt clear register. Busy state can be self cleared after read out.

78.4.1.4 Video Decoding Process Flow Example

The figure below shows a process flow example for decoding an H.264 bitstream and decoding an MPEG4 bitstream simultaneously. At first both decoding process are created and initialized, then each process is executed with DEC_PIC_RUN command alternately. More details are described as below.

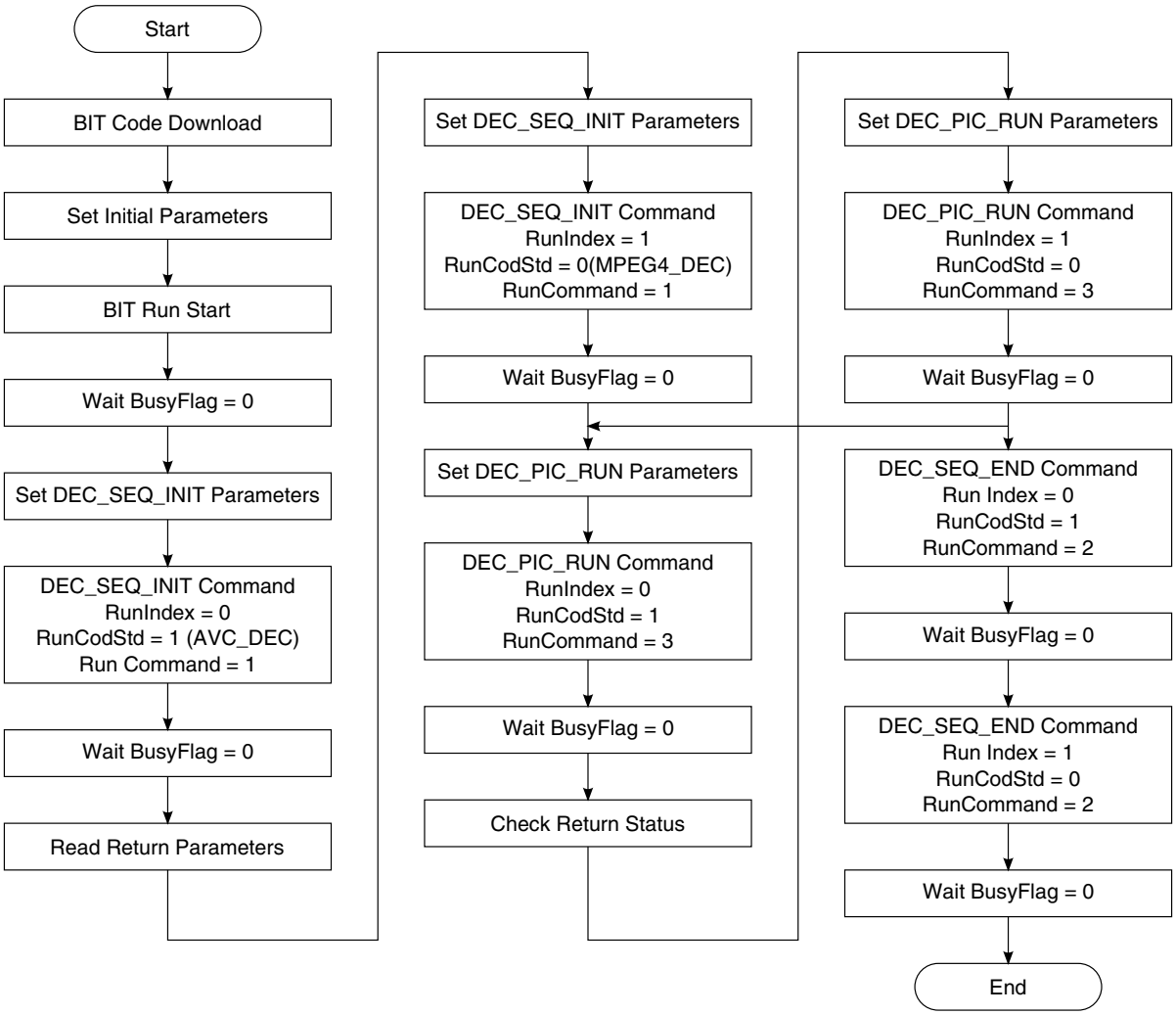


Figure 78-7. H.264 decoding and MPEG4 decoding process flow

NOTE

*RunCommand = 1 (DEC_SEQ_INIT); RunCommand = 2 (DEC_SEQ_END); RunCommand = 3 (DEC_PIC_RUN)

1. Initialize VPU
   - BIT Code Download: Load BIT Processor firmware to memory.
   - Set Initial Parameters: General configuration for BIT processor, setting working buffer base address, BIT Code memory address, bitstream buffer control and so on.
   - BIT Run Start: Run BIT processor to initialize VPU.
2. Create and initialize an H.264 decoding process
   - Set DEC_SEQ_INIT parameters: Configure base address and size of bitstream buffer, base address of frame buffers and so on.
   - Run DEC_SEQ_INIT command: Initiate an H.264 decoding process.

- Wait BusyFlag=0: Wait BIT processor completes DEC_SEQ_INIT command execution.
- Read Return Parameters: Read the features of decoded bitstream, such as the picture resolution and number of reference frames, this can be done by reading the RetSrcFormat and Ret264Info register through VPU API. In this way, the host can prepare the required frame buffers.

3. Create and initialize an MPEG4 decoding process
   - The flow is similar to the H.264 decoding process except the run standard setting.
4. Run the H.264 decoding process
   - Set DEC_PIC_RUN Parameters: Configure the frame destination address.
   - Run DEC_PIC_RUN command: Start the H.264 decoding process.
   - Wait BusyFlag=0: Wait the BIT processor completes DEC_PIC_RUN command execution. It also means one frame process is finished. (The finish detection can be implemented in other way described in Video Decoding Process Finish Detection.) The decoded frame can be sent to the Image Processing Unit for post-processing and display. The actual operation is dependant on the application.
5. Run the MPEG4 decoding process
   - The flow is similar to the H.264 decoding process. The decoding process should configure frame source address in addition to destination address.
6. Execute step 4 and step 5 alternately.
   - Before running decoding process, the host should load new bitstream to the bitstream buffer if it is empty, and update the frame destination address according to the application.
7. Stop the decoding processes
   - Run DEC_SEQ_END command to each process to terminate it.

Basically, the process flow for decoding is similar, though it may have minor change for different firmware version. The detailed implementation is done in the VPU driver.

78.4.2 Video Encoding Processing Control

This section describes how BIT processor controls the video encoding and communicates with the host.

78.4.2.1 The Pipeline for Encoding

The following figure shows the 6 pipeline stages for H.264 encoding.

In encoding, there are two separate data paths. The upper four stages form the encoding path, and the lower four stages form the reconstruction path. .

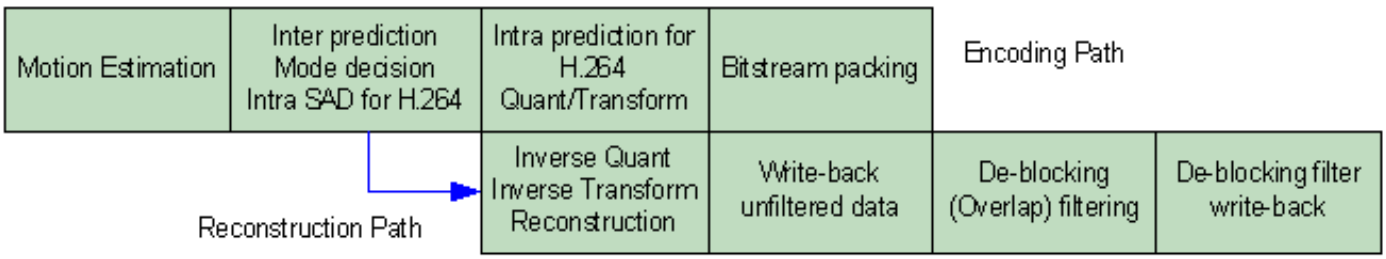


Figure 78-8. Encoding pipeline stages for H.264 Encoding

The sub-block in each stage is controlled by BIT processor and main controller (macroblock sequencer). The detailed H.264 encoding pipeline stage is as follows.

- Pipeline Stage 1
  a. The Motion Estimation (ME) block searches the motion vectors.
  b. The BIT processor reads the motion vectors from ME.
  c. The used data (reference luminance and current macroblock luminance data) is moved to the MC block for reuse.
- Pipeline Stage 2
  a. The MC block writes the predicted data from ME motion vector.
  b. The Intra predictor calculates the Intra SAD and cost for the mode decision.
  c. The Inter/Intra prediction block writes the predicted data to predicted local memory for subtraction in the next pipeline stage.
- Pipeline Stage 3
  a. Execute Intra Prediction and transfer the results to Quantizer/Transform.
  b. Quantized and transformed data are written to the residual buffer for bitstream packing.
  c. Quantized and transformed data are transferred to Inverse Quantizer/ Inverse Transform.
  d. Inverse quantized and inverse transformed data are added (reconstructed) with prediction block data in order to make a reconstructed frame.
- Pipeline Stage 4
  a. The BIT processor reads the residual buffer and makes the bitstream.
  b. Write back the un-filtered pixel data
- Pipeline Stage 5
  a. The deblocking filter reads the reconstructed data from its local memory and runs deblocking filtering for H.264/VC-1 or overlap smoothing for VC-1 intra block.

b. The filtered pixel data, that is the final reconstructed picture, is stored in the local memory of deblocking filter block for write-back.

- Pipeline Stage 6
  a. The write-back DMA writes the decoded picture into external frame buffer and it will be the reference picture of next picture decoding.

shows the 5 pipeline stages for MPEG-4/H.263 encoding when the deblocking filter is disabled and shows the 6 pipeline stages for H.263 when the deblocking filter is enabled. The encoding data path in MPEG-4/H.263 is similar to the H.264 encoding data path, except for the addition of an AC/DC prediction step.

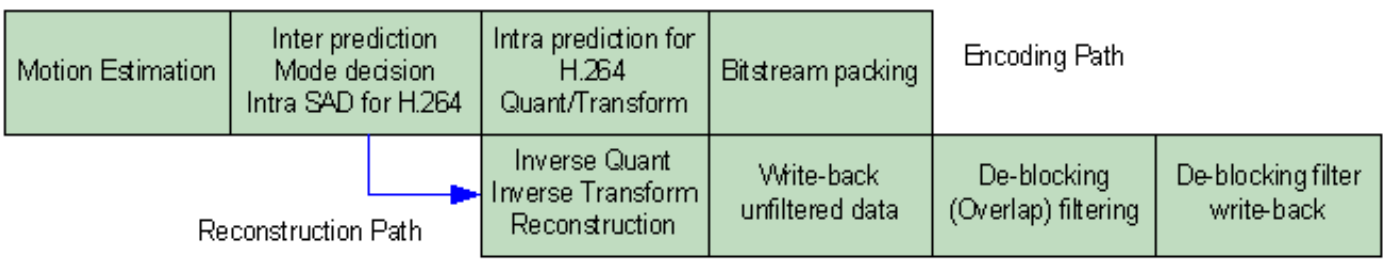


Figure 78-9. Encoding pipeline stages for H.264 encoding

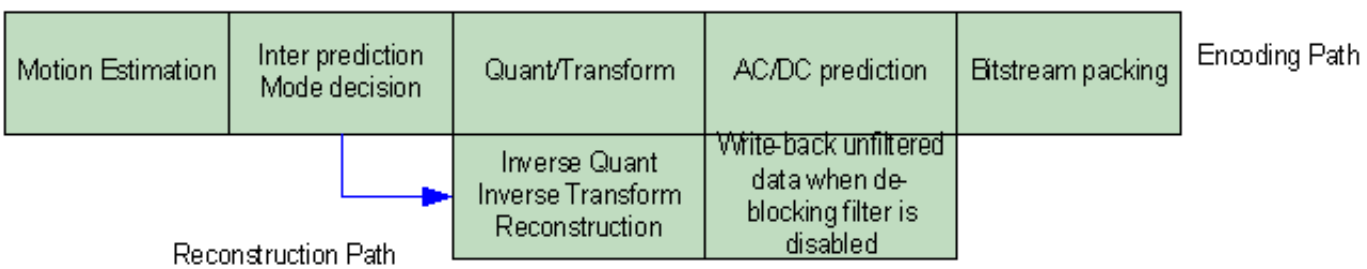


Figure 78-10. MPEG-4/H.263 encoding pipeline stage when deblocking filter is disabled

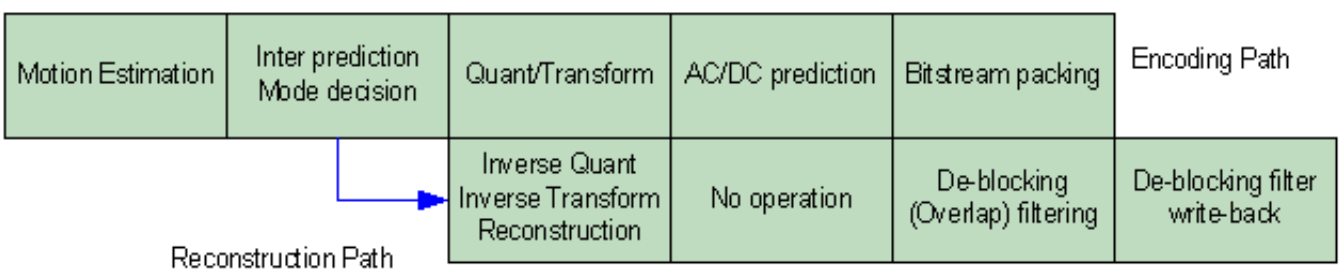


Figure 78-11. H.263 encoding pipeline stage when deblocking filter is enabled

78.4.3 Video Codec Processing Buffer Requirement

VPU has full access to the entire external memory. It uses external memory to load or store image frame, bitstream, program and data for the BIT processor.

AC/DC predication and de-blocking filtering also uses external memory. The buffer size requirement is dependent on the standard and target applications. For example, the H.264 decoding uses multiple reference frames up to 16. MPEG-4 and H.263 decoding uses only one reference frame. Each standard requires a different temporary memory size when it processes de-blocking or overlap-smoothing filtering.

VPU uses five kinds of buffers, as follows:

- Frame Buffer: for storing image frame.
- BIT processor program memory: for boot code and firmware.
- Working Buffer: for intermediate data from the BIT processor and the video decoding hardware.
- Bitstream Buffer: for loading bitstream.
- Parameter Buffer: for BIT processor command execution argument and return data.
- Search RAM: for the use of ME to reduce SDRAM bus loading.

Different buffers can be noncontiguous in external memory, though each buffer must be contiguous.

VPU also supports an optional secondary AXI bus which is connected to internal SRAM for storing temporal data of some sub-blocks, such as de-blocking filter and intra prediction. This decreases the total bandwidth to the external memory.

78.4.3.1 Memory Map Types of Frame Buffer

There are 7 map types of frame buffer:

- Type 0 : linear map
- Type 1 : Frame based tiled map, horizontal addressing
- Type 2 : Frame based tiled map, vertical addressing
- Type 3 : Field based tiled map, vertical addressing
- Type 4 : Frame/Field mixed tiled map, vertical addressing
- Type 5 : Tiled MB Raster Frame Map
- Type 6 : Tiled MB Raster Field Map

In i.MX 6Dual/6Quad, only Type 0, 5 and 6 are supported.

In the linear map, a pixel is read out from or written to the frame buffer in a raster scan order for a frame. However, VPU requires pixels on an MB basis and such address access cannot be continuous on the physical memory. Meanwhile in the tiled

memory region is logically split into a specific number of tile-shaped segments. A tile can be an access unit in which addresses are sequential and have efficiently accessible structure. The size of tile can be configurable according to applications. It might be an MB at least or a group of MBs whose addresses belong to the same row and bank address and are consecutive column address in SDRAM.

Meanwhile in the tiled map, the whole memory region is logically split into a specific number of tile-shaped segments. A tile can be an access unit in which addresses are sequential and have efficiently accessible structure. The size of tile can be configurable according to applications. It might be an MBat least or a group of MBs whose addresses belong to the same row and bank address and are consecutive column address in SDRAM. For more details, please refer to VPU API document.

78.4.3.2 Frame Buffer

This section describes the memory map of the frame buffer and the size requirement.

- Only 4:2:0 for all standards except (M)JPEG codec: in case of MJPEG, 4:0:0, 4:2:0, 4:2:2, and 4:4:4 are supported.
- Luminance, Cb and Cr frame buffer base addresses in 16-byte alignment. In case of the interleaved Cb/Cr map, a base address for Cr is ignored because a base address for the Cb is used to store or load the interleaved Cb/Cr samples.
- Stride for the width of the luminance frame buffer should be equal or greater than the width of picture and multiple of 16.
- Endian of the frame buffers is configurable as little or big endian.
- Both the interleaved Cb/Cr map and the separate Cb/Cr map are supported.
- A field pair is stored in a single frame buffer.

As shown in Figure 78-12, a frame buffer is specified with the base address and the stride line. A complete image consists of Y, U, and V components. There are two kinds of configuration, separate Cb/Cr map and interleaved Cb/Cr map.

For the separate Cb/Cr configuration, each chroma component has its own buffer. Therefore, one frame buffer needs 3 buffers for Y, U, and V components, and these buffers can be noncontiguous in memory.

For the interleaved Cb/Cr configuration, only one buffer is needed for chroma components. Therefore, one frame buffer needs 2 buffers for Y, U, and V components, and these buffers can be noncontiguous in memory.

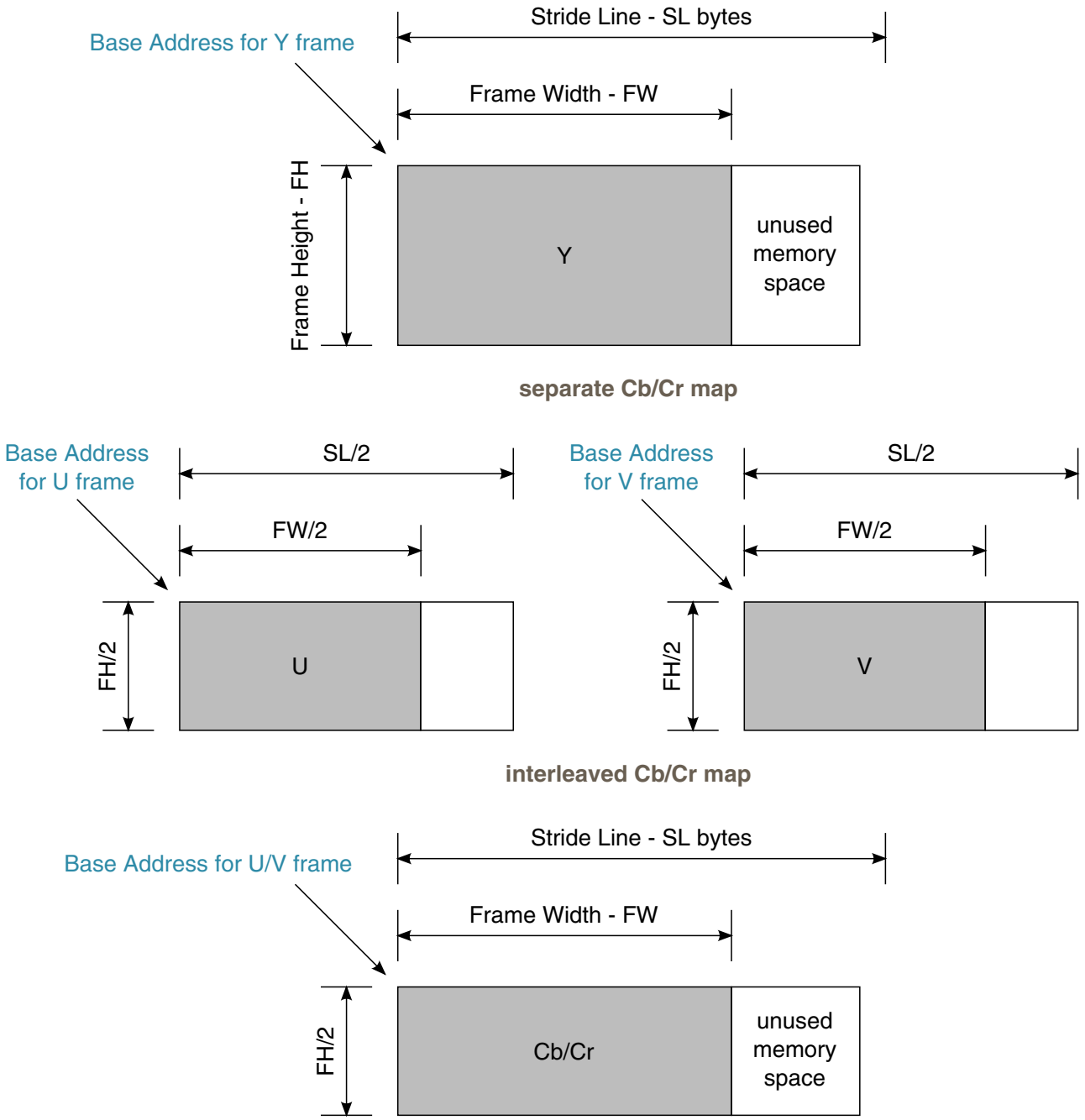


Figure 78-12. Frame Buffer configuration

Figure 78-13 shows the memory map of the frame buffer. In separate Cb/Cr map, for V frame buffer, the memory map is the same as the U frame buffer except for the base address.

VPU supports both little and big endian systems. Y(0,0) could be located in the bit[31:24]. User can specify the endianness through VPU API.

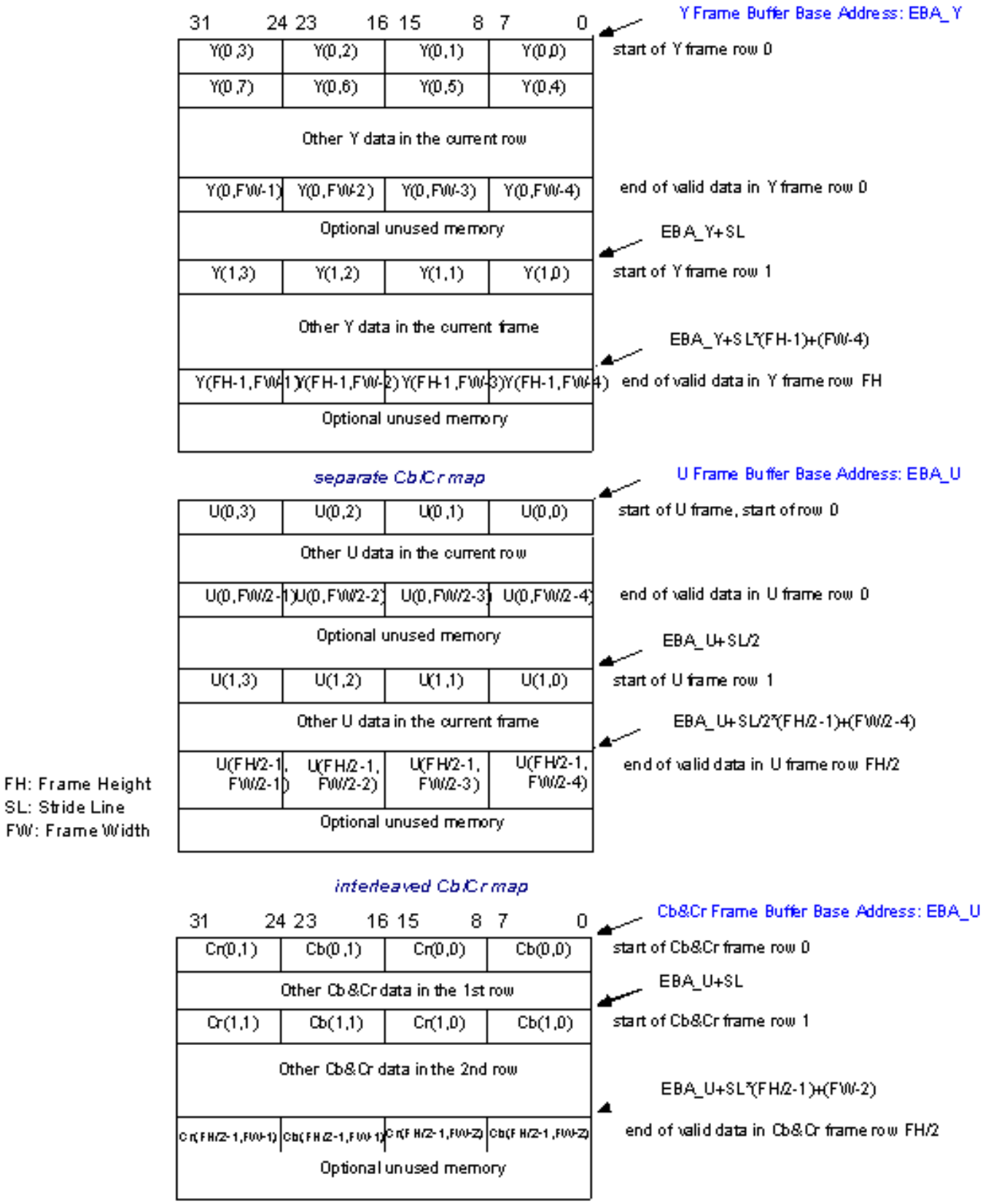


Figure 78-13. Frame Buffer address map in little endian

The table below shows the frame buffer requirement for MPEG-4, H.264 and VC-1. The H.264 decoder requires multiple reference frames up to 16. In the case of VC-1 main profile, the decoder needs two reference frames to decode a B picture. Also, VC-1 requires two more frames that it stores for range reduction or multi-resolution.

The table below also shows the memory requirement in case of QCIF/CIF/VGA resolution image. In the case of H.264 decoding, the required size for the reference frame is dependent on the level being supported. The VPU supports up to H.264 decoding level 3.0, which the maximum decoded picture buffer size is defined as 3037.5Kbyte in the standard. To support H.264 CIF at level 3.0, 2524Kbyte is needed if 16 reference frames are used.

Table 78-3. Frame Buffer Requirement

		MEPG-4 Decoder	H.264 Decoder	VC-1 Decoder
QCIF	Reference Frames	2	16	2
	Instant Frames	1	1	2
	Display Frames	1	1	2
	Total Frames	4	18	6
	Picture Size[1]	37Kbyte	37Kbyte	37Kbyte
	Total Frame Size	148Kbyte	668Kbyte	222Kbyte
CIF	Reference Frames	2	16	2
	Instant Frames	1	1	2
	Display Frames	1	1	2
	Total Frames	4	18	6
	Picture Size	148Kbyte	148Kbyte	148Kbyte
	Total Frame Size	592Kbyte	2672Kbyte	891Kbyte
VGA	Reference Frames	2	5	2
	Instant Frames	1	1	2
	Display Frames	1	1	2
	Total Frames	4	7	6
	Picture Size	450Kbyte	450Kbyte	450Kbyte
	Total Frame Size	1800Kbyte	3150Kbyte	2700Kbyte

1. The Picture Size is the minimum size of one frame buffer with assumption that the picture is YUV 4:2:0 format and the stride line is equal to frame width.

78.4.3.3 BIT Processor Program Buffer

At the initialization stage of VPU, the host processor must download boot code to the BIT processor. After initialization, the BIT processor loads a program corresponding to the activated standard.

The program size of the boot code is 1Kbyte. There are total 48 Kbyte sizes for the current version of BIT firmware to support multi-standards decoding.

78.4.3.4 Working Buffer

Besides buffers for frames and firmware, additional working buffer for intermediate data from the BIT processor and decoding is needed.

The buffers are such as the reconstructed pixel row buffer for MPEG-4 AC/DC prediction or H.264 intra-prediction, context saving buffer for running multiple processes and various temporal storage buffer for decoding process.

The required working buffer size varies according to decode size, decoding standard, decoding capability. For example, AC/DC prediction buffer size is determined by picture width and the maximum bitstream re-ordering buffer for data partition is determined by the maximum bitstream size of one picture. Working buffer size may change for different firmware version. The current version of firmware requires max 256 Kbyte for working. Its size can be set through VPU API. The detailed working buffer is organized as indicated in the table below.

Table 78-4. Working buffer organization

Working Buffer Type	Name	Description	size
static buffer	STATIC_PRC_DMEM	BIT processor data memory of each process for context switching.	5Kbyte for each process
	STATIC_PRC_SEQ	static data storage of each sequence.	8Kbyte for each process

Table continues on the next page...

Table 78-4. Working buffer organization (continued)

Working Buffer Type	Name	Description	size
temp mpeg-4 decoding/encoding buffer	MP4_DEC_ACDC	AC/DC prediction buffer of Y/Cb/Cr	204Kbyte
	MP4_DEC_DP	bitstream reordering buffer for data partition.	
	MP4_ENC_ACDC	AC/DC prediction buffer of Y/Cb/Cr	
	MP4_ENC_DP	bitstream reordering buffer for data partition.	
temp avc decoding/ebcidubg buffer	AVC_DEC_FMO	FMO group status buffer	
	AVC_DEC_IP	intra-prediction buffer of Y/Cb/Cr	
	AVC_DEC_SLICE_INFO	slice information buffer. maximum 1280 slices per picture.	
	AVC_DEC_SLICE	slice data RBSP buffer. All slice data RBSP of one picture is stored.	
	AVC_ENC_IP	intra-prediction buffer of Y/Cb/Cr	

78.4.3.5 Bitstream Buffer

The host processor has to assign buffers for bitstreams on a per instance basis.

If VPU handles N-bitstreams simultaneously in an application, the host should assign N bitstream buffers, and specify the base address and size. The External bitstream buffer is "ring buffer" type. The start address of ring buffer and buffer size must be written by host to BIT processor. The current read or write address of ring buffer is automatically wrapped-around by firmware.

In decoding case, the host writes the bitstream to be decoded then BIT processor reads bitstream. In this case, the bitstream overwriting or underflow may occur and if it occurs, decoding will fail. To prevent overwriting or underflow, current bitstream read/write pointer must be exchanged between the host and the BIT processor.

The BIT processor writes current read pointer of ring buffer to internal register and the host must write current write pointer of ring buffer to internal register. The BIT processor checks the bit buffer empty (underflow) status by comparing current read pointer and write pointer. If no more bitstream data is available to be decoded (buffer empty status), the BIT processor stops bitstream decoding to prevent mis-reading the bitstream and waits until the host writes more bitstream data and updates write pointer. The host must check the current read pointer and write pointer before writing more bitstream data to ring buffer to prevent overwriting bitstream data.

78.4.3.6 Parameter Buffer

Host processor must reserve parameter buffer in external memory for BIT processor command execution argument and return data.

78.4.3.7 Search RAM

VPU's motion estimation sub-block uses a search RAM to reduce the bandwidth on the external SDRAM.

Generally, motion estimation reads a reference pixel data several times. To avoid this B/W overhead, the motion estimation sub-block loads the reference pixel data form the external SDRAM one time and stores them to the search RAM through AXI bus. The stored reference pixel data is loaded several times to search the motion vector, but these operations are conducted through AMBA AXI bus. Therefore the B/W of loading reference pixel data will be reduced using search RAM in AXI bus.

The size of search RAM is (# of horizontal pixel x 36 + 8 macroblocks) bytes.

For example, in CIF encoding, the search RAM size is 352x36 + 2048 = 14720 bytes = 14.375Kbyte.

78.4.3.8 Buffer Requirement Summary

Table 78-5 shows a summary of buffer requirement for each decoding instance, where bitstream buffer size is not considered because the size is not limited. The total size may change for different firmware version. Except the BIT processor program memory, other kinds of buffer has to be assigned on a per instance basis. The overall buffer size for a multiparty call application is nearly the sum of each instanced decoding.

Table 78-5. Summary of Buffer Requirement

		MEPG-4 Decoder	H.264 Decoder	VC-1 Decoder
QCIF	Frames buffer	148KB	668Kbyte	222Kbyte
	BIT processor program buffer	128Kbyte		
	Parameter buffer	8Kbyte		
	Bitstream buffer	1000Kbyte		
	Working Buffer	13Kbyte	256Kbyte	13Kbyte
	Total	1297Kbyte	2060Kbyte	1371Kbyte

Table continues on the next page...

Table 78-5. Summary of Buffer Requirement (continued)

		MEPG-4 Decoder	H.264 Decoder	VC-1 Decoder
CIF	Frames Size	592Kbyte	2672Kbyte	891Kbyte
	BIT processor program buffer	128Kbyte		
	Parameter buffer	8Kbyte		
	Bitstream buffer	1000Kbyte		
	Working Buffer	13Kbyte	256Kbyte	13Kbyte
	Total	1741Kbyte	4064Kbyte	2040Kbyte
VGA	Frame Size	1800Kbyte	3150Kbyte	2700Kbyte
	BIT processor program buffer	128Kbyte		
	Parameter buffer	8Kbyte		
	Bitstream buffer	1000Kbyte		
	Working Buffer	13Kbyte	256Kbyte	13Kbyte
	Total	2949Kbyte	4542Kbyte	3849Kbyte

78.5 Programmable Registers

VPU registers are all 32-bit wide, support only 32bit aligned read/write operation. VPU registers are grouped into several regions corresponding to different decoding process step. They are used for decoding process configuration and control. They can only be accessed through IP bus interface.

Please note that there are some undefined address space in VPU memory map, any read/write accessing to this register address space is ignored in the VPU. Read accessing to write only register returns ZERO value. Write accessing to read only register is ignored.

The BIT processor registers' memory map in VPU is 0xBASE_0000~0xBASE_01FC. The BIT processor registers are divided into 2 categories.

- Address 0xBASE_0000~0xBASE_00FC (64 registers address space) are hardware registers. These registers have reset values and their functions are fixed (not configurable).
- Address 0xBASE_0100~0xBASE_01FC (64 registers address space) are software registers. They have no reset values and can be configured by internal BIT processor. Their definitions may change for different firmware version, so they are not provided

here. This type of registers can be used as general parameter registers between host processor and BIT processor.

- The first 32 parameter registers (address 0xBASE_0100~0xBASE_017C) are used as static parameters. Definition and functions of those registers are same for all kinds of run commands.
- The second 32 parameter registers (address 0xBASE_0180~0xBASE_01FC) are used as temporal parameters. The definition and functions of those registers may differ in different run commands.

The memory map for the hardware registers of VPU is shown in the table below.

Please refer to the VPU API document for descriptions on software registers access.

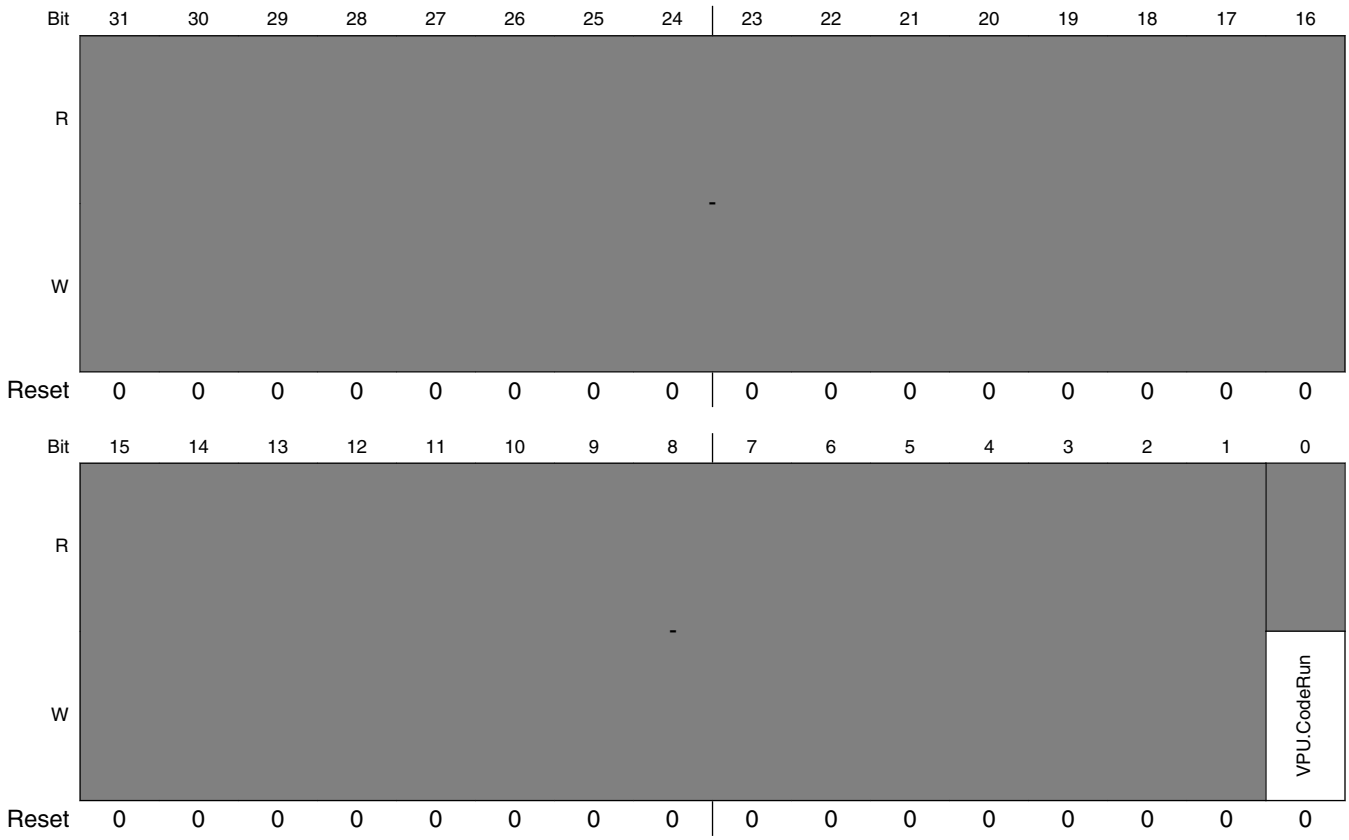
VPU memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
63FF_4000	BIT Processor run start (VPU_CodeRun)	32	W	0000_0000h	78.5.1/ 4994
63FF_4004	BIT Boot Code Download Data register (VPU_CodeDown)	32	W	0000_0000h	78.5.2/ 4994
63FF_4008	Host Interrupt Request to BIT (VPU_HostIntReq)	32	W	0000_0000h	78.5.3/ 4995
63FF_400C	BIT Interrupt Clear (VPU_BitIntClear)	32	W	0000_0000h	78.5.4/ 4996
63FF_4010	BIT Interrupt Status (VPU_BitIntSts)	32	R	0000_0000h	78.5.5/ 4997
63FF_4014	BIT Code Reset (VPU_BitCodeReset)	32	W	0000_0000h	78.5.6/ 4998
63FF_4018	BIT Current PC (VPU_BitCurPc)	32	R	0000_0000h	78.5.7/ 4999
63FF_4020	BIT CODEC Busy (VPU_BitCodecBusy)	32	R	0000_0000h	78.5.8/ 5000

78.5.1 BIT Processor run start (VPU_CodeRun)

See the figure below for illustration of valid bits in VPU Code Run Register and the table below for description of the bit fields in the register.

Address: VPU_CodeRun is 63FF_4000h base + 0h offset = 63FF_4000h



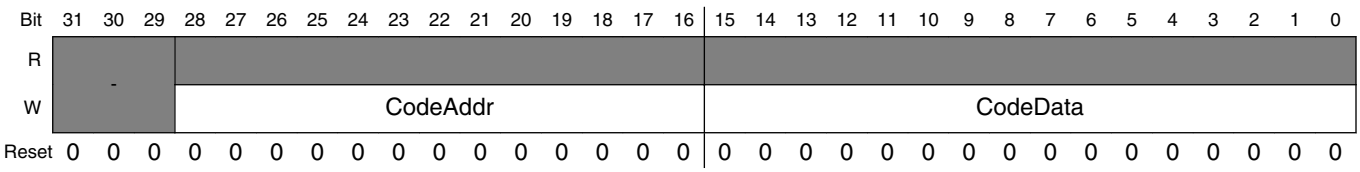
VPU_CodeRun field descriptions

Field	Description
31–1 -	Reserved
0 VPU.CodeRun	VPU_CodeRun. BIT processor run start bit. 0 BIT Processor stop execution. 1 BIT Processor start execution.

78.5.2 BIT Boot Code Download Data register (VPU_CodeDown)

See the figure below for illustration of valid bits in VPU BIT Boot Code Download Data Register and the following table for description of the bit fields in the register.

Address: VPU_CodeDown is 63FF_4000h base + 4h offset = 63FF_4004h



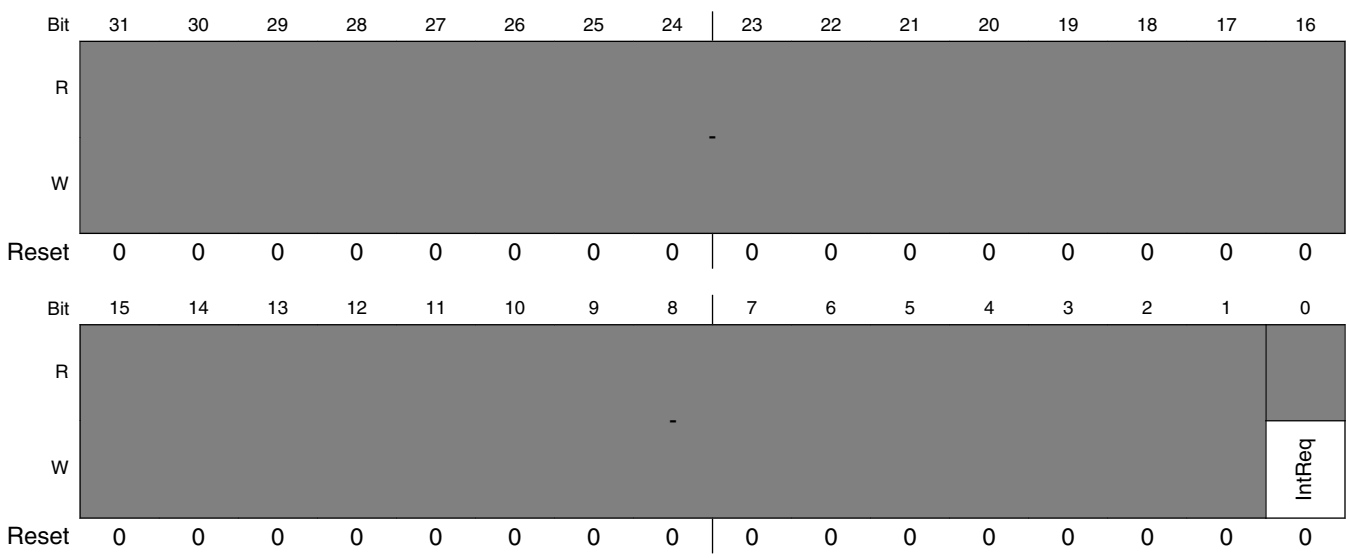
VPU_CodeDown field descriptions

Field	Description
31–29 -	Reserved
28–16 CodeAddr	CodeAddr[12:0] Download address of VPU BIT boot code, which is VPU internal address of BIT processor.
15–0 CodeData	CodeData[15:0] Download data of VPU BIT boot code.

78.5.3 Host Interrupt Request to BIT (VPU_HostIntReq)

See the figure below for illustration of valid bits in VPU Host Interrupt Request Register and the following table for description of the bit fields in the register.

Address: VPU_HostIntReq is 63FF_4000h base + 8h offset = 63FF_4008h



VPU_HostIntReq field descriptions

Field	Description
31–1 -	Reserved

Table continues on the next page...

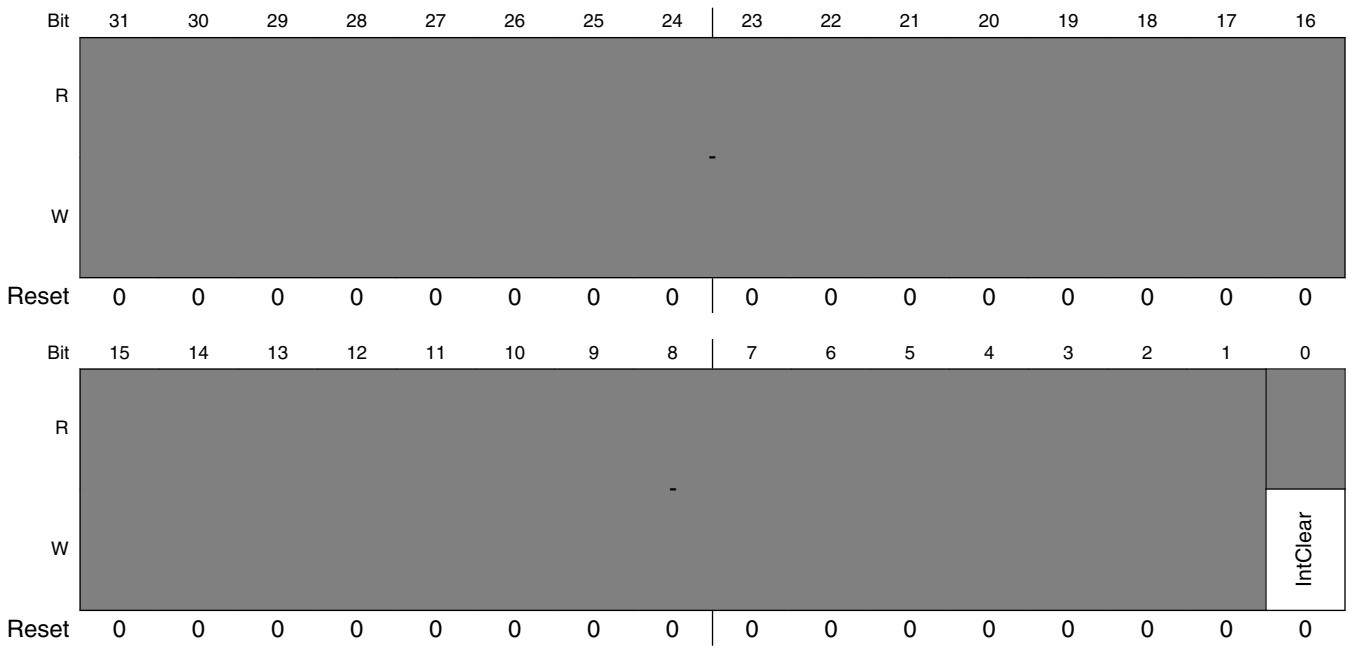
VPU_HostIntReq field descriptions (continued)

Field	Description
0 IntReq	IntReq. The host interrupt request bit. 0 No host interrupt is requested. 1 The host processor request interrupt to the BIT processor.

78.5.4 BIT Interrupt Clear (VPU_BitIntClear)

See the figure below for illustration of valid bits in VPU BIT Interrupt Clear Register and the following table for description of the bit fields in the register.

Address: VPU_BitIntClear is 63FF_4000h base + Ch offset = 63FF_400Ch



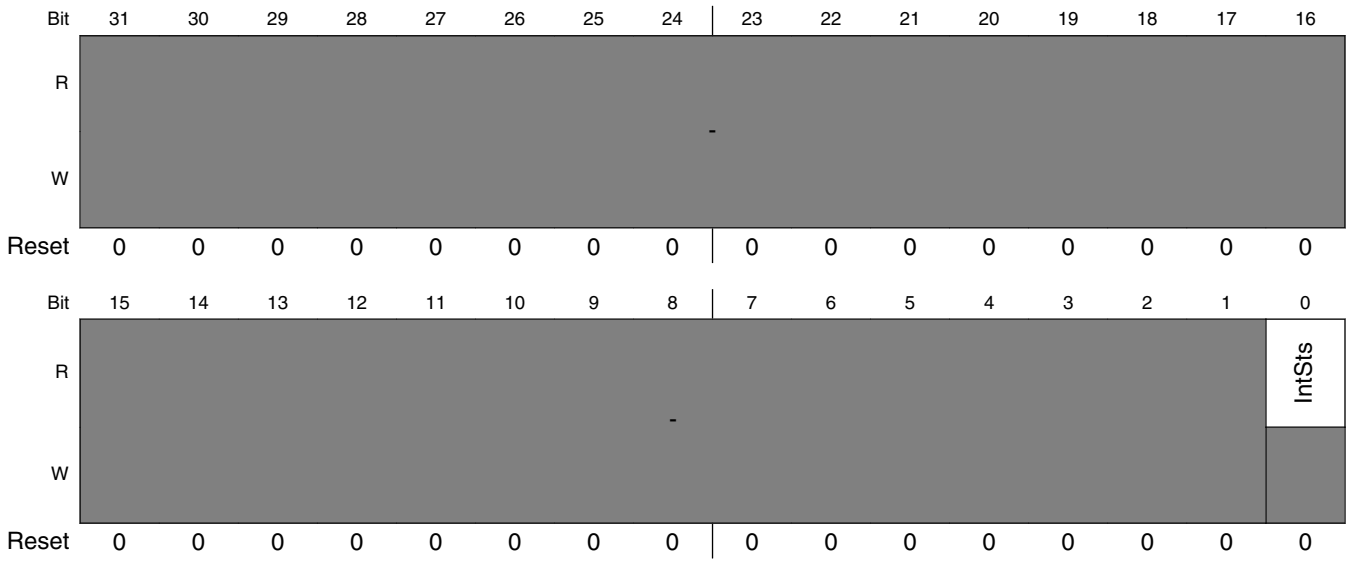
VPU_BitIntClear field descriptions

Field	Description
31–1 -	Reserved
0 IntClear	IntClear. BIT interrupt clear bit. 0 No operation is issued. 1 Clear the BIT interrupt to the host.

78.5.5 BIT Interrupt Status (VPU_BitIntSts)

See the figure below for illustration of valid bits in VPU BIT Interrupt Status Register and the following table for description of the bit fields in the register.

Address: VPU_BitIntSts is 63FF_4000h base + 10h offset = 63FF_4010h



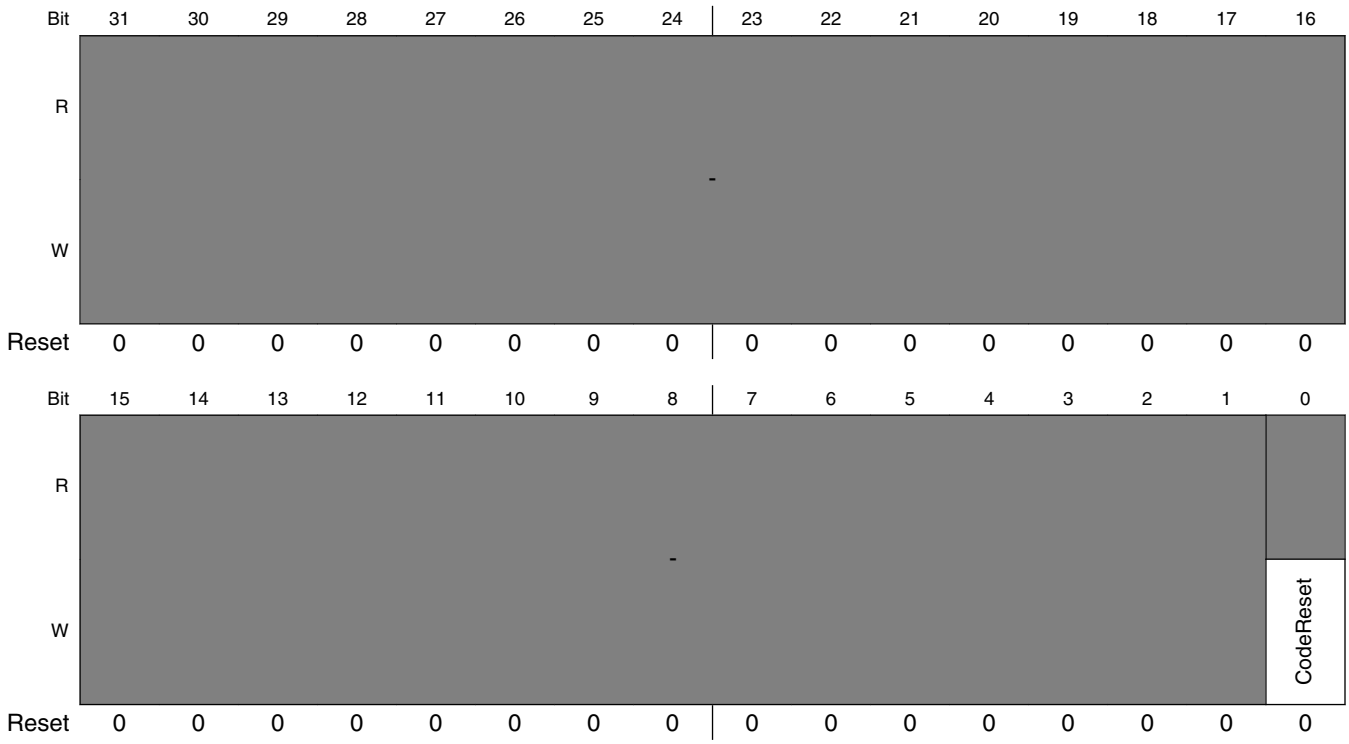
VPU_BitIntSts field descriptions

Field	Description
31–1 -	Reserved
0 IntSts	IntSts. BIT interrupt status bit. 0 No BIT interrupt is asserted. 1 The BIT interrupt is asserted to the host. It is cleared when the host processor write "1" to VPU_BitIntClear register.

78.5.6 BIT Code Reset (VPU_BitCodeReset)

See the figure below for illustration of valid bits in VPU BIT Code Reset Register and the following table for description of the bit fields in the register.

Address: VPU_BitCodeReset is 63FF_4000h base + 14h offset = 63FF_4014h



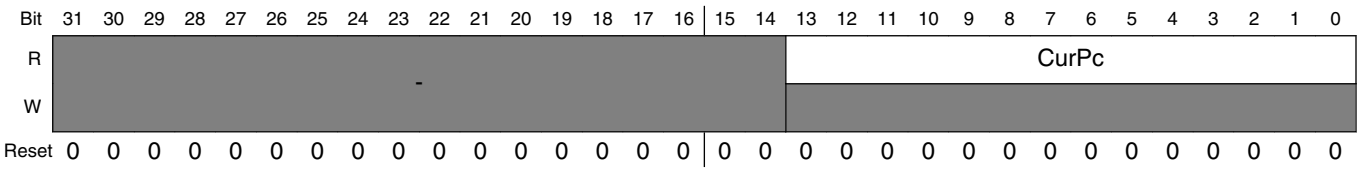
VPU_BitCodeReset field descriptions

Field	Description
31–1 -	Reserved
0 CodeReset	CodeReset. BIT code reset bit. 0 No operation is issued. 1 The program counter of BIT processor is set to "0", BIT processor restart at initial routine.

78.5.7 BIT Current PC (VPU_BitCurPc)

See the figure below for illustration of valid bits in VPU BIT Current PC Register and the following table for description of the bit fields in the register.

Address: VPU_BitCurPc is 63FF_4000h base + 18h offset = 63FF_4018h



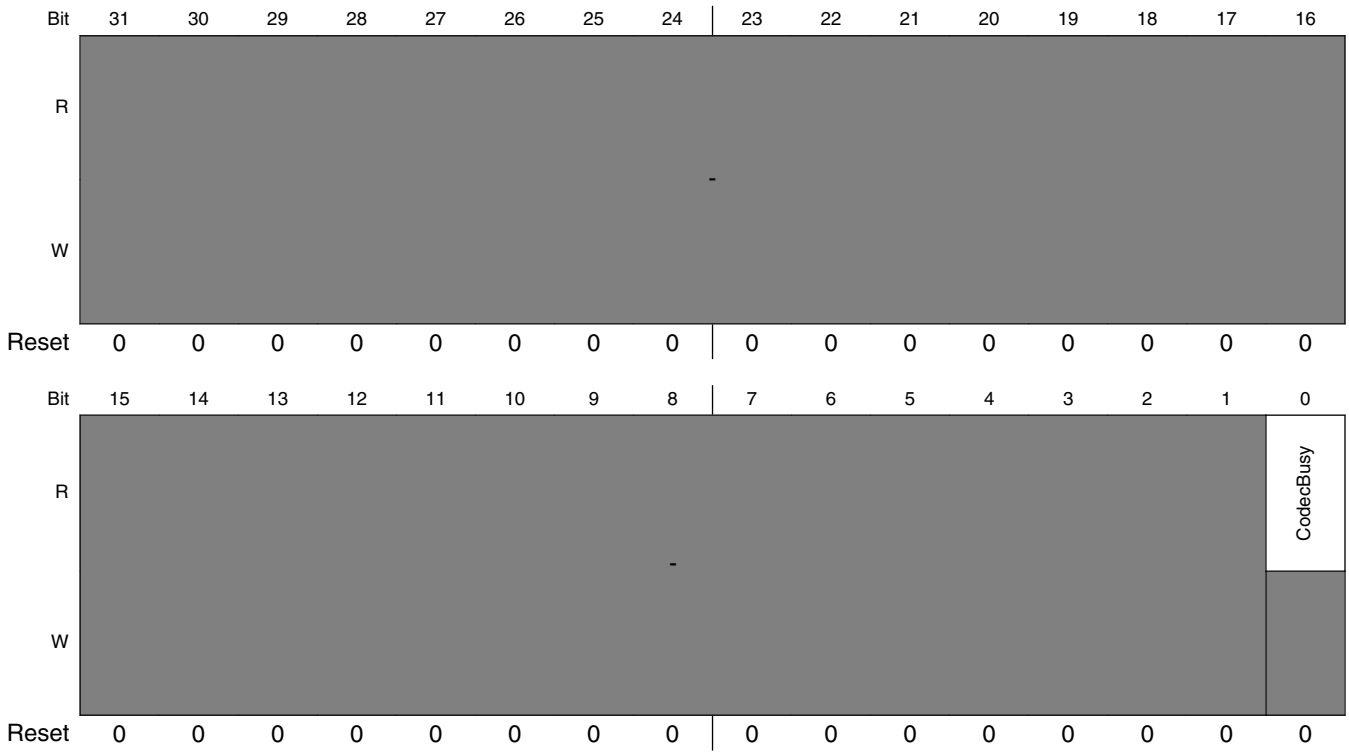
VPU_BitCurPc field descriptions

Field	Description
31–14 -	Reserved
13–0 CurPc	CurPc[13:0]. BIT current PC value. Returns the current program counter of BIT processor by reading this register.

78.5.8 BIT CODEC Busy (VPU_BitCodecBusy)

See the figure below for illustration of valid bits in VPU BIT Codec Busy Register and the following table for description of the bit fields in the register.

Address: VPU_BitCodecBusy is 63FF_4000h base + 20h offset = 63FF_4020h



VPU_BitCodecBusy field descriptions

Field	Description
31–1 -	Reserved
0 CodecBusy	Codec busy flag for Bit processor.BIT processor write "1"to this register when the processor is running."0"means processor is waiting for a command.This value is connected to the o_vpu_idle.

Chapter 79
Watchdog Timer (WDOG-1)

79.1 Introduction

This chapter describes a block integrated into an SoC. The chapter is intended for a block driver software developer. It describes block-level operation and programming. To understand how the block is integrated at the SoC level, a system software developer should see discussions of the block in the appropriate SoC-level chapter(s).

79.2 Overview

This section briefly introduces the block. The full description of the block is in Functional Description.

This section includes a top level diagram that shows the functional organization of the block, including all off-chip signals. The figure below is the block diagram.

The Watchdog Timer (WDOG-1) protects against system failures by providing a method of escaping from unexpected events or programming errors. Once the WDOG-1 is activated, it must be serviced by the software on a periodic basis. If servicing does not take place, the timer times out. Upon a time-out, the WDOG-1 asserts the internal system reset signal, $\overline{\text{wdog\_rst}}$ which will be going to System Reset Controller . There is also a provision for WDOG-1 signal assertion by time-out counter expiration.There is an option of programmable interrupt generation before the counter actually times out. The time at which the interrupt needs to be generated prior to counter time-out is programmable. There is a power down counter which gets enabled out of any reset (POR, Warm / Cold). This counter has a fixed time out period of 16 seconds upon which it will assert the WDOG-1signal. Flow diagrams for the time-out counter, power down counter and interrupt operations are shown in Figure 79-6, Figure 79-7 and Figure 79-8.

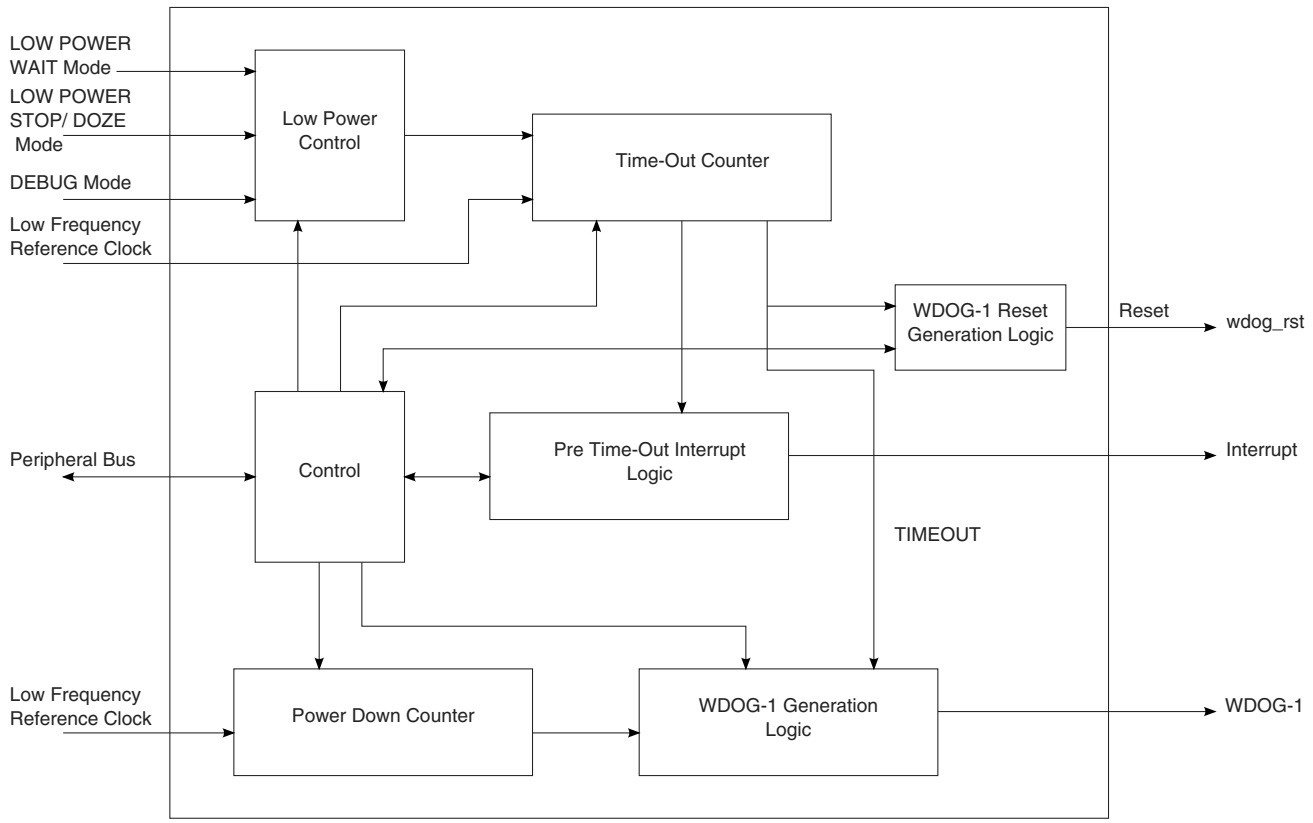


Figure 79-1. WDOG-1 Diagram

79.2.1 Features

The WDOG-1 features are as follows:

- A Configurable time-out counter with Time-out periods from 0.5 seconds up to 128 seconds and after time-out expiration results in assertion of $\overline{\text{wdog\_rst}}$ Reset signal .
- Time resolution of 0.5 seconds.
- Configurable time-out counter that can be programmed to run or stop during low-power modes.
- Configurable time-out counter that can be programmed to run or stop during DEBUG mode.
- Programmable Interrupt generation prior to time-out.
- The time duration between interrupt and time-out events can be programmed from 0 to 127.5 seconds in steps of 0.5 seconds.
- A power down counter with fixed time-out period of 16 seconds which if not disabled after reset will assert $\overline{\text{WDOG-1}}$ signal low.
  - Power down counter will be enabled out of any reset (POR, Warm / Cold reset) by default.

79.2.2 Modes and Operations

The WDOG-1 supports the following modes described in the indicated sections:

- Low Power Modes
- Debug Mode

As described in Operations, the WDOG supports the operations described in the indicated sections:

- Watchdog Reset Generation
- WDOG_B Generation

79.3 External Signals

Table 79-1. Off-Chip Block Signals

Signal	I/O	Description	Reset State[1]	Pull-Up/Down[1]
WDOG-1	O	This signal will powerdown the Chip. Refer to WDOG_B Generation.	1	-
wdog_rst	O	This signal is a reset source for the chip. Refer to Watchdog Reset Generation.	1	-

1. The reset state values and pull-up/down requirements provided in this table are from the block-level perspective. To understand how the block is integrated at the SoC level, the system software developer must see discussions of the block in the appropriate SoC-level chapter(s). For example, a block signal that requires a pull-up could be integrated with a pull-up option built into the SoC. In this case, the system software developer must ensure the proper programming at the SoC level.

79.4 Functional Description

79.4.1 Time-Out Event

The WDOG-1 provides time-out periods from 0.5 seconds up to 128 seconds with a time resolution of 0.5 seconds.

The user can determine the time-out period by writing to the WDOG-1 Time-out field (WT[7:0]) in the Watchdog Control Register (WDOG_WCR). The WDOG-1 has to be enabled by setting the WDE bit of Watchdog Control Register (WDOG_WCR) for the time-out counter to start running. After the WDOG-1 is enabled, the counter is activated, loads the time-out value and begins to count down from this programmed value. The

timer will time-out when the counter reaches zero and the WDOG-1 outputs a system reset signal, wdog_rst and asserts WDOG-1 (WDT bit should be set in Watchdog Control Register (WDOG_WCR)).

However, the Time-out condition can be prevented by reloading the counter with the new time-out value (WT[7:0] of WDOG-1.WCR) if a service routine is performed before the counter reaches zero. If any system errors occur that prevent the software from servicing the Watchdog Service Register (WDOG_WSR), then the time-out condition occurs. By performing the service routine, the WDOG-1 reloads its counter to the time-out value indicated by bits WT[7:0] of the Watchdog Control Register (WDOG_WCR) and it restarts the countdown.

A system reset will reset the counter and place it in the idle state at any time during the countdown. The counter flow diagram is shown in Figure 79-6.

NOTE

The time-out value is reloaded to the counter at the time when WDOG-1 is enabled or after the service routine has been performed.

79.4.1.1 Servicing WDOG-1 To Reload The Counter

The proper service sequence to reload a time-out value to the counter begins by writing 0x 5555 followed by 0x AAAA to the Watchdog Service Register (WDOG_WSR). Any number of instructions can be executed between the two writes. If the WDOG-1.WSR is not loaded with 0x 5555 prior to writing 0x AAAA to the WDOG-1.WSR, the counter is not reloaded. If any value other than 0x AAAA is written to the WDOG-1.WSR after 0x 5555, the counter is not reloaded. This service sequence will reload the counter with the time-out value WT[7:0] of Watchdog Control Register (WDOG_WCR). The time-out value can be changed at any point of the time and the counter reloads this value whenever the core services the Watchdog.

79.4.2 Interrupt Event

Prior to time-out the WDOG-1 can generate an interrupt which can be considered a warning signal to indicate that the time-out will occur shortly.

The time duration between interrupt event and time-out event can be controlled by writing to the WICT field of Watchdog Interrupt Control Register (WDOG_WICR). It can vary between 0 and 127.5 seconds. If the WDOG-1 is serviced before the interrupt generation then the counter will be reloaded with the time-out value WT[7:0] of Watchdog Control Register (WDOG_WCR) and interrupt will not be t

79.4.3 Power-Down Counter Event

The Power down Counter inside WDOG-1 will be enabled out of reset. This counter has a fixed time-out value of 16 seconds, after which it will drive the $\overline{\text{WDOG-1}}$ signal low.

To prevent this, the software must disable this counter by clearing the PDE bit of Watchdog Miscellaneous Control Register (WDOG_WMCR) within 16 seconds of reset de-assertion. Once disabled, this counter cannot be enabled again until the next system reset occurs. This feature is provided to prevent the hanging up of cores after reset, as WDOG-1 is not enabled out of reset.

79.4.4 Low Power Modes

79.4.4.1 STOP and DOZE Mode

If the WDOG-1 Timer Disable bit for Low power STOP and DOZE mode (WDZST) bit in the Watchdog Control Register (WDOG_WCR), is '0", the WDOG-1 Timer continues to operate using the Low Frequency Reference clock. If Low power Enable (WDZST) bit is set to "1", then the WDOG-1 Timer operation will be suspended in Low power STOP or DOZE mode. Upon exiting Low power STOP or DOZE mode, the WDOG-1 operation returns to what it was prior to entering the STOP or DOZE mode.

79.4.4.2 WAIT Mode

If the WDOG-1 Timer Disable bit for low power WAIT mode (WDW) bit in the Watchdog Control Register (WDOG_WCR), is '0", the WDOG-1 Timer continues to operate using the Low Frequency Reference clock . If low power WAIT Enable (WDW) bit is set to "1", then the WDOG-1 Timer operation will be suspended. Upon exiting low power WAIT mode, the WDOG-1 operation returns to what it was prior to entering the WAIT mode.

NOTE

WDOG-1 Timer will not be able to detect events which happen for periods less than one Low Frequency Reference clock cycle. For example in repeated WAIT mode entry/exit, if the RUN mode time is less than one Low Frequency Reference clock cycle and if WDW bit is set then WDOG-1 Timer may never time out even though the system is in RUN mode for a finite

duration because WDOG-1 may not see any Low Frequency Reference clock edge during its wake time.

79.4.5 Debug Mode

The WDOG-1 Timer can be configured for continual operation, or the operation can be suspended during debug mode. If the WDOG-1 debug enable (WDBG) bit is set to "1" in the Watchdog Control Register (WDOG_WCR), the WDOG-1 Timer operation is suspended in debug mode. In-case of WDBG bit being set to "1" and the Debug mode is entered, WDOG-1 Timer operation is suspended after 2 Low Frequency Reference clocks and similarly WDOG-1 Timer operation is continued after 2 Low frequency reference clocks of Debug mode Exit. Register read and write accesses in Debug mode continue to function normally. Also, while in DEBUG mode, the WDE bit of Watchdog Control Register (WDOG_WCR) can be enabled-disabled directly. If the WDOG-1 debug enable (WDBG) bit is cleared then Watchdog Timer operation is not suspended. Power-down counter is not affected by debug mode entry/exit.

NOTE

If the WDE bit of Watchdog Control Register (WDOG_WCR) is set/cleared while in DEBUG mode, it remains set/cleared even after exiting DEBUG mode.

79.4.6 Operations

This section describes the block's operations.

79.4.6.1 Watchdog Reset Generation

The WDOG-1 generated reset signal $\overline{\text{wdog\_rst}}$ is asserted by the following operations:

- A software write to the Software Reset Signal (SRS) bit of the Watchdog Control Register (WDOG_WCR).
- WDOG-1 time-out.

The $\overline{\text{wdog\_rst}}$ will be asserted for one clock cycle of Low frequency Reference clock for both the time-out Condition and software write occurrence. It remains asserted for 1 clock cycle of Low frequency Reference clock even if a system reset is asserted in between. Figure 79-3 shows the timing diagram of this signal due to time-out condition.

79.4.6.2 $\overline{\text{WDOG\_B}}$ Generation

The WDOG-1 asserts $\overline{\text{WDOG}}$ under the following scenarios:

- Software write to WDA bit of Watchdog Control Register (WDOG_WCR). $\overline{\text{WDOG}}$ Signal remains asserted as long as the WDA bit is "0".
- WDOG-1 time-out condition, WDT bit of Watchdog Control Register (WDOG_WCR) must be set for this scenario. $\overline{\text{WDOG-1}}$ Signal remains asserted until a Power-on Reset (POR) occurs. It gets cleared after the POR occurs and not due to any other system reset. Figure 79-4 shows the timing diagram of $\overline{\text{WDOG-1}}$ due to time-out condition.
- WDOG-1 Power down Counter time-out, PDE bit of Watchdog Miscellaneous Control Register (WDOG_WMCR) should not be cleared for this scenario. $\overline{\text{WDOG-1}}$ Signal remains asserted for one clock cycle of Low frequency Reference clock.

Figure 79-2 shows the scenarios under which $\overline{\text{WDOG-1}}$ gets asserted.

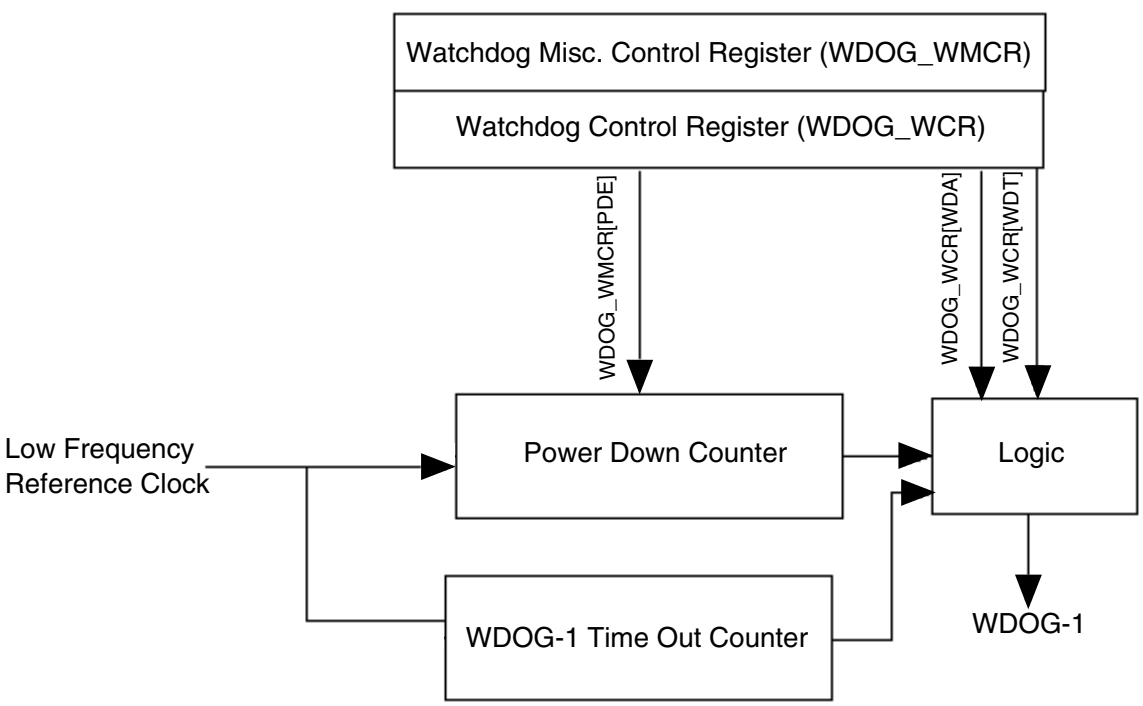


Figure 79-2. $\overline{\text{WDOG-1}}$ Generation

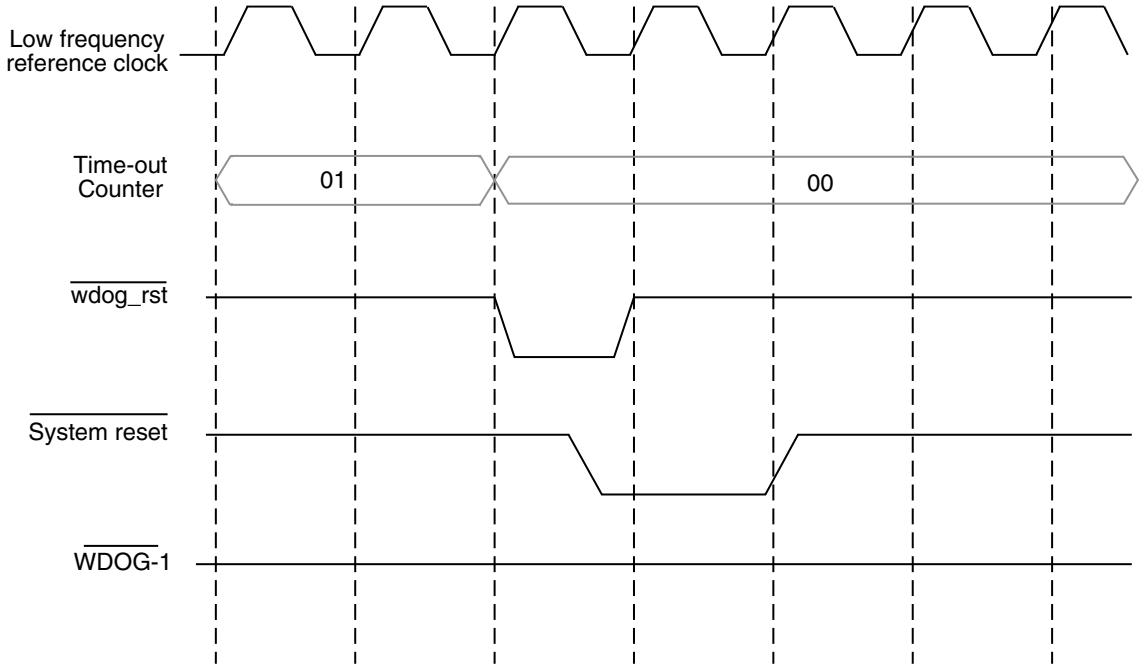


Figure 79-3. WDOG-1 Time-Out Condition/WDT Bit Is Not Set

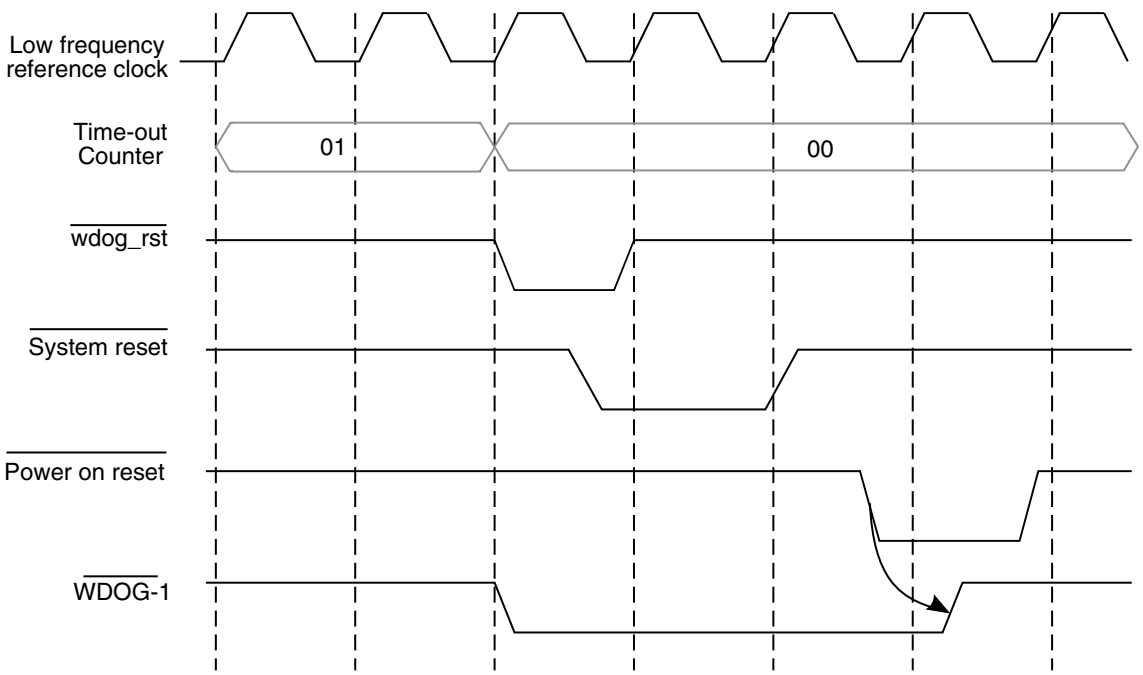


Figure 79-4. WDOG-1 Time-Out Condition/WDT Bit Is Set

79.4.7 Clocks

This section describes clocks and special clocking requirements of the block.

The WDOG-1 uses the Low frequency Reference clock for its counter and control operations. Peripheral Bus clock is used for register Read/Write operations.

Low frequency Reference clock is a free running clock and cannot be gated. Peripheral Bus clock cannot be gated and is selectively switched ON whenever Read/Write operations takes place.

79.4.8 Reset

This section describes how to reset the block and explains special requirements related to reset.

The block is reset by a system reset and will have the following consequences: WDOG-1 counter is disabled.

Power-Down counter is enabled and starts counting.

79.4.9 Interrupt

The WDOG-1 has the feature of Interrupt generation before time-out.

The interrupt will be generated only if the WIE bit in Watchdog Interrupt Control Register (WDOG_WICR) is set. The exact time at which the interrupt should occur prior to time-out depends on the value of WICT field of Watchdog Interrupt Control Register (WDOG_WICR). As an example if the WICT field has a value 0x04, then the interrupt will be generated 2 seconds prior to time-out. Once the interrupt is triggered the WTIS bit in Watchdog Interrupt Control Register (WDOG_WICR) will be set. The software need to clear this bit to de-assert the Interrupt. If the WDOG-1 is serviced before the interrupt generation then the counter will be reloaded with the time-out value WT[7:0] of Watchdog Control Register (WDOG_WCR) and interrupt would not be triggered.

79.4.10 Flow Diagrams

A flow diagram of watchdog operation is shown in Figure 79-6, Figure 79-7 and Figure 79-8.

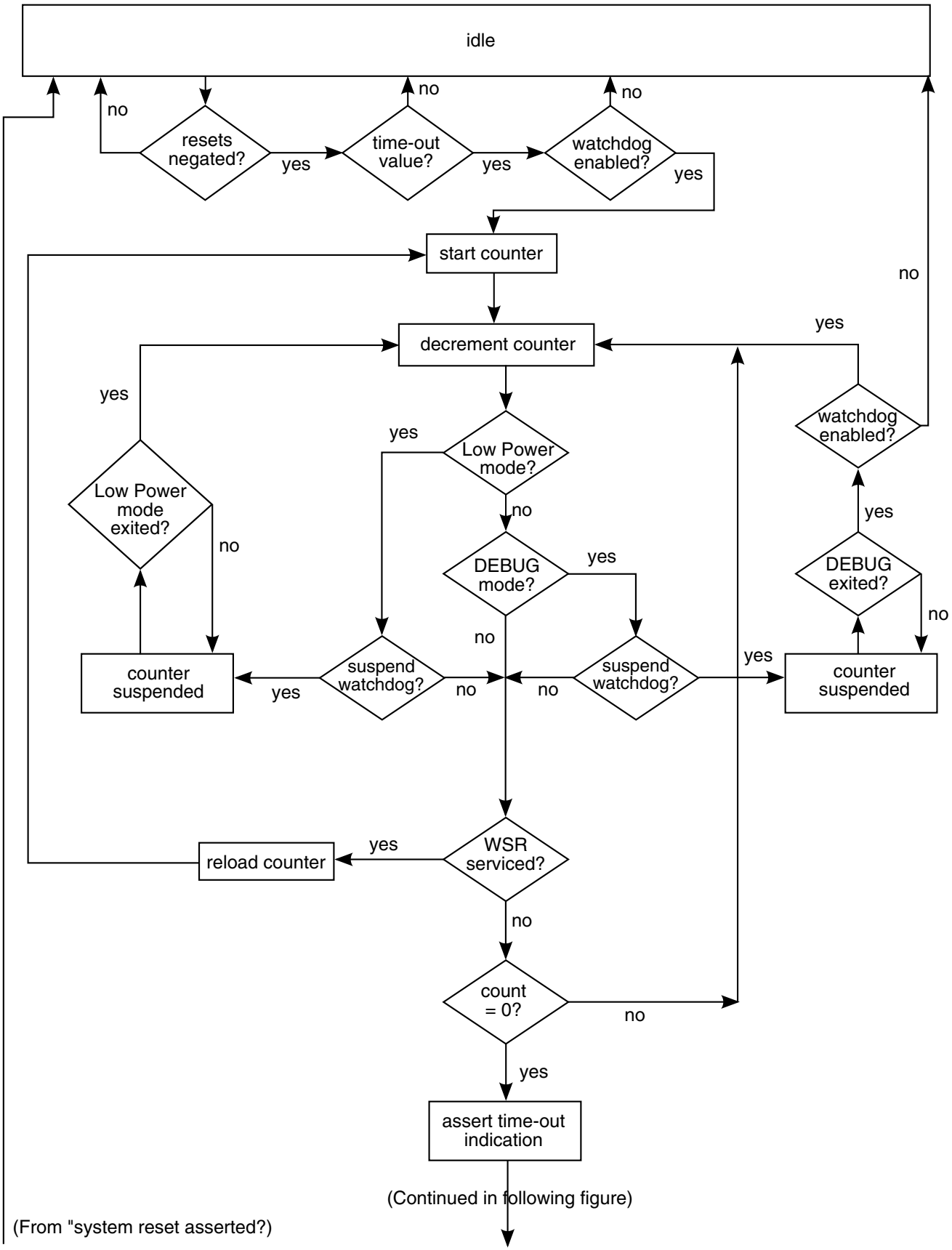
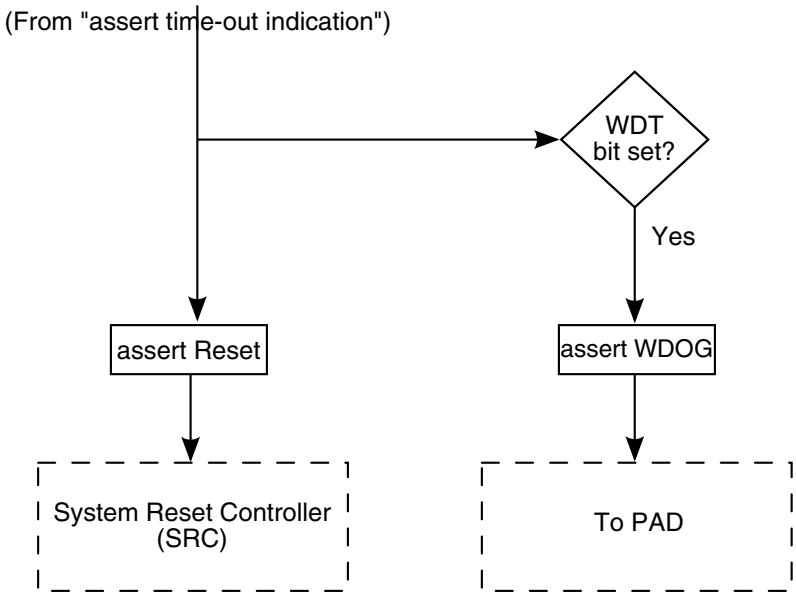


Figure 79-5. Time-Out Counter Flow Diagram



Note: A system reset will force the state machine to "idle" at any time during countdown.

Figure 79-6. Time-Out Counter Flow Diagram

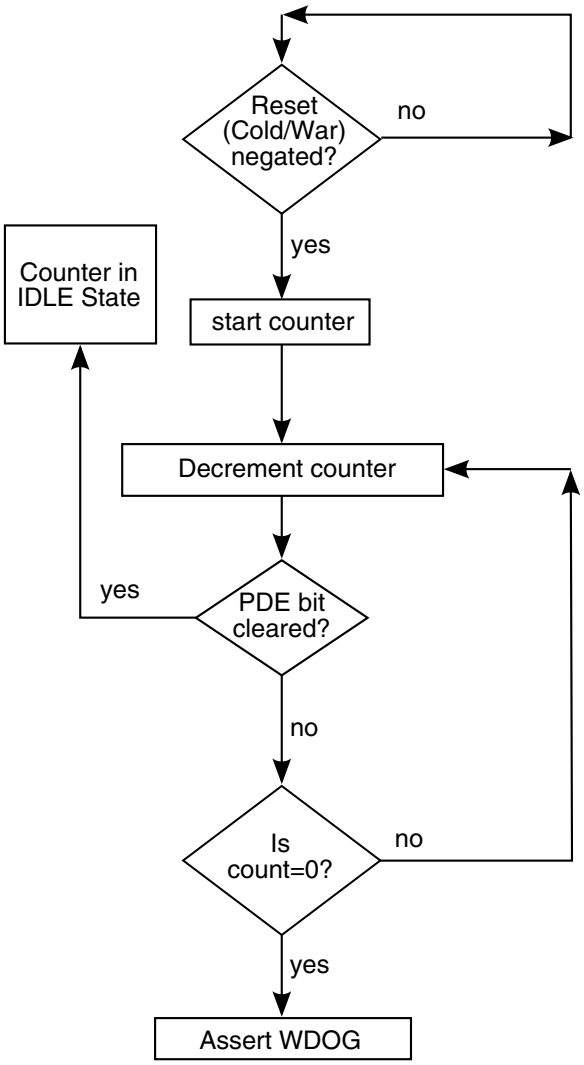


Figure 79-7. Power-Down Counter Flow Diagram

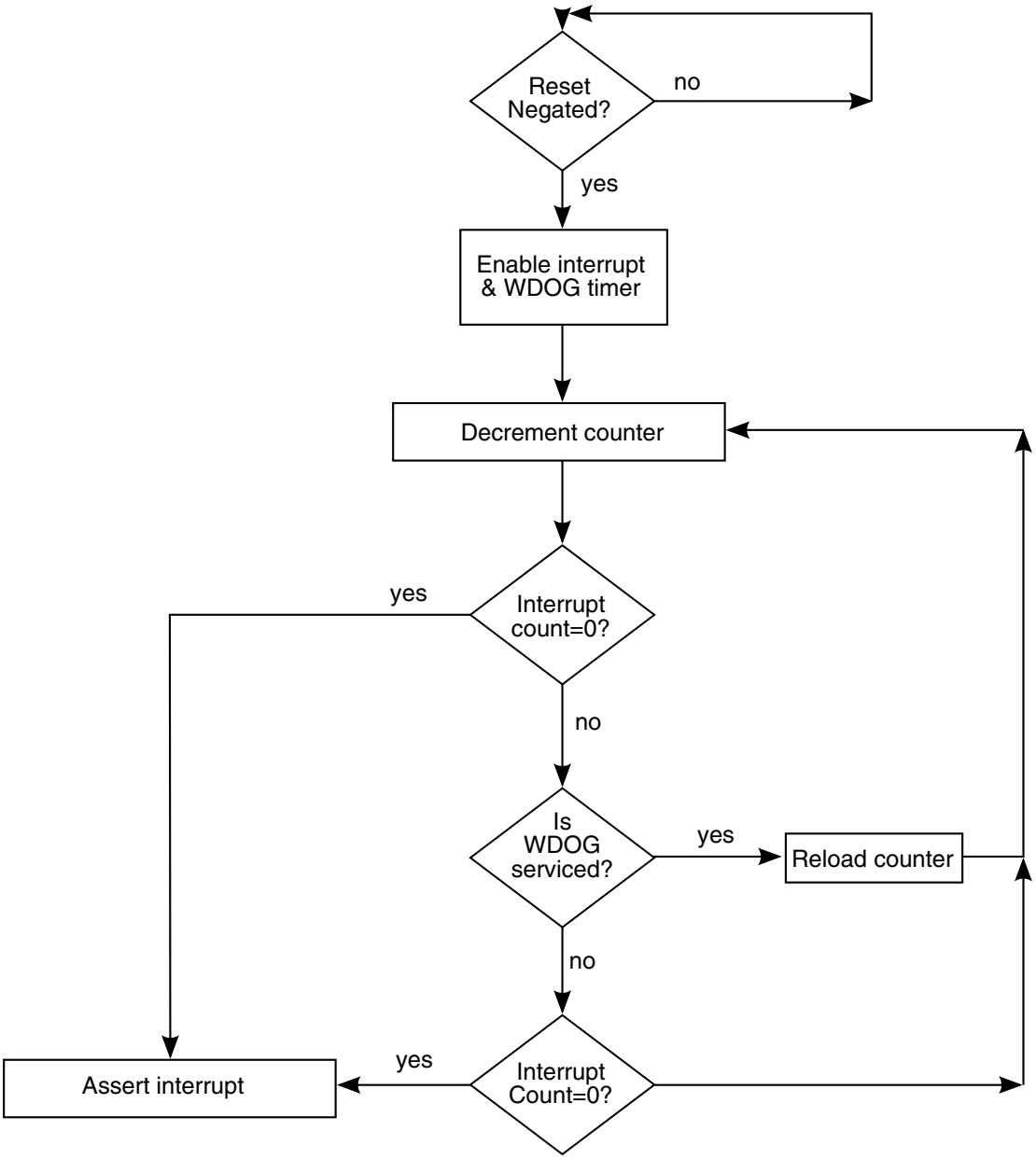


Figure 79-8. Interrupt Generation Flow Diagram

79.5 Initialization

The following sequence of programming should be performed for WDOG-1 initialization:

- PDE bit of Watchdog Miscellaneous Control Register (WDOG_WMCR) should be cleared for disabling the power down counter.

- WDT field of Watchdog Control Register (WDOG_WCR) should be programmed for sufficient time-out value.
- WDOG-1 should be enabled by setting WDE bit of Watchdog Control Register (WDOG_WCR) so that the Time-out counter loads the WDT field value of Watchdog Control Register (WDOG_WCR) and starts counting.

79.6 Programmable Registers

The WDOG-1 has user-accessible, 16-bit registers used to configure, operate, and monitor the state of the Watchdog Timer. Byte operations can be performed on these registers. If a 32-bit access is performed on these registers then the WDOG-1 will not generate any Peripheral Bus error and will behave normally, like a 16-Bit access, making Read/Write possible. A 32-Bit access should be avoided, as the system might go to an unknown state.

WDOG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
53F9_8000	Watchdog Control Register (WDOG_WCR)	16	R/W	0030h	79.6.1/ 5014
53F9_8002	Watchdog Service Register (WDOG_WSR)	16	R/W	0000h	79.6.2/ 5016
53F9_8004	Watchdog Reset Status Register (WDOG_WRSR)	16	R	0000h	79.6.3/ 5017
53F9_8006	Watchdog Interrupt Control Register (WDOG_WICR)	16	R/W	0004h	79.6.4/ 5018
53F9_8008	Watchdog Miscellaneous Control Register (WDOG_WMCR)	16	R/W	0001h	79.6.5/ 5019

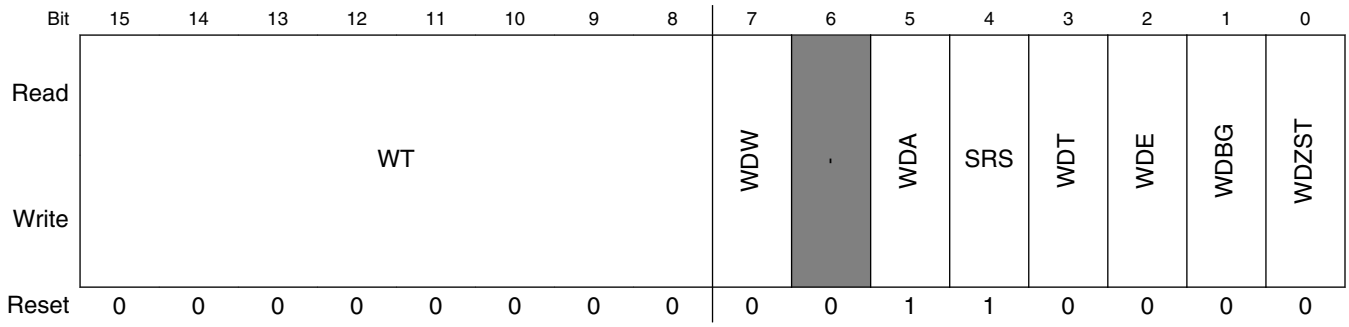
79.6.1 Watchdog Control Register (WDOG_WCR)

The Watchdog Control Register (WDOG_WCR) controls the WDOG-1 operation.

- WDZST, WDBG and WDW bits are write-once only bits. Once the software does a write access to these bits, all these bits will get locked and cannot be reprogrammed till the next system reset assertion.

- WDE is a write one once only bit. Once software performs a write "1" operation to this bit it cannot be reset/cleared till the next system reset.
- WDT is also a write one once only bit. Once software performs a write "1" operation to this bit it cannot be reset/cleared till the next POR (Power-on Reset). This bit does not gets reset/cleared due to any system reset.

Address: WDOG_WCR is 53F9_8000h base + 0h offset = 53F9_8000h



WDOG_WCR field descriptions

Field	Description
15–8 WT	Watchdog Time-out Field. This 8-bit field contains the time-out value that is loaded into the Watchdog counter after the service routine has been performed or after the Watchdog is enabled. After reset, WT[7:0] must have a value written to it before enabling the Watchdog otherwise count value of zero which is 0.5 seconds is loaded into the counter. **NOTE:** The time-out value can be written at any point of time but it is loaded to the counter at the time when WDOG-1 is enabled or after the service routine has been performed. For more information see Time-Out Event . 0x00 - 0.5 Seconds (Default). 0x01 - 1.0 Seconds. 0x02 - 1.5 Seconds. 0x03 - 2.0 Seconds. 0xff - 128 Seconds.
7 WDW	Watchdog Disable for Wait. This bit determines the operation of WDOG-1 during Low Power WAIT mode. This is a write once only bit. 0 Continue WDOG-1 timer operation (Default). 1 Suspend WDOG-1 timer operation.
6 -	Reserved adopt Reserved
5 WDA	$\overline{\text{WDOG}}$ assertion. Controls the software assertion of the $\overline{\text{WDOG}}$ signal. 0 Assert $\overline{\text{WDOG}}$ output. 1 No effect on system (Default).
4 SRS	Software Reset Signal. Controls the software assertion of the WDOG-generated reset signal $\overline{\text{wdog\_rst}}$. This bit automatically resets to "1" after it has been asserted to "0". **NOTE:** This bit does not generate the software reset to the block.

Table continues on the next page...

WDOG_WCR field descriptions (continued)

Field	Description
	0 Assert system reset signal. 1 No effect on the system (Default).
3 WDT	$\overline{\text{WDOG}}$ Time-out assertion. Determines if the $\overline{\text{WDOG}}$ gets asserted upon a Watchdog Time-out Event. This is a write-one once only bit. **NOTE:** There is no effect on $\overline{\text{wdog\_rst}}$ (WDOG Reset) upon writing on this bit. $\overline{\text{WDOG}}$ gets asserted along with $\overline{\text{wdog\_rst}}$ if this bit is set. 0 No effect on $\overline{\text{WDOG}}$ (Default). 1 Assert $\overline{\text{WDOG}}$ upon a Watchdog Time-out event.
2 WDE	Watchdog Enable. Enables or disables the WDOG-1 block. This is a write one once only bit. It is not possible to clear this bit by a software write, once the bit is set. **NOTE:** This bit can be set/reset in debug mode (exception). 0 Disable the Watchdog (Default). 1 Enable the Watchdog.
1 WDBG	Watchdog DEBUG Enable. Determines the operation of the WDOG-1 during DEBUG mode. This bit is write once-only. 0 Continue WDOG-1 timer operation (Default). 1 Suspend the watchdog timer.
0 WDZST	Watchdog Low Power. Determines the operation of the WDOG-1 during low-power modes. This bit is write once-only. **NOTE:** The WDOG-1 can continue/suspend the timer operation in the low-power modes (STOP and DOZE mode). 0 Continue timer operation (Default). 1 Suspend the watchdog timer.

79.6.2 Watchdog Service Register (WDOG_WSR)

When enabled, the WDOG-1 requires that a service sequence be written to the Watchdog Service Register (WSR) to prevent the time-out condition.

NOTE

Executing the service sequence will reload the WDOG-1 time out counter.

Address: WDOG_WSR is 53F9_8000h base + 2h offset = 53F9_8002h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	WSR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDOG_WSR field descriptions

Field	Description
15–0 WSR	Watchdog Service Register. This 16-bit field contains the Watchdog service sequence. Both writes must occur in the order listed prior to the time-out, but any number of instructions can be executed between the two writes. The service sequence must be performed as follows: 0x5555 Write to the Watchdog Service Register (WDOG_WSR). 0xAAAA Write to the Watchdog Service Register (WDOG_WSR).

79.6.3 Watchdog Reset Status Register (WDOG_WRSR)

The WRSR is a read-only register that records the source of the output reset assertion. It is not cleared by a hard reset. Therefore, only one bit in the WRSR will always be asserted high. The register will always indicate the source of the last reset generated due to WDOG-1. Read access to this register is with one wait state. Any write performed on this register will generate a Peripheral Bus Error .

A reset can be generated by the following sources, as listed in priority from highest to lowest:

- Watchdog Time-out
- Software Reset

Address: WDOG_WRSR is 53F9_8000h base + 4h offset = 53F9_8004h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0											POR	0		TOUT	SFTW
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDOG_WRSR field descriptions

Field	Description
15–5 Reserved	This read-only field is reserved and always has the value zero. Reserved.
4 POR	Power On Reset. Indicates whether the reset is the result of a power on reset. 0 Reset is not the result of a power on reset. 1 Reset is the result of a power on reset.
3–2 Reserved	This read-only field is reserved and always has the value zero. Reserved.
1 TOUT	Time-out. Indicates whether the reset is the result of a WDOG-1 time-out. 0 Reset is not the result of a WDOG-1 time-out. 1 Reset is the result of a WDOG-1 time-out.

Table continues on the next page...

WDOG_WRSR field descriptions (continued)

Field	Description
0 SFTW	Software Reset. Indicates whether the reset is the result of a WDOG-1 software reset by asserting SRS bit 0 Reset is not the result of a software reset. 1 Reset is the result of a software reset.

79.6.4 Watchdog Interrupt Control Register (WDOG_WICR)

The WDOG_WICR controls the WDOG-1 interrupt generation.

Address: WDOG_WICR is 53F9_8000h base + 6h offset = 53F9_8006h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	WIE	WTIS	0						WICT							
Write		w1c														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

WDOG_WICR field descriptions

Field	Description
15 WIE	Watchdog Timer Interrupt enable bit. Reset value is 0. **NOTE:** This bit is a write once only bit. Once the software does a write access to this bit, it will get locked and cannot be reprogrammed till the next system reset assertion 0 Disable Interrupt (Default). 1 Enable Interrupt.
14 WTIS	Watchdog TImer Interrupt Status bit will reflect the timer interrupt status, whether interrupt has occurred or not.Once the interrupt has been triggered software must clear this bit by writing 1 to it. 0 No interrupt has occurred (Default). 1 Interrupt has occurred
13–8 Reserved	This read-only field is reserved and always has the value zero. Reserved.
7–0 WICT	Watchdog Interrupt Count Time-out (WICT) field determines, how long before the counter time-out must the interrupt occur. The reset value is 0x04 implies interrupt will occur 2 seconds before time-out. The maximum value that can be programmed to WICT field is 127.5 seconds with a resolution of 0.5 seconds. **NOTE:** This field is write once only. Once the software does a write access to this field, it will get locked and cannot be reprogrammed till the next system reset assertion. 0x00 WICT[7:0] = Time duration between interrupt and time-out is 0 seconds. 0x01 WICT[7:0] = Time duration between interrupt and time-out is 0.5 seconds. 0x04 WICT[7:0] = Time duration between interrupt and time-out is 2 seconds (Default). 0xff WICT[7:0] = Time duration between interrupt and time-out is 127.5 seconds.

79.6.5 Watchdog Miscellaneous Control Register (WDOG_WMCR)

WDOG_WMCR Controls the Power Down counter operation.

Address: WDOG_WMCR is 53F9_8000h base + 8h offset = 53F9_8008h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0															PDE
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

WDOG_WMCR field descriptions

Field	Description
15–1 Reserved	This read-only field is reserved and always has the value zero. Reserved.
0 PDE	Power Down Enable bit. Reset value of this bit is 1, which means the power down counter inside the WDOG-1 is enabled after reset. The software must write 0 to this bit to disable the counter within 16 seconds of reset de-assertion. Once disabled this counter cannot be enabled again. See Power-Down Counter Event for operation of this counter. **NOTE:** This bit is write-one once only bit. Once software sets this bit it cannot be reset till the next system reset. 0 Power Down Counter of WDOG-1 is disabled. 1 Power Down Counter of WDOG-1 is enabled (Default).

Chapter 80
Crystal Oscillator 24 MHz (XTALOSC)

80.1 Introduction

The crystal oscillator contains an Automatic Level Controller (ALC), which detects the amplitude of the oscillation and adjusts the amount of current sourced to the crystal accordingly.

NOTE

As the oscillator is initially turned on there is no oscillation. The oscillator will source its maximum current of a few milliamps (mA) to kick-start the crystal.

NOTE

Once the oscillation starts and the amplitude of oscillation is detected and monitored by the ALC circuitry, the oscillator slowly cuts back its current to a stable value of a few hundreds microamps (uA).

80.1.1 Interface Specification

Most signals in the XTALOSC interface are either analog in nature or clock signals with no specific timing.

They are represented in the figure and table below.

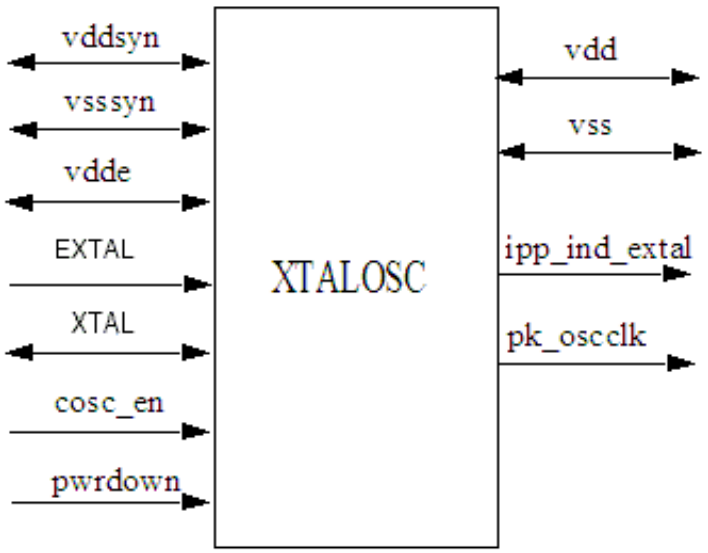


Figure 80-1. XTALOSC Interface

Table 80-1. Block I/O Signals

Signal	I/O	Description
XTALI	I/O	Crystal inout or 24MHz clock source input
XTALO	O	XTAL out pin to crystal

NOTE

If wish to use a single ended source, drive input on XTALI and float XTALO. (XTALI refers to the input of the integrated amplifier while XTALO is the output). Please refer to part data sheet, for max voltage limitation on XTALI.

80.1.2 Crystal Operating Frequency

This oscillator uses crystals with fundamental frequency 24MHz.

80.1.3 Power Supply

This oscillator is powered by a 2.5V power supply +/-10% tolerance. The core supply is 1.2V +/-10% tolerance.

80.1.4 Operating Temperature

For safety application, the temperature range is from -20C to 125C. The same temperature range applies when choosing a crystal for the oscillator.

80.1.5 Input Clock in Bypass Mode

Input clock voltage range in bypass mode is from 1.8V to 2.5V (50% duty cycle). It is recommended that the input clock voltage be the same as Vdde for proper output clock duty cycle.

80.1.6 Input Control Signal

To prevent voltage mismatch problem, input control voltage range can be high (2.5V) or low voltage (1.2V for cmos065lp)

80.1.7 Output Clock

The output clocks that are the same as the 2.5V power supply are pk_oscclk. The output clocks that are the same as the core logic voltage are ipp_ind_extal. All output clocks, either 2.5V or 1.2V, are required to have duty cycle in the range of 45% to 55% for 50% clock inputs

80.2 External Signals

The following table describes all XTALOSC (24MHz) I/Os.

Table 80-2. Block I/O Signals

Signal	I/O	Description
XTALI	I/O	Crystal input or 24 MHz clock source input
XTALO	O	XTAL out pin to crystal

80.3 Operation Modes

80.3.1 Crystal Osc Mode

In this mode, both extal and xtal are connected to a crystal and "cosc_en" is asserted with "pwrdown" deasserted. Two output clocks are available: "pk_oscclk" at 2.5V for 2.5V PLL and ipp_ind_extal at 1.2V for 1.2V PLL.

80.3.2 Bypass Mode

In this mode, the crystal oscillator amplifier is off with "cosc_en" deasserted. Only "extal" can be driven by an external clock source.

Any clock at voltage level of 1.2V to 2.5V is level-shifted to 2.5 V for 2.5 V PLL or 1.2 V for 1.2 V logic. The level shifters are actually comparators.

80.3.3 Low Power Mode

The "pwrdown" signal turns most of the circuitry in the osc, including the bias circuit.

80.3.4 Oscillator Safety Margin Requirements

Oscillator safety margin requirements are normally 5 or 10 times the maximum crystal ESR, depending on the crystal vendor. NDK requires 5x and KDS requires 10x.

The oscillator safety margin depends heavily on the capacitive loads on "extal" and "xtal" pins, so care must be taken when doing simulation to include parasitic loads on-chip as well as parasitic caps on boards.

Chapter 81
Crystal Oscillator 32K (XTALOSC32K)

81.1 Introduction

The Crystal Oscillator 32K (XTALOSC32K) is an integrated building block, that, when coupled with pads and an external crystal and load capacitors, can implement a crystal oscillator.

The cell features a power multiplexer allowing it to switch between a NiMH battery source and a nominal 1.2 V source. The multiplexer is controlled by the vdd1p2good signal from the Low-dropout regulator (LDO).

81.2 Features

Table 81-1. Features

	Min	Typ	Max	Comments
Fosc		32.768KHz		This frequency is nominal and determined mainly by the crystal selected. 32.0K would work as well.
Supply Voltage	1.0V	1.2V	1.4V	The target battery is a 1.2V NiMH button cell.
Temperature range	- 40C		120C	This is the simulated junction temperature. Battery specifications may practically limit this to a smaller range when used in an oscillator.

Table continues on the next page...

Table 81-1. Features (continued)

	Min	Typ	Max	Comments
Current consumption		1.5uA		This is the consumption of the oscillator alone (osc32k). Total supply consumption will depend on what the digtal portion of the RTC consumes. Target is for an additional 2uA. TBD
Bias resistor		14Mohm		This the integrated bias resistor that sets the amplifier into a high gain state. Any leakage through the ESD network, external board leakage, or even a scope probe that is significant relative to this value will debias the amp. The debiasing will result.
Crystal Properties				
Cload		10pF		Usually crystals can be purchased tuned for different Cloads. A higher Cload with decrease oscillation margin, but increases current oscillating through the crystal. See graph below.
ESR		50KHz		Equivalent series resistance of the crystal. Choosing a crystal with a higher value will decrease the oscillating margin. See graph below.

81.3 External Signals

Table below describes all XTAL32K Block I/Os.

Table 81-2. Block I/O Signals

Signal	I/O	Description
RTC_XTALI	I/O	Crystal inout or 32KHz clock source input
RTC_XTALO	O	XTAL out pin to crystal

NOTE

If wish to use a single ended source, drive input on RTC_XTALI and float RTC_XTALO . (RTC_XTALI refers to the input of the integrated amplifier while RTC_XTALO is the output). Please refer to part data sheet, for max voltage limitation on RTC_XTALI.

81.4 Memory Map/Register Definition

No programming model exists.

81.5 Functional Description

The following figure shows the Simulated Oscillator Gain Margin Vs Input Leakage Resistor referred to in Table 81-1.

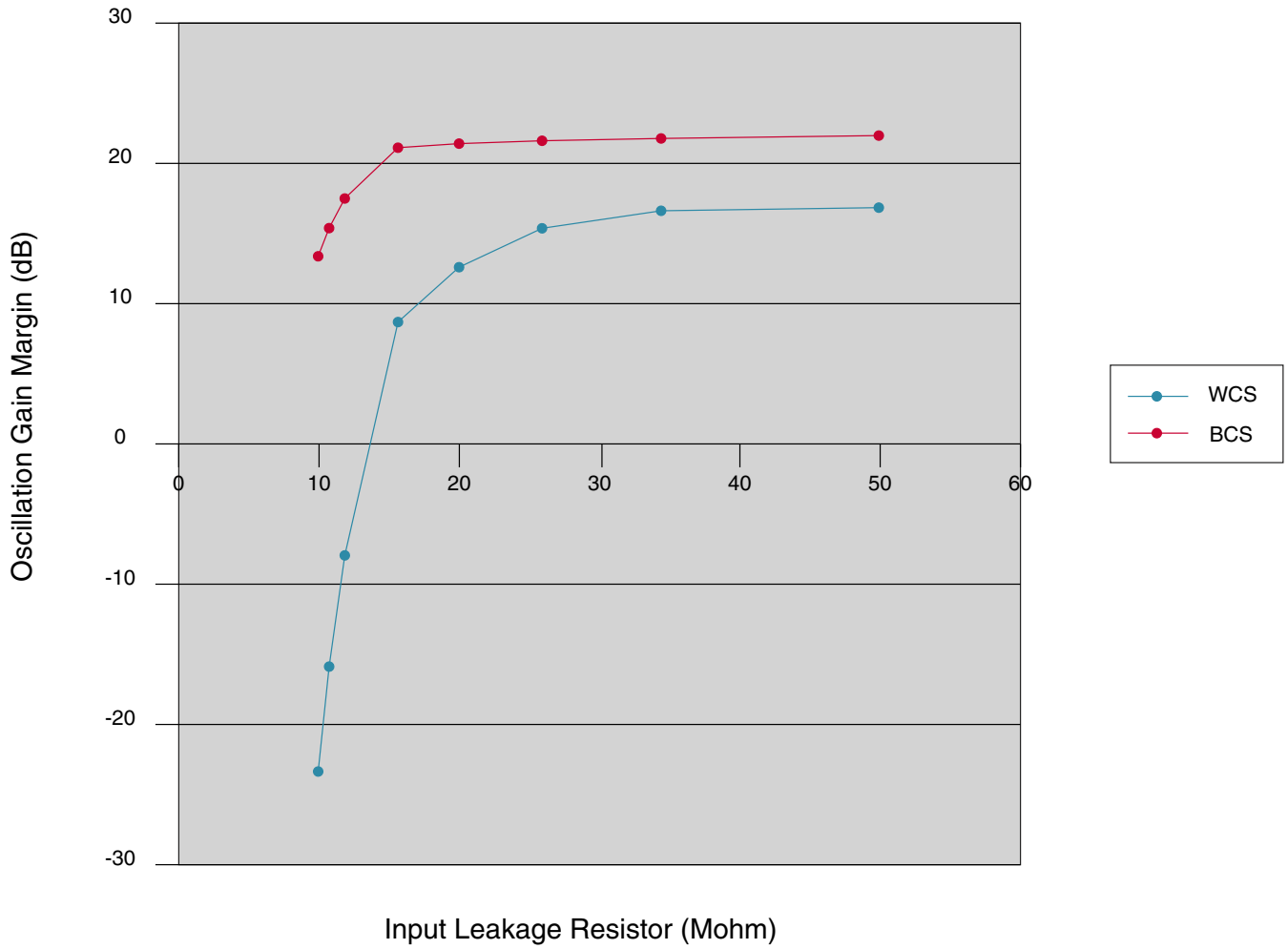


Figure 81-1. Simulated Oscillator Gain Margin Vs Input Leakage Resistor

The following figure shows the Simulated Oscillator Gain Margin Vs Crystal ESR referred to in Table 81-1.

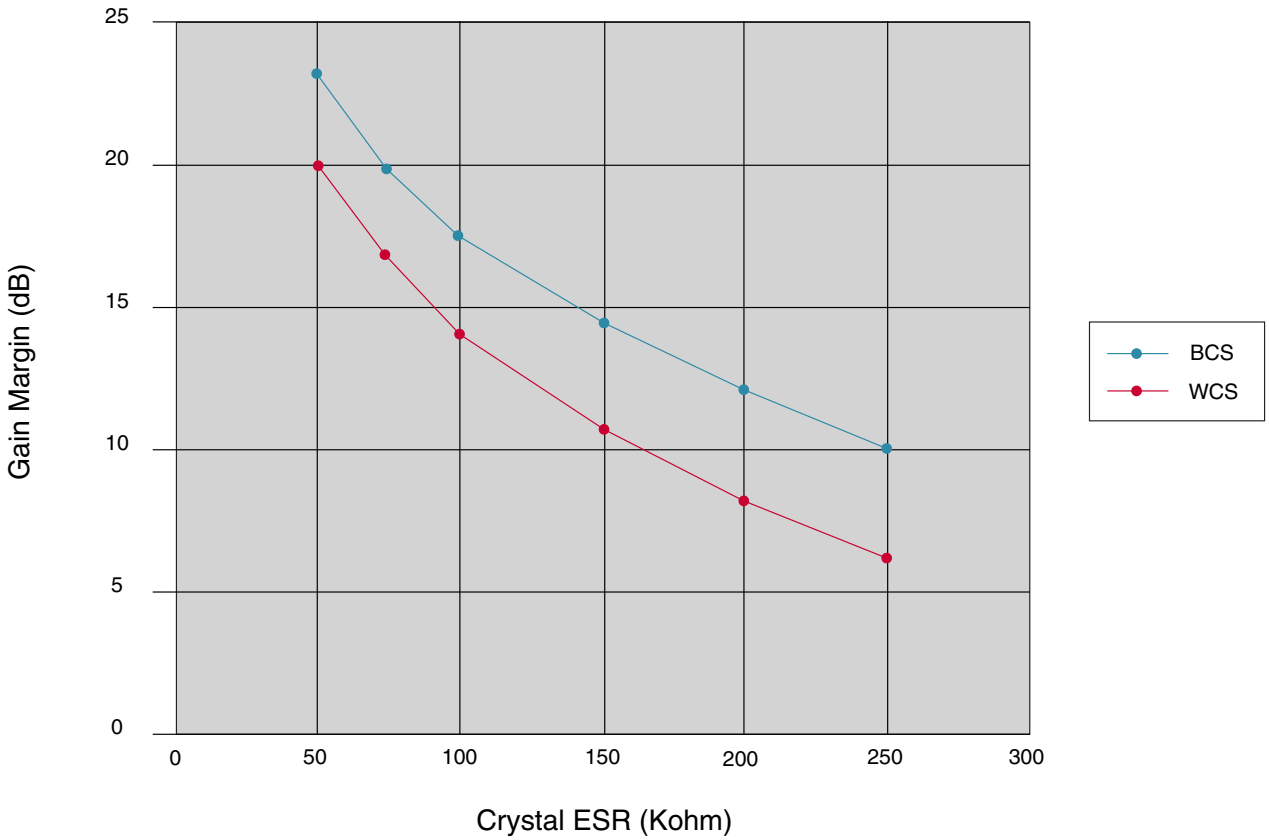


Figure 81-2. Simulated Oscillator Gain Margin Vs Crystal ESR

Appendix A
SDMA Scripts

A.1 Introduction

This appendix provides descriptions of scripts that may be used to perform data transfers using the Smart DMA (SDMA) block of the SoC. The SDMA block supports data transfer from core memory space to core memory, from core memory space to peripherals, and vice versa.

A.2 SDMA Scripts Overview

Table A-1 gives an overview of the SDMA scripts.

Table A-1. SDMA Scripts Overview

Data Transfer	Script to Use	Use Case	Parameters Through Context	Parameters Through Buffer Descriptor
Memory	ap_2_ap	Memory Copy	None	Word length (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter Fist address is memory source address, second address is memory destination address
FIRI	mcu_2_firi	Firi Tx	FIRI Tx FIFO address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter Memory source address, Second address not used
	firi_2_mcu	Firi Rx	FIRI Tx FIFO address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size(2'b01 or 2'b10 or 2'b11 or 2'b00), Counter Memory source address, Second address not used
Shared UART	uartsh_2_mcu	Uart Rx	UART Rx FIFO address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01), Counter Mem destination address, Second address not used

Table continues on the next page...

Table A-1. SDMA Scripts Overview (continued)

Data Transfer	Script to Use	Use Case	Parameters Through Context	Parameters Through Buffer Descriptor
UART	uart_2_mcu	Uart Rx	Uart Rx FIFO address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01), Counter Memory source address, Second address not used
SPDIF	mcu_2_spdif	Spdif Playback	SPDIF Tx FIFO address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b10 or 2'b00), Counter Memory source address, Second address not used
	spdif_2_mcu	Spdif Record	SPDIF Rx FIFO address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b10 or 2'b00), Counter Memory source address, Second address not used
SSI	mcu_2_app, mcu_2_shp	Audio Playback	SSI Tx FIFO address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter Memory source address Second address not used
	app_2_mcu, shp_2_mcu	Audio Record	SSI Rx FIFO address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory source address Second address not used
	mcu_2_ssiapp, mcu_2_ssish	Audio Playback	SSI Tx FIFO 0 address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter Memory source address, Second address not used
	ssiapp_2_mcu, ssish_2_mcu	Audio Record	SSI Rx FIFO 0 address (r6) Event_mask (r1) Event2_mask (r0) Watermark level (r7)	Periph size (2'b01 or 2'b10 or 2'b11 or 2'b00), Counter, Memory source address Second address not used
Peripheral	p_2_p	Peripheral Transfer	Destination address (r6) Event_mask (r1) Event2_mask (r0) Source address (r2) Information (r7)	Counter, First address and Second address not used

In Table A-1, the data transfer column lists the possible types of DMA channels that involve the SDMA and a specific script is attached to each channel. It must be noted that some scripts cover several DMA channels. The scripts deal with the channels that have generic peripherals where only the watermark level is needed (no aging timer, no end_of_packet feature, and so on). The location refers to whether a script resides in ROM or in RAM and the size of the script is given in the instructions. The parameters columns are explained in the next section.

The following terms are used in Table A-1 which lists all the supported data transfers.

EMI or External Memory:

The external memories are connected to the External Memory Interface. Thus, a data transfer to EMI means a data transfer to any of the external memories (NAND Flash, NOR Flash, PSRAM, SDRAM, and so on).

Host mem or ARM internal memory:

The memory space accessible through the MAX of the ARM platform. It does not correspond to the peripherals, although they are accessible through the MAX as well. A data transfer from Host mem means data is read from the internal memory of the ARM or from the external memory. It is possible to point to an external memory through the MAX because it is also connected to EMI module. The next diagram replaces the SDMA and its DMA port in the hardware architecture. The Host mem is accessible through the Per DMA port of the SDMA, the EMI is accessible through the Burst DMA and the DSP mem through the DSP port.

Shared Peripheral:

A same peripheral can be connected to the shared peripheral bus (output of SPBA module) and to the ARM platform. This is the case for some UART, CSPI, and SSI. Therefore, "shared UART" indicates that the UART is connected to the shared peripheral, whereas "UART" means the UART is connected to the ARM platform.

Figure A-1 gives an overview of the SDMA in its hardware environment.

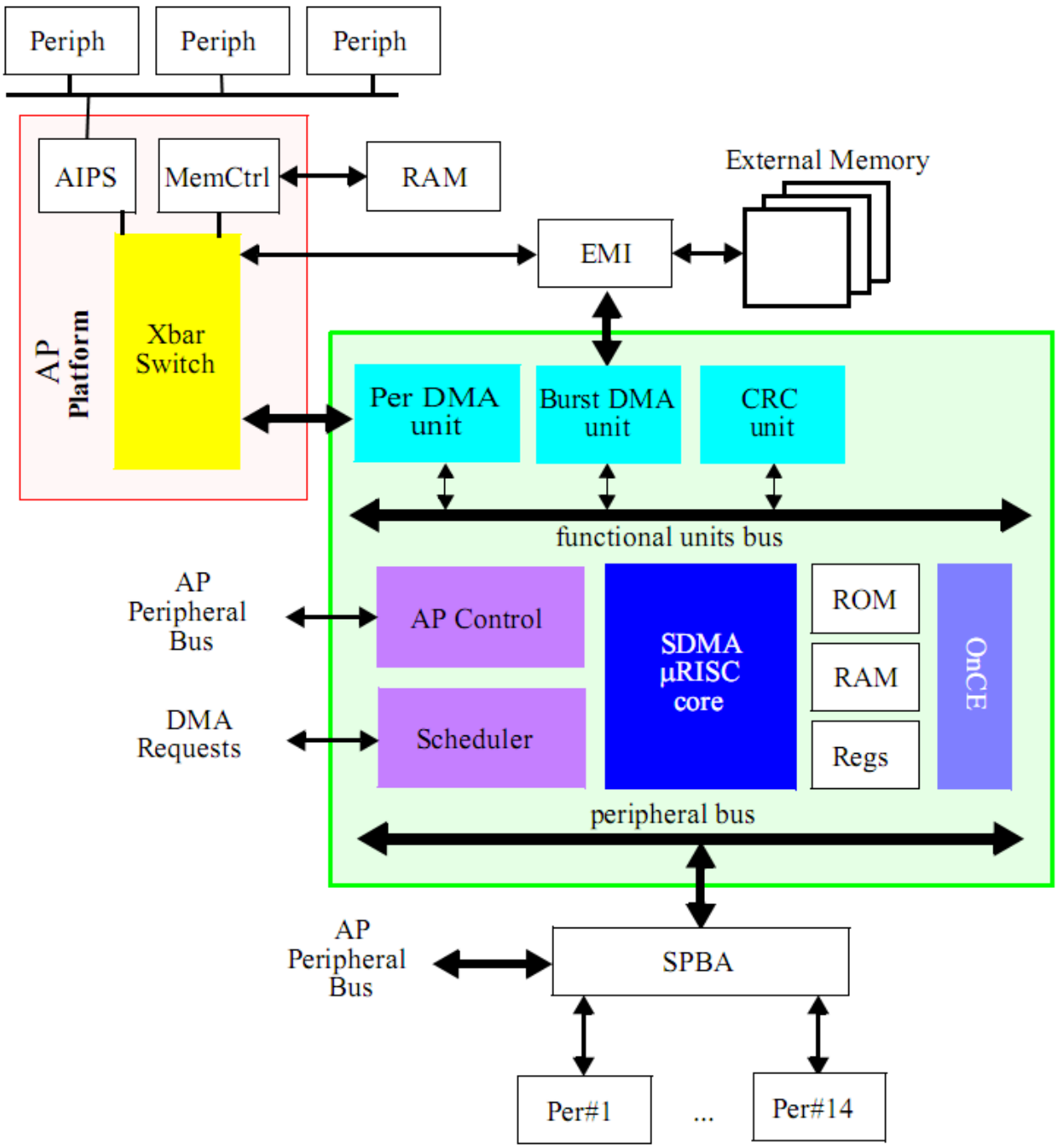


Figure A-1. SDMA Connections

A.3 Scripts Parameters: Definition

A.3.1 Parameters Definition

A.3.1.1 Parameters Required by Data Transfer Script Involving a Peripheral

The scripts that have a peripheral as a player in the transfer of data need to have the following input parameters:

A.3.1.1.1 Watermark Level (WML)

It is the upper or the lower threshold of the receive or transmit FIFO of the peripheral that triggers a DMA request to the SDMA. The WML determines the data transfer loop size, meaning the number of bytes that will be read/write from/to the receive/transmit FIFO. This parameter must be a multiple of the peripheral FIFO data size. As an example, if UART1 is the data transfer destination, data must be written to the Tx FIFO of the peripheral. For instance, the watermark level can be set to 8 bytes (the UART is a 1 byte FIFO data size), meaning the UART_TX DMA request will be sent to the SDMA each time there are more than 8 free locations in the FIFO; therefore SDMA loop size will be set to 8 and 8 bytes will be written to the UART_TX FIFO. In fact, the loop size is the minimum between the WML and the count field of the buffer descriptor attached to the channel. This concept is explained later on in the chapter. Similarly, if the UART is the source of a data transfer and if the WML is set to 16, each time the DMA request is received by the SDMA, which triggers the associated channel, the data transfer loop size is set to 16. This means 16 bytes will be read from the UART_RX FIFO. The watermark level is a "long" (32-bit data) programmed by the ARM and is not supposed to evaluate a lot during application. The WML is always given in bytes.

NOTE

For the transmit FIFO, the WML = FIFO_SIZE - FIFOTX_THRESHOLD For the receive FIFO, the WML = FIFORX_THRESHOLD The FIFOTX_THRESHOLD/ FIFORX_THRESHOLD are the values of the peripheral transmit and receive FIFO level at which a DMA request is triggered. For instance, if TX FIFO is 32 bytes depth and if the DMA request is sent when the number of bytes present in the FIFO is below a threshold of 10, the WML will be 22. It means that when the SDMA will receive the DMA request from the peripheral, it will be possible to store 22 bytes in the TX FIFO. For the receive FIFO, the WML equals the threshold.

A.3.1.1.2 Event_mask and Event2_mask

As previously said, when the WML threshold is over or under passed, a DMA request is sent to the scheduler part of the SDMA. The SDMA scheduler has 48 input pins, called "events", which are connected to the different DMA request. The mapping between the DMA request name and the event number is SoC dependent and defined in the "Interrupts and SDMA Events" chapter. For instance, in SoC A, the UART_RX DMA request is connected to events pin 5; whereas for SoC B, it may be connected to events pin 15. The script reads the EVENTS(32 events requests) and EVENTS2(remaining 16 events requests) register to check if the received DMA request is compliant with the expected one and it guaranteed that a second DMA request sent during the data transfer is not lost.

For instance, while SDMA is reading the UART_RX FIFO (again, the number of data to be read is given by WML value), the peripheral may receive new data from its input ports. Therefore after the first data transfer, the number of data available in the UART received FIFO can be still higher than the WML threshold, so a second DMA request may be sent.

The event_mask value is generally a 1-bit high value, which means that if the script attached to the channel must be triggered by DMA request number I, event_mask[I] must be set to 1, if the script attached to the channel must be triggered by DMA request number 32+I, event2_mask[I] must be set to 1. Some scripts (like ata_2_mcu) may be triggered by several DMA requests; in that case several bits of the event_mask must be asserted. The event_mask and event2_mask are long (32-bit) data.

A.3.1.1.3 Request sources

The SDMA has 48 sources of hardware requests coming from various peripherals either integrated inside the SoC, or connected outside the SoC (external DMA requests). Each request, or event, is able to trigger a single or several channels. A channel transfer can be also simply triggered in software if the script does not use the even_mas/event_mask2 variables or if the user wants to force a transfer.

Typically, the scripts (like uart_2_mcu) working with peripherals FIFO use the event_mask/event_mask2 variables to perform a data burst upon FIFO requests until count is reached. Whereas other scripts (like ap_2_ap) are just triggered once through an event (hardware or software request), and do not stop until count is reached.

A.3.1.1.4 Peripheral Address

The scripts of the SDMA scripts library are SoC independent but as their goal is to address peripheral, they required the base address or the FIFO address of the peripheral. Passing FIFO or peripheral base address depends on the scripts.

A.3.1.1.5 Data Length

The scripts whose name contains the key word "app" or "shp" are generic: they are used to transfer data from/to a 8, 16, 24 or 32-bit peripheral data size. Some jumps to some routines of these scripts depend on this peripheral size. This parameter is required as input of these scripts. This parameter is passed through the command field of the buffer descriptor and is coded on bits 25 and 24 of the mode:

00: 32-bit data transfer.

01: 8-bit data transfer.

10: 16-bit datatransfer.

11: 24-bit data transfer.

A.4 SDMA Scripts on i.MX53

The following scripts are all based on buffer descriptor mechanism.

A.4.1 SDMA ROM Scripts

The ROM holds the generic memory to memory scripts (ap_2_ap, ap_2_bp, bp_2_ap, bp_2_bp) and the peripheral most useful scripts (app_2_mcu, mcu_2_app, uart_2_mcu, uartsh_2_mcu, mcu_2_shp, shp_2_mcu).

In these memory to memory scripts, the number of bytes to transfer from the source memory to the destination is divided by 4 to get the number of 32-bit words that can be transferred. Most of the time, the transfer will be a transfer of words. There will be a byte or half data transfer at the end of the transfer if the total number of bytes to transfer is not an multiple of 4.

A.4.1.1 ap_2_ap

This script is used to transfer data inside the application processor memory using BurstDMA.

Using this script there is no restriction on the number of bytes to transmit, the source address and destination address alignment, or on the source memory and destination memory type. But word alignment and source/destination memory type will impact performance. In case that the source and destination are in the same memory type, being word aligned allows the SDMA to use the copy capability of the DMA units. When it is not the case the SRAM of the SDMA is used as temporary buffer.

A.4.1.1.1 Parameters Transmitted Through the Context ap_2_ap

None.

A.4.1.1.2 Parameters Transmitted Through the Buffer Descriptor ap_2_ap

The number of bytes to transmit is stored in the first Buffer descriptor word

The source address is stored in the second Buffer descriptor word (address field).

The destination address is stored in the Extended Buffer address

A.4.1.1.3 Use of the General Register During the Execution ap_2_ap

- r0: nb of bytes to transfer, counter during loop
- r2: channel cb address, scratch
- r3: current buffer descriptor pointer, AP destination address, scratch
- r4: mode
- r5: source address loaded by getbd,
- r6: destination address loaded by getcb, scratch
- r7:

A.4.1.1.4 Overview of Script Functionality ap_2_ap

The parameters of the transfer (number of bytes to transfer, source address and destination address) are retrieved from the buffer descriptor.

Depending on the address alignment and the DMA units involved, if possible, the transfer is performed using the copy capability of the DMA unit, or using the SDMA SRAM as a temporary buffer.

When the transfer is finished, this algorithm restarts (a new buffer descriptor is read).

The count in the BD mode word after the script is over is meaningful only if error was reported, else it has the same value as was fed as input to the script.

A.4.1.2 uartsh_2_mcu

This script is used to transfer data from the shared UART to the External memory. The UART is connected to the ARM platform. The UART is an 8-bit peripheral, but when reading the FIFO, 16 bits are retrieved. In fact, the 8 MSB are the status bytes for the current data. At each access the value of this byte is checked to verify that a valid data was retrieved. If an error is detected the transfer is stopped and the information is sent back to the Host through the buffer descriptor with the byte count field updated.

A.4.1.2.1 Parameters Transmitted Through the Context uartsh_2_mcu

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check event - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the peripheral Rx FIFO (project dependent)

r7: Watermark level - Used to determine the maximum of data that can be sent to the peripheral each time the channel is started.

A.4.1.2.2 Parameters Transmitted Through the Buffer Descriptor uartsh_2_mcu

The number of bytes to transmit is stored in the first Buffer descriptor word

The destination address is stored in the second Buffer descriptor word (address field).

The Extended Buffer address is not used

A.4.1.2.3 Use of the General Register During the Execution uartsh_2_mcu

r0 = mask to check events2

r1 = mask to check events

r2 = AP M3 destination address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.4.1.2.4 Overview of Script Functionality uartsh_2_mcu

When the UART sends a DMA request, it can be for three different reasons:

When the RX threshold is over passed. It deals with the most frequent event.

When the AGEING timer has reached its final value, meaning there is less data than the RX threshold in the RX FIFO, and they have been present for too long. It is called the aging DMA request.

When IDLE timer reached its final value; meaning the RX FIFO has been empty for too long. It is called the iddle dma request. The IDLE timer is not used, so only normal and AGEING DMA requests are supported by the script.

The first time the script is executed, a buffer descriptor is opened, the buffer destination address and its size are retrieved from the BD. Then the script checks if the RRDY bit of USR1 is set. If yes, it means that the Watermark level has been reached, there are "WML" bytes in RX FIFO. If the count field of the opened BD is higher than WML-1, the SDMA script will perform WML-1 read accesses to the RX FIFO and WML write access to the destination buffer (a read access then a write access is call a data transfer loop). If the count field is lesser than the WML-1 value, the count field will be the data transfer loop size. So the data transfer loop size equals min(WML-1,BD count)

So there will be always one data remaining in the UART RX FIFO after every data transfer loop and the ageing timer will always trigger a DMA request. It means that the channel on which the script is run will be always closed on an AGEING DMA request.

Now assume the script is triggered by a DMA request, but RRDY is not set. It means that it deals with an AGEING DMA request; there is at least one data in the RX FIFO. The data is read and written into the destination buffer, and then the RDR is checked again. If it is set and if the destination data buffer is not full (BD count is not 0), a new data is read and so on until RDR is 0. When RDR is 0, the ageing timer interrupt flag is disabled by setting the AGTIM bit, the count field of BD is updated and the current BD is closed. Then the script tries to open the next BD if the Continuous bit of the current BD was set, if there is no more BD to open, the channel is stopped.

When reading a data from the UART RX FIFO, the MSB are check to verify that a valid data has been retrieved. If an error is detected the transfer is stopped, the error bit is set in the BD, the count is updated, the BD is closed and an interrupt is sent to the ARM.

When the buffer is full, this algorithm restarts (a new buffer descriptor is read, if it exists).

The next diagram illustrates the script algorithm, it is quite complex but things become more obvious when we notice that there is a loop for the normal DMA request (in sky-blue) and on when the AGEING DMA request was detected (in yellow).

A.4.1.3 shp_2_mcu

This generic script is used to transfer data from a 8, 16, 24 or 32-bit peripheral connected to the shared peripheral bus accessed through the SPBA to memories accessed by the BurstDMA (External memories). It can be used for SSI (8, 16, 24 or 32-bit data size), for CSPI (32-bit data size) or for SDHC(MMC)/SIM (16-bit data size), these peripherals being connected to the shared peripheral bus.

A.4.1.3.1 Parameters Transmitted Through the Context shp_2_mcu

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check event - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the peripheral Rx FIFO (project dependent)

r7: Watermark level - Used to determine the maximum of data that can be sent to the peripheral each time the channel is started.

A.4.1.3.2 Parameters Transmitted Through the Buffer Descriptor shp_2_mcu

The peripheral size/data length is set in the command field of the first Buffer descriptor word, specially bits 24 and 25.

The number of bytes to transmit is stored in the first Buffer descriptor word (count field)

The destination address in the external memory is stored in the second Buffer descriptor word (address field).

The Extended Buffer address is not used.

A.4.1.3.3 Use of the General Register During the Execution shp_2_mcu

r0 = mask to check events2

r1 = mask to check events

r2 = BD destination address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.4.1.3.4 Overview of Script Functionality shp_2_mcu

The parameters of the transfer (peripheral size, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the peripheral sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" data in the FIFO.

During this data transfer loop, the number of bytes that are transferred from the RX FIFO of the peripheral to the EMI is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory write access is fixed by the parameter peripheral size.

The necessary number of transfer loops are executed in order to fill the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is filled, this algorithm restarts (a new buffer descriptor is read).

A.4.1.4 mcu_2_shp

This generic script is used to transfer data from memories accessed by the BurstDMA (External memories) to a 8, 16, 24 or 32-bit peripheral connected to the shared peripheral bus accessed through the SPBA. It can be used for SSI (8, 16, 24 or 32-bit data size), for CSPI (32-bit data size), for SDHC(MMC)/SIM (16-bit data size) or for UART3,4 (8-bit data size), these peripherals being connected to the shared peripheral bus.

A.4.1.4.1 Parameters Transmitted Through the Context mcu_2_shp

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check events - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the peripheral Tx FIFO (project dependent)

r7: Watermark level - Used to determine the maximum of data that can be retrieved from the peripheral each time the channel is started.

A.4.1.4.2 Parameters Transmitted Through the Buffer Descriptor mcu_2_shp

The peripheral size/data length is set in the command field of the first Buffer descriptor word, specially bits 24 and 25.

The number of bytes to transmit is stored in the first Buffer descriptor word (count field)

The source address in the external memory is stored in the second Buffer descriptor word (address field).

The Extended Buffer address is not used.

A.4.1.4.3 Use of the General Register During the Execution mcu_2_shp

r0 = mask to check events2

r1 = mask to check events

r2 = EVENTS / EVENTS2

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.4.1.4.4 Overview of Script Functionality mcu_2_shp

The parameters of the transfer (peripheral size, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the peripheral sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" empty rooms in the FIFO.

During this data transfer loop, the number of bytes that are transferred from the EMI to the TX FIFO of the peripheral is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory read access is fixed by the parameter peripheral size.

The necessary number of transfer loops are executed in order to empty the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is empty, this algorithm restarts (a new buffer descriptor is read).

A.4.1.5 uart_2_mcu

This script is similar to uartsh_2_mcu script. The only difference between these scripts is the way SDMA access UART RX FIFO. In case of uartsh_2_mcu, the UART RX FIFO can directly be accessed by SDMA as it is on Shared Peripheral Bus. In case of uart_2_mcu, SDMA script uses one of its DMA (Peripheral DMA) to access UART RX FIFO.

A.4.1.6 app_2_mcu

This generic script is used to transfer data from a 8, 16, 24 or 32-bit peripheral connected to the AIPS accessed through the Periphera DMA of SDMA, to memories accessed by the BurstDMA (External memories). It can be used for SSI (8, 16, 24 or 32-bit data size), for CSPI (32-bit data size) or for SDHC(MMC)/SIM (16-bit data size).

A.4.1.6.1 Parameters Transmitted Through the Context app_2_mcu

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check event - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the peripheral Rx FIFO (project dependent)

r7: Watermark level - Used to determine the maximum of data that can be sent to the peripheral each time the channel is started.

A.4.1.6.2 Parameters Transmitted Through the Buffer Descriptor app_2_mcu

The Data Length peripheral size/data length is set in the command field of the first buffer descriptor word, specially bits 24 and 25.

The number of bytes to transmit is stored in the first buffer descriptor word (count field)

The destination address in the external memory is stored in the second buffer descriptor word (address field).

The Extended Buffer address is not used.

A.4.1.6.3 Use of the General Register During the Execution app_2_mcu

r0 = mask to check events2

r1 = mask to check events

r2 = AP M3 destination address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.4.1.6.4 Overview of Script Functionality app_2_mcu

The parameters of the transfer (peripheral size, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the peripheral sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" data in the FIFO.

During this data transfer loop, the number of bytes that are transferred from the RX FIFO of the peripheral to the EMI is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory write access is fixed by the parameter peripheral size.

The necessary number of transfer loops are executed in order to fill the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is filled, this algorithm restarts (a new buffer descriptor is read).

A.4.1.7 mcu_2_app

This generic script is used to transfer data from memories accessed by the BurstDMA (External memories), to a 8, 16, 24 or 32-bit peripheral connected to the AIPS accessed through the Periphera DMA of SDMA. It can be used for SSI (8, 16, 24 or 32-bit data size), for CSPI (32-bit data size) or for SDHC(MMC)/SIM (16-bit data size).

A.4.1.7.1 Parameters Transmitted Through the Context mcu_2_app

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check event - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the peripheral Tx FIFO (project dependent)

r7: Watermark level - Used to determine the maximum of data that can be sent to the peripheral each time the channel is started.

A.4.1.7.2 Parameters Transmitted Through the Buffer Descriptor mcu_2_app

The peripheral size/data length is set in the command field of the first buffer descriptor word, specially bits 24 and 25.

The number of bytes to transmit is stored in the first buffer descriptor word (count field).

The source address in the external memory is stored in the second buffer descriptor word (address field).

The Extended Buffer address is not used.

A.4.1.7.3 Use of the General Register During the Execution mcu_2_app

r0 = mask to check events2

r1 = mask to check events

r2 = TX FIFO address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.4.1.7.4 Overview of Script Functionality mcu_2_app

The parameters of the transfer (peripheral size, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the peripheral sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" empty room in the FIFO.

During this data transfer loop, the number of bytes that are transferred from the EMI to the TX FIFO of the peripheral is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory read access is fixed by the parameter peripheral size.

The necessary number of transfer loops are executed in order to fill the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is filled, this algorithm restarts (a new buffer descriptor is read).

A.4.1.8 mcu_2_spdif

This script performs a data move from AP buffer to 24 bits FIFO of the SPDIF. This peripheral is on the peripheral bus, the SDMA accesses it through the SPBA. The MCU buffer is accessed through the burst DMA.

A.4.1.8.1 Parameters Transmitted Through the Context mcu_2_spdif

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check events - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: Address of SPDIF STL register

r7: Watermark level - Used to determine the maximum data that can be sent to the peripheral each time the channel is started.

A.4.1.8.2 Parameters Transmitted Through the Buffer Descriptor mcu_2_spdif

The data length (Either 32-bit or 16-bit) is set in the command field of the first Buffer descriptor word, especially bits 24, 25.

The number of bytes to transmit is stored in the first Buffer descriptor word (count field)

The source address in the external memory is stored in the second Buffer descriptor word (address field).

The Extended Buffer address is not used.

A.4.1.8.3 Use of the General Register During the Execution mcu_2_spdif

r0 = mask to check events2

r1 = mask to check events

r2 = EVENTS / EVENTS2

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.4.1.8.4 Overview of Script Functionality mcu_2_spdif

The parameters of the transfer (data length, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the spdif sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" of empty room in the FIFO (Sum of left and right FIFO's).

During this data transfer loop, the number of bytes that are transferred from the EMI to the TX FIFO of the spdif is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory read access is fixed by the parameter data length.

The necessary numbers of transfer loops are executed in order to empty the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is empty, this algorithm restarts (a new buffer descriptor is read).

A.4.1.9 spdif_2_mcu

This script performs a data move from 24 bits FIFO of the SPDIF to AP buffer. This peripheral is on the peripheral bus, the SDMA accesses it through the SPBA. The MCU buffer is accessed through the burst DMA.

A.4.1.9.1 Parameters Transmitted Through the Context spdif_2_mcu

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check events - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: Address of SPDIF SRL register

r7: Watermark level - Used to determine the maximum data that can be sent to the peripheral each time the channel is started.

A.4.1.9.2 Parameters Transmitted Through the Buffer Descriptor spdif_2_mcu

The data length (Either 32 bit or 16 bit) is set in the command field of the first Buffer descriptor word, especially bits 24, 25.

The number of bytes to transmit is stored in the first Buffer descriptor word (count field)

The destination address in the external memory is stored in the second Buffer descriptor word (address field).

The Extended Buffer address is not used.

A.4.1.9.3 Use of the General Register During the Execution spdif_2_mcu

r0 = mask to check events2

r1 = mask to check events

r2 = EVENTS / EVENTS2

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.4.1.9.4 Overview of Script Functionality spdif_2_mcu

The parameters of the transfer (data length, number of bytes to transfer and destination address) are retrieved from the buffer descriptor.

When the spdif sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" date in the FIFO (Sum of left and right FIFO's).

During this data transfer loop, the number of bytes that are transferred from the RX FIFO of the spdif to the EMI is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory write access is fixed by the parameter data length.

The necessary numbers of transfer loops are executed in order to empty the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is empty, this algorithm restarts (a new buffer descriptor is read).

A.4.1.10 mcu_2_firi

This script performs a data move from AP buffer to 8, 16, 24 or 32-bit FIFO of the FIRI. Both MCU buffer and FIRI are accessed through the burst DMA.

A.4.1.10.1 Parameters Transmitted Through the Context mcu_2_firi

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check events - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: Address of FIRI Transmit FIFO register

r7: Watermark level - Used to determine the maximum data that can be sent to the peripheral each time the channel is started.

A.4.1.10.2 Parameters Transmitted Through the Buffer Descriptor mcu_2_firi

The data length (8, 16, 24 or 32-bit) is set in the command field of the first buffer descriptor word, especially bits 24, 25.

The number of bytes to transmit is stored in the first buffer descriptor word (count field)

The source address in the memory is stored in the second buffer descriptor word (address field).

The Extended Buffer address is not used.

A.4.1.10.3 Use of the General Register During the Execution mcu_2_firi

r0 = mask to check events2

r1 = mask to check events

r2 = EVENTS / EVENTS2

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.4.1.10.4 Overview of Script Functionality mcu_2_firi

The parameters of the transfer (data length, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the firi sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" of empty room in the FIFO (Sum of left and right FIFO's).

During this data transfer loop, the number of bytes that are transferred from the EMI to the TX FIFO of the firi is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory read access is fixed by the parameter data length.

The necessary numbers of transfer loops are executed in order to empty the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is empty, this algorithm restarts (a new buffer descriptor is read).

A.4.1.11 firi_2_mcu

This script performs a data move from 8, 16, 24, 32-bit FIFO of the FIRI to the AP buffer. Both the MCU buffer and FIRI are accessed through the burst DMA.

A.4.1.11.1 Parameters Transmitted Through the Context firi_2_mcu

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check event - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the FIRI RX FIFO

r7: Watermark level - Used to determine the maximum of data that can be sent to the peripheral each time the channel is started.

A.4.1.11.2 Parameters Transmitted Through the Buffer Descriptor firi_2_mcu

The data length (8, 16, 24 or 32 bits) is set in the command field of the first buffer descriptor word, especially bits 24 and 25.

The number of bytes to transmit is stored in the first buffer descriptor word (count field)

The destination address is stored in the second buffer descriptor word (address field).

The Extended Buffer address is not used

A.4.1.11.3 Use of the General Register During the Execution firi_2_mcu

r0 = mask to check events2

r1 = mask to check events

r2 = AP M3 destination address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.4.1.11.4 Overview of Script Functionality firi_2_mcu

When the FIRI sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" of data in the FIFO.

During this data transfer loop, the number of bytes that are transferred from the RX FIFO of the FIRI to the EMI is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory write access is fixed by the parameter data length.

The necessary numbers of transfer loops are executed in order to fill the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is full, this algorithm restarts (a new buffer descriptor is read).

A.4.2 SDMA RAM scripts

A.4.2.1 mcu_2_ssiapp

This script performs a data move from AP buffer to 8, 16, 24 or 32-bit FIFO of the SSI in the AP region with 2 FIFOs enabled. The MCU buffer is accessed through the burst DMA, while SSI is accessed with Per DMA.

A.4.2.1.1 Parameters Transmitted Through the Context mcu_2_ssiapp

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check events - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: Address of SSI Transmit FIFO 0 register

r7: Watermark level - Used to determine the maximum data that can be sent to the peripheral each time the channel is started.

A.4.2.1.2 Parameters Transmitted Through the Buffer Descriptor mcu_2_ssiapp

The data length (8, 16, 24 or 32-bit) is set in the command field of the first buffer descriptor word, especially bits 24 and 25.

The number of bytes to transmit is stored in the first buffer descriptor word (count field).

The source address in the memory is stored in the second buffer descriptor word (address field).

The Extended Buffer address is not used.

A.4.2.1.3 Use of the General Register During the Execution mcu_2_ssiapp

r0 = mask to check events2

r1 = mask to check events

r2 = Tx FIFO address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.4.2.1.4 Overview of Script Functionality mcu_2_ssiapp

The parameters of the transfer (data length, number of bytes to transfer and source address) are retrieved from the buffer descriptor.

When the SSI sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" of empty room in the FIFO (Sum of left and right FIFO's).

During this data transfer loop, the number of bytes that are transferred from the EMI to the TX FIFO of the SSI is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory read access is fixed by the parameter data length.

The necessary numbers of transfer loops are executed in order to empty the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is empty, this algorithm restarts (a new buffer descriptor is read).

A.4.2.2 ssiapp_2_mcu

This script performs a data move from the 8, 16, 24 or 32-bit FIFO of the SSI to the AP buffer. The SSI is in the AP region and 2 FIFO enabled. The MCU buffer is accessed through the burst DMA while the SSI is accessed through Per DMA.

A.4.2.2.1 Parameters Transmitted Through the Context ssiapp_2_mcu

r0: mask to check events2 - If script is triggered by event 32+I, r0[I] must be set to 1. (Project dependent)

r1: mask to check event - If script is triggered by event I, r1[I] must be set to 1. (Project dependent)

r6: address of the SSI RX FIFO 0

r7: Watermark level - Used to determine the maximum of data that can be sent to the peripheral each time the channel is started.

A.4.2.2.2 Parameters Transmitted Through the Buffer Descriptor ssiapp_2_mcu

The data length (8, 16, 24, 32 bits) is set in the command field of the first buffer descriptor word, especially bits 24 and 25.

The number of bytes to transmit is stored in the first buffer descriptor word (count field)

The destination address is stored in the second buffer descriptor word (address field).

The Extended Buffer address is not used

A.4.2.2.3 Use of the General Register During the Execution ssiapp_2_mcu

r0 = mask to check events2

r1 = mask to check events

r2 = AP M3 destination address

r3 = Ram channel context address

r4 = Bd mode

r5 = Bd mode Count

r6 = SDMA memory mapped regs base address

r7 = Watermark

A.4.2.2.4 Overview of Script Functionality ssiapp_2_mcu

When the SSI sends an event, a data transfer loop is executed. This event is set when there is at least "watermark level" of data in the FIFO(sum of left and right FIFOs).

During this data transfer loop, the number of bytes that are transferred from the RX FIFO of the SSI to the EMI is equal to the minimum value between the "watermark level" and the number of data that have not been transferred yet. For each data transfer (one read, one write), the size of the memory write access is fixed by the parameter data length.

The necessary numbers of transfer loops are executed in order to fill the buffer, but they are executed only if the event is set. While the event is not set, the channel is not executable.

When the buffer is full, this algorithm restarts (a new buffer descriptor is read).

A.4.2.3 mcu_2_ssish

This script is similar to mcu_2_ssiapp. The only difference is the way SDMA core accesses the SSI TX FIFO. For script of mcu_2_ssiapp, SDMA core accesses the TX FIFO through Functional Unit Bus and regards it as the peripheral DMA, whereas for the mcu_2_ssish, the access is through the Shared Peripheral Bus and SSI can be directly accessed by SDMA.

A.4.2.4 ssish_2_mcu

This script is similar to ssiapp_2_mcu. The only difference is the way SDMA core accesses the SSI RX FIFO. For script of ssiapp_2_mcu, SDMA core accesses the RX FIFO through Functional Unit Bus and regards it as the peripheral DMA, whereas for the ssish_2_mcu, the access is through the Shared Peripheral Bus and SSI can be directly accessed by SDMA.

A.4.2.5 p_2_p

This script performs a data move of 32 bits from the peripheral's FIFO connected either on SPBA or AIPS bus to another. If destination FIFO belongs to ASRC then pad adding is performed after every N samples from the source device. If the source FIFO belongs to ASRC then after transmitting N samples a pad is swallowed. In case there is a transaction between ASRC(source) and SPDIF (destination) no pad adding or swallowing is required because SPDIF expects even number of samples.

This script presently does not deal with the transactions involving two devices connected to AIPS bus.

A.4.2.5.1 Parameters Transmitted Through the Context p_2_p

r0: LWML event mask

r1: HWML event mask

r2: source address

r6: destination address

r7: INFO. See details in Table A-1 :

p_2_p (r7 INFO)

Bits	Name	Description
0-7	Lower WML	Watermark Level
8	PS	1 : Pad Swallowing 0 : No Pad swallowing
9	PA	1 : Pad Adding 0 : No Pad swallowing
10	SPDIF	If this bit is set both source and destination are on SPBA
11	Source Bit(SP)	1 : Source on SPBA 0 : Source on AIPS

Table continues on the next page...

Bits	Name	Description
12	Destination Bit (DP)	1 : Destination on SPBA 0 : Destination on AIPS
13-15	----------	MUST BE 0
16-23	Higher WML	HWML
24-27	N	Total number of samples after which Pad adding/Swallowing must be done. It must be odd.
28	Lower WML Event (LWE)	SDMA events reg to check for LWML event mask 0 : LWE in EVENTS register 1 : LWE in EVENTS2 register
29	Higher WML Event(HWE)	SDMA events reg to check for HWML event mask 0 : HWE in EVENTS register 1 : HWE in EVENTS2 register
30	----------	MUST BE 0
31	CONT	1 : Amount of samples to be transferred is unknown and script will keep on transferring samples as long as both events are detected and script must be manually stopped by the application 0 : The amount of samples to be is equal to the count field of mode word

A.4.2.5.2 Parameters Transmitted Through the Buffer Descriptor p_2_p

In the first buffer descriptor word, the mode and number of types to transfer is included.

The Buffer address and Extended Buffer address are not used.

A.4.2.5.3 Use of the General Register During the Execution p_2_p

The register usage for each case(spba_2_spba, asrc_2_spba) are different, so don't present here.

A.4.2.5.4 Overview of Script Functionality p_2_p

The script get the number of data to transfer per event according to the bit 31(CONT) of INFO. If CONT is 0, each transfer loop will transmit the number of data as in the mode count; Or else, it means the number of samples to be transferred is not known and the transfer must take place as long as both events are detected and the script must be manually stopped by the application in case the transfer needs to be stopped.

When PS or PA is set, then after each N words transfer, a pad swallowing or pad adding will be done. This is to handle the ASRC involved transfer.

The transfer is executed only when the event is set. After each transfer for the event, the script will try to obtain a new BD.

Appendix B
i.MX53 Revision History

B.1 Arch Overview Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
Block List	In Table 1-1, "Digital and Analog Blocks," removed "(Pop Only)" in the description of the EXTMC row.
Features	In Section "Features," replaced the text "One CSPI, up to 26 Mbps in master mode (up to 10MHz in slave mode)" with "One CSPI; refer to data sheet for performance parameters."
Features	In Section "Features," replaced the text "Two ECSPI (enhanced CSPI), up to 52 Mbps each" with "Two eCSPI (enhanced CSPI); refer to data sheet for performance parameters."
Block List	In Table 1-1, "Digital and Analog Blocks," updated the description of the SAHARA row.
Architectural Partitioning	In Section, "Architectural Partitioning, added a NOTE referring to SATA Temperature Sensor application note.

B.2 Memory Map Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
ARM Platform System Memory Map	In Table 2-1 "System Memory Map," removed "(TBD)."

B.3 Interrupts Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
ARM Platform Interrupts	In Table 3-1, "ARM Domain Interrupt Summary," added the PWRFAIL_INT signal in the description of IOMUXC row.
ARM Platform Interrupts	In Table 3-1, "ARM Domain Interrupt Summary," updated the description of P_PLATFORM_NE_32K_256K row.

B.4 External Memory Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
Features	Updated Table 5-2, "EIM multiplexing," as per i.MX53 datasheet.
Features	Removed "EIM/NANDF multiplexing diagram."
Overview	Replaced "Each AXI bud master" with "Each AXI bus master."
Features	In Table 5-2, "EIM multiplexing," corrected the DSZ values in 8 bit columns.

B.5 Multimedia Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
Clocking Architecture	Updated Section "Clocking Architecture."
Software	Updated first sentence of Section "Software."

B.6 AHBMAX Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
System Connectivity	Added Section "System Connectivity."

B.7 ASRC Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
Introduction	In Section "Introduction ," updated Figure "General System Overview."

B.8 AUDMUX Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
Internal Network Mode	In Section "Internal Network Mode," corrected the dimensions of images.
Port Receive Data Modes	In Section "Port Receive Data Modes," corrected the dimensions of images.

B.9 CSU Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
Features	In Section "Features," added an introductory paragrah.
Features	In Section "Features," updated the bullet point related to peripheral access policy and added a note.
Features	In Section "Features," added a note.
Peripheral Access Policy	Changed the title of the Section "Execution Mode Access Policy" to " Peripheral Access Policy."

B.10 DPLLC Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
DPLLC Memory Map/Register Definition	In Section "DPLLC Memory Map/Register Definition," corrected the positions of bits.
DPLLC Memory Map/Register Definition	In Section "DPLLC Control Register," corrected restes of UPEN and ref_clk_sel[1:0]. Also, replaced "10 clock2 is selected" with "10 clock2 is selected (24MHz oscillator clock)." in the field description of ref_clk_sel[1:0].
Overview	In Section "Overview," updated Figure "Block Diagram of the DPLLC."
Feature Description	In Section "Feature Description," updated the third bullet point.
DPLLC Memory Map/Register Definition	In Table "DPLLC_CTL field descriptions," updated the description of ref_clk_sel[1:0].
Normal Mode	In Section "Normal Mode," updated Figure "Clock MUXing and Gating/Divider Circuitry."

B.11 EIM Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
Features	Changed six chip selects to four chip selects.
Multiplexed Address/Data Mode	Updated second sentence of Section "Multiplexed Address/ Data Mode"
Multiplexed Address/Data Mode	Updated Table 25-1 as per the IMX53 Data sheet.
EIM Memory Map/Register Definition	In Section "EIM Memory Map/Register Definition" updated offset/base addresses for "Programmable Registers."
EIM Memory Map/Register Definition	In Table "EIM_CSnWCR1 field descriptions," updated WWSC row.
Boot Mode	In Section "Boot Mode," added a cross reference to Chapter "System Boot."
Boot Mode	In Table "Table 25-2," removed first 7 rows.
Other Important Block I/O Signals Internal to the SoC	Updated EIM_BOOT row of the table in Section "Other Important Block I/O Signals Internal to the SoC."

Table continues on the next page...

Topic Cross-Reference	Customer-facing Description
EIM Memory Map/Register Definition	Updated NOTEs in AUS, CSREC, DSZ in EIM_CSnGCR1 field descriptions in Table "EIM_CSnGCR1 field descriptions."
EIM Memory Map/Register Definition	Removed the NOTE in ERRST row of Table "EIM_WIAR field descriptions."
Booting from EIM	Updated second paragraph of Section "Booting from EIM."
Access to AMD Flash	Removed Section "AMD Flash Boot."
Access to Intel Sibley Flash	Removed Section "Intel Sibley Flash Boot."
MDOC Device Boot	Updated Section "MDOC Device Boot."
Samsung OneNAND Boot	Updated Section "Samsung OneNAND Boot."
Access to Spansion Flash	Removed Section "Spansion Flash Boot."
Overview	Replaced "EIM_BOOT[12:0]" with "EIM_BOOT[2:0]" in Figure "EIM Diagram."
EIM Memory Map/Register Definition	In Section "Chip Select n Read Configuration Register 1," added footnotes for RWSC and OEA bit fields.
EIM Memory Map/Register Definition	In Section "Chip Select n Write Configuration Register 1," added a footnote for WWSC bit field.
EIM Memory Map/Register Definition	In Section "Chip Select n General Configuration Register 1," added a footnotes for CSREC, DSZ, and MUM bit fields.
EIM Memory Map/Register Definition	In Section "Chip Select n General Configuration Register 1," updated the NOTE in in the field description of CSREC row.
Initialization Information	Removed Section "Active Chip Selects and Address Space Configuration."
EIM Memory Map/Register Definition	Removed note in WIAR row EIM IP Access Register.
EIM Memory Map/Register Definition	Removed note in WAL row of Chip Select n Write Configuration Register 1.
EIM Memory Map/Register Definition	Updated note in WWSC row of Chip Select n Write Configuration Register 1.
EIM Memory Map/Register Definition	Removed note in RAL row of Chip Select n Read Configuration Register 1.
EIM Memory Map/Register Definition	Updated note in RWSA row of Chip Select n Read Configuration Register 1.
EIM Memory Map/Register Definition	Removed note in MUX16_BYP_GRANT row of Chip Select n General Configuration Register 2.
EIM Memory Map/Register Definition	Removed note in AUS row of Chip Select n General Configuration Register 1.

B.12 ESDCTL Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
ESDCTL	Added a NOTE to Section "ESDCTL MR4 Derating Register (ESDCTL_ESDMR4)."
ESDCTL	Added Table 28-34, "ODT Values," and updated the descriptions of ODT3_INT_RES, ODT2_INT_RES, ODT1_INT_RES, and ODT0_INT_RES bits with references to the same table.
ESDCTL	Added a line in the NOTE to Section "ESDCTL MR4 Derating Register (ESDCTL_ESDMR4)," clarifying the temperature sensor location.
ESDCTL Memory Map/Register Definition	Updated definitions for bit fields "SDE_0" and "SDE_1" in the ESDCTL register.

B.13 ESDHCv2 Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
ESDHCV2	Removed the text "In the case of command with busy pending: This status indicates that a busy state follows the command and the data line is in use. This bit will be cleared when the DAT0 line is released." from DLA bit field row of Table "ESDHCV2x_PRSSTAT field descriptions."

B.14 EXTMC Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
EIM and NFC IO pin sharing	Updated Section "EIM and NFC IO pin sharing."
EIM and NFC IO pin sharing	Updated Table 31-1, "EIM multiplexing."
EIM and NFC IO pin sharing	Removed Figure "EIM/NANDF multiplexing diagram."
IP Bus Memory Map	In Table 31-3, "EXTMC Block memory map," updated Start ADDR and End ADDR columns for base addresses.

B.15 GPIO Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
External Signal Description	Removed Sections "External Signal Description" and "Signals Overview", which described only internal signals.

B.16 IOMUXC Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
CH0_MODE[10]	In Section "General Purpose Register 2," changed "29kOhm" to "28 kOhms," for BGREF_ RRMODE in Table "IOMUXC_GPR2 field descriptions."
DDR_SEL	In Section "IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE," updated the field description for DR_SEL.
IOMUXC_memoryMap	In Section "Programmable Registers," Added Table 43-2, "DDR Output Driver Average Impedance," and placed cross references to it in DSE field descriptions of DQM3, SDQS3, DQM2, SDODT1, SDQS2, RESET, SDCLK_1, CAS, SDCLK_0, SDQS0, SDODT0, DQM0, RAS, SDQS1, DQM1, ADDDS, B0DS, B1DS, CTLDS, B2DS, and B3DS registers, replacing existing field description.

B.17 IPU Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
IPU Memory Map/Register Definition	In Section, "Programmable Registers," added the missing text in the last sentence."
IPU Memory Map/Register Definition	The register base address for IPU has been changed from "0x1800_0000" to "0x1E00_0000".
IPU Memory Map/Register Definition	Updated duplicate bit 22 offset in IPU_INT_CTRL_1, 3, 5, 13, IPU_SDMA_EVENT_1, 3, 13, IPU_CH_DB_MODE_SEL0, IPU_INT_STAT_1, 3, 5, 13, IPU_CUR_BUF_0, and IPU_CH_BUF0_RDY0 to correct offsets.
IPU Memory Map/Register Definition	Updated bit names in IPU_IDMAC_CH_EN_2 and IPU_CH_BUF0_RDY0 registers.

B.18 KPP Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
Ghost Key Problem and Correction	In Section "Ghost Key Problem and Correction," updated the Figure "Matrix with "Ghost" Key Protections."

B.19 M4IF Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
M4IF	In Table Watermark Interrupt and Status 0 Register (M4IF_WMIS0), corrected typos in the descriptions of WS0_7 and WS0_6 rows.

B.20 NFC Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
NFC	In Section "AXI Memory Map," corrected the base addresses of NFC registers.

B.21 OCRAM Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
Memory Map	Removed Section "Memory Map."

B.22 PLARB1 Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
Overview	Added Section System Connectivity

B.23 PWM Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
PWM	In Section "PWM Period Register," replaced "Writing 0xFFFF to this register will achieve the same result as writing 0xFFFE," with "Maximum allowed value for this register is 0xFFFE."
PWM	In Section "PWM Sample Register," replaced "The FIFO can be written and read when the PWM is disabled" with "The FIFO can be written at any time, but can be read only when the PWM is enabled."
FIFO	In Section "FIFO," replaced "The FIFO can be written and read when the PWM is disabled" with "The FIFO can be written at any time, but can be read only when the PWM is enabled."
Overview	In Section "Overview," updated Figure "Pulse-Width Modulator Block Diagram."

B.24 SAHARA Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
Features	Updated Section "Features."
Overview	In Section "Overview," updated Figure "Block Diagram of SAHARA."

B.25 SATA Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
-	Restructured presentation of information in this chapter.
	Added Section "."
Features	In Section, "Features," updated the bullet point related to SATA speeds.
Hot Plug	In Section, "Hot Plug," removed last two bullet points.
Link Layer Tx OOB Sequence Generation	In Section, "Link Layer Tx OOB Sequence Generation," removed second sentence of the NOTE.
PHY Initialization Details	In Section, "PHY Initialization Details," removed footnote.
Port DMA	In Section, "Port DMA," removed NOTE.
Bus Interface Unit	Removed Section "AMBA AXI Interface."
Features	Updated Section "Features."
Block Diagram	Updated Section "Block Diagram."

B.26 SATA PHY Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
-	Removed "Top metal options for active section", "Naming conventions", "Static timing constraint files and setup", and "Core sizes" sections.
-	Removed "Signaling completion of an operation" and "Digital scan operation" sections.
-	Removed "Boundary scan and jtag interface connections", "SoC layout considerations", "Back-to-back diodes", and "Using deliverables in physical design tools" sections.
-	Removed "Simulation", "Verilog model files and simulators", "Gate-level models", "Macros affecting simulation", "Running example tests", "Spice models descriptions", and "Using the spice models" sections.
Overview	Added Chapter "DesignWare Cores SATA2 PHY."
ID Code	In Table "Table 64-29" replaced value "0011" with "1017."
Control Register Bus Interface	In Section "Control Register Bus Interface," corrected a crossreference.
Signal Descriptions Information	Removed "Section Boundary Scan Port Signals," and its references in sections "Common Signals" and "Signal Descriptions Overview."

B.27 SDMA Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
ARM Platform Memory Map and Control Register Definitions	In Section "ARM platform Channel 0 Pointer (SDMAARM_MC0PTR)," replaced the text "Refer to the API document i.MX53 SDMA Scripts User Manual for the use of this register" with "Appendix A fully describes the SDMA Application Programming Interface (API) in SDMAARM_MC0PTR field descriptions."
Standard Boot Sequence	In Section "Standard Boot Sequence," replaced the text "4. Perform any necessary setup as required by the standard boot script in ROM (this is described in i.MX53 SDMA Scripts User Manual)" with "4. Perform any necessary setup as required by the standard boot script in ROM (this is described in Appendix A)."
SDMA Initialization	In Section "SDMA Initialization," replaced the text "The API document i.MX53 SDMA Scripts User Manual describes the setup of the SDMA" with "The Appendix A describes the setup of the SDMA."
SDMA Internal (Core) Memory Map and Internal Register Definitions	Changed reset value of SDMACORE_ENDIANNESS[APEND] to 1 and added that bit is tied to logic '1'.
Overview	Removed Section "References," and pointed cross references directly to appendix Section "SDMA Scripts Overview."

B.28 SJC Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
External Signal Overview	In Section "External Signal Overview," removed a duplicate Figure "SJC Connections."
	Various updates to figures in several sections.

B.29 SPBA Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
Introduction	In Section "Introduction," updated Figure "SPBA Block Diagram."

B.30 SRC Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
Boot Output Signals	In Section "Boot Output Signals ," updated the boot address from F000_0000 to 0800_0000.

B.31 SSI Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
Features	Added min/max audio sampling rates for I2S. Added min/max frame rates for AC97.

B.32 USB Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
Queue Head	Updated C bit field position at offset 07-04H.
USB	In "Non-core Registers" section, removed the duplicate rows of USBNC memory map.
USB	In "Core Registers" section, updated the base addresses of USB_UOG_ENDPTCTRL1-7 registers.
	Updated Section "."
Entering Low Power Suspend Mode	Updated Section "Entering Low Power Suspend Mode."
Wake-Up Events	Updated Section "Wake-Up Events."
	Updated Section "."
USB Non-Core Memory Map/Register Definition	Added bit UH1_PHY_CTRL_0[UTMI_ON_CLOCK] entry and description
USB Non-Core Memory Map/Register Definition	Added bit USB_OTG_PHY_CTRL_0[conf2] entry and description
Peripheral (Device) Mode	Added Section "Peripheral (Device) Mode."

Table continues on the next page...

Topic Cross-Reference	Customer-facing Description
Pins Used for Host Controller 1	Added Section "Pins Used for Host Controller 1."
Overview	In Section "Overview."

B.33 WDOG Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
WDOG_B Generation	In Section, "WDOG_B Generation," updated Figure "WDOG-1 Generation."

B.34 XTALOSC Revision History

The following table contains changes made to this block guide.

Topic Cross-Reference	Customer-facing Description
Introduction	Updated Section "Introduction."

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or
+1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064
Japan
0120 191014 or
+81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 010 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
+1-800 441-2447 or
+1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

Document Number: IMX53RM
Rev. 2, 12/2011

Addendum to Rev. 2 of the i.MX53 Applications Processor Reference Manual

Document Number: IMX53RMAD
Rev. 2.1, 5/2012

Contents

Section Number **Title** **Page**

Chapter 1
Introduction

Chapter 2
Fusemap

Chapter 1
Introduction

1.1 About This Document

The following chapter is a supplement to the *i.MX53 Multimedia Applications Processor Reference Manual, Rev. 2*.

Chapter 2
Fusemap

2.1 Fusemap

Table 2-1 is a summary of the fuse locks used in the chip.

Table 2-1. Fuse Lock Summary

Fuse Name	Fuse Bank	Rows Affected
LOCK	0, 1, 3	0800, 0C00, 1400
BOOT_LOCK	0	0804 - 0818, 0840-0844
TESTER_LOCK	0	0820 - 083C
GP_LOCK	0	0860 - 087C
SRK_LOCK	1	0C04
SJC_RESP_LOCK	1	0C08 - 0C20
MAC_ADDR_LOCK	1	0C24 - 0C38
SRK_LOCK88	3	1404 - 142C
SRK_LOCK160	3	1430 - 147C
Bank 4 Locks	4	1808 - 1814

Table 2-2 details the lock fuses, which control the fuse accessibility in an irreversible mode.

NOTE

Setting the lock fuses irreversibly locks the affected rows against any further modification.

Table 2-2. Lock Fuses

FB	Addr	7	6	5	4	3	2	1	0
0	0800	FBWP	FBOP	FBRP	TESTER_LOCK	FBESP	TESTER_LOCK2	GP_LOCK	BOOT_LOCK
0	0804	Reserved	JTAG_SMODE[1:0]		BT_FUSE_SEL	JTAG_HEO	KTE	SEC_JTAG_RE	JTAG_BP

The following section details the various modes and selection of the required boot devices. A separate map is given for each and every boot device. The device select is specified by BOOT_CFG1[7:4] fuses listed in Table 2-3.

Table 2-3. Boot Device Select

Boot Device	BOOT _CFG1[7]	BOOT_CFG1[6]	BOOT_CFG1[5]	BOOT_CFG1[4]
WEIM (Table 2-4)	0	0	0	0
HD (SATA/PATA) (Table 2-5)	0	0	1	0
Serial ROM (Table 2-6)	0	0	1	1
SD/eSD (Table 2-7)	0	1	0	X
MMC/eMMC (Table 2-8)	0	1	1	X
NAND Flash (Table 2-9)	1	X	X	X

Table 2-4. WEIM Boot Fusemap

FB	Addr	7	6	5	4	3	2	1	0
0	080C	**BOOT _CFG1[7]**	**BOOT_CFG 1[6]**	**BOOT_CFG 1[5]**	**BOOT_CFG 1[4]**	**BOOT_CFG 1[3]**	**BOOT_CFG 1[2]**	**BOOT_CFG 1[1]**	**BOOT_CFG 1[0]**
		0	0	0	0	Memory Type: 0 - NOR Flash 1 - OneNAND	Reserved	BT_FREQ 0-800 1-400	BT_MMU_ENABLE
0	0810	**BOOT_CFG 2[7]**	**BOOT_CFG 2[6]**	**BOOT_CFG 2[5]**	**BOOT_CFG 2[4]**	**BOOT_CFG 2[3]**	**BOOT_CFG 2[2]**	**BOOT_CFG 2[1]**	**BOOT_CFG 2[0]**
		Muxing Scheme: 00 - Muxed, 16-bit data (low half) interface 01 - Not muxed, 16-bit data (high half) interface 10 - Reserved 11 - Reserved		Reserved	AXI / DDR Freq 0 - PLL2: 400MHz 1 - PLL2: 333MHz	OSC_FREQ _SEL 0 - 19.2, 24, 26, 27 MHz - auto detect. 1 - Oscillator frequency is 24 MHz	Reserved	SEC_CONFIG[1:0]	
0	0814	**BOOT_CFG 3[7]**	**BOOT_CFG 3[6]**	**BOOT_CFG 3[5]**	**BOOT_CFG 3[4]**	**BOOT_CFG 3[3]**	**BOOT_CFG 3[2]**	**BOOT_CFG 3[1]**	**BOOT_CFG 3[0]**
		OneNand Page Size: 00 - 1KB 01 - 2KB 10 - 4KB 11 - Reserved		Reserved	Reserved	Reserved	Reserved	Reserved	DIR_BT_DIS

Table continues on the next page...

Table 2-4. WEIM Boot Fusemap (continued)

FB	Addr	7	6	5	4	3	2	1	0
0	0818	BT_LPB_DIV[1:0] 00 – Divide by 1 01 – Divide by 2 10 – Divide by 3 11 – Divide by 4		Reserved	BT_LPB	BT_LPB_POLARITY	Reserved	Reserved	WDOG_ENABLE

Table 2-5. HD Boot Fusemap

FB	Addr	7	6	5	4	3	2	1	0
0	**080C**	**BOOT_CFG1[7]**	**BOOT_CFG1[6]**	**BOOT_CFG1[5]**	**BOOT_CFG1[4]**	**BOOT_CFG1[3]**	**BOOT_CFG1[2]**	**BOOT_CFG1[1]**	**BOOT_CFG1[0]**
		0	0	1	0	HD Type: 0 - PATA 1 - SATA	Reserved	BT_FREQ 0-800 1-400	BT_MMU_ENABLE
0	**0810**	**BOOT_CFG2[7]**	**BOOT_CFG2[6]**	**BOOT_CFG2[5]**	**BOOT_CFG2[4]**	**BOOT_CFG2[3]**	**BOOT_CFG2[2]**	**BOOT_CFG2[1]**	**BOOT_CFG2[0]**
		Reserved	Reserved	Reserved	AXI / DDR Freq 0 - PLL2: 400MHz 1 - PLL2: 333MHz	OSC_FREQ_SEL 0 - 19.2, 24, 26, 27 MHz - auto detect. 1 - Oscillator frequency is 24 MHz	Reserved	SEC_CONFIG[1:0]	
0	**0814**	**BOOT_CFG3[7]**	**BOOT_CFG3[6]**	**BOOT_CFG3[5]**	**BOOT_CFG3[4]**	**BOOT_CFG3[3]**	**BOOT_CFG3[2]**	**BOOT_CFG3[1]**	**BOOT_CFG3[0]**
		Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	DIR_BT_DIS
0	0818	BT_LPB_DIV[1:0] 00 – Divide by 1 01 – Divide by 2 10 – Divide by 3 11 – Divide by 4		Reserved	BT_LPB	BT_LPB_POLARITY	Reserved	Reserved	WDOG_ENABLE

Table 2-6. Serial ROM Boot Fusemap

FB	Addr	7	6	5	4	3	2	1	0
0	**080C**	**BOOT_CFG 1[7]**	**BOOT_CFG 1[6]**	**BOOT_CFG 1[5]**	**BOOT_CFG 1[4]**	**BOOT_CFG 1[3]**	**BOOT_CFG 1[2]**	**BOOT_CFG 1[1]**	**BOOT_CFG 1[0]**
		0	0	1	1	Serial ROM select: 0 - I2C 1- SPI	Reserved	BT_FREQ 0-800 1-400	BT_MMU_ENABLE
0	**0810**	**BOOT_CFG 2[7]**	**BOOT_CFG 2[6]**	**BOOT_CFG 2[5]**	**BOOT_CFG 2[4]**	**BOOT_CFG 2[3]**	**BOOT_CFG 2[2]**	**BOOT_CFG 2[1]**	**BOOT_CFG 2[0]**
		Reserved	Reserved	SPI Addressing: 0 - 2-bytes (16-bit) 1 - 3-bytes (24-bit)	AXI / DDR Freq 0 - PLL2: 400MHz 1 - PLL2: 333MHz	OSC_FREQ_SEL 0 - 19.2, 24, 26, 27 MHz - auto detect. 1 - Oscillator frequency is 24 MHz	Reserved	SEC_CONFIG[1:0]	
0	**0814**	**BOOT_CFG 3[7]**	**BOOT_CFG 3[6]**	**BOOT_CFG 3[5]**	**BOOT_CFG 3[4]**	**BOOT_CFG 3[3]**	**BOOT_CFG 3[2]**	**BOOT_CFG 3[1]**	**BOOT_CFG 3[0]**
		Reserved	Reserved	Port Select: 00 - I2C1 / eCSPI1 01 - I2C2 / eCSPI2 10 - I2C3 / CSPI 11 - Reserevd		CS select (SPI only): 00 - CS#0 (default) 01 - CS#1 10 - CS#2 11 - CS#3		Reserved	DIR_BT_DIS
0	0818	BT_LPB_DIV[1:0] 00 – Divide by 1 01 – Divide by 2 10 – Divide by 3 11 – Divide by 4		Reserved	BT_LPB	BT_LPB_POLARITY	Reserved	Reserved	WDOG_ENABLE

Table 2-7. SD/eSD Boot Fusemap

FB	Addr	7	6	5	4	3	2	1	0
0	**080C**	**BOOT_CFG 1[7]**	**BOOT_CFG 1[6]**	**BOOT_CFG 1[5]**	**BOOT_CFG 1[4]**	**BOOT_CFG 1[3]**	**BOOT_CFG 1[2]**	**BOOT_CFG 1[1]**	**BOOT_CFG 1[0]**
		0	1	0	Fast Boot: 0 - Regular 1 - Fast Boot	SD/MMC Speed 0 - High 1- Normal	Reserved	BT_FREQ 0-800 1-400	BT_MMU_ENABLE

Table continues on the next page...

Table 2-7. SD/eSD Boot Fusemap (continued)

FB	Addr	7	6	5	4	3	2	1	0
0	**0810**	**BOOT_CFG 2[7]**	**BOOT_CFG 2[6]**	**BOOT_CFG 2[5]**	**BOOT_CFG 2[4]**	**BOOT_CFG 2[3]**	**BOOT_CFG 2[2]**	**BOOT_CFG 2[1]**	**BOOT_CFG 2[0]**
		Reserved	Reserved	Bus Width: 0 - 1-bit 1 - 4-bit	AXI / DDR Freq 0 - PLL2: 400MHz 1 - PLL2: 333MHz	OSC_FREQ _SEL 0 - 19.2, 24, 26, 27 MHz - auto detect. 1 - Oscillator frequency is 24 MHz	Reserved	SEC_CONFIG[1:0]	
0	**0814**	**BOOT_CFG 3[7]**	**BOOT_CFG 3[6]**	**BOOT_CFG 3[5]**	**BOOT_CFG 3[4]**	**BOOT_CFG 3[3]**	**BOOT_CFG 3[2]**	**BOOT_CFG 3[1]**	**BOOT_CFG 3[0]**
		Reserved	Reserved	Port Select: 00 - eSDHC1 01 - eSDHC2 10 - eSDHC3 11 - eSDHC4		Reserved	Reserved	Reserved	DIR_BT_DIS
0	0818	BT_LPB_DIV[1:0] 00 – Divide by 1 01 – Divide by 2 10 – Divide by 3 11 – Divide by 4		Reserved	BT_LPB	BT_LPB_PO LARITY	Reserved	Reserved	WDOG_EN ABLE

Table 2-8. MMC/eMMC Boot Fusemap

FB	Addr	7	6	5	4	3	2	1	0
0	**080C**	**BOOT_CFG 1[7]**	**BOOT_CFG 1[6]**	**BOOT_CFG 1[5]**	**BOOT_CFG 1[4]**	**BOOT_CFG 1[3]**	**BOOT_CFG 1[2]**	**BOOT_CFG 1[1]**	**BOOT_CFG 1[0]**
		0	1	1	Fast Boot: 0 - Regular 1 - Fast Boot	SD/MMC Speed 0 - High 1- Normal	Reserved	BT_FREQ 0-800 1-400	BT_MMU_E NABLE
0	**0810**	**BOOT_CFG 2[7]**	**BOOT_CFG 2[6]**	**BOOT_CFG 2[5]**	**BOOT_CFG 2[4]**	**BOOT_CFG 2[3]**	**BOOT_CFG 2[2]**	**BOOT_CFG 2[1]**	**BOOT_CFG 2[0]**
		Bus Width: 000 - 1-bit 001 - 4-bit 010 - 8-bit 101 - 4-bit DDR (MMC 4.4) 110 - 8-bit DDR (MMC 4.4) Else - reserved.			AXI / DDR Freq 0 - PLL2: 400MHz 1 - PLL2: 333MHz	OSC_FREQ _SEL 0 - 19.2, 24, 26, 27 MHz - auto detect. 1 - Oscillator frequency is 24 MHz	Reserved	SEC_CONFIG[1:0]	

Table continues on the next page...

Table 2-8. MMC/eMMC Boot Fusemap (continued)

FB	Addr	7	6	5	4	3	2	1	0
0	**0814**	**BOOT_CFG 3[7]**	**BOOT_CFG 3[6]**	**BOOT_CFG 3[5]**	**BOOT_CFG 3[4]**	**BOOT_CFG 3[3]**	**BOOT_CFG 3[2]**	**BOOT_CFG 3[1]**	**BOOT_CFG 3[0]**
		Reserved	Reserved	Port Select: 00 - eSDHC1 01 - eSDHC2 10 - eSDHC3 (eMMC4.4) 11 - eSDHC4		DLL Override: 0 - Boot ROM default. 1 - Apply value per fuse field MMC_DLL_DLY[3:0]	Fast Boot Acknowledge Disable: 0 - Boot Ack Enabled 1 - Boot Ack Disabled	Reserved	DIR_BT_DIS
0	0818	BT_LPB_DIV[1:0] 00 – Divide by 1 01 – Divide by 2 10 – Divide by 3 11 – Divide by 4		Reserved	BT_LPB	BT_LPB_POLARITY	Reserved	Reserved	WDOG_ENABLE

Table 2-9. NAND Flash Fusemap

FB	Addr	7	6	5	4	3	2	1	0
0	**080C**	**BOOT_CFG 1[7]**	**BOOT_CFG 1[6]**	**BOOT_CFG 1[5]**	**BOOT_CFG 1[4]**	**BOOT_CFG 1[3]**	**BOOT_CFG 1[2]**	**BOOT_CFG 1[1]**	**BOOT_CFG 1[0]**
		1	Muxed On: 0 - PATA 1 - WEIM	Interleave Scheme: 00 - No Interleaving 01- 2 device 10- 4 device 11- Reserved		Address Cycles: 00 - 3 01 - 4 10 - 5 11 - 6		BT_FREQ 0-800 1-400	BT_MMU_ENABLE
0	**0810**	**BOOT_CFG 2[7]**	**BOOT_CFG 2[6]**	**BOOT_CFG 2[5]**	**BOOT_CFG 2[4]**	**BOOT_CFG 2[3]**	**BOOT_CFG 2[2]**	**BOOT_CFG 2[1]**	**BOOT_CFG 2[0]**
		Page Size: 00 - 512 + 16 Bytes (4-bit ECC) 01 - 2KB + 64 Bytes 10 - 4KB + 128 Bytes 11 - 4KB + 218 Bytes		Nand Interface: 0 - 8 bit 1- 16 bit	AXI / DDR Freq 0 - PLL2: 400MHz 1 - PLL2: 333MHz	OSC_FREQ_SEL 0 - 19.2, 24, 26, 27 MHz - auto detect. 1 - Oscillator frequency is 24 MHz	NFC_FREQ_SEL (Nand Flash clock frequency) 0 - AXI DDR divide by 12 1 - AXI DDR divide by 28	SEC_CONFIG[1:0]	

Table continues on the next page...

Table 2-9. NAND Flash Fusemap (continued)

FB	Addr	7	6	5	4	3	2	1	0
0	**0814**	**BOOT_CFG 3[7]**	**BOOT_CFG 3[6]**	**BOOT_CFG 3[5]**	**BOOT_CFG 3[4]**	**BOOT_CFG 3[3]**	**BOOT_CFG 3[2]**	**BOOT_CFG 3[1]**	**BOOT_CFG 3[0]**
		Stride Size (Bad Block skip step 0 - 1 Block 1 - 8 Blocks	LBA-Nand 0 - Non LBA (11ms delay) 1 - LBA (22ms delay)	NAND Use R/B Signals: 0 - No 1 - Yes	ECC / Spare select: 00 - 8-bit ECC 01 - 14-bit ECC 10 - 16-bit ECC 11 - ECC OFF		Pages In Block: 00 - 32 01 - 64 10 - 128 11 - 256		DIR_BT_DIS
0	0818	BT_LPB_DIV[1:0] 00 – Divide by 1 01 – Divide by 2 10 – Divide by 3 11 – Divide by 4		Reserved	BT_LPB	BT_LPB_POLARITY	Reserved	Reserved	WDOG_ENABLE

Table 2-10. Common Fusemap

FB	Addr	7	6	5	4	3	2	1	0
0	0820	SJC_CHALL[63:56] / UNIQUE_ID[63:56]							
0	0824	SJC_CHALL[55:48] / UNIQUE_ID[55:48]							
0	0828	SJC_CHALL[47:43] / UNIQUE_ID[47:40]							
0	082C	SJC_CHALL[39:32] / UNIQUE_ID[39:32]							
0	0830	SJC_CHALL[31:24] / UNIQUE_ID[31:24]							
0	0834	SJC_CHALL[23:16] / UNIQUE_ID[23:16]							
0	0838	SJC_CHALL[15:8] / UNIQUE_ID[15:8]							
0	083C	SJC_CHALL[7:0] / UNIQUE_ID[7:0]							
0	0840	SRTC_MCOUNT[2:0] The number of updates per 512 second: 000 - 512 (1 per second in average) 001 - 1024 (2 per second in average) 010 - 2048 (4 per second in average) 011 - 4096 (8 per second in average) 100 - 8192 (16 per second in average) 101 - reserved 110 - reserved 111 - No restriction			SJC_DISABLE 0 - Secure JTAG Controller is enabled 1 - Secure JTAG Controller is disabled	LDO_DIS[1:0] The two bits controls the Analog and Digital supplies of the DPLL's: 0 - PLL supply is provided by internal LDO source. 1 - PLL supply is provided by the external pad.		SRTC_SECMODE[1:0] 00 - Low Security 01 - Medium Security 10 - High Security 11 - Reserved	

Table continues on the next page...

Table 2-10. Common Fusemap (continued)

FB	Addr	7	6	5	4	3	2	1	0
0	0844	Reserved			CSU_FA_OUT[1] 0 - The CSU alarm inputs that routed to csu_alarm_out1 are specified by Alarm Routing Registers 1 - All CSU alarm inputs are routed to csu_alarm_out1	CSU_FA_OUT[0] 0 - The CSU alarm inputs that routed to csu_alarm_out0 are specified by Alarm Routing Registers 1 - All CSU alarm inputs are routed to csu_alarm_out0	CSU_AM_DIA[1] 0 - CSU alarm inputs, csu_alarm_in[N:N/2], can be masked 1 - CSU alarm inputs, csu_alarm_in[N:N/2], can't be masked	CSU_AM_DIS[0] 0 - CSU alarm inputs, csu_alarm_in[N/2-1:2], can be masked 1 - CSU alarm inputs, csu_alarm_in[N/2-1:2], can't be masked	CSU_FA_COUNT 0 - CSU alarm counter can be disabled and stoped by SW 1 - CSU alarm counter can't be disabled or stoped by SW
0	0860	GP[63:56] - General Purpose user fuses							
0	0864	GP[55:48] - General Purpose user fuses							
0	0868	GP[47:40] - General Purpose user fuses							
0	086C	GP[39:32] - General Purpose user fuses							
0	0870	GP[31:24] - General Purpose user fuses							
0	0874	GP[23:16] - General Purpose user fuses							
0	0878	GP[15:8] - General Purpose user fuses							
0	087C	GP[7:0] - General Purpose user fuses							
0	0C00	Reserved			MAC_ADDR_LOCK 0 - Unlock 1 - Lock	Reserved	SRK_LOCK 0 - Unlock 1 - Lock	SJC_RESP_LOCK 0 - Unlock 1 - Lock	Reserved
1	0C04	SRK_HASH[255:248]							
1	0C08	SJC_RESP[55:48]							
1	0C0C	SJC_RESP[47:40]							
1	0C10	SJC_RESP[39:32]							
1	0C14	SJC_RESP[31:24]							
1	0C18	SJC_RESP[23:16]							
1	0C1C	SJC_RESP[15:8]							
1	0C20	SJC_RESP[7:0]							
1	0C24	MAC_ADDR[47:40]							
1	0C28	MAC_ADDR[39:32]							
1	0C2C	MAC_ADDR[31:24]							
1	0C30	MAC_ADDR[23:16]							
1	0C34	MAC_ADDR[15:8]							
1	0C38	MAC_ADDR[7:0]							

Table continues on the next page...

Table 2-10. Common Fusemap (continued)

FB	Addr	7	6	5	4	3	2	1	0
3	1400	Reserved						SRK_LOCK88 0 - Unlock 1 - Lock	SRK_LOCK160 0 - Unlock 1 - Lock
3	1404-142C	SRK_HASH[247:160]							
3	1430-147C	SRK_HASH[159:0]							
4	1808	MMC_DLL_DLY[3:0]				Reserved			
4	180C	Reserved		NFC_DLL_DLY 000 - Zero delay (default) XXX - TBD			SATA_ALT_CLK_REF 00 - 100MHz (External) 01 - 50MHz (External) 10 - 120MHz, internal (USB PHY) 11 - Reserved		SATA_RST_SRC 0 - SRC reset 1 - Orig SATA reset controller
4	1810	Reserved				NFC_RD_ST_TIME 0 - Use new (fast) read status timing. (status_samp_sel=1) 1 - Old / Legacy implementation (status_samp_sel=0)	SRK_REVOKE[2:0]		
4	1814	GP[87:80] - General Purpose user fuses							
4	1818	GP[79:72] - General Purpose user fuses							
4	181C	GP[71:64] - General Purpose user fuses							

2.2 Fusemap Description Table

This section covers the fusemap descriptions of the chip.

Table 2-11. Fusemap Descriptions

IIM addr	IIM Bank	Fuses Name	Number of Fuses	Fuses Function	Setting	Locked by/Is Lock?
0800[0]	0	BOOT_LOCK	1	Boot fuses Locking rows 0804 080C-0818 0840-0844 0854 fusebank0.	0 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded IIM register) 1 - Lock (The controlled field can be read or sensed only)	LOCK
0800[1]	0	GP_LOCK	1	Locking rows 0860-087C, fusebank0	0 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded IIM register) 1 - Lock (The controlled field can be read or sensed only)	LOCK
0804[0]	0	JTAG_BP	1	JTAG Debug Security Bypass. Blowing this fuse semi-permanently returns the device to the “JTAG Enable mode”	0 - JTAG Security bypass is not active 1 - JTAG Security bypass is active	BOOT_LOCK
0804[1]	0	SEC_JTAG_RE	1	Secure JTAG Re-enable. Overrides the Secure JTAG Bypass fuse (JTAG_BP, in this register), to limit JTAG access according to the JTAG_SMODE. The JTAG_RE signal permanently overrides JTAG security bypass, returning the device to “Secure JTAG mode.	0 - Secure JTAG Bypass fuse is not overridden (secure JTAG bypass is allowed) 1 - Secure JTAG Bypass fuse is overridden (secure JTAG bypass is not allowed)	BOOT_LOCK
0804[2]	0	KTE	1	Kill Trace Enable. Enables tracing capability on ETM, and other modules.	0 - Bus tracing is allowed 1 - Bus tracing is allowed in case security state as defined by Secure JTAG allows it (for example, JTAG_ENABLE or NO_DEBUG)	BOOT_LOCK
0804[3]	0	JTAG_HEO	1	JTAG HAB Enable Override. Disallows HAB JTAG enabling. The HAB may normally enable JTAG debugging by means of the HAB_JDE-bit in the IIM SCS0 register. The JTAG_HEO-bit can override this behavior.	0 - HAB may enable JTAG debug access 1 - HAB JTAG enable is overridden (HAB may not enable JTAG debug access)	BOOT_LOCK

Table continues on the next page...

Table 2-11. Fusemap Descriptions (continued)

IIM addr	IIM Bank	Fuses Name	Number of Fuses	Fuses Function	Setting	Locked by/Is Lock?
0804[4]	0	BT_FUSE_SEL	1	Determines, whether using fuses for boot configuration, or GPIO /Serial loader.	If boot_mode=""00"" (Development) 0=Boot mode configuration is taken from GPIOs. 1=Boot mode configuration is taken from fuses. If boot_mode=""10"" (Production) 0 - Boot using Serila Loader (UART/USB) 1- Boot mode configuration is taken from fuses.	BOOT_LOCK
0804[6:5]	0	JTAG_SMODE[1:0]	2	JTAG Security Mode. Controls the security mode of the JTAG debug interface	00 - JTAG enable mode 01 - Secure JTAG mode 11 - No debug mode	BOOT_LOCK
080C[0]	0	BT_MMU_ENABLE (BOOT_CFG1[0])	1	Option to enable MMU initializing during the HAB boot. (Include L2 cache enable), Enable L1 during start of boot processing. Have a corresponded GPIO pin.	0 - Not initializing MMU during the boot 1 - Initializing MMU with L1 cache (write through) during the boot	BOOT_LOCK
080C[1]	0	BT_FREQ (BOOT_CFG1[1])	1	ARM core frequency at boot time.	0 - 800MHz 1 - 400MHz	BOOT_LOCK
080C[7:4]	0	BOOT_CFG1[7:2]	4	Defines specific boot mode configurations. Refer to the respective boot fusemap tables for details.	0x0000 - WEIM (NOR/OneNAND) boot 0x0010 - HD (PATA/SATA) boot 0x0011 - Serial ROM (I2C/SPI) boot 0x1XXX - NAND FLASH boot 0x01XX - SD/MMC (eSD/eMM) boot 0x0001 - Reserved Refer to the respective boot fusemap tables for details.	BOOT_LOCK
0810[1:0]	0	SEC_CONFIG[1:0] (BOOT_CFG2[1:0])	2	Security Configuration	00 - FAB (Open) 01 - Open - allows any code to be flashed and executed, even if it has no valid signature. 1x - Closed (Security On)	BOOT_LOCK

Table continues on the next page...

Table 2-11. Fusemap Descriptions (continued)

IIM addr	IIM Bank	Fuses Name	Number of Fuses	Fuses Function	Setting	Locked by/Is Lock?
0810[2]	0	NFC_FREQ_SEL	1	NFC Clock Freq selection, to meet device speed characteristics. (Delay line set to '0' delay)	NAND Flash clock frequency, (in brackets freq for 400MHz / 300MHz DDR AXI clock setting respectively): 0 - AXI DDR divide by 12, (33.33MHz / 27.75MHz) 1 - AXI DDR divide by 28 (14.28MHz / 11.89MHz)	BOOT_LOCK
0810[3]	0	OSC_FREQ_SEL (BOOT_CFG2[3])	1	Oscillator frequency select. It is used by boot code for PLL programming.	0 - Auto detect (~10ms cost in boot time) Selects between 19.2, 24, 26 and 27 MHz. 1 - Oscillator frequency is 24 MHz	BOOT_LOCK
0810[4]	0	AXI / DDR Freq (BOOT_CFG2[4])	1	DDR / AXI default frequency settings	0 - PLL2: 400MHz 1 - PLL2: 333MHz	BOOT_LOCK
0810[7:5]	0	BOOT_CFG2[7:5]	3	Various boot options configurations, specific to the selected boot device. Refer to the respective boot fusemap tables for details.	Refer to the respective boot fusemap tables for details.	BOOT_LOCK
0814[0]	0	DIR_BT_DIS	1	Direct External Memory Boot Disable	0 - Direct boot from external memory is allowed 1 - Direct boot from external memory is not allowed	BOOT_LOCK
0814[7:1]	0	BOOT_CFG3[7:1]	7	Various boot options configurations, specific to the selected boot device. Refer to the respective boot fusemap tables for details.	Refer to the respective boot fusemap tables for details.	BOOT_LOCK
0818[0]	0	WDOG_ENABLE	1	Watchdog Enable	Used to specify whether to enable / not watchdog at boot. 0 - Watch-Dog is disabled. 1 - Watch-Dog is enabled.	BOOT_LOCK
0818[3]	0	BT_LPB_POLARITY	1	Select polarity of GPIO pin used for LPB indication.	0 - Positive logic ('1' on GPIO - means LPB boot) 1 - Inverse logic ('0' on GPIO - means LPB boot)	BOOT_LOCK
0818[4]	0	BT_LPB	1	Options for Low Power Boot mode. Have a corresponded GPIO pin.	0 - Low power boot is not activated 1 - Low Power boot is enabled	BOOT_LOCK

Table continues on the next page...

Table 2-11. Fusemap Descriptions (continued)

IIM addr	IIM Bank	Fuses Name	Number of Fuses	Fuses Function	Setting	Locked by/Is Lock?
0818[7:6]	0	BT_LPB_DIV	2	In Low Power Boot mode - control of DVFS divider factors.	00 – Divide by 1 (No devision) 01 – Divide by 2 10 – Divide by 3 11 – Divide by 4	BOOT_LOCK
0820[7:0]	0	SJC_CHALL[63:56] / UNIQUE_ID[63:56]	8	SJC CHALLENGE/ Unique ID	Burned by Freescale	TESTER_LOCK
0824[7:0]	0	SJC_CHALL[55:48] / UNIQUE_ID[55:48]	8	SJC CHALLENGE/ Unique ID	Burned by Freescale	TESTER_LOCK
0828[7:0]	0	SJC_CHALL[47:43] / UNIQUE_ID[47:40]	8	SJC CHALLENGE/ Unique ID	Burned by Freescale	TESTER_LOCK
082C[7:0]	0	SJC_CHALL[39:32] / UNIQUE_ID[39:32]	8	SJC CHALLENGE/ Unique ID	Burned by Freescale	TESTER_LOCK
0830[7:0]	0	SJC_CHALL[31:24] / UNIQUE_ID[31:24]	8	SJC CHALLENGE/ Unique ID	Burned by Freescale	TESTER_LOCK
0834[7:0]	0	SJC_CHALL[23:16] / UNIQUE_ID[23:16]	8	SJC CHALLENGE/ Unique ID	Burned by Freescale	TESTER_LOCK
0838[7:0]	0	SJC_CHALL[15:8] / UNIQUE_ID[15:8]	8	SJC CHALLENGE/ Unique ID	Burned by Freescale	TESTER_LOCK
083C[7:0]	0	SJC_CHALL[7:0] / UNIQUE_ID[7:0]	8	SJC CHALLENGE/ Unique ID	Burned by Freescale	TESTER_LOCK
0840[1:0]	0	SRTC_SECMODE[1:0]	2	Security Mode for Secure RTC. Determines the level of security for Secure Real Time Clock (SRTC) module	00 - Low Security 01 - Medium Security 10 - High Security 11 - Reserved	BOOT_LOCK

Table continues on the next page...

Table 2-11. Fusemap Descriptions (continued)

IIM addr	IIM Bank	Fuses Name	Number of Fuses	Fuses Function	Setting	Locked by/Is Lock?
0840[3:2]	0	LDO_DIS[1:0]	2	LDO_DIS[0]: Controls vddpllana - Analog PLL supply. LDO_DIS[1]: Controls vddplldig - Digital PLL supply.	The two bits controls the Analog and Digital supplies of the DPLL's: 0 - PLL supply is provided by internal LDO source. 1 - PLL supply is provided by the external pad.	BOOT_LOCK
0840[4]	0	SJC_DISABLE	1	Disable/Enable the Secure JTAG Controller module. This fuse is used to create highest JTAG security level, where JTAG is totally blocked.	0 - Secure JTAG Controller is enabled 1 - Secure JTAG Controller is disabled	BOOT_LOCK
0840[7:5]	0	SRTC_MCOUNT[2:0]	3	Monotonic Counter Roll-Over Protection Mechanism. In order to prevent a case, where a malicious SW rolls over the monotonic counter, counter update limitation mechanism is implemented. Updates beyond the specified number is ignored and counter is not incremented.	The number of updates per 512 second: 000 - 512 (1 per second in average) 001 - 1024 (2 per second in average) 010 - 2048 (4 per second in average) 011 - 4096 (8 per second in average) 100 - 8192 (16 per second in average) 101 - reserved 110 - reserved 111 - No restriction	BOOT_LOCK
0844[0]	0	CSU_FA_COUNT	1	CSU Force Alarm Counter. When set, the Alarm Counter is always enabled (can not be disabled by SW) and the option to stop the Counter by SW is disabled.	0 - CSU alarm counter can be disabled and stoped by SW 1 - CSU alarm counter can't be disabled or stoped by SW	BOOT_LOCK
0844[1]	0	CSU_AM_DIS[0]	1	CSU Alarm Mask Disable 0. When this fuse is set, the first half of the CSU alarm input signals can't be masked. Note that the first two alarm input signals can't be masked by definition as this fuse has no effect on them.	0 - CSU alarm inputs, csu_alarm_in[N/2-1:2], can be masked 1 - CSU alarm inputs, csu_alarm_in[N/2-1:2], cannot be masked	BOOT_LOCK

Table continues on the next page...

Table 2-11. Fusemap Descriptions (continued)

IIM addr	IIM Bank	Fuses Name	Number of Fuses	Fuses Function	Setting	Locked by/Is Lock?
0844[2]	0	CSU_AM_DIS[1]	1	CSU Alarm Mask Disable 1. When this fuse is set, the second half of the CSU alarm input signals can't be masked.	0 - CSU alarm inputs, csu_alarm_in[N:N/2], can be masked 1 - CSU alarm inputs, csu_alarm_in[N:N/2], can't be masked	BOOT_LOCK
0844[3]	0	CSU_FA_OUT[0]	1	CSU Force Alarm Output 0. When this fuse is set, all CSU alarm inputs are routed to csu_alarm_out0 regardless of settings in Alarm Routing Registers. Note that this fuse doesn't affect routing policy of other alarm outputs.	0 - The CSU alarm inputs that routed to csu_alarm_out0 are specified by Alarm Routing Registers 1 - All CSU alarm inputs are routed to csu_alarm_out0	BOOT_LOCK
0844[4]	0	CSU_FA_OUT[1]	1	CSU Force Alarm Output 1. When this fuse is set, all CSU alarm inputs are routed to csu_alarm_out1 regardless of settings in Alarm Routing Registers. Note that this fuse doesn't affect routing policy of other alarm outputs.	0 - The CSU alarm inputs that routed to csu_alarm_out1 are specified by Alarm Routing Registers 1 - All CSU alarm inputs are routed to csu_alarm_out1	BOOT_LOCK
0860	0	GP[63:56]	8	General Purpose user fuses	0	GP_LOCK
0864	0	GP[55:48]	8	General Purpose user fuses	0	GP_LOCK
0868	0	GP[47:40]	8	General Purpose user fuses	0	GP_LOCK
086C	0	GP[39:32]	8	General Purpose user fuses	0	GP_LOCK
0870	0	GP[31:24]	8	General Purpose user fuses	0	GP_LOCK
0874	0	GP[23:16]	8	General Purpose user fuses	0	GP_LOCK
0878	0	GP[15:8]	8	General Purpose user fuses	0	GP_LOCK
087C	0	GP[7:0]	8	General Purpose user fuses	0	GP_LOCK
0C00[4]	1	MAC_ADDR_LOCK	1	Lock for rows 0024-0038, fusebank1.	0 - Unlock 1 - Lock	LOCK
0C00[2]	1	SRK_LOCK	1	Locking row 0004, fusebank1 SRK_HASH[255:248]	0 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded IIM register) 1 - Lock (The controlled field can be read or sensed only)	LOCK
0C00[1]	1	SJC_RESP_LOCK	1	Locking 0008-0020, fusebank1. SJC_RESP[55:0] When locked, the fuses cannot be read, sensed, overridden nor programmed	0 - Unlock (SJC_RESP[55:0] can be read, sensed, burned or overridden in the corresponded IIM register) 1 - Lock (SJC_RESP[55:0] cannot be read, sensed, overridden nor written)	LOCK

Table continues on the next page...

Table 2-11. Fusemap Descriptions (continued)

IIM addr	IIM Bank	Fuses Name	Number of Fuses	Fuses Function	Setting	Locked by/Is Lock?
0C04	1	SRK_HASH[255:248]	8	Most Significant Byte of the AP super root key hash	0	SRK_LOCK
0C08	1	SJC_RESP[55:48]	8	Response reference value for the secure JTAG controller	0	SJC_RESP_LOCK
0C0C	1	SJC_RESP[47:40]	8	Response reference value for the secure JTAG controller	0	SJC_RESP_LOCK
0C10	1	SJC_RESP[39:32]	8	Response reference value for the secure JTAG controller	0	SJC_RESP_LOCK
0C14	1	SJC_RESP[31:24]	8	Response reference value for the secure JTAG controller	0	SJC_RESP_LOCK
0C18	1	SJC_RESP[23:16]	8	Response reference value for the secure JTAG controller	0	SJC_RESP_LOCK
0C1C	1	SJC_RESP[15:8]	8	Response reference value for the secure JTAG controller	0	SJC_RESP_LOCK
0C20	1	SJC_RESP[7:0]	8	Response reference value for the secure JTAG controller	0	SJC_RESP_LOCK
0C24	1	MAC_ADDR[47:40]	8	Reserved for users/SW	0	MAC_ADDR_LOCK
0C28	1	MAC_ADDR[39:32]	8	Reserved for users/SW	0	MAC_ADDR_LOCK
0C2C	1	MAC_ADDR[31:24]	8	Reserved for users/SW	0	MAC_ADDR_LOCK
0C30	1	MAC_ADDR[23:16]	8	Reserved for users/SW	0	MAC_ADDR_LOCK
0C34	1	MAC_ADDR[15:8]	8	Reserved for users/SW	0	MAC_ADDR_LOCK
0C38	1	MAC_ADDR[7:0]	8	Reserved for users/SW	0	MAC_ADDR_LOCK
1400[1]	3	SRK_LOCK88	1	Lock for SRK_HASH[247:160] fuses in rows 1404-142C, fusebank3	0 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded IIM register) 1 - Lock (The controlled field can be read or sensed only)	LOCK
1400[0]	3	SRK_LOCK160	1	Lock for SRK_HASH[159:0] fuses in rows 1430-147C	0 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded IIM register) 1 - Lock (The controlled field can be read or sensed only)	LOCK
1404-142C	3	SRK_HASH[247:160]	88	AP Super Root Key hash, bits [247:160] Most significant byte SRK_HASH[255:248] is in the fuse bank #1, 1428	0	SRK_LOCK88

Table continues on the next page...

Table 2-11. Fusemap Descriptions (continued)

IIM addr	IIM Bank	Fuses Name	Number of Fuses	Fuses Function	Setting	Locked by/Is Lock?
1430-147C	3	SRK_HASH[159:0]	160	AP Super Root Key hash, bits [159:0] Most significant byte SRK_HASH[255:248] is in the fuse bank #1, 1428	0	SRK_LOCK160
1808[7:4]	4	MMC_DLL_DLY[3:0]	3	eMMC 4.4 delay line default value (set by boot rom), used in conjunction with "DLL Override" = 1 (BOOT_CFG3[3])	Connected to LVDS module.	Bank4 Locks
180C[0]	4	SATA_RST_SRC	1	Controls mux to select 'herset' to SATA, from either SATA reset controller, or SRC module.	SATA_RST_SRC 0 - SRC reset 1 - Orig SATA reset controller	Bank4 Locks
180C[2:1]	4	SATA_ALT_REF_CLK[1:0]	2	SATA PHY source clock and frequency select	00 - 100 MHz (External) 01 - 50 MHz (External) 10 - 120 MHz, internal (USB PHY) 11 - Reserved	Bank4 Locks
180C[5:3]	4	NFC_DLL_DLY	3	NFC read-status timing (NFC status_samp_sel control)	000 - Zero delay (default) XXX - TBD	Bank4 Locks
1810[3]	4	NFC_RD_ST_TIME	1	NFC read-status timing (NFC status_samp_sel control)	0 - Use new (fast) read status timing. (status_samp_sel=1) 1 - Old / Legacy implementation (status_samp_sel=0)	Bank4 Locks
1810[2:0]	4	SRK_REVOKE[2:0]	3	SRK keys revocation	000 - Key 1 001 - Key 2 011 - Key 3 111 - Key 4 Else - TBD	
1814	0	GP[87:80]	8	General purpose user fuses	0	Bank4 Locks
1818	0	GP[79:72]	8	General purpose user fuses	0	Bank4 Locks
181C	0	GP[71:64]	8	General purpose user fuses	0	Bank4 Locks

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Document Number: IMX53RMAD
Rev. 2.1, 5/2012